Precision Lattice Calculation of Kaon Decays with Möbius Domain Wall Fermions

Hantao Yin

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2013

Abstract

Precision Lattice Calculation of Kaon Decays with Möbius Domain Wall Fermions

Hantao Yin

We report our recent development in algorithms and progress in measurements in lattice QCD. The algorithmic development includes the forecasted force gradient integrator, and further theoretical development and implementation of the Möbius domain wall fermions. These new technologies make it practical to simulate large $48^3 \times 96$ and $64^3 \times 128$ lattice ensembles with $(5.5\text{fm})^3$ boxes and 140MeV pion. The calculation was performed using the Möbius domain wall fermions and the Iwasaki gauge action. Simulated directly at physical quark masses, these ensembles are of great value for our ongoing and future lattice measurement projects.

With the help of measurement techniques such as the eigCG algorithm and the all mode averaging method, we perform a direct, precise lattice calculation of the semileptonic kaon decay $K \longrightarrow \pi l \nu$ using these newly generated high quality lattice ensembles. Our main result is the form factor $f^{\pm}_{K\pi}(q^2)$ evaluated directly at zero momentum transfer $q^2 = 0$. Free of various systematic errors, this new result can be used to determine the CKM matrix element V_{us} to a very high precision when combined with experimental input.

The calculation also provides results for various low energy strong interaction constants such as the pseudoscalar decay constants f_K and f_π, and the neutral kaon mixing matrix element B_K. These calculations are naturally performed by reusing the propagators calculated for the kaon semileptonic decay mentioned above. So they come with no or very low additional cost. The results allow us to also determine these important low energy constants on the lattice to unprecedented accuracy.

Contents

List of Tables

List of Figures

ACKNOWLEDGMENTS

First of all I would like to thank my advisor Prof. Robert Mawhinney for his guidance, insightful discussions and support during my graduate research. The weekly discussion is an reservoir that constantly produces new ideas during the development of this project.

I would also like to thank Prof. Norman Christ for many helpful suggestions and of course help in understanding many physical details. I am grateful to Peter Boyle, Taku Izubuchi and Chulwoo Jung for guidance, help on testing and advice on numerical analysis and programming.

I enjoyed and benefited from discussions with my colleagues: Tom Blum, Luchang Jin, Chris Kelly, Zhongjie “Jasper” Lin, Qi Liu, Jianglei Yu, Daiqian Zhang and many others. The results presented here received refinement from many people in the collaboration. I feel myself very fortunate to work in such a great team.

Finally, I would like to thank my family and friends for their uncountable support and understanding throughout these years.

Chapter 1

Introduction

Since its inception in the 1920s, quantum field theory has become the standard approach to describe subatomic particles. Following the quantization of the electromagnetic field, quantum electrodynamics (QED) has withstood a few most stringent experimental tests in the history of physics. The excellent agreement between the experimental and theoretical values of the anomalous magnetic dipole moment has usually been quoted to show the glory of theoretical development of quantum physics during the 20th century.

The discovery of the quark model 40 years later led to the construction of quantum chromodynamics (QCD), the branch of quantum field theory that describes strong interactions. It took a few decades to fully establish the theory, partly because of the complexity of the theory, and also because higher energy is required to generate the heavier quarks. Nevertheless, there are few parameters in QCD other than the quark masses (in fact, there are no free parameters. The parameter Λ_{QCD} occurs only because we need to set its scale when combine QCD with other parts of the standard model). It is widely believed that QCD has been understood well.

Although quantum chromodynamics is believed to be well understood, its analytic or perturbative treatment in the low energy regime is very difficult if not entirely impossible. The strong interaction has a very large coupling constant in the low energy limit, effectively making the perturbative approach untenable. In addition, the underlying gauge group of QCD is the non-Abelian SU(3) group. This introduces complicated interactions between gluons, rendering analytic treatment very difficult. Many analytic works on QCD are based on various symmetry analysis, low energy effective theories such as the chiral perturbation theory and various other approximations.

On the other hand, lattice discretization is the only known method to perform non-perturbative QCD calculations from first principles. The development in lattice algorithms and supercomputers has progressed greatly during the last few decades. The results are a series of significant successes on various aspects of QCD. The advancement has reached a point where direct calculation using physical light and strange quark masses has just become tenable. This is a very important breakthrough in the field. Since most previous calculations are performed using unphysical parameters, they rely on various extrapolations to obtain physical results. This usually requires more theoretical input such as chiral perturbation theory. As such they are not pure first principle calculations. More importantly, various extrapolations can increase errors significantly. Performing lattice calculations directly at the physical point eliminates the need of extrapolations, greatly reduces the errors of the predictions. The results from such direct calculations will be reported in this work.

Combining QCD with electroweak theory produces the standard model of particle physics. The weak interaction bosons $W^{\pm}$ and Z were not discovered until 1983. The standard model also features many parameters. Out of 19 parameters, 16 are closely related to weak interactions. Even more parameters are required considering the recent discovery of neutrino masses. The weak interaction also includes many new processes that can not happen in other sectors in the standard model. For example, processes that violate CP symmetry must involve weak interactions. For these reasons, weak interaction is usually considered the least understood part in the standard model.

Although an enormous success experimentally, weak interaction is considered by many to be the place where new physics may be discovered. It has many fundamental parameters. Some of these parameters are related because of constraints within the theory. One research direction is to test these constraints, and any discrepancy would call for explanations from new theories. For example, the standard model Cabibbo-Kobayashi-Maskawa (CKM) matrix can be parametrized by 4 constants, assuming that it is a unitary matrix. The unitarity test of this matrix has become a highly sought research project. A negative answer (i.e., the CKM matrix is not unitary) would be extremely interesting. It can potentially become the playground for many new physical models. New sources of CP violations many be found or confirmed and this may lead to new research directions that can answer many open questions in the standard model. A positive answer would also be very interesting, since a stringent agreement of the unitarity condition constrains the ways new physics may be built. Especially, it sets up the scale where new physics may contribute to the elements of the CKM matrix [2].

Weak interactions can be computed using perturbation theory, due to their "weak" nature[1]. However, many interesting weak interaction processes also involve mesons and baryons. QCD interactions can thus mix the weak interaction vertices in nontrivial ways. A full calculation thus requires QCD input. Consequently, lattice QCD calculations are required to give precise predictions for these processes. The standard approach in lattice calculation is to integrate out the weak interaction off the lattice, obtaining an effective Hamiltonian that includes only the strong interaction operators. Lattice QCD is then used to obtain any remaining strong interaction matrix elements.

This work is a combined lattice calculation that produces predictions for many strong and weak interaction parameters. One primary objective is the precise calculation of the $K \longrightarrow \pi$ QCD matrix element via a vector current vertex. This matrix element appears

[1] Interestingly, there is no known lattice theory for weak interactions. In addition, even if such a theory exists, it will still be difficult to carry out any calculation for low energy physics. Since the high energy scale of the weak interaction bosons requires a fine lattice, whereas describing low energy processes requires a large box. The combination of these 2 factors leads to an prohibitively large number of lattice points.

in the semileptonic kaon decays $K \longrightarrow \pi l \nu$. This decay mode is very interesting in that it can be used to precisely determine the CKM matrix element $|V_{us}|$. V_{us} enters the decay amplitude through the weak interaction current $J_W^{\mu +} \propto V_{us}\overline{u}_L \gamma_\mu s_L$. Since V_{us} enters the decay amplitude as an overall constant, it is necessary to integrate over the entire phase space to obtain the full decay width and extract V_{us}. This involves both experimental data and theoretical input. It requires us to compute the QCD matrix element $\left\langle \pi^+ \middle| \overline{u}\gamma_\mu s \middle| \overline{K}^0 \right\rangle$. Chiral perturbation theory can be used to estimate this matrix element with a few percent errors. However, lattice input is necessary to determine this quantity to subpercent accuracy. Precise determination of V_{us} is a major goal of this project.

The calculation also generates other important QCD constants. Among all the results there are pseudoscalar decay constants f_π and f_K, and the neutral kaon mixing matrix element B_K. As mentioned above, calculations presented in this work can improve the accuracy of these parameters to unprecedented levels.

To study low energy QCD, a lattice theory with good chiral symmetry is desired. A chiral theory not only simplifies the extrapolation to the continuum limit[2], but also eases parameter tuning. However, implementing chiral fermions on the lattice is surprisingly difficult. A naive approach induces the “fermion species doubling” problem. A treatment proposed by Wilson solves the doubling problem but breaks chiral symmetry in a nontrivial way that can not be easily cured. Exact chiral fermions on the lattice such as the Ginsperg-Wilson fermion or overlap fermion are usually much more difficult to simulate, and they also pose other practical problems. In fact, the Nielsen-Ninomiya no-go theorem states that there is no lattice fermion that is Hermitian, unitary and also free of fermion doubling problem[3]. So a compromise has to be made when constructing lattice fermions.

In this work we use the domain wall fermion (DWF) and also its generalization, the Möbius fermion actions. Both still break chiral symmetry, but in a well controlled manner.

[2] such extrapolations are always necessary in lattice calculations, for the simple reason that computers can not simulate continuous field.

The residual chiral symmetry breaking effect only introduces a small additive mass renormalization constant. And its effect in the continuum extrapolation is also well under control. The price we pay for good chiral symmetry is that these actions introduce a fictitious fifth dimension in addition to the four dimensional spacetime into the action. The extent of this extra dimension has a direct impact on the quality of the chiral symmetry the action has. We achieve better chiral symmetry with larger sizes in the fifth dimension but the simulation also becomes increasingly difficult. In general, such actions are much more expensive to simulate comparing to simpler actions such as the Wilson action.

Simulating QCD with physical parameters turns out to be a very difficult task. In real QCD the u and d quarks are nearly degenerate with very small masses. Lattice simulations tend to become more expensive when approaching this physical limit. This is due to a few reasons,

1. As the masses of the two light quarks become smaller, the associated lattice Dirac operator becomes aggressively more singular. This effect makes the associated Dirac equation more difficult to solve since the effectiveness of Krylov space solvers is highly dependent on the condition number of the matrix. Sophisticated algorithms such as all mode averaging and eigCG have been developed to mitigate the problem.

2. As the u and d masses become smaller, pion mass also becomes smaller. Larger boxes are needed to control the finite volume effect. Such lattices contain more spacetime points and consequently require more computational resources.

Nevertheless, with the help of new algorithms such as the force gradient integrator, low modes deflation in Krylov space solvers and the all mode averaging technique we are able to perform a direct calculation using large lattices such as $48^3 \times 96 \times 24$ and $64^3 \times 128 \times 12$. These lattices have $(5.5\text{fm})^3$ boxes, large enough for us to simulate at physical pion and kaon masses[3]. This is made possible also because of the introduction of the new IBM BlueGene/Q

[3] On the lattice we usually require $M_\pi L \geq 4$ to have finite volume effects well under control.

supercomputers.

The primary physical results in this work are computed on the $48^3 \times 96$ $(5.5\text{fm}, 140\text{MeV})$ and $64^3 \times 128$ $(5.5\text{fm}, 140\text{MeV})$ lattices[4]. Due to the expensive nature of these lattices, we performed an exploratory calculation on a smaller $32^3 \times 64$ $(4.6\text{fm}, 170\text{MeV})$ lattice. The small lattice serves as a playground to test various algorithms and parameter settings for the larger lattices. Thus, the results on the $32^3 \times 64$ $(4.6\text{fm}, 170\text{MeV})$ lattice are shown here primarily to compare the effectiveness of various algorithms, and the results on the larger lattices are used to predict physical values.

We used the Möbius fermion formalism to generate ensembles for the large $48^3 \times 96$ and $64^3 \times 128$ lattices. This can gain a factor of around 2 over Shamir domain wall actions, which we previously used. The ensembles are generated using the newly invented force gradient integrator, which is more effective on larger lattices. Two additional error/cost reduction techniques are used in the measurements of the QCD matrix elements. First, we use the all mode averaging method to reduce the cost when solving the fermion propagators. This allows us to solve more propagators using the same amount of resources and improve the signal to noise ratio. In addition, we use the eigCG algorithm to improve the condition number of the discretized Dirac operator. This greatly helps to improve the efficiency of the Krylov space solvers we are using.

This work is organized as follows. In chapter 2 we discuss how to establish the lattice framework and setup the gauge and fermion fields on the lattice. In chapter 3 we review the ensemble generation process on the lattice, which is the first important step of any lattice measurements. Chapter 4 discusses our development on Möbius fermions. Chapter 5 then proceeds and discusses various techniques that can be applied to lattice measurements. Chapter 6 addresses various details in the $Kl3$ matrix element calculation. We present our main result in chapter 7.

[4] The two numbers in the parenthesis are the spatial extent and dynamic pion mass. Appendix B has more details about the parameters on these lattice ensembles.

Chapter 2

The Lattice Actions

We discuss in the following briefly how continuum actions are discretized on a 4 dimensional spacetime grid. We start our discussion by assuming the Euclidean metric instead of the Minkowski metric. The details about how a Minkowski action can be transformed into an Euclidean action can be found in e.g. [4].

2.1 Gauge Actions on the Lattice

Unlike the continuum theory where the gauge fields are represented by elements of the Lie algebra of the corresponding gauge group, lattice gauge fields are commonly represented by elements of the gauge group itself. For example, on the lattice the gluon fields are elements of the $SU(3)$ gauge group. Commonly denoted by symbol U, these lattice gauge fields link *2 neighboring spacetime sites.* This fact reflects the original motivation of introducing gauge fields as building blocks for parallel transports and covariant derivatives. In other words, gauge links are primitive Wilson lines that connect 2 end points on the lattice. In the continuum limit such links connect 2 infinitesimally separated spacetime sites, and there is no difference between a gauge group description and a corresponding Lie algebra description.

The simplest quantum chromodynamics gauge action on the lattice is the Wilson gauge action

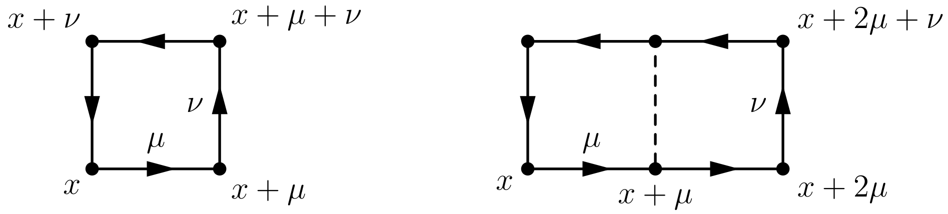
$$S_G^W(U) = \beta \sum_{x,\mu<\nu} \left(1 - \frac{1}{3}\mathrm{ReTr}\left(U_P(x,\mu,\nu)\right)\right), \tag{2.1}$$

where U_P represents a general 1×1 plaquette

$$U_P(x,\mu,\nu) = U_\mu(x)U_\nu(x+\mu)U_\mu^\dagger(x+\nu)U_\nu^\dagger(x). \tag{2.2}$$

The action is a sum over traces of all possible plaquettes with proper normalization. By expanding U_P with respect to a the leading $\mathcal{O}(a^2)$ term represents the usual Lagrangian density in the continuum limit

$$\mathcal{L} = -\frac{1}{4}\mathrm{Tr}\left(F_{\mu\nu}F^{\mu\nu}\right). \tag{2.3}$$



A square plaquette

A rectangle plaquette

Figure 2.1: Square (1×1) and rectangular (1×2) plaquettes in the Wilson and Iwasaki gauge actions.

We use the Iwasaki gauge action [5, 6] in this work. This gauge action includes both 1×1 and 1×2 plaquettes

$$S_G^R(U) = \beta \sum_{x,\mu<\nu} \left(1 - \frac{c_0}{3}\mathrm{ReTr}(U_P(x,\mu,\nu))\right) + \beta \sum_{x,\mu\neq\nu} \left(1 - \frac{c_1}{3}\mathrm{ReTr}(U_R(x,\mu,\nu))\right), \tag{2.4}$$

where $U_R(x,\mu,\nu)$ is the rectangle plaquette

$$U_R(x,\mu,\nu) = U_\mu(x)U_\mu(x+\mu)U_\nu(x+2\mu)U_\mu^\dagger(x+\mu+\nu)U_\mu^\dagger(x+\nu)U_\nu^\dagger(x), \tag{2.5}$$

c_0 and c_1 are 2 constants. The Iwasaki gauge action uses $c_1 = -0.331$ and $c_0 = 1 - 8c_1$. This action has better chiral behavior when used together with domain wall fermions. Both the square and the rectangular plaquettes are shown in figure 2.1.

2.2 The Naive Lattice Fermions

It turns out that representing fermions on the lattice raises certain subtle issues. Simple discretization of the continuum fermion action incurs the "fermion species doubling" problem.

To discretize the continuum Euclidean action

$$S_F\left(\psi,\overline{\psi}\right) = \int d^4x\, \overline{\psi}(x)\left(\gamma_\mu\partial_\mu + m\right)\psi(x) \tag{2.6}$$

we need to replace the derivatives by finite differences. The simplest choice is the following symmetric finite difference

$$\partial_\mu \longleftarrow \frac{1}{2a}\left(\delta_{x+\mu,x'} - \delta_{x-\mu,x'}\right). \tag{2.7}$$

We choose symmetric discretization since asymmetric discretization produces non-Hermitian Hamiltonians. The resulting action on the lattice takes the following form

$$S_F = \sum_{x,y}\overline{\psi}(x)\left(m\delta_{x,y} + \frac{1}{2}\sum_\mu\left(\gamma_\mu\delta_{x+\mu,y} - \gamma_\mu\delta_{x-\mu,y}\right)\right)\psi(y). \tag{2.8}$$

Computing the fermion propagator

$$\left\langle\psi(x)\overline{\psi}(y)\right\rangle = \frac{1}{Z}\int\left(\prod_x d\psi(x)d\overline{\psi}(x)\right)\psi(x)\overline{\psi}(y)e^{-S_F} \tag{2.9}$$

in Fourier space yields the following result

$$\left\langle\psi(x)\overline{\psi}(y)\right\rangle = \frac{1}{N^4}\sum_p\frac{1}{m + i\sum_\mu\gamma_\mu\sin\left(\frac{2\pi}{N}p_\mu\right)}e^{i2\pi p(x-y)/N}. \tag{2.10}$$

For simplicity we assume that the lattice extent is N in all spacetime directions. In the continuum limit the lattice spacing a approaches 0 while the total size in each direction remains fixed, and we obtain the following limit

$$\left\langle \psi(x)\overline{\psi}(y)\right\rangle = \int_B \frac{d^4p}{(2\pi)^4} \frac{1}{\frac{m}{a} + \frac{i}{a}\sum_\mu \gamma_\mu \sin\left(p_\mu a\right)} e^{ip(x-y)}, \tag{2.11}$$

where the integration is performed in the first Brillouin zone. This solution poses the famous "fermion doubling" problem. For values of p_μ close to the origin, $\sin p_\mu a \approx p_\mu a$ and the above propagator behaves like the continuum propagator. However, near the boundary of the first Brillouin zone where $p_\mu \approx \pm\pi/a$ the behavior is also similar to the region near the origin. Thus the propagator receives contributions when all $p_\mu \approx 0, \pm\pi/a$. There are a total of $2^4 = 16$ such regions. So instead of representing a single fermion, the naive fermion action represents $2^4 = 16$ degenerate fermion species. The situation can be seen in figure 2.2.

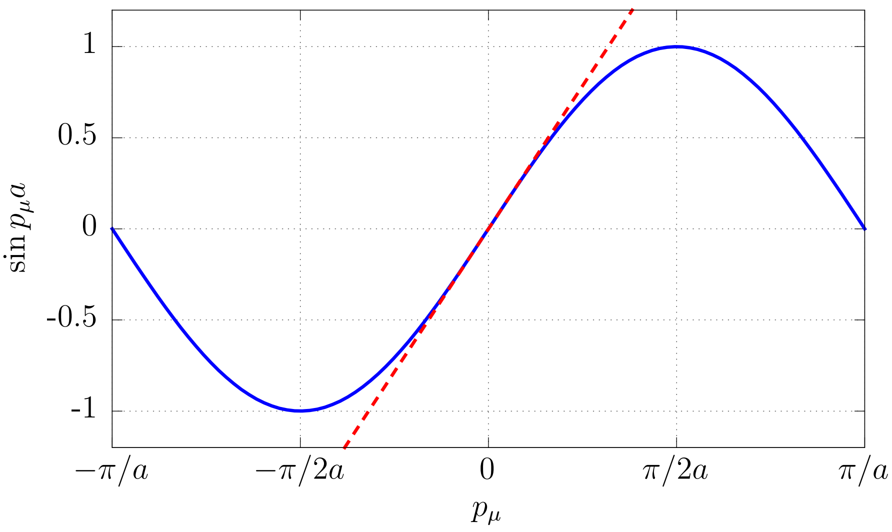


Figure 2.2: Sketch showing the fermion doubling problem near the boundary of the first Brillouin zone. The behavior of $\sin p_\mu a$ in the first Brillouin zone is plotted. The dashed line corresponds to the continuum behavior.

2.3 The Wilson Fermions

Since the continuum limit is obtained via the limit $a \longrightarrow 0$, adding terms that vanish in this limit will not have any continuum remnant. Wilson [7] proposed adding a cutoff to the fermion action that vanishes in the continuum limit, obtaining the following free field Wilson fermion action

$$S_F^W(\psi, \overline{\psi}) = \sum_{xy} \overline{\psi}(x) D_W(x,y) \psi(y). \tag{2.12}$$

Where $D_W(x,y)$ is the *Wilson Dirac operator*

$$D_W(x,y) = (4+m)\delta_{x,y} - \frac{1}{2}\sum_{\mu} \left((1-\gamma_\mu)\delta_{x+\mu,y} + (1+\gamma_\mu)\delta_{x-\mu,y}\right). \tag{2.13}$$

The Wilson propagator is then equal to

$$\left\langle \psi(x)\overline{\psi}(y)\right\rangle = \int_{-\pi/a}^{\pi/a} \frac{d^4p}{(2\pi)^4} \frac{1}{m(p) + \frac{i}{a}\sum_\mu \gamma_\mu \sin(p_\mu a)} e^{ip(x-y)}, \tag{2.14}$$

where

$$m(p) = m + \frac{2}{a}\sum_\mu \sin^2 \frac{p_\mu a}{2}. \tag{2.15}$$

For any fixed p_μ, $m(p)$ approaches m when $a \longrightarrow 0$. Thus the action behaves correctly near the origin. However, near the boundary of the first Brillouin zone it behaves differently from the naive approach. One has $p_\mu \approx \pm\pi/a$ and $m(p)$ diverges

$$\frac{1}{a}\sum_\mu \sin^2 \frac{p_\mu a}{2} \propto \frac{1}{a}. \tag{2.16}$$

This added divergence eliminates any contribution to the continuum propagator from regions other than the center of the first Brillouin zone, removing all doublers from the theory.

Adding gauge links to the action is straightforward and we obtain the following dynamic

Wilson Dirac operator

$$D_W(x,y;U) = (4+m)\delta_{x,y} - \frac{1}{2}\sum_{\mu}\big((1-\gamma_\mu)U_\mu(x)\delta_{x+\mu,y} + (1+\gamma_\mu)U_\mu^\dagger(x-\mu)\delta_{x-\mu,y}\big)\,. \quad (2.17)$$

Though free of fermion doublers, the additional terms added to the Wilson fermion formalism break chiral symmetry since these terms commute with γ_5. The symmetry breaking terms are in general $\mathcal{O}(1)$ quantities. Consequently quantities such as the bare quark mass m receives $\mathcal{O}(1)$ additive renormalization, meaning that it must be finely tuned to represent physical values correctly.

We can derive the representation of the axial current using the Wilson fermion action to see the chiral effect of the added terms. By varying the fermion path integral

$$Z = \int \mathcal{D}\psi\mathcal{D}\overline{\psi}\; e^{-\overline{\psi}D_W\psi} \quad (2.18)$$

using the following local transformation

$$\begin{cases} \psi(x) & \longleftarrow \quad e^{i\gamma_5\theta^a(x)t^a}\psi(x) \\ \overline{\psi}(x) & \longleftarrow \quad \overline{\psi}(x)e^{i\gamma_5\theta^a(x)t^a} \end{cases}, \quad (2.19)$$

we obtain the following conserved quantity

$$\sum_y \big\langle \delta\overline{\psi}(x)D_W(x,y)\psi(y)\big\rangle + \sum_y \big\langle \overline{\psi}(y)D_W(y,x)\delta\psi(x)\big\rangle = 0. \quad (2.20)$$

The resulting axial current and its divergence take the following form

$$A_\mu(x) = \frac{1}{2}\overline{\psi}(x)\gamma_\mu\gamma_5 t^a U_\mu(x)\psi(x+\mu) + \frac{1}{2}\overline{\psi}(x+\mu)\gamma_\mu\gamma_5 t^a U_\mu^\dagger(x)\psi(x). \quad (2.21)$$

$$\begin{aligned}\sum_\mu \Delta_\mu A_\mu(x) =& 2(4+m)\overline{\psi}(x)\gamma_5 t^a \psi(x) \\ & -\frac{1}{2}\sum_\mu \overline{\psi}(x)\gamma_5 t^a U_\mu(x)\psi(x+\mu) \\ & -\frac{1}{2}\sum_\mu \overline{\psi}(x)\gamma_5 t^a U^\dagger_\mu(x-\mu)\psi(x-\mu) \\ & -\frac{1}{2}\sum_\mu \overline{\psi}(x-\mu)\gamma_5 t^a U_\mu(x-\mu)\psi(x) \\ & -\frac{1}{2}\sum_\mu \overline{\psi}(x+\mu)\gamma_5 t^a U^\dagger_\mu(x)\psi(x).\end{aligned} \tag{2.22}$$

Clearly in addition to the continuum term $2m\overline{\psi}\gamma_5 t^a\psi$, there are $\mathcal{O}(1)$ terms that must be absorbed. This yields $\mathcal{O}(1)$ additive renormalizations to the fermion mass.

2.4 The Domain Wall Fermions

Kaplan proposed the domain wall fermion formulation [8] to solve the fermion doubling problem while still maintaining good chiral symmetry. This is achieved by introducing an extra dimension into the action. This extra dimension is usually labeled by s in addition to the 4 dimensional spacetime labeled by x, y, z and t. Suppose the size of this extra dimension is L_s. Written explicitly, the domain wall action looks like L_s copies of the Wilson fermion action with extra hopping terms in the s direction

$$S_F^{\mathrm{DWF}} = \sum_{x,s;x's'} \overline{\psi}(x,s) D^{\mathrm{DWF}}(x,s;x',s')\psi(x',s'), \tag{2.23}$$

where the domain wall Dirac operator is defined as follows

$$D^{\mathrm{DWF}}(x,s;x',s') = (\delta_{x,x'} + D_W(x,x'))\delta_{s,s'} - m(s)P_+\delta_{s,s'+1} - m(s+1)P_-\delta_{s,s'-1}. \tag{2.24}$$

$D_W(x,x')$ is the regular Wilson action with a special mass M_5, the domain wall height. $m(s)$ is defined as follows

$$m_s = \begin{cases} 1 & s = 1, 2, \cdots, L_s - 1 \\ -m & s = 0 \end{cases} . \tag{2.25}$$

The additional hopping terms in s direction can also be thought of as a complicated mass matrix that mixes L_s otherwise degenerate Wilson fermions.

The property of the domain wall fermion action and its generalizations will be discussed in chapter 4. Where we will see that this action shows very good chiral symmetry.

2.5 Pseudofermion Field

An additional problem associated with all fermion actions is that any Grassmann integrals must be evaluated analytically, before we apply any numerical techniques. This is simply because computers can only deal with real numbers. Integrating the fermion part of the action produces fermionic determinants in the path integral,

$$\int \mathcal{D}\psi\mathcal{D}\overline{\psi}\; e^{-\sum_{xy}\overline{\psi}(x)D(x,y)\psi(y)} \propto \det\left(D(x,y)\right). \tag{2.26}$$

On the lattice the u and d quarks are usually treated as degenerate species. So a full quantum chromodynamics path integral with 2 degenerate light quarks and 1 strange quark takes the following form

$$Z = \int \mathcal{D}U\; \det\left(\frac{D(U;m_l)}{D(U;1)}\right)^2 \det\left(\frac{D(U;m_s)}{D(U;1)}\right) \exp\left(-S_G(U)\right). \tag{2.27}$$

Where $D(U;m)$ is the fermion Dirac operator with bare quark mass m. We put a Pauli-Villars factor $D(U;1)$ for each fermion determinant. This heavy bosonic factor is necessary for domain wall like fermions. Since we would have a divergent number of massive fermions in the $L_s \longrightarrow \infty$ limit and the Pauli-Villars factors cancel out this divergence [9]. We will transform these determinants into suitable forms of actions, i.e., into the following form

$$e^{-S_F(U)}. \tag{2.28}$$

$S_F(U)$ is a fermion action suitable for use in the ensemble generation. The reason for doing this is that such exponential factors are the only form that the hybrid Monte Carlo algorithm can work on directly.

2.5.1 Even-odd Preconditioning

Since fermion Dirac operators enter the path integral only through their determinants, we are free to replace the fermion Dirac operator D with any equivalent form as long as its determinant remains the same as the original operator. Even-odd preconditioning is one such method to rewrite the fermion Dirac operator which can later accelerate the calculation.

The 4D even-odd preconditioning rewrites the Dirac operator as

$$D = \begin{pmatrix} M_{ee} & M_{eo} \\ M_{oe} & M_{oo} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ M_{oe}M_{ee}^{-1} & 1 \end{pmatrix} \begin{pmatrix} M_{ee} & M_{eo} \\ 0 & M_{oo} - M_{oe}M_{ee}^{-1}M_{eo} \end{pmatrix}. \tag{2.29}$$

Where the subscripts e and o represent even and odd sites, respectively. Whether a lattice site is even or odd is determined by the parity of the sum of its spacetime coordinates $x + y + z + t$. M_{ee} is the part in D that connects 2 even sites, M_{eo} is the part that starts from an odd site and ends at an even site. The gauge field always connects 2 neighboring spacetime sites, so it can only appear in M_{eo} and M_{oe}, M_{ee} and M_{oo} are both constant matrices. We define the *preconditioned* fermion Dirac operator to be

$$\mathcal{M} = M_{oo} - M_{oe}M_{ee}^{-1}M_{eo}. \tag{2.30}$$

Using this new operator, the determinant of the fermion ratio is equal to

$$\det\left(\frac{D(U;m)}{D(U;1)}\right) = \det\left(\frac{\mathcal{M}(U;m)}{\mathcal{M}(U;1)}\right) \tag{2.31}$$

2.5.2 The Quotient and Rational Quotient Actions

We still need to transform the fermion determinant into the form $e^{-S_F(U)}$ to incorporate it into the hybrid Monte Carlo algorithm. The canonical method of doing this is through the Gaussian integral

$$\int dz^\dagger dz\; e^{-z^\dagger A z} \propto \frac{1}{\det A}, \tag{2.32}$$

where A must be a positive definite matrix. However, only the determinant in equation (2.31) is guaranteed to be positive. So we can not apply (2.32) directly to the fermionic determinant. Nevertheless, this can be promptly applied to the case of 2 degenerate light quarks. Since $\det(\mathcal{M}) = \det(\mathcal{M}^\dagger)$,

$$\begin{aligned}\det\left(\frac{\mathcal{M}(m)}{\mathcal{M}(1)}\right)^2 &= \det\left(\frac{\mathcal{M}(m)^\dagger\mathcal{M}(m)}{\mathcal{M}(1)^\dagger\mathcal{M}(1)}\right) && (2.33)\\ &= \int \mathcal{D}\phi^\dagger\mathcal{D}\phi\; \exp\left(-\phi^\dagger\mathcal{M}(1)\frac{1}{\mathcal{M}(m)^\dagger\mathcal{M}(m)}\mathcal{M}(1)^\dagger\phi\right). && (2.34)\end{aligned}$$

Where ϕ is the *pseudofermion field*, a complex valued auxiliary field that does not have any direct physical meaning. The only purpose of such fields is to cast the fermion determinants into exponential factors.

We can then define the quotient fermion action as

$$S_{\mathrm{Q}}(m_a, m_b) = \phi^\dagger\mathcal{M}(m_b)\frac{1}{\mathcal{M}^\dagger(m_a)\mathcal{M}(m_a)}\mathcal{M}^\dagger(m_b)\phi. \tag{2.35}$$

The fermion action in (2.34) is then $S_{\mathrm{Q}}(m, 1)$.

For the single strange quark determinant the above trick can not be applied. However, we can square it and take the square root via rational approximations

$$\det\left(\frac{\mathcal{M}(m)}{\mathcal{M}(1)}\right) = \det\left(\frac{\mathcal{M}^\dagger(m)\mathcal{M}(m)}{\mathcal{M}^\dagger(1)\mathcal{M}(1)}\right)^{\frac{1}{2}} \tag{2.36}$$

$$= \int \mathcal{D}\phi^\dagger \mathcal{D}\phi \, \exp\left(-\phi^\dagger \left(\mathcal{M}^\dagger(1)\mathcal{M}(1)\right)^{\frac{1}{4}} \frac{1}{\left(\mathcal{M}^\dagger(m)\mathcal{M}(m)\right)^{\frac{1}{2}}} \left(\mathcal{M}^\dagger(1)\mathcal{M}(1)\right)^{\frac{1}{4}} \phi\right). \tag{2.37}$$

Where rational approximations of $x^{1/4}$ and $x^{-1/2}$ are used to evaluate the fractional powers of matrices. A general rational function for use in lattice calculation has the following form

$$r(x) = \sum_k \frac{\alpha_k}{x + \beta_k}, \tag{2.38}$$

where α_k and β_k are real numbers. There are stock algorithms such as the Remez algorithm [10] that calculate rational approximations to functions such as x^λ. In what follows we will use the symbol $S_{\mathrm{R}}(m_1, m_2)$ to represent this rational action

$$S_{\mathrm{R}}(m_1, m_2) = \phi^\dagger r_1\left(\mathcal{M}^\dagger(m_2)\mathcal{M}(m_2)\right) r_2\left(\mathcal{M}^\dagger(m_1)\mathcal{M}(m_1)\right) r_1\left(\mathcal{M}^\dagger(m_2)\mathcal{M}(m_2)\right)\phi, \tag{2.39}$$

where $r_1(x)$ and $r_2(x)$ are rational approximations to functions $x^{1/4}$ and $x^{-1/2}$, respectively.

Chapter 3

The Hybrid Monte Carlo Algorithm

In this chapter we describe the hybrid Monte Carlo algorithm and related algorithmic development. The hybrid Monte Carlo algorithm is our algorithm of choice for generating ensembles in lattice calculations. It has been widely used since its initial development [11].

3.1 Path Integrals and the Monte Carlo Method

The path integral

$$Z = \int \mathcal{D}U\mathcal{D}\psi\mathcal{D}\overline{\psi}\; e^{-S(U,\psi,\overline{\psi})} \tag{3.1}$$

plays a central role in quantum field theory. Any observable $\mathcal{O}(U)$ can in principle be measured using the path integral

$$\big\langle \mathcal{O}(U,\psi,\overline{\psi})\big\rangle = \frac{1}{Z}\int \mathcal{D}U\mathcal{D}\psi\mathcal{D}\overline{\psi}\; \mathcal{O}(U,\psi,\overline{\psi})e^{-S(U,\psi,\overline{\psi})}. \tag{3.2}$$

The lattice approach of quantum chromodynamics directly makes use of the path integral to compute observables from first principles. The integration domain of Z is a very high dimensional object after discretization. Due to its very high dimensionality many conventional numerical integration methods do not work for such integrals (the so called "curse of dimensionality" problem). Such problems are usually tackled using randomized algorithms such as Monte Carlo integration.

The Monte Carlo method estimates an integral by generating a series of random samples in the integration domain. Taking the following 1 dimensional integral as an example

$$I = \int_D dx\ f(x)\omega(x). \tag{3.3}$$

Where we assume that $\omega(x)$ can be used as a distribution,

$$\omega(x) \geq 0,\ \forall x \in D, \tag{3.4}$$

$$\int_D dx\ \omega(x) = 1. \tag{3.5}$$

The Monte Carlo integration method finds N random samples of x according to the distribution $\omega(x)$. I can then be approximated by

$$I \approx \frac{1}{N} \sum_i f(x_i). \tag{3.6}$$

The best case scenario for the above Monte Carlo method is such that $f(x)$ is a slowly varying function of x. This is the idea of importance sampling.

The integration domain of a QCD partition function is the configuration space of the gauge field. On the lattice, one generates a set of random samples U_i within the integration domain according to the following distribution

$$\omega(U) = \frac{1}{Z} e^{-S(U)}. \tag{3.7}$$

So an observable $\mathcal{O}(U)$ is approximated by

$$\langle \mathcal{O}(U) \rangle = \frac{1}{N} \sum_i \mathcal{O}(U_i). \tag{3.8}$$

A typical lattice calculation thus consists of the following 2 stages,

1. (Ensemble generation) Generate a set of gauge field configurations U_i according to the distribution (3.7).

2. (Measurement) Calculate the observable on the generated configurations using (3.8).

There are challenging problems associated with both ensemble generations and measurements. This chapter discusses problems related to ensemble generations. Methods related to measurements will be discussed in chapter 5.

The problem of generating appropriate gauge field samples is challenging due to the following reasons

1. The QCD path integral (2.27) is a very high dimensional integral. For example, there are a total of $48^3 \times 96 = 10616832$ sites for our $48^3 \times 96$ lattice. On each site there are 4 link variables, each is integrated over the 8 dimensional $SU(3)$ gauge group space. So the total dimensionality of the integration domain is $48^3 \times 96 \times 4 \times 8 = 339738624$. At such high dimensions, most methods based on grid discretization of the integration domain can not produce any reasonable result since the efficiency of these algorithms degrades as the dimensionality of the sampling space increases.

2. The density distribution $\omega(U_i) = e^{-S_G(U)-S_F(U)}$ becomes non-local once fermions are included. So local updating methods such as the heat bath method do not work well. Global updating methods (i.e., methods that refresh all gauge links simultaneously) are usually required.

The hybrid Monte Carlo algorithm is a global updating method built on top of the Metropolis-Hastings algorithm.

3.2 The Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm is a Markov chain method. It solves the ensemble generation problem in the Monte Carlo integration. We describe its continuum version here.

Suppose we want to generate a random sample within domain D and distribution $\omega(x)$, $x \in D$. Starting from an arbitrary point $x_0 \in D$, the Metropolis algorithm returns a new sample x_{i+1} from a previous sample x_i by performing the following procedure [12]

1. Pick up a new point $y_i \in D$ based on the distribution $P(y|x_i)$. $P(y|x)$ is called the "jumping distribution" which specifies how to obtain a new sample from a given old sample. This transition function must be reversible

$$P(y|x)dx = P(x|y)dy, \tag{3.9}$$

where dx or dy is the volume element at point x or y, respectively.

2. Compute the *acceptance ratio*

$$\lambda = \frac{\omega(y_i)}{\omega(x_i)}, \tag{3.10}$$

3. Accept the new point y_i with probability λ. If $\lambda \geq 1$ then the new point is always accepted. Return y_i as the next sample x_{i+1} if it is accepted, otherwise *duplicate the old sample* x_i as the next sample x_{i+1}.

The Metropolis algorithm is an extremely adaptive method because of a variety of choices can be made to construct the jumping distribution $P(y|x)$. On the lattice, the molecular dynamics evolution is the predominant choice when constructing $P(y|x)$. This is the hybrid Monte Carlo algorithm.

3.3 The Hybrid Monte Carlo Algorithm

As stated above, the hybrid Monte Carlo algorithm is an adaption of the Metropolis algorithm in which a new sample is obtained by molecular dynamics evolution of the old sample. Its basic idea can be succinctly described by the following example.

Consider the problem of generating a random variable q with normal distribution

$$\omega(q) \propto e^{-q^2/2}. \tag{3.11}$$

The partition function is

$$Z = \int_{-\infty}^{\infty} dq\, \omega(q). \tag{3.12}$$

Using the hybrid Monte Carlo algorithm, q is treated as a generalized coordinate of a Hamiltonian system and $S(q) = q^2/2$ can be thought as the action. We add a fictitious kinetic term $T(p)$ to the action so we have a complete Hamiltonian. Usually we choose the following quadratic form for $T(p)$

$$T(p) = \frac{1}{2}p^2. \tag{3.13}$$

The new partition function is

$$Z' = \int dqdp\, e^{-H(q,p)}, \tag{3.14}$$

where $H(q,p) = T(p) + S(q)$. The new problem now consists of generating random samples of (q,p) pairs following distribution $e^{-H(q,p)}$. By integrating p out we get the desired sample q. Operationally this simply means dropping p from each of the pairs.

The hybrid Monte Carlo algorithm generates p by using a direct method. In this example, we generate a new p each time following the distribution *without any considerations of the previous sample in the sequence*

$$\omega(p) \propto e^{-T(p)} = e^{-p^2/2}. \tag{3.15}$$

This is simple to do and there are many stock algorithms to perform the task.

The next step in the hybrid Monte Carlo algorithm moves both q and p by evolving the state (q, p) according to the Hamilton equations

$$\frac{dq}{dt} = \{H, q\} = -p, \tag{3.16}$$

$$\frac{dp}{dt} = \{H, p\} = q. \tag{3.17}$$

The above Hamiltonian system is integrated for a certain duration and the resulting end point in the phase space is regarded as the new proposed sample. The duration that we integrate equations (3.16, 3.17) is the *molecular dynamics time*, usually denoted by τ.

In this example we can integrate the Hamiltonian system (3.16,3.17) exactly. Consequently the acceptance ratio will always be 1 and the proposed new point will always be accepted. This is simply because Hamiltonian is conserved when evolving the system. For practical Hamiltonian systems it is usually infeasible or not economical to perform exact integration. The accept/reject step is then necessary to eliminate any bias for such cases.

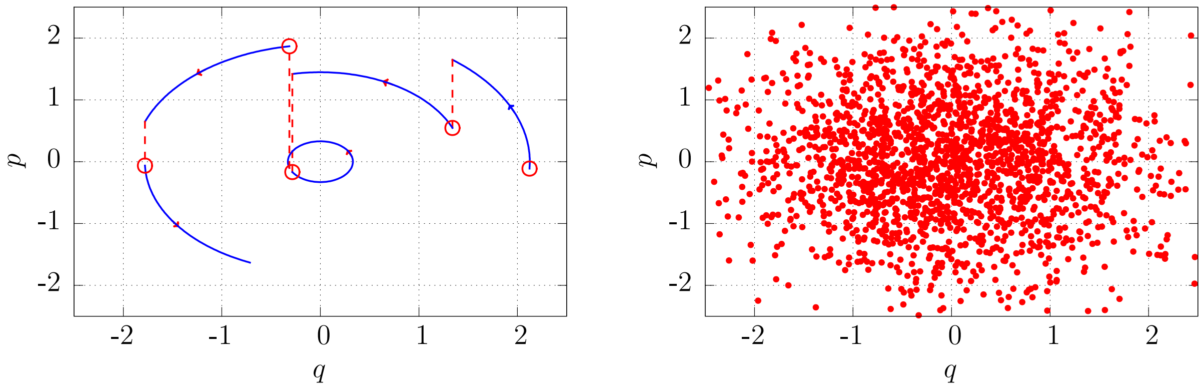


Figure 3.1: Left: 5 sample steps in the hybrid Monte Carlo algorithm. Blue solid curves: trajectories of (q, p) following the Hamilton equation, the corresponding molecular dynamics time duration is 2. Red dashed lines: "jumps" in p due to refreshment of the generalized momentum. Right: Distribution of (q, p) after 2000 steps.

Putting these steps together, the full hybrid Monte Carlo algorithm generates a new point q_{i+1} from the previous sample q_i using the following steps.

1. Generate a new p_i according to the normal distribution

$$\omega(p) \propto e^{-T(p)} = e^{-p^2/2} \tag{3.18}$$

   using any direct method.

2. Evolve the point (q_i, p_i) for a certain amount of molecular dynamics time according to some proposed Hamilton equations to obtain a new point in the phase space (q_{i+1}, p_{i+1}).

3. Compute the acceptance ratio

$$\lambda = \frac{e^{-H(q_{i+1},p_{i+1})}}{e^{-H(q_i,p_i)}}. \tag{3.19}$$

4. Perform the accept/reject test as described above and either use the new point q_{i+1} or return the old point q_i as the next sample in the generated sequence.

Figure 3.1 shows how the hybrid Monte Carlo works for this example. Note that the accept/reject step is not shown in the figure since it is not necessary for this simple example.

It may seem circular that we generate p using a direct method and use it to generate a sample of q with the same distribution. This is simply because the distribution we want for q happens to be the normal distribution. The hybrid Monte Carlo method works equally well for essentially any distribution of the following form

$$\omega(q) \propto e^{-S(q)}. \tag{3.20}$$

In addition, since the choice for $T(p)$ is always $T(p) = p^2/2$, the direct method used to generate p_i is not related to the actual form of $S(q)$.

3.4 Hamiltonian Mechanics on the Lattice

The key ingredient of the hybrid Monte Carlo algorithm is the molecular dynamics evolution of the gauge links. In this section, we discuss how it is implemented using the language of differential geometry.

To form a Hamiltonian system we need to construct a symplectic manifold (the mathematical synonym for phase space). A symplectic manifold is an even dimensional manifold M endowed with a closed 2-form ω (i.e., $d\omega = 0$) [13]. The states of the system are described by points on the manifold, which in turn are described by their local coordinates (q, p). The time development of a state following the Hamilton equations is interpreted as the movement of the corresponding point on the manifold following the integration curve of a certain vector field. This vector field (the Hamiltonian field) is derived from the Hamiltonian via the 2-form ω. It can be thought as the "gradient" of the Hamiltonian, though the analogy is not exact. ω is also used in the construction of the Poisson bracket. The Poisson bracket describes the structure of the phase space and defines the volume form in the phase space. The latter plays an essential role in the detailed balance condition. The triplet (M, ω, H) fully defines the dynamics of the system.

3.4.1 The Symplectic Manifold (M, ω)

A symplectic manifold or phase space is an even dimensional manifold M. As such a point on this manifold is described by $2N$ local coordinates. In Hamiltonian mechanics N local coordinates are interpreted as generalized coordinates and the remaining N local coordinates are interpreted as generalized momenta.

The configuration space of the link variables is the starting point to construct the phase space. A single link variable is an element of the $SU(3)$ gauge group. So the configuration space for the entire lattice is the direct product of many copies of $SU(3)$ gauge groups, one for each link

$$G = \bigotimes_{\mu x} SU(3)_{\mu x}, \tag{3.21}$$

where $\mu = 0, 1, 2, 3$ labels the spacetime directions. x labels an arbitrary site on the lattice.

A common choice to construct a phase space from a configuration space M is to use its cotangent bundle T^*M. In this approach all link variables are treated as generalized coordinates, and the generalized momenta are objects in the cotangent space. In this setup the 2-form ω has a natural construction. First of all we need to generalize the Poisson bracket

$$\{f(q,p), g(q,p)\} = \frac{\partial f}{\partial q}\frac{\partial g}{\partial p} - \frac{\partial f}{\partial p}\frac{\partial g}{\partial q} \tag{3.22}$$

so it is well defined on the cotangent bundle T^*G. The partial differential operator $\partial/\partial p$ does not need special treatment since the space for p is the flat cotangent space. However, we need to find a definition for $\partial/\partial q$ since $SU(3)$ is not flat and unlike the Euclidean space $\mathbb{R}^n$, there is no predefined coordinate system for us to introduce the partial differential operator $\partial/\partial q$.

Since vector fields can be naturally regarded as differential operators on a manifold, we define $\partial/\partial q$ as a suitable set of vector fields on the group manifold G. Most frequently, the left invariant vector field e_i is used for this purpose. e_i is defined as the push-forward of the corresponding basis vectors in the underlying Lie algebra. In other words, we pick up a basis E_i in the tangent space at the origin (the unit element of G) and translate it to all other points to form the left invariant vector field. The specific translation here is the left translation defined by the multiplication rule of the group

$$L_g : G \longrightarrow G \qquad h \longmapsto L_g h := gh. \tag{3.23}$$

And the field e_i is defined as the push-forward of the vector E_i

$$e_i(g) = L_{g*}E_i. \tag{3.24}$$

The details can be found in [13].

The field e_i has the following nice property

$$[e_i, e_j] = c_{ij}^k e_k, \tag{3.25}$$

where the coefficients c_{ij}^k are the structure constants of the corresponding Lie group. From this relation we obtain the following very useful formula (the *Maurer-Cartan formula*)

$$de^i + \frac{1}{2}c_{jk}^i e^j \wedge e^k = 0. \tag{3.26}$$

By using the covector of the left invariant vector field e^i, we can readily introduce ω as

$$\omega = d(p_i e^i). \tag{3.27}$$

ω as defined above is guaranteed to be closed since the exterior derivative operator d is nilpotent: $dd = 0$. Using the Maurer-Cartan formula, we have

$$\omega = d(p_i e^i) = dp_i \wedge e^i + p_i de^i = dp_i \wedge e^i - \frac{1}{2}p_i c_{jk}^i e^j \wedge e^k. \tag{3.28}$$

When deriving the equations of motion, it is easier to use the Poisson antisymmetric tensor $\mathcal{P}$ instead of ω. $\mathcal{P}$ can be thought as the “inverse” of ω

$$\mathcal{P} \circ \omega = -\mathbb{1},\ \mathcal{P}^{ij}\omega_{jk} = -\delta_k^i. \tag{3.29}$$

By this definition

$$\mathcal{P} = \frac{\partial}{\partial p_i} \otimes e_i - e_i \otimes \frac{\partial}{\partial p_i} + p_i c^i_{jk} \frac{\partial}{\partial p_j} \otimes \frac{\partial}{\partial p_k} = \frac{\partial}{\partial p_i} \wedge e_i + \frac{1}{2} p_i c^i_{jk} \frac{\partial}{\partial p_j} \wedge \frac{\partial}{\partial p_k}. \tag{3.30}$$

For example, the Poisson bracket of 2 scalar functions $F(U,p)$ and $G(U,p)$ is the following

$$\{F, G\} = \mathcal{P}(dF, dG) = \frac{\partial F}{\partial p_i} e_i(G) - \frac{\partial G}{\partial p_i} e_i(F) + p_i c^i_{jk} \frac{\partial F}{\partial p_j} \frac{\partial G}{\partial p_k}. \tag{3.31}$$

This concludes the definition of the Poisson bracket.

We still need a kinetic term $T(p)$ to fully construct the Hamiltonian. In effect any function $T(p)$ can be used if it is bounded from below.[1] The most common choice[2] is a quadratic form,

$$H = T(p_i) + S(U) = \frac{1}{2} p_i p^i + S(U). \tag{3.32}$$

To be exact, the number of different momenta p_i is equal to the dimension of the group G in equation (3.21). We use the convention that indices such as i, j, k represent coordinates for the entire group G, while indices such as a, b, c represent coordinates on a single link. Equivalently i can be thought as the combination of 3 indices $i = (a, \mu, x)$.

Now that we have a complete Hamiltonian system, we can build the equations of motion for the system. By definition the Hamiltonian H gives rise to the following vector field called the *Hamiltonian field* ζ_H,

$$\zeta_H = \mathcal{P}(dH, \,\cdot\,) = p^i e_i - e_i(S) \frac{\partial}{\partial p_i}. \tag{3.33}$$

The Hamilton equations describe the evolution of the state in the phase space along the integration curve of the Hamiltonian field. So the time evolution of an observable $F(U,p)$ is

[1] One restriction apply: the choice of $T(p)$ has to make sure that the algorithm is ergodic. For example, setting $T(p) = 0$ will not produce a usable algorithm.

[2] Other choices do exist. For example see [14].

given by

$$\frac{dF}{dt} = \zeta_H F = \{H, F\}\,. \tag{3.34}$$

Treating the coordinates $(U_\mu(x), p_i)$ themselves as observables, we obtain the standard Hamilton equations

$$\begin{aligned}\frac{dU_\mu(x)}{dt} =& \{H, U_\mu(x)\} = \sum_a p^a_{\mu x} T_a U_\mu(x), & (3.35)\\ \frac{dp_i}{dt} =& -\, e_i(S). & (3.36)\end{aligned}$$

Where T_i is the generator associated with e_i in its fundamental representation.

Treating T, S or $\{S, \{S, T\}\}$ as Hamiltonians, their corresponding Hamiltonian fields are

$$\begin{aligned}\zeta_T =& \mathcal{P}(dT,\ \cdot\ ) = p^i e_i & (3.37)\\ \zeta_S =& \mathcal{P}(dS,\ \cdot\ ) = -e_i(S)\frac{\partial}{\partial p_i} & (3.38)\\ \zeta_{\{S\{S,T\}\}} =& \mathcal{P}(d\{S\{S,T\}\},\ \cdot\ ) = -2e^j(S)e_i e_j(S)\frac{\partial}{\partial p_i} & (3.39)\end{aligned}$$

As can be seen below, ζ_T and ζ_S are natural building blocks for symmetric symplectic integrators. $\zeta_{\{S\{S,T\}\}}$ plays an essential role in the force gradient integrator.

3.5 Symmetric Symplectic Integrators

The integration curve of the vector field (3.33) is obtained by solving the Hamilton equations

$$\begin{aligned}\frac{dU_\mu(x)}{dt} =& p^a_\mu(x) T_a U_\mu(x),\\ \frac{dp^a_\mu(x)}{dt} =& -\, e^a_{\mu x}(S).\end{aligned} \tag{3.40}$$

Exact numeric integration of the above equations is generally very difficult. Since the corresponding time evolution operator

$$e^{\tau\zeta_H} = e^{\tau(\zeta_T+\zeta_S)} \tag{3.41}$$

involves non-commuting operators ζ_T and ζ_S,

$$\zeta_T = \mathcal{P}(dT,\ \cdot\ ) = p^i e_i \tag{3.42}$$

$$\zeta_S = \mathcal{P}(dS,\ \cdot\ ) = -e_i(S)\frac{\partial}{\partial p_i}. \tag{3.43}$$

Various numerical methods (integrators) can be applied to the above equations. In principle all numerical methods that solve partial differential equations can be used.

For use with the hybrid Monte Carlo algorithm, the integrator has to satisfy the detailed balance condition. This ensures that the subsequent accept/reject step corrects any numerical errors introduced by integrating equations (3.40). This is usually fulfilled by two additional properties of the integrator, namely symplecticity and reversibility[3].

1. A symplectic integrator preserves the volume of a domain while evolving the domain on the symplectic manifold. In other words, the time evolution according to a symplectic integrator can be interpreted as an incompressible flow. In particular, Liouville's theorem guarantees that the time evolution operator of any Hamiltonian preserves the phase volume [13].

2. A reversible integrator can move precisely from the final point to the initial point when integrated backwards. As an simple example, the linear multistep method is *not* reversible because it refers to its history when moving forward.

Integrators that are both symplectic and reversible are called symmetric symplectic integrators. such integrators can be conveniently constructed using the Baker-Campbell-Hausdorff

[3] Being symplectic and reversible is sufficient but not required for the detailed balance condition. However, this is usually the easiest way to enforce the detailed balance condition.

(BCH) formula. The BCH formula is an expansion of

$$\log\left(e^{A}e^{B}\right) \tag{3.44}$$

while A and B are elements of some Lie algebra. The initial few terms of the formula are as follows

$$\log\left(e^{A}e^{B}\right) = A + B + \frac{1}{2}\left[A,B\right] + \frac{1}{12}\left[A,\left[A,B\right]\right] - \frac{1}{12}\left[B,\left[A,B\right]\right] + \cdots . \tag{3.45}$$

Using this formula, it is immediately clear that

$$U(\tau) = e^{\tau\zeta_T}e^{\tau\zeta_S} \tag{3.46}$$

is an approximation to the full time evolution operator (3.41) with local error equal to $\mathcal{O}(\tau^2)$. Since

$$\log\left(e^{\tau\zeta_T}e^{\tau\zeta_S}\right) = \tau\zeta_H + \tau^2\left[\zeta_T,\zeta_S\right]/2 + \mathcal{O}(\tau^3). \tag{3.47}$$

In addition, $U(\tau)$ is symplectic since $e^{\tau\zeta_T}$ and $e^{\tau\zeta_S}$ are time evolution operators corresponding to Hamiltonians T and S.

The above integrator is the simplest symplectic integrator one can construct. Unfortunately, it is not reversible

$$U(\tau)U(-\tau) = e^{\tau\zeta_T}e^{\tau\zeta_S}e^{-\tau\zeta_T}e^{-\tau\zeta_S} \neq 1. \tag{3.48}$$

This problem can be cured by making the integrator symmetric in 2 ways

$$U_{\mathrm{QPQ}}(\tau) = e^{\tau\zeta_T/2}e^{\tau\zeta_S}e^{\tau\zeta_T/2}, \tag{3.49}$$

$$U_{\mathrm{PQP}}(\tau) = e^{\tau\zeta_S/2}e^{\tau\zeta_T}e^{\tau\zeta_S/2}. \tag{3.50}$$

Both integrators are reversible,

$$U_{\mathrm{QPQ}}(\tau)U_{\mathrm{QPQ}}(-\tau) = 1, \tag{3.51}$$

$$U_{\mathrm{PQP}}(\tau)U_{\mathrm{PQP}}(-\tau) = 1. \tag{3.52}$$

An additional benefit is that any $\mathcal{O}(\tau^{2n})$ local errors are eliminated. For example, by using the BCH formula it can be shown that

$$U_{\mathrm{QPQ}}(\tau) = \exp\left(\tau\left(\zeta_T + \zeta_S\right) + \mathcal{O}(\tau^3)\right). \tag{3.53}$$

So the leading term in the local error is $\mathcal{O}(\tau^3)$. These are the leapfrog integrators.

By adding more alternating steps one can construct progressively more sophisticated integrators. The popular Omelyan integrator has 5 integration steps

$$U_{\mathrm{QPQPQ}}(\tau) = e^{\alpha\tau\zeta_T} e^{\frac{1}{2}\tau\zeta_S} e^{(1-2\alpha)\tau\zeta_T} e^{\frac{1}{2}\tau\zeta_S} e^{\alpha\tau\zeta_T}, \tag{3.54}$$

where α is a free parameter. We can obtain its error terms by using the BCH formula

$$\begin{aligned} &\log\left(U_{\mathrm{QPQPQ}}(\tau)\right) \\ =&\tau\left(\zeta_S + \zeta_T\right) + \tau^3\left(\frac{6\alpha - 1}{24}\zeta_{\{S,\{S,T\}\}} + \frac{-6\alpha^2 + 6\alpha - 1}{12}\zeta_{\{T,\{S,T\}\}}\right) + \mathcal{O}\left(\tau^5\right). \end{aligned} \tag{3.55}$$

If $\alpha = 0$ or 1 then this integrator becomes the above mentioned leapfrog integrators. The Omelyan integrator has $\mathcal{O}\left(\tau^3\right)$ local errors that are formally the same size as the leapfrog integrator. However, the presence of the free parameter α allows one to tune the integrator for a specific problem to minimize the $\mathcal{O}\left(\tau^3\right)$ errors.

By adding more alternating time evolution operators we can introduce more free parameters into the integrators, thus eliminating more error terms. For example, it is possible to eliminate both $\mathcal{O}\left(\tau^3\right)$ terms in (3.55) by integrating over the vector field ζ_T and ζ_S alternately in 7 steps. We thus obtain the Campostrini integrator [15]

$$U_{\mathrm{QPQPQPQ}}(\tau) = e^{C_1\tau\zeta_T} e^{C_2\tau\zeta_S} e^{C_3\tau\zeta_T} e^{C_4\tau\zeta_S} e^{C_3\tau\zeta_T} e^{C_2\tau\zeta_S} e^{C_1\tau\zeta_T}, \tag{3.56}$$

where

$$\begin{aligned} C_1 =& \frac{\sqrt[3]{4} + 2\sqrt[3]{2} + 4}{12},\ C_2 = \frac{\sqrt[3]{4} + 2\sqrt[3]{2} + 4}{6}, \\ C_3 =& \frac{-\sqrt[3]{4} - 2\sqrt[3]{2} + 2}{12},\ C_4 = -\frac{\sqrt[3]{4} + 2\sqrt[3]{2} + 1}{3}. \end{aligned} \tag{3.57}$$

This integrator has the following error terms (cf. [16])

$$\begin{aligned} \log\left(U_{\mathrm{QPQPQPQ}}(\tau)\right) = \tau\left(\zeta_S + \zeta_T\right) + \frac{\tau^5}{34560} \Big(& -\left(48 + 40\sqrt[3]{2} + 40\sqrt[3]{4}\right) \zeta_{\{S,\{S,\{S,\{S,T\}\}\}\}} \\ & + \left(8 - 20\sqrt[3]{4}\right) \zeta_{\{S,\{S,\{T,\{S,T\}\}\}\}} \\ & + \left(32 + 20\sqrt[3]{2}\right) \zeta_{\{S,\{T,\{T,\{S,T\}\}\}\}} \\ & + \left(8 + 5\sqrt[3]{2}\right) \zeta_{\{T,\{T,\{T,\{S,T\}\}\}\}} \\ & + \left(304 + 240\sqrt[3]{2} + 200\sqrt[3]{4}\right) \zeta_{\{\{S,T\},\{S,\{S,T\}\}\}} \\ & + \left(72 + 60\sqrt[3]{2} + 60\sqrt[3]{4}\right) \zeta_{\{\{S,T\},\{T,\{S,T\}\}\}} \Big). \end{aligned} \tag{3.58}$$

Theoretically one can construct integrators of arbitrarily high orders using this approach. The following section illustrates a simple constructive way to do this.

3.5.1 Triplet Concatenations

Triplet concatenation [17] is an interesting method to generate an $\mathcal{O}(\tau^{n+2})$ integrator from an arbitrary $\mathcal{O}(\tau^n)$ symmetric symplectic integrator. Integrators of arbitrarily high orders can be constructed by applying this method recursively.

These integrators would be very useful if the goal of the calculation was to keep the error of the overall energy (on the lattice, the Hamiltonian) very small. However, unlike many other numerical applications, in a typical lattice calculation the goal is *not* to keep the error as small as possible but rather to keep it as an $\mathcal{O}(1)$ quantity[4]. In such cases, this class of integrators is less stable compared to previously mentioned integrators due to their use of large negative time steps. Nevertheless, we shall describe the method here since it is simple and a few notable integrators such as the fourth order Campostrini integrator (3.56) can be easily constructed in this way.

Starting from an nth order integrator

$$U_n(\tau) = e^{\tau\zeta_H + \tau^{n+1}\zeta_F + \mathcal{O}(\tau^{n+3})}, \tag{3.59}$$

where $\tau^{n+1}\zeta_F$ is the local error term, we try to construct a higher order integrator by the following ansatz

$$U_n(\tau)U_n(-\lambda\tau)U_n(\tau). \tag{3.60}$$

By using the BCH formula, it is easy to prove that [5]

$$U_n(\tau)U_n(-\lambda\tau)U_n(\tau) = e^{\tau(2-\lambda)\zeta_H + (2-\lambda^{n+1})\tau^{n+1}\zeta_F + \mathcal{O}(\tau^{n+3})}. \tag{3.61}$$

So choosing $2 - \lambda^{n+1} = 0$ yields a higher order integrator. In case of symmetric symplectic

[4] If the error is much larger than 1, then the acceptance ratio (3.19) will be very small and we move the state in the phase space very slowly. On the other hand, there is not much difference in the computed ensemble if the acceptance ratio is close to 1. The HMC method with 90% acceptance ratio has just slightly larger autocorrelation time than that with 99% (or any higher value) acceptance ratio. So it is usually preferable to use a cheaper integrator as long as it maintains reasonable acceptance ratio.

[5] One may raise the question that why $U_n(\tau)U_n(-\lambda\tau)U_n(\tau) \neq U_n((2-\lambda)\tau)$. This is because $U_n(\tau)$ does not commute with $U_n(-\lambda\tau)$. For a “normal” time evolution operator $e^{\tau\zeta_H}$, ζ_H does not depend on τ and $e^{\tau\zeta_H}$ commutes with $e^{-\lambda\tau\zeta_H}$ for any λ. This is not the case for $U_n(\tau)$.

integrators, we get an integrator of order $n+2$ from an integrator of order n.

$$U_n(\tau)U_n(-2^{1/(n+1)}\tau)U_n(\tau) = e^{\tau(2-2^{1/(n+1)})\zeta_H + \mathcal{O}(\tau^{n+3})}. \tag{3.62}$$

After normalizing the step size, the higher order integrator can be written as

$$V(\tau) = U_n\left(\frac{1}{2-\lambda}\tau\right) U_n\left(-\frac{\lambda}{2-\lambda}\tau\right) U_n\left(\frac{1}{2-\lambda}\tau\right) \tag{3.63}$$

with $\lambda = 2^{1/(n+1)}$.

As a simple example, the 4th order Campostrini integrator (3.56) can be constructed from the leapfrog integrator (3.53) in this way. By setting $\lambda = \sqrt[3]{2}$, we have

$$U_{\text{QPQ}}\left(\frac{1}{2-\sqrt[3]{2}}\tau\right) U_{\text{QPQ}}\left(-\frac{\sqrt[3]{2}}{2-\sqrt[3]{2}}\tau\right) U_{\text{QPQ}}\left(\frac{1}{2-\sqrt[3]{2}}\tau\right) = U_{\text{QPQPQPQ}}(\tau). \tag{3.64}$$

Integrators obtained in this way relies heavily on cancellations between forward steps of size $\tau/(2-\lambda)$ and a backward step of size $-\lambda\tau/(2-\lambda)$ to eliminate the $\mathcal{O}(\tau^{n+1})$ error. In the large n limit, both forward and backward steps have their step sizes approaching 1. When applying this class of integrators to lattice Hamiltonian systems we frequently find that the use of backward steps render the integrator less stable in the region where the error ΔH is an order 1 quantity. So they do not necessarily outperform their underlying integrators.

3.6 The Force Gradient Integrator

Both the leapfrog integrator and the Omelyan integrator are second order integrators. For both integrators, the global error scales according to $\mathcal{O}(\tau^2)$. The Campostrini integrator is an $\mathcal{O}(\tau^4)$ integrator. However, practice shows that the large coefficients in the error terms of the Campostrini integrator (3.58) and the extra integration costs usually offset the benefit of this $\mathcal{O}(\tau^4)$ behavior in the errors, for reasons described in the previous section.

It is also possible to eliminate the $\mathcal{O}(\tau^3)$ terms using an alternative approach. This approach exploits the specific form (3.32) we chose for the kinetic term $T(p)$. To see the trick, we substitute $T(p) = p_i p^i/2$ into the 2 error terms of the Omelyan integrator (3.55)

$$\{S, \{S, T\}\} = e^i(S) e_i(S) \tag{3.65}$$

$$\{T, \{S, T\}\} = -\, p^i p^j e_i e_j(S). \tag{3.66}$$

An important observation is that $\{S, \{S, T\}\}$ does not depend on the canonical momentum p^i. Since it is a function of $U_\mu(x)$, it can be treated as a potential. Consequently its time evolution operator evolves only the canonical momentum and can thus be evaluated exactly for any given finite time step.

So it is possible to eliminate both $\mathcal{O}(\tau^3)$ error terms (3.55) in the Omelyan integrator without introducing additional alternating steps,

1. The term $\zeta_{\{T,\{S,T\}\}}$ can be eliminated by tuning α in the Omelyan integrator. For example, setting $\alpha = (3 - \sqrt{3})/6$ in (3.55) eliminates this error term in the Omelyan QPQPQ integrator.

2. The term $\zeta_{\{S,\{S,T\}\}}$ can be eliminated by incorporating an appropriate time evolution operator into the integrator. This extra time evolution operator uses the Hamiltonian field of $\{S, \{S, T\}\}$.

The resulting integrator takes the following 2 forms [16],

$$\begin{aligned} U_{\text{FGI-QPQPQ}}(\tau) = &\exp\left(\frac{3-\sqrt{3}}{6}\tau\zeta_T\right) \exp\left(\frac{1}{2}\tau\zeta_S - \frac{2-\sqrt{3}}{48}\tau^3\zeta_{\{S,\{S,T\}\}}\right) \cdot \\ &\exp\left(\frac{\sqrt{3}}{3}\tau\zeta_T\right) \exp\left(\frac{1}{2}\tau\zeta_S - \frac{2-\sqrt{3}}{48}\tau^3\zeta_{\{S,\{S,T\}\}}\right) \exp\left(\frac{3-\sqrt{3}}{6}\tau\zeta_T\right), \end{aligned} \tag{3.67}$$

$$\begin{aligned}U_{\text{FGI-PQPQP}}(\tau) =& \exp\left(\frac{1}{6}\tau\zeta_S\right)\exp\left(\frac{1}{2}\tau\zeta_T\right)\cdot \\ &\exp\left(\frac{2}{3}\tau\zeta_S - \frac{1}{72}\tau^3\zeta_{\{S,\{S,T\}\}}\right)\exp\left(\frac{1}{2}\tau\zeta_T\right)\exp\left(\frac{1}{6}\tau\zeta_S\right).\end{aligned} \tag{3.68}$$

For both integrators, the global errors scale as $\mathcal{O}\left(\tau^4\right)$. As will be seen below, evaluating the force gradient term $\zeta_{\{S,\{S,T\}\}}$ for fermions requires solving one extra Dirac equation. Consequently we need to solve 1 additional Dirac equation if we apply the PQPQP force gradient integrator to fermions and 2 additional Dirac equations if the QPQPQ form is used instead.

As an explicit example, the force gradient QPQPQ integrator thus involves the following 5 steps

$$U_\mu(x) \longleftarrow \exp\left(\frac{3-\sqrt{3}}{6}\tau p_\mu^a(x)T_a\right)U_\mu(x), \tag{3.69}$$

$$p_\mu^a(x)T_a \longleftarrow p_\mu^a(x)T_a - \frac{1}{2}\tau e_{\mu x}^a(S)T_a + \frac{2-\sqrt{3}}{24}\tau^3 e^j(S)e_{\mu x}^a e_j(S)T_a, \tag{3.70}$$

$$U_\mu(x) \longleftarrow \exp\left(\frac{\sqrt{3}}{3}\tau p_\mu^a(x)T_a\right)U_\mu(x), \tag{3.71}$$

$$p_\mu^a(x)T_a \longleftarrow p_\mu^a(x)T_a - \frac{1}{2}\tau e_{\mu x}^a(S)T_a + \frac{2-\sqrt{3}}{24}\tau^3 e^j(S)e_{\mu x}^a e_j(S)T_a, \tag{3.72}$$

$$U_\mu(x) \longleftarrow \exp\left(\frac{3-\sqrt{3}}{6}\tau p_\mu^a(x)T_a\right)U_\mu(x). \tag{3.73}$$

3.6.1 Implementing the Force Gradient Integrator

We use the PQPQP form of the force gradient integrator (3.68) as an example when discussing its implementations. Most of the discussion extends trivially to the force gradient QPQPQ integrator (3.67).

The key step when implementing the FGI-PQPQP integrator involves the following update procedure for the momentum field

$$p_i \longleftarrow p_i - \frac{2}{3}\tau e_i(S) + \frac{1}{36}\tau^3 e^j(S)e_ie_j(S). \tag{3.74}$$

For reference purpose we will refer such a step as the force gradient step.

The naive implementation of the force gradient step is quite complicated, since it requires evaluating the force twice and the second order differential $e_je_i(S)$ generates many individual force terms. This is especially the case when the force gradient integrator is used inside a nested integration scheme. In addition, the fermion force gradient step generally requires solving more Dirac equations than a usual fermion force term, increasing the cost substantially. Fortunately there is a simpler approach.

This approach has a close relation to the well known Horner scheme for evaluating polynomials. Take a quadratic function

$$f(x) = a + bx + cx^2 \tag{3.75}$$

with known coefficients a,b and c as an example. Horner scheme replaces evaluation of each power of x by applying a linear function of x in a nested manner:

$$f(x) = a + x(b + cx). \tag{3.76}$$

So evaluating the linear function of x twice is equivalent to evaluating a quadratic function.

Exactly the same idea can be used to implement the second order derivative term $e^j(S)e_je_i(S)$. As the first step we notice that

$$e_ie_j - e_je_i = [e_i, e_j] = c_{ij}^k e_k, \tag{3.77}$$

so

$$e^j(S)e_ie_j(S) - e^j(S)e_je_i(S) = e^j(S)c_{ij}^k e_k(S) = c_{ijk}e^j(S)e^k(S) = 0. \tag{3.78}$$

The last equality holds since c_{ijk} is antisymmetric with respect to interchanging any 2 indices. This implies we are free to exchange the order of the 2 derivatives e_i, e_j in $e^j(S)e_je_i(S)$.

Since e_i and e_j are interchangeable, the following approximation holds

$$\frac{2}{3}\tau e_i(S) - \frac{1}{36}\tau^3 e^j(S)e_ie_j(S) = \frac{2}{3}\tau\left(1 - \frac{1}{24}\tau^2 F^j e_j\right)e_i(S) = \frac{2}{3}\tau e^{-\frac{1}{24}\tau^2 F^j e_j}e_i(S) + \mathcal{O}(\tau^5). \tag{3.79}$$

Where $F^j = e^j(S)$ and it must be treated as a field independent of link variables in the above equation. In other words, we define $e_i(F_j) = 0$ despite the fact that F^j depends on the gauge field. This sequence of transformation preserve the important feature that the local error of the force gradient integrator is of the order $\mathcal{O}(\tau^5)$.

Effectively we obtain from the last expression

$$e^{-\frac{\tau^2}{24}F^j e_j}e_i(S) = e_i(S[U']) \tag{3.80}$$

with

$$U' = e^{-\frac{\tau^2}{24}F^j e_j}U = e^{-\frac{\tau^2}{24}F^j T_j}U \tag{3.81}$$

So the force gradient step can be integrated using the following method

1. (the preparation step) Calculate $F^j = e^j(S[U])$ and compute a set of auxiliary link variables $U' = e^{-\frac{\tau^2}{24}F^j T_j}U$.

2. (the update step) Calculate again $2\tau e_i(S[U'])/3$ using the auxiliary gauge field. The latter calculation is to be added to the momentum field.

The above implementation of the force gradient integrator differs from the original force gradient integrator (3.68) [16], since the approximation in (3.79), though being $\mathcal{O}(\tau^5)$ locally, changes the error term. This approximation still preserves the symmetricity and symplecticity of the original force gradient integrator.

1. It is symmetric since the force gradient step depends only on the current gauge field. The preparation step does not update the state of the system, but rather computes an auxiliary field. If we reverse the molecular dynamics time τ, then preparation step computes the same U' and the update step then computes a force term that is reversed.

2. Since the update step can be thought as using the time evolution operator $\exp(2\tau\zeta_S/3)$ computed on the field U', Liouville's theorem guarantees that it is symplectic.

Here we present the scaling test of the integrator on the $16^3 \times 32$ (1.8fm, 420MeV) lattice with 2+1 flavor dynamical DWF fermions. With the PQPQP force gradient integrator, we were able to raise the top level step size from 1/4 as in the Omelyan to 1/3[6]. However, due to the extra cost in the extra force gradient solve, the use of the force gradient integrator by itself is not enough for a significant speed up.

3.6.2 The Forecasted Force Gradient Integrator

For fermion actions each time $e_i(S)$ is evaluated we need to solve a Dirac equation with the preconditioned Dirac operator, as can be seen in appendix A. This part of calculation can easily consume a large portion of the computation time in the hybrid Monte Carlo algorithm. The situation for the force gradient step is even worse since we need to solve more (typically, twice as many) Dirac equations to evaluate the second order force gradient.

However, for the above implementation of the force gradient integrator it is possible to use the solution from the preparation step as an initial guess when solving the Dirac equation in the update step. Consider a simple fermion action $S(U) = \phi^\dagger \mathcal{M}(U)^{-1}\phi$, its force term

[6] Without further notice, the full trajectory length is always 1 in all the tests performed.

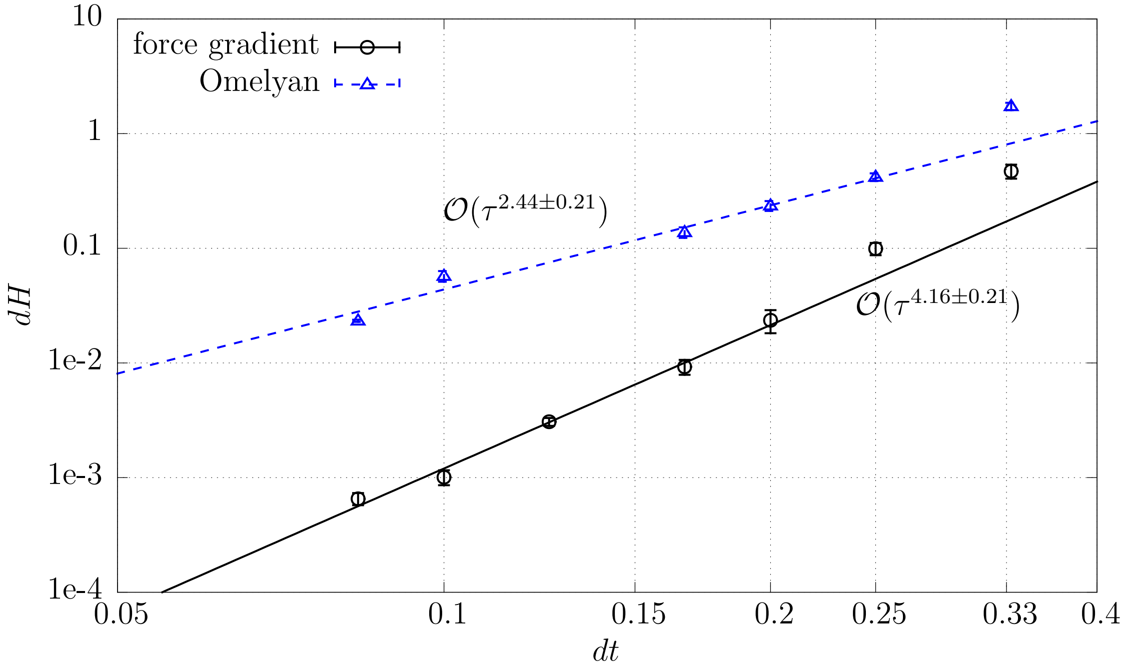


Figure 3.2: The absolute value of the error dH as a function of the step size τ, plotted for both the force gradient integrator and the Omelyan integrator. We integrate for a total of 1 molecular dynamics time unit so we are effectively comparing the global behavior of both integrators. Data are obtained on the $16^3 \times 32$ $(1.8\text{fm}, 420\text{MeV})$ ensemble. We fit both data sets using the ansatz $f(x) = Ax^{\beta}$, where data points with $dt \leq 0.2$ are used. The result shows that $\beta = 4.16(21)$ for the force gradient integrator and 2.44(21) for the Omelyan integrator.

takes the following form

$$e_i(S(U)) = -\phi^\dagger \mathcal{M}(U)^{-1} e_i(\mathcal{M}(U)) \mathcal{M}(U)^{-1}\phi. \tag{3.82}$$

We need to solve the linear system $\mathcal{M}(U)\psi = \phi$ to evaluate this force term. Two such solves are needed to implement the force gradient step (3.74) due the presence of the second order derivative. However, the above implementation of the force gradient integrator features 2 almost identical steps. The only difference in the 2 steps is in the gauge field, the first step uses the original field U while the second step uses U' computed from the first step. Notice that U' is only slightly different from U, we can simply use the solution obtained in the first step as an initial guess when solving the Dirac equation in the second step. The difference between U' and U is proportional to $\tau^2/24$. In practice this factor is usually small despite a relative large step size, making the forecasting very effective.

The force gradient integrator with forecasting is still a symmetric symplectic integrator. As mentioned in the previous text, the molecular dynamics time does not evolve within the two steps. Unlike the chronological inverter, the forecasting always produces exactly the same solution in the update step regardless of the direction of the molecular dynamics time.

Figure (3.3) shows the difference in the Hamiltonian when we perform a forward integration followed by a backward integration with the MD time equal to 1 in both steps. In other words, we are applying the following operator to the system,

$$U(-1)U(1). \tag{3.83}$$

If there is no numerical error and the integrator is symmetric then we should reach exactly the same state as where we started, thus any deviation from zero provides a measure for how far we are from being perfectly reversible. The figure shows the behavior of the Omelyan integrator and the force gradient integrator with/without the forecasting. Clearly there is

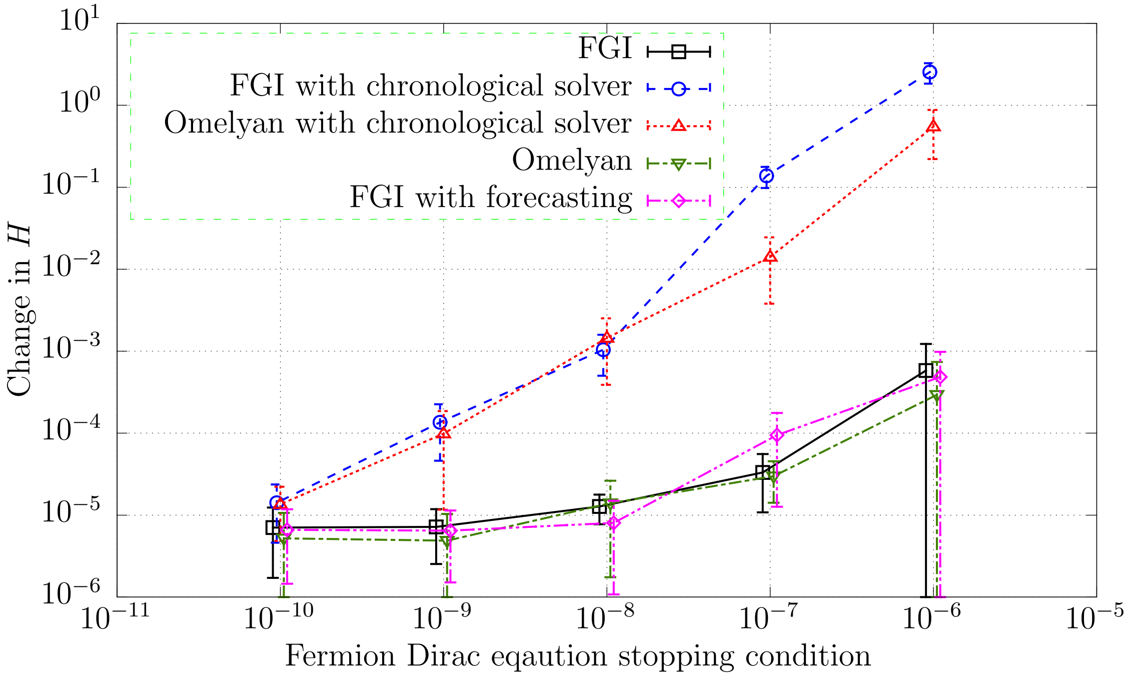


Figure 3.3: Reversibility of different integration strategies plotted as a function of different stopping conditions used in molecular dynamics evolution. The data were obtained on ensemble $16^3 \times 32$ $(1.8\text{fm}, 420\text{MeV})$. “Change in H” is the change in the Hamiltonian when we perform a forward integration followed immediately by a backward integration, both for 1 molecular dynamics time unit. The graph shows the absolute value of this difference, plotted as average value/standard deviation pairs.

no appreciable difference between the Omelyan integrator and the force gradient integrator, either with or without forecasting. For reference purposes we also show the behavior of these integrators combined with the chronological solver[7]. Clearly integrators which make use of its history are more susceptible to changes in stopping condition.

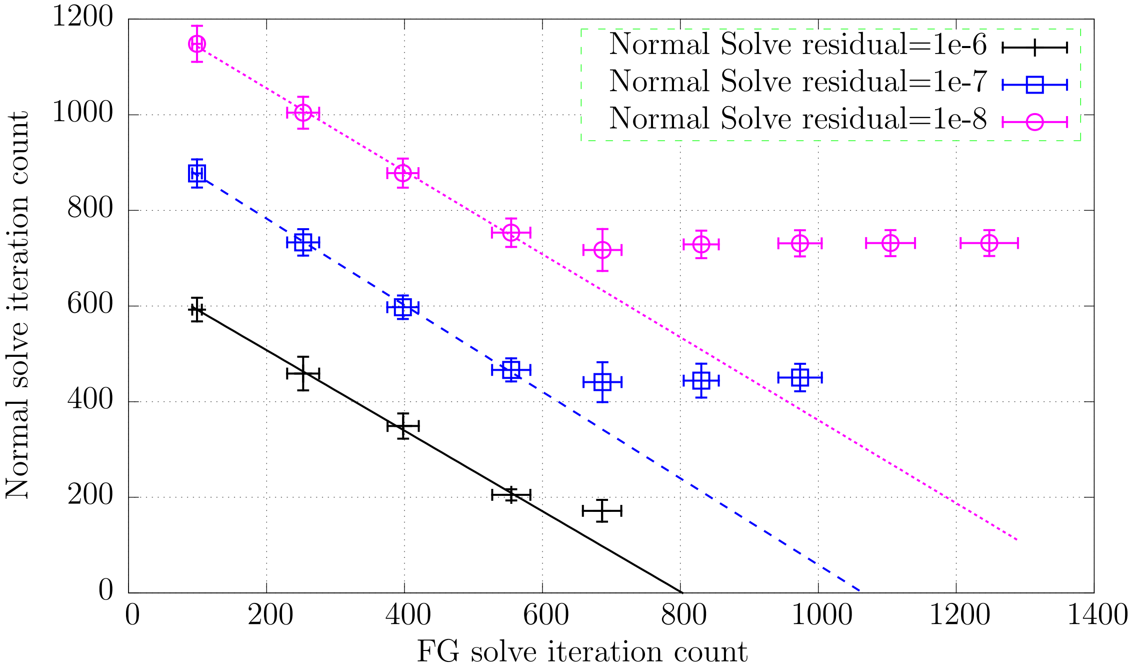


Figure 3.4: The iteration count of the Krylov space solver in the update step as a function of the iteration count of the solver in the preparation step.

Practically the effectiveness of the force gradient forecasting is restricted by the difference between U and U', and also the accuracy of the force gradient step itself. Figure (3.4) shows the iteration count of the Krylov space solver (in this case, the conjugate gradient method) in the update step as a function of the iteration count of the same solver in the preparation step. The graph suggests that as we solve the Dirac equation in the preparation step more accurately, force gradient forecasting becomes more effective. There is also a turning point associated with each curve. If we solve the Dirac equation in the preparation step more

[7] The idea of chronological solver is to use linear combinations of the past solutions of fermion Dirac equations as an initial guess when solving new fermion Dirac equations in HMC. Further details can be found in [18].

accurately than indicated by the turning point, the cost of the second Krylov space solver remains to be a constant.

With the help of the force gradient forecasting, the force gradient integrator starts to outperform the Omelyan integrator on large ensembles. In table (3.1) we compare the forecasted force gradient integrator with the Omelyan integrator on a production $48^3 \times 96$, $\beta = 2.25$ ensemble[8]. Using our settings the 2 integrators finish a trajectory within roughly the same amount of computer time. However, the errors in the forecasted force gradient integrator are a factor of 3 smaller than the corresponding Omelyan integrator, allowing a better acceptance ratio.

integrator	$\langle \lvert \Delta H \rvert \rangle$	acceptance ratio	time per trajectory (seconds)	total number of configurations
forecasted FGI	0.30(2)	0.85(18)	5.1e3	105
Omelyan($\alpha = 0.22$)	0.99(13)	0.58(6)	5.3e3	30

Table 3.1: Comparison between the forecasted force gradient integrator and the Omelyan integrator on the $48^3 \times 96$, $\beta = 2.25$ ensemble. The full trajectory length is 2 for both cases.

3.7 Sexton-Weingarten Integration

In practice the Hamiltonian (3.32) involves both the gauge action and the fermion action(s)

$$H = T(p) + S_{\mathrm{G}}(U) + S_{\mathrm{F}}(U). \tag{3.84}$$

A practical problem arises when both actions are involved. The gauge force is usually larger than the fermion force by an order of magnitude, while much easier to evaluate. Furthermore, the fermion action $S_{\mathrm{F}}(U)$ itself can usually be divided into separate parts with different costs and magnitude of forces. If both the gauge action and various parts of the fermion action are integrated in the same step then the step size τ has to be chosen to accommodate the largest

[8] This is *not* the ensemble $48^3 \times 96$ $(5.5\text{fm}, 140\text{MeV})$ listed in appendix B.

force among them. This usually means small yet difficult to calculate force terms have to be computed more frequently than necessary, introducing unnecessary overhead. Consequently it is desirable to have a method to integrate different actions using different step sizes.

The Sexton-Weingarten integration scheme can be used to mitigate this problem [19]. Define

$$H = T' + S_{\mathrm{F}}(U), \tag{3.85}$$

$$T' = T(p) + S_{\mathrm{G}}(U). \tag{3.86}$$

T' and $S_{\mathrm{F}}(U)$ can be fit into one integrator. When integrating T', its 2 parts $T(p)$ and $S_{\mathrm{G}}(U)$ can be fit into a separate integrator. For example, when using the leapfrog QPQ integrator for both levels one has the following integration scheme

$$\exp\left(\tau\zeta_H\right) \approx \exp\left(\frac{1}{2}\tau\zeta_{T'}\right)\exp\left(\tau\zeta_{S_{\mathrm{F}}}\right)\exp\left(\frac{1}{2}\tau\zeta_{T'}\right) \tag{3.87}$$

$$\exp\left(\frac{1}{2}\tau\zeta_{T'}\right) \approx \left(\exp\left(\frac{1}{4n}\tau\zeta_T\right)\exp\left(\frac{1}{2n}\tau\zeta_{S_{\mathrm{G}}}\right)\exp\left(\frac{1}{4n}\tau\zeta_T\right)\right)^n, \tag{3.88}$$

where n can be chosen as any positive integer. In this way different time steps can be assigned to $S_{\mathrm{G}}(U)$ and $S_{\mathrm{F}}(U)$. n can be tuned to balance the size of the force and the cost, maximizing the effectiveness of the composite integrator.

3.8 Hasenbusch Mass Splitting

The Hasenbusch mass splitting [20] is yet another method that offers fine grained control to distribute the fermion forces to different parts in the action. Urbach *et al.* [21] proposed that by assigning a smaller force to the more expensive part and putting it on a coarser time scale using a multilevel integrator, an acceleration of the molecular dynamics evolution can be achieved.

The fermion action is derived from the following fermion determinant

$$\det\left(\frac{\mathcal{M}^\dagger(m)\mathcal{M}(m)}{\mathcal{M}^\dagger(1)\mathcal{M}(1)}\right) = \int \mathcal{D}\phi^\dagger\mathcal{D}\phi \;\exp\left(-\phi^\dagger\mathcal{M}(1)\frac{1}{\mathcal{M}^\dagger(m)\mathcal{M}(m)}\mathcal{M}^\dagger(1)\phi\right). \tag{3.89}$$

Using the Hasenbusch factorization [20], we rewrite the above quotient action as a product of quotient actions by introducing a series of intermediate masses

$$\det\left(\frac{\mathcal{M}^\dagger(m)\mathcal{M}(m)}{\mathcal{M}^\dagger(1)\mathcal{M}(1)}\right) = \prod_{i=1}^{k+1}\det\left(\frac{\mathcal{M}^\dagger(m_{i-1})\mathcal{M}(m_{i-1})}{\mathcal{M}^\dagger(m_i)\mathcal{M}(m_i)}\right) \tag{3.90}$$

$$=\prod_{i=1}^{k+1}\int \mathcal{D}\phi_i^\dagger\mathcal{D}\phi_i \;\exp\left(-\phi_i^\dagger\mathcal{M}(m_i)\frac{1}{\mathcal{M}^\dagger(m_{i-1})\mathcal{M}(m_{i-1})}\mathcal{M}^\dagger(m_i)\phi_i\right), \tag{3.91}$$

where $m = m_0 < m_1 < \cdots < m_{k+1} = 1$.

The size of the fermion force generated by the action

$$S_{\mathrm{Q}}(m_{i-1}, m_i) = -\phi_i^\dagger\mathcal{M}(m_i)\frac{1}{\mathcal{M}^\dagger(m_{i-1})\mathcal{M}(m_{i-1})}\mathcal{M}^\dagger(m_i)\phi_i \tag{3.92}$$

is controlled by the mass parameters m_{i-1} and m_i. In general closer masses produce smaller forces. Thus by tuning these masses we can tune the size of the fermion forces in each part continuously.

3.8.1 Tuning the Intermediate Masses

Tuning the intermediate masses proves to be challenging in this strategy. Empirically, we found that the L^∞ norm of the force (the largest force across the lattice) tends to be strongly correlated with the acceptance ratio. In practice, this means comparing the L^∞ norm of each quotient action while doing the tuning work. Fig. 3.5 shows the L^∞ norm distribution of each part of the action from both the original Omelyan integration scheme and the tuned scheme, the data was collected from a few hundred trajectories for each scheme.

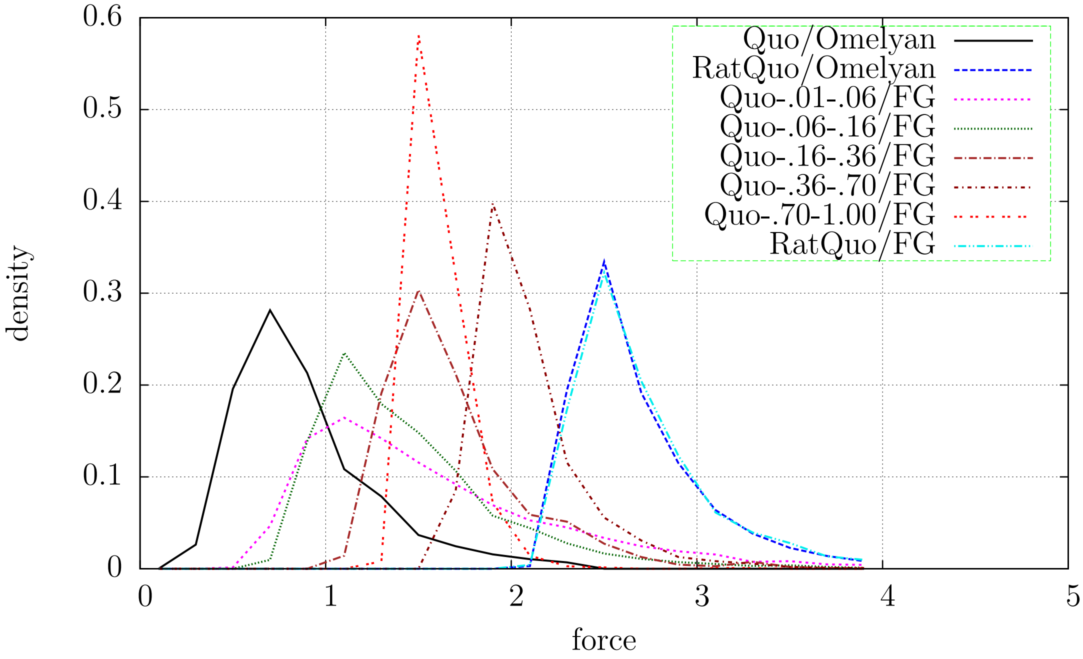


Figure 3.5: Force distribution

We found that by making all the quotient actions at the same level with roughly the same average value of the L^∞, the whole integrator has a generally better acceptance ratio. We believe this shows L^∞ norm can be used as an indicator when tuning intermediate masses. In addition, keeping a smaller L^∞ norm for the force in the light quotient sector seems to be better than keeping them all equal. One possible explanation is that the L^∞ distribution of the light quotient force tends to be more spread out compared to the heavy ones, as seen from figure 3.5.

3.9 Practical Integration Schemes

By combining previously mentioned methods, we can rewrite the Hamiltonian used for 2+1 flavor lattice QCD simulation in the following manner

$$H = T + S_{\mathrm{G}} + \sum_{a=1}^{k} S_{\mathrm{Q}}(m_{a-1}, m_a) + S_{\mathrm{R}}(m_s, 1), \tag{3.93}$$

where $m_0 = m_l$, $m_k = 1$, m_l and m_s are the light quark and strange quark masses, respectively.

We use multiple levels of nested integrators to separate different parts of the action. A general multilevel Sexton-Weingarten integration scheme can be written as follows

$$H = T'_0 = T'_1 + S_1 \tag{3.94}$$

$$T'_i = T'_{i+1} + S_{i+1} \quad i = 1, 2, \cdots, k-1, \tag{3.95}$$

where $T'_k = T(p)$. The above equations separate the entire action into k levels.

Practically how various parts in (3.93) are assigned to different levels is mainly determined by their relative costs and the sizes of corresponding forces. In its simplest form, we can separate the fermion action and the gauge action and construct a 2 level Sexton-Weingarten Integration scheme

level(i)	S_i
1	$\sum_a S_{\mathrm{Q}}(m_{a-1}, m_a) + S_{\mathrm{R}}(m_s, 1)$
2	S_{G}

Table 3.2: The simplest Sexton-Weingarten multilevel integration scheme. Gauge action is separated from the fermion actions.

More sophisticated schemes may further separate fermion actions into multiple levels, as shown in table 3.3. This is the integration scheme used to evolve the $48^3 \times 96$ (5.5fm, 140MeV) and $64^3 \times 128$ (5.5fm, 140MeV) ensemble.

level(i)	S_i
1	$\sum_a S_{\mathrm{Q}}(m_{a-1}, m_a)$
2	$S_{\mathrm{R}}(m_s, 1)$
3	S_{G}

Table 3.3: Multilevel integrator used to evolve the $48^3 \times 96$ $(5.5\mathrm{fm}, 140\mathrm{MeV})$ and $64^3 \times 128$ $(5.5\mathrm{fm}, 140\mathrm{MeV})$ DWF+I ensembles. These ensembles are simulated with $(5.5\mathrm{fm})^3$ boxes. The details can be found in table B.1 in appendix B.

Additional pieces may be added to the action for specific purposes. For example, one may add the Dislocation Suppressing Determinant Ratio (DSDR) to the path integral [22] when generating ensembles for use in QCD thermodynamics. In such situations, even more sophisticated integrators may be built to maximize the efficiency [23].

Chapter 4

The Möbius Fermions

4.1 General Möbius Action - Motivation

The domain wall fermion action in its original form [8] maintains good chiral symmetry at the cost of introducing an extra fictitious dimension into the action. The size of this extra dimension (usually labeled as the s direction) is a direct multiplicative factor when counting the cost of a lattice simulation. The situation becomes worse when we approach the physical limit, in which the size of the s direction becomes so large that cost of the entire simulation can be prohibitively expensive.

There are ongoing efforts to alleviate this problem. Among them are the optimal domain wall fermions introduced by Ting-Wai Chiu [24], and the Möbius domain wall fermions introduced by Richard C. Brower *et al.*[25, 26] as a generalization. In this work we mainly use the Möbius domain wall fermion action. The Möbius fermions achieve good chiral symmetry usually at a much smaller size in the 5th dimension, essentially by exploiting the inefficiencies in the way the plain domain wall action uses this dimension.

To see why the plain domain wall action uses the s dimension inefficiently, we can examine the overlap equivalent of the domain wall action

$$D_{OV}(H_T) = \frac{1}{2}\left(1 + m + (1-m)\gamma_5 \epsilon(H_T)\right), \tag{4.1}$$

$$\epsilon(x) = \frac{(x+1)^{L_s} - (x-1)^{L_s}}{(x+1)^{L_s} + (x-1)^{L_s}}, \tag{4.2}$$

where L_s is an even number that denotes the size of the 5th dimension and H_T is the Shamir kernel: $H_T = \gamma_5 D_W (2 + D_W)^{-1}$.

$\epsilon(x)$ would be the sign function in an ideal overlap action. In domain wall actions $\epsilon(x)$ are instead approximations to the sign function. The difference between $\epsilon(x)$ and a true sign function is a measure of the residual chiral symmetry breaking effect.

The domain wall $\epsilon(x)$ in (4.2) approximates the sign function only within a certain range. For very small x, $\epsilon(x)$ approaches 0 linearly. For very large x, $\epsilon(x)$ also falls to 0, since $\epsilon(x) = \epsilon(x^{-1})$ for any even L_s. We can analyze how fast/slow $\epsilon(x)$ deviates from the sign function by plotting $1 - \epsilon(x)$ on $(0, +\infty)$. The result is shown in figure (4.1).

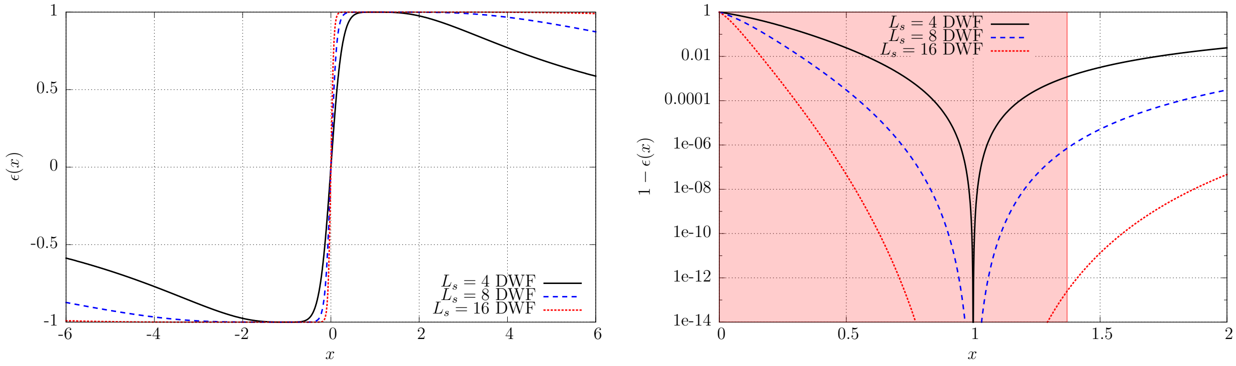


Figure 4.1: Left: plot of $\epsilon(x)$ for the domain wall fermion actions with $L_s = 4, 8, 16$. Right: plot of $\log(1 - \epsilon(x))$ with different L_s values. The shaded region denotes the approximate range of the eigenvalues of the H_T kernel.

Figure (4.1) shows that there are 2 separate branches for which $\epsilon(x)$ deviates from 1. One has $x < 1$ and the other has $x > 1$. $1 - \epsilon(x)$ is always exactly 0 for $x = 1$. The deviation on both branches approaches 1 when x approaches 0 or ±∞. However, the distribution of the eigenvalues of H_T does not extend to ∞. In fact, the eigenvalues of H_T can be close to 0 but not ∞. There is usually a sharp upper bound in the spectrum of H_T at some finite x. Due to this upper bound, the right branch introduces less deviation from the ideal overlap action compared to the left branch. The situation is more prominent when larger sizes are used in the s direction.

Since deviations on both branches contribute to the residual chiral symmetry breaking effect, the right branch in the domain wall action is unnecessarily accurate compared to the left branch. In the large L_s limit, the residual chiral symmetry breaking effect receives its main contribution from the nonzero values of $1 - \epsilon(x)$ near the origin.

The Möbius fermion action improves the situation by introducing a scaling factor λ into $\epsilon(x)$. The Möbius approximation to the sign function is the following

$$\epsilon_\lambda(x) = \frac{(\lambda x + 1)^{L_s} - (\lambda x - 1)^{L_s}}{(\lambda x + 1)^{L_s} + (\lambda x - 1)^{L_s}}. \tag{4.3}$$

The effect of any $\lambda > 1$ is to “squeeze” the curves in in figure (4.1). The location for $1 - \epsilon(x) = 0$ is no longer 1 but $1/\lambda$. Figure (4.2) shows the situation for different λ values.

Larger λ values effectively make the left branch smaller and right branch larger, thus increasing the “usefulness” of the right branch since this branch now has more overlap with the spectrum of the H_T kernel. The smaller left branch also implies a better approximation to the sign function. Apparently λ can not be too large, otherwise some of the large eigenvalues of the H_T kernel falls on the region of the right branch where the approximation to the sign function is no longer good.

Once we know the maximum eigenvalue of the H_T kernel, we can determine how large λ can be made. For example, the largest eigenvalue of H_T for the $16^3 \times 32$ $(1.8\text{fm}, 420\text{MeV})$ ensemble is known to be fluctuate around 1.37. We can plot $1 - \epsilon(x)$ at this specific point to see how $\epsilon(x)$ deviates from 1. The result can be seen in (4.2). This can help us gaining knowledge about how large we can set λ for a particular choice of L_s.

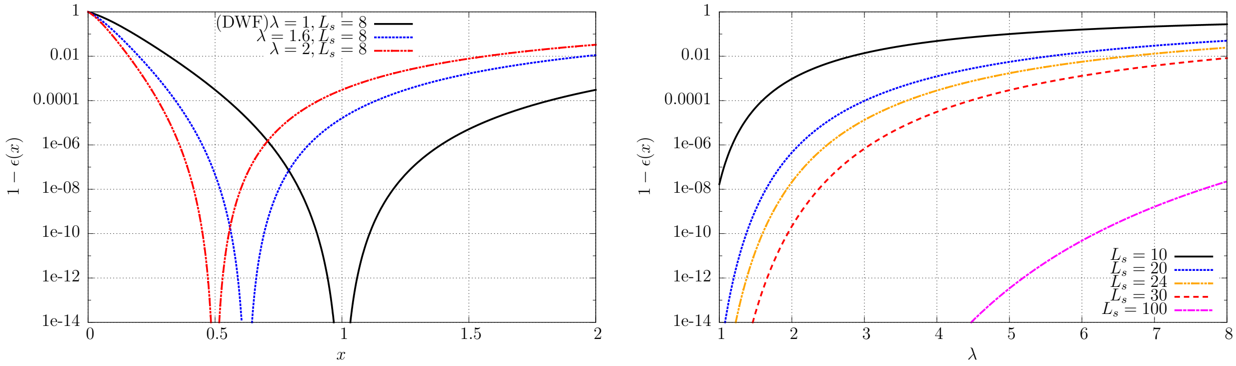


Figure 4.2: Left: plot of $1 - \epsilon(x)$ with N fixed to 8 and various λ values. There are 2 branches for each curve, the left branch shows the deviation of $\epsilon(x)$ from 1 near the origin, the right branch shows the deviation of $\epsilon(x)$ from 1 in the large x limit. Note that the largest eigenvalue of the H_T kernel is roughly 1.4. Right: Plot of $1 - \epsilon(x)$ at $x = 1.37$. The x axis is the c parameter in the Möbius fermion.

It is also possible to introduce more free parameters into $\epsilon(x)$

$$\epsilon_{\{\lambda_i\}}(x) = \frac{\prod_i(\lambda_i x + 1) - \prod_i(\lambda_i x - 1)}{\prod_i(\lambda_i x + 1) + \prod_i(\lambda_i x - 1)} \tag{4.4}$$

and optimize λ_i such that $\epsilon_{\{\lambda_i\}}(x)$ is the best approximation to the sign function for a given size in s direction. This is the general Möbius fermion which we will not discuss in this document.

Now we present the full definition of the Möbius domain wall fermion action. Written as a 5 dimensional action, it takes the following form

$$S_F(\overline{\psi}, \psi, U) = \sum_{xs;ys'} \overline{\psi}_{xs} D_{xs;ys'} \psi_{ys'}, \tag{4.5}$$

where $D_{xs;ys'}$ is the Möbius Dirac operator

$$D_{xs;ys'} = D^s_+(x,y)\delta_{s,s'} + D^s_-(x,y) m_s P_+ \delta_{s,s'+1} + D^s_-(x,y) m_{s+1} P_- \delta_{s,s'-1}. \tag{4.6}$$

The symbols D_+^s, D_-^s and m_s in the above expression are defined as follows,

$$\begin{aligned} D_+^s(x,y) &= b_s D(x,y) + 1 & (4.7)\\ D_-^s(x,y) &= c_s D(x,y) - 1, & (4.8) \end{aligned}$$

$$m_s = \begin{cases} 1 & s = 1, 2, \cdots, L_s - 1 \\ -m & s = 0 \end{cases}, \tag{4.9}$$

and $D(x,y)$ is the Wilson Dirac operator,

$$D(x,y) = (4 + M_5)\delta_{xy} - \frac{1}{2}\sum_{\mu}\big((1-\gamma_\mu)U_\mu(x)\delta_{x+\hat{\mu},y} + (1+\gamma_\mu)U_\mu^\dagger(x-\hat{\mu})\delta_{x-\hat{\mu},y}\big)\,. \tag{4.10}$$

The Möbius Dirac operator as defined above is the most general form, with $2L_s$ free parameters b_s and c_s in addition to the original domain wall fermion parameters. This operator can be transformed into its 4 dimensional equivalent overlap form using the domain wall - overlap transformation [25]

$$LD(m)R = FD_{OV}^5(m), \tag{4.11}$$

where the definition of L, R and F can be found in [25]. $D_{OV}^5(m)$ takes the following form

$$D_{OV}^5(m) = \mathrm{diag}\left\{D_{OV}(m), 1, \cdots, 1\right\}, \tag{4.12}$$

where $D_{OV}(m)$ has a definition similar to (4.1)

$$\begin{aligned} D_{OV}(m) =& \frac{1}{2}\left(1 + m + (1-m)\gamma_5\frac{S-1}{S+1}\right), & (4.13)\\ S =& \prod_s \frac{H_T^s + 1}{H_T^s - 1}, & (4.14)\\ H_T^s =& (b_s + c_s)\gamma_5 D_W \left(2 + (b_s - c_s)D_W\right)^{-1}. & (4.15) \end{aligned}$$

One important result from the domain wall - overlap transformation is

$$R^{-1}D(1)^{-1}D(m)R = D^5_{OV}(m), \tag{4.16}$$

and hence

$$\det\left(D(1)^{-1}D(m)\right) = \det\left(D^5_{OV}(m)\right) = \det\left(D_{OV}(m)\right). \tag{4.17}$$

4.2 Möbius Conserved Current

In this section we discuss how the vector and axial conserved currents are implemented in Möbius fermions and we will define the physical quark fields based on the discussion. The method here is analogous to [27].

4.2.1 The Conserved Möbius Fermion Current

We start by varying the partition function

$$Z = \int \mathcal{D}U\mathcal{D}\overline{\psi}\mathcal{D}\psi\ e^{-S_F}, \tag{4.18}$$

where S_F is the Möbius fermion action (4.5). Consider a slightly generalized case where the fermion field ψ can have multiple flavors (the flavor index is not shown in the above formula). The fermion Dirac operator D is diagonal in this flavor space since QCD does not mix fermions of different flavors. Such a theory will have a flavor $SU(N)$ symmetry. Z is unchanged under the following local fermion field transformation

$$\left\{\begin{array}{lcl} \psi_{xs} & \longleftarrow & \exp\left(\varepsilon^a_{xs}\lambda^a\right)\psi_{xs} \\ \overline{\psi}_{xs} & \longleftarrow & \overline{\psi}_{xs}\exp\left(-\varepsilon^a_{xs}\lambda^a\right) \end{array}\right. . \tag{4.19}$$

The conserved current can be obtained from the following identity

$$0 = \delta Z = - \langle \delta S_F \rangle . \tag{4.20}$$

Expanding the above equation yields the following current conservation equation

$$- \sum_{ys'} \left\langle \overline{\psi}_{xs} \lambda^a D_{xs;ys'} \psi_{ys'} \right\rangle + \sum_{ys'} \left\langle \overline{\psi}_{ys'} D_{ys';xs} \lambda^a \psi_{xs} \right\rangle = 0. \tag{4.21}$$

or

$$\begin{aligned} - \sum_{y} \left\langle \overline{\psi}_{xs} \lambda^a D_+^s(x,y) \psi_{ys} \right\rangle + \sum_{y} \left\langle \overline{\psi}_{ys} D_+^s(y,x) \lambda^a \psi_{xs} \right\rangle \\ - \sum_{y} \left\langle \overline{\psi}_{xs} \lambda^a D_-^s(x,y) m_s P_+ \psi_{y,s-1} \right\rangle + \sum_{y} \left\langle \overline{\psi}_{y,s+1} D_-^{s+1}(y,x) m_{s+1} P_+ \lambda^a \psi_{xs} \right\rangle \\ - \sum_{y} \left\langle \overline{\psi}_{xs} \lambda^a D_-^s(x,y) m_{s+1} P_- \psi_{y,s+1} \right\rangle + \sum_{y} \left\langle \overline{\psi}_{y,s-1} D_-^{s-1}(y,x) m_s P_- \lambda^a \psi_{xs} \right\rangle = 0 \end{aligned} \tag{4.22}$$

The formula becomes quite lengthy if we substitute the explicit form of $D(x,y)$ and expand this equation. So instead we use the following shorthand notations to keep equations simple,

$$F(x,\mu,s;+) = \frac{1}{2} \left\langle \overline{\psi}_{xs} \lambda^a (1-\gamma_\mu) U_\mu(x) \psi_{x+\mu,s} \right\rangle \tag{4.23}$$

$$F(x,\mu,s;-) = \frac{1}{2} \left\langle \overline{\psi}_{xs} \lambda^a (1+\gamma_\mu) U_\mu^\dagger(x-\mu) \psi_{x-\mu,s} \right\rangle \tag{4.24}$$

$$S(x,s;+) = \left\langle \overline{\psi}_{xs} \lambda^a P_+ \psi_{x,s-1} \right\rangle \tag{4.25}$$

$$S(x,s;-) = \left\langle \overline{\psi}_{xs} \lambda^a P_- \psi_{x,s+1} \right\rangle \tag{4.26}$$

$$P(x,\mu,s;++) = \frac{1}{2} \left\langle \overline{\psi}_{xs} \lambda^a (1-\gamma_\mu) U_\mu(x) P_+ \psi_{x+\mu,s-1} \right\rangle \tag{4.27}$$

$$P(x,\mu,s;-+) = \frac{1}{2} \left\langle \overline{\psi}_{xs} \lambda^a (1+\gamma_\mu) U_\mu^\dagger(x-\mu) P_+ \psi_{x-\mu,s-1} \right\rangle \tag{4.28}$$

$$P(x,\mu,s;+-) = \frac{1}{2} \left\langle \overline{\psi}_{xs} \lambda^a (1-\gamma_\mu) U_\mu(x) P_- \psi_{x+\mu,s+1} \right\rangle \tag{4.29}$$

$$P(x,\mu,s;--) = \frac{1}{2} \left\langle \overline{\psi}_{xs} \lambda^a (1+\gamma_\mu) U_\mu^\dagger(x-\mu) P_- \psi_{x-\mu,s+1} \right\rangle . \tag{4.30}$$

The equality (4.22) can then be written explicitly as

$$
\begin{aligned}
& b_s \sum_\mu \left[F(x,\mu,s;+) + F(x,\mu,s;-) - F(x-\mu,\mu,s;+) - F(x+\mu,\mu,s;-)\right] \\
& +c_s m_s \sum_\mu \left[P(x,\mu,s;++) + P(x,\mu,s;-+)\right] \\
& +c_s m_{s+1} \sum_\mu \left[P(x,\mu,s;+-) + P(x,\mu,s;--)\right] \\
& -c_{s+1} m_{s+1} \sum_\mu \left[P(x-\mu,\mu,s+1;++) + P(x+\mu,\mu,s+1;-+)\right] \\
& -c_{s-1} m_s \sum_\mu \left[P(x-\mu,\mu,s-1;+-) + P(x+\mu,\mu,s-1;--)\right] \\
= & -m_s\left(1-c_s(4+M_5)\right) S(x,s;+) - m_{s+1}\left(1-c_s(4+M_5)\right) S(x,s;-) \\
& +m_{s+1}\left(1-c_{s+1}(4+M_5)\right) S(x,s+1;+) + m_s\left(1-c_{s-1}(4+M_5)\right) S(x,s-1;-)
\end{aligned}
\tag{4.31}
$$

4.2.2 5 Dimensional Vector and Axial Current

If we sum the equation (4.31) over all s slices on both sides, the right hand side sums up to zero after a shift of dummy s variables.

$$
\begin{aligned}
& \sum_{s\mu} b_s \left[F(x,\mu,s;+) + F(x,\mu,s;-) - F(x-\mu,\mu,s;+) - F(x+\mu,\mu,s;-)\right] \\
& +\sum_{s\mu} c_s m_s \left[P(x,\mu,s;++) + P(x,\mu,s;-+)\right] \\
& +\sum_{s\mu} c_s m_{s+1} \left[P(x,\mu,s;+-) + P(x,\mu,s;--)\right] \\
& -\sum_{s\mu} c_{s+1} m_{s+1} \left[P(x-\mu,\mu,s+1;++) + P(x+\mu,\mu,s+1;-+)\right] \\
& -\sum_{s\mu} c_{s-1} m_s \left[P(x-\mu,\mu,s-1;+-) + P(x+\mu,\mu,s-1;--)\right] = 0
\end{aligned}
\tag{4.32}
$$

Since we are summing over all s slices, we are free to shift any dummy variables in in the last 2 sums such that they contain c_s only

$$\begin{aligned}
&\sum_{s\mu} b_s\left[F(x,\mu,s;+)+F(x,\mu,s;-)-F(x-\mu,\mu,s;+)-F(x+\mu,\mu,s;-)\right]\\
+&\sum_{s\mu} c_s m_s\left[P(x,\mu,s;++)+P(x,\mu,s;-+)-P(x-\mu,\mu,s;++)-P(x+\mu,\mu,s;-+)\right]\\
+&\sum_{s\mu} c_s m_{s+1}\left[P(x,\mu,s;+-)+P(x,\mu,s;--)-P(x-\mu,\mu,s;+-)-P(x+\mu,\mu,s;--)\right]=0
\end{aligned} \tag{4.33}$$

At this point we can define the conserved vector current. First define

$$\left\langle j^a_{b\mu}(x,s)\right\rangle = b_s\left[F(x,\mu,s;+)-F(x+\mu,\mu,s;-)\right] \tag{4.34}$$

$$\left\langle j^a_{c1\mu}(x,s)\right\rangle = c_s m_s\left[P(x,\mu,s;++)-P(x+\mu,\mu,s;-+)\right] \tag{4.35}$$

$$\left\langle j^a_{c2\mu}(x,s)\right\rangle = c_s m_{s+1}\left[P(x,\mu,s;+-)-P(x+\mu,\mu,s;--)\right]. \tag{4.36}$$

Then the vector current is defined as

$$\mathscr{V}^a_\mu(x)=\sum_s\left[j^a_{b\mu}(x,s)+j^a_{c1\mu}(x,s)+j^a_{c2\mu}(x,s)\right]. \tag{4.37}$$

$\mathscr{V}^a_\mu(x)$ follows the following conservation identity,

$$\left\langle\sum_\mu \Delta_\mu\mathscr{V}_\mu(x)\right\rangle=\left\langle\sum_\mu\left(\mathscr{V}^a_\mu(x)-\mathscr{V}^a_\mu(x-\hat{\mu})\right)\right\rangle=0. \tag{4.38}$$

Similar to the plain domain wall case [27] we define the axial current by introducing a -1 factor for half of the 5th dimensional slices (where we require that L_s to be even: $L_s=2N$),

$$\mathscr{A}^a_\mu(x)=-\sum_{s=0}^{2N-1}\operatorname{sign}\left(N-s-\frac{1}{2}\right)\left(j^a_{b\mu}(x,s)+j^a_{c1\mu}(x,s)+j^a_{c2\mu}(x,s)\right). \tag{4.39}$$

After some computation we get the following approximate conservation identity for the divergence of the axial current

$$\left\langle \sum_{\mu} \Delta_{\mu} \mathscr{A}_{\mu}^{a}(x) \right\rangle = \left\langle 2mJ_5^a(x) + 2J_{5q}^a(x) \right\rangle , \tag{4.40}$$

where J_5^a and J_{5q}^a take the following form

$$J_5^a(x) = \sum_y \overline{\psi}_0(y)\lambda^a D_-^0(y,x)P_+\psi_{2N-1}(x) - \sum_y \overline{\psi}_{2N-1}(y)\lambda^a D_-^{2N-1}(y,x)P_-\psi_0(x), \tag{4.41}$$

$$J_{5q}^a(x) = \sum_y \overline{\psi}_N(y)\lambda^a D_-^N(y,x)P_+\psi_{N-1}(x) - \sum_y \overline{\psi}_{N-1}(y)\lambda^a D_-^{N-1}(y,x)P_-\psi_N(x). \tag{4.42}$$

It is interesting to observe that J_5^a only depends on the field on the 2 walls, while J_{5q}^a only depends on the field on the 2 middle planes, just like their counterparts in the plain domain wall formalism. This is the case even though the Möbius fermions have an extra type of hopping term. For the domain wall fermions there are hopping terms that connect sites differing by 1 in *one* of the 5 directions. While for the Möbius fermions there are additional hopping terms that connect sites differing by 1 in *both* the s direction and one of the 4 spacetime directions.

The above $J_5^a(x)$ and $J_{5q}^a(x)$ formulas suggest the following definition for physical fermions

$$q(x) = P_-\psi_0(x) + P_+\psi_{2N-1}(x) \tag{4.43}$$

$$\overline{q}(x) = -\sum_y \overline{\psi}_{2N-1}(y)D_-^{2N-1}(y,x)P_- - \sum_y \overline{\psi}_0(y)D_-^0(y,x)P_+. \tag{4.44}$$

4.3 Möbius Residual Mass

We can define the residual mass for Möbius fermions by simply computing the ensemble average of J^a_{5q} and J^a_5 and then fitting the plateau of the following ratio

$$R(t) = \frac{\left\langle \sum_x J^a_{5q}(x,t) J^a_5(x',0) \right\rangle}{\left\langle \sum_x J^a_5(x,t) J^a_5(x',0) \right\rangle}. \tag{4.45}$$

This requires us to calculate

$$\begin{aligned}
&\left\langle \sum_x J^a_{5q}(x,t) J^a_5(x',0) \right\rangle \\
=&\sum_{yz} \big\langle \big\{ \overline{\psi}_N(y) D^N_-(y;x,t) P_+ \psi_{N-1}(x,t) - \overline{\psi}_{N-1}(y) D^{N-1}_-(y;x,t) P_- \psi_N(x,t) \big\} \\
&\quad \cdot \big\{ \overline{\psi}_0(z) D^0_-(z;x',0) P_+ \psi_{2N-1}(x',0) - \overline{\psi}_{2N-1}(z) D^{2N-1}_-(z;x',0) P_- \psi_0(x',0) \big\} \big\rangle
\end{aligned} \tag{4.46}$$

and a similar contraction $\langle J^a_5(x,t) J^a_5(x',0)\rangle$ for Möbius fermions.

After expanding the above equation and performing all contractions for the Grassmann variables, we get the following formula

$$\begin{aligned}
&\langle J_{5q}(x,t) J_5(x',0)\rangle \\
=&\sum_{xx'} \big\langle \mathrm{Tr}\big\{ (D^{-1}D_-)(x',0,0;x,t,N-1) P_- (D^{-1}D_-)(x,t,N;x',0,2N-1) P_- \big\} \big\rangle \\
-&\sum_{xx'} \big\langle \mathrm{Tr}\big\{ (D^{-1}D_-)(x',0,2N-1;x,t,N-1) P_- (D^{-1}D_-)(x,t,N;x',0,0) P_+ \big\} \big\rangle \\
-&\sum_{xx'} \big\langle \mathrm{Tr}\big\{ (D^{-1}D_-)(x',0,0;x,t,N) P_+ (D^{-1}D_-)(x,t,N-1;x',0,2N-1) P_- \big\} \big\rangle \\
+&\sum_{xx'} \big\langle \mathrm{Tr}\big\{ (D^{-1}D_-)(x',0,2N-1;x,t,N) P_+ (D^{-1}D_-)(x,t,N-1;x',0,0) P_+ \big\} \big\rangle
\end{aligned} \tag{4.47}$$

Where $D_- = \mathrm{diag}\left\{D_-^0, D_-^1, \cdots, D_-^{2N-1}\right\}$.

A remarkable property of the above equation is that the correlator looks exactly like the counterpart in the plain domain wall fermion case if we treat $D^{-1}D_-$ as the equivalent of the D^{-1} in the plain domain wall fermion. The formula reduces naturally to the plain domain wall fermion case in which $D_- = -1$.

Another noteworthy issue is the definition of the Hermitian Dirac operator D_H. Unlike the plain domain wall fermion, $(\gamma_5 R_5 D)^\dagger \neq \gamma_5 R_5 D$ for general Möbius fermions with non-zero off-diagonal c_s. So the simple definition of the Hermitian Dirac operator in plain domain wall fermion does not apply. However, if all D_-^s are invertible and $c_s = c_{2N-1-s}$ for all s, then

$$(D_-)^{-1}D = \begin{pmatrix} (D_-^0)^{-1}D_+^0 & P_- & 0 & \cdots & -mP_+ \\ P_+ & (D_-^1)^{-1}D_+^1 & P_- & \cdots & 0 \\ 0 & P_+ & (D_-^2)^{-1}D_+^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -mP_- & 0 & P_+ & \cdots & (D_-^{2N-1})^{-1}D_+^{2N-1} \end{pmatrix} \tag{4.48}$$

is Hermitian: $(\gamma_5 R_5 (D_-)^{-1}D)^\dagger = \gamma_5 R_5 (D_-)^{-1}D$. The inverse of of this Hermitian operator is the following

$$D_H^{-1} = \left(\gamma_5 R_5 D^{-1} D_-\right)^\dagger = \gamma_5 R_5 D^{-1} D_-. \tag{4.49}$$

If one or more D_-^s are not invertible, then $(D_-)^{-1}D$ is not well defined. However, D_H^{-1} as shown in (4.49) is still Hermitian and well defined.

By combining (4.43, 4.44), (4.48) and (4.49) we conclude that $D^{-1}D_-$ is the Möbius counterpart of D^{-1} in the plain domain wall action. For many physical quantities, we can use exactly the same formula in the Möbius formalism by making the substitution $D^{-1} \longleftarrow D^{-1}D_-$ in the corresponding domain wall formula. For example, the Möbius quark propagator takes the following form,

$$\underbrace{q(x)\overline{q}}(y) = \sum_z D^{-1}(x,z)D_-(z,y) = \left(D^{-1}D_-\right)(x,y). \tag{4.50}$$

Since the substitution rule $D^{-1} \longleftarrow D^{-1}D_-$ is valid for a single fermion propagator, it is immediately clear that it can be applied to all quark contractions with vertices made of the physical fields defined in (4.43, 4.44). Exceptions include vertices made of the 5 dimensional vector and axial currents (4.37, 4.40). This is because the $\overline{q}$ field involved in the definitions of $\mathscr{V}(x)$ and $\mathscr{A}(x)$ does not follow (4.44). As a consequence, there is no D_- associated with the outgoing quark line starting from the vector or axial current vertex.

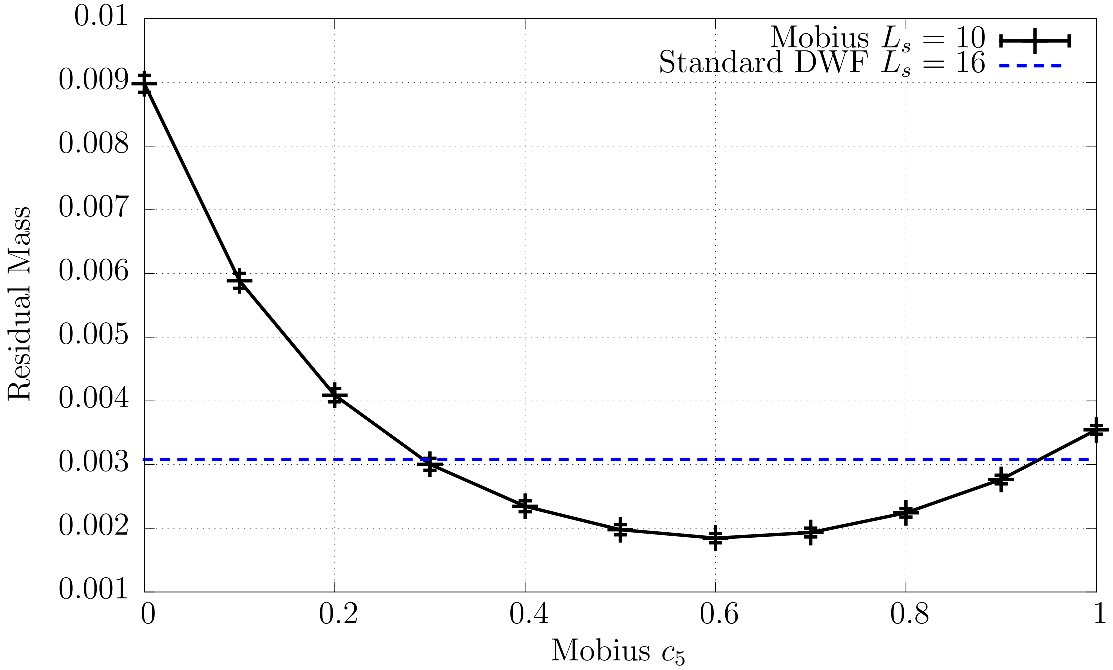


Figure 4.3: The residual mass for the Möbius fermions, measured on the $16^3 \times 32$ (1.8fm, 420MeV) lattice. For the Möbius fermions we use $L_s = 10$.

Figure 4.3 shows the residual mass of a $L_s = 10$ Möbius action. This action sets all c_s equal and $b_s = c_s + 1$. The figure shows the residual mass as a function of c_s, measured with input quark mass $m = 0.01$. The blue line is the residual mass of the $L_s = 16$ domain wall action. Using the Möbius formalism this action is equivalent to an $L_s = 16$, $c_s = 0$ and $b_s = 1$ Möbius action. Notice that the blue line crosses the black curve at $c_s = 0.3$, this will be explained later.

The graph shows that the residual mass exhibits a local minimum as a function of c. This is typical for Möbius actions in general. The existence of a local minimum can be explained using exactly the same reasoning in section 4.1. For small and large c, either the left branch or the right branch significantly breaks the chiral symmetry. The residual chiral symmetry breaking effect of the 2 branches is minimized and balanced by a suitably chosen c neither too large nor too small. m_{res} thus reaches a local minimum at this point.

4.4 The Axial Current Renormalization Factor Z_A

The vector and axial currents as defined in (4.37, 4.40) both have conservation laws associated with them. So they receive either no extra multiplicative renormalization factor or a renormalization factor very close to 1, since the axial current is not strictly conserved. It is possible to use it directly in lattice calculations such as the measurement of the kaon and pion decay constants. However both are 5 dimensional objects and are not particularly easy to use. An additional problem in the Möbius formalism is that these vertices do not have a built in D_- in their $\overline{q}$ part, so care must be taken for any propagators starting from these vertices.

In practice it is more convenient to define physical quantities using the definition of 4 dimensional physical field operators (4.43, 4.44). Following the definition of the axial current in continuum, a fine choice for the local vector and axial currents on the lattice is

$$V_\mu(x) = \overline{q}(x)\gamma_\mu q(x) \tag{4.51}$$

$$A_\mu(x) = \overline{q}(x)\gamma_\mu\gamma_5 q(x). \tag{4.52}$$

These objects are easier to use because they are 4 dimensional and are constructed directly from the interpolating operators (4.43, 4.44). They give identical results for low energy Green's functions when properly normalized. However, since they do not directly follow the conservation laws (in other words, they are only approximately conserved), both of the above operators have renormalization factors associated with them. These factors need to be accounted for when computing physical quantities.

The usual method to compute these factors involves measuring the ratio of the ensemble averages of each operator in otherwise identical low energy Green's functions. For example, the axial current renormalization factor can be calculated in the following way

$$\left\langle \sum_{\boldsymbol{x}} \mathscr{A}_3(\boldsymbol{x}, t-1/2) P(0) \right\rangle = N_5 \left(e^{-m_\pi t} - e^{-m_\pi (T-t)} \right), \tag{4.53}$$

$$\left\langle \sum_{\boldsymbol{x}} A_3(\boldsymbol{x}, t) P(0) \right\rangle = N_4 \left(e^{-m_\pi t} - e^{-m_\pi (T-t)} \right), \tag{4.54}$$

$$Z_A = N_5/N_4. \tag{4.55}$$

Where $P(0)$ is an interpolating operator that couples with the axial current operator evaluated at source time slice 0. For simplicity we can choose $P(0)$ to be the pion interpolating operator. We only use the temporal component since spatial components do not contribute when summed over the spatial volume.

As mentioned above, computing the conserved 5 dimensional axial current for Möbius fermions in this way requires inverting the 5 dimensional D_- matrix. This is quite inconvenient and may be ill defined since D_- can be near singular. To bypass this apparent difficulty we can compute the divergences of the 2 forms of axial currents instead,

$$\left\langle \sum_{\boldsymbol{x}} \Delta_\mu \mathscr{A}_\mu(\boldsymbol{x},t) P(0) \right\rangle = N_5' \left(e^{-m_\pi t} + e^{-m_\pi (T-t)} \right), \tag{4.56}$$

$$\left\langle \sum_{\boldsymbol{x}} \Delta_\mu A_\mu(\boldsymbol{x},t) P(0) \right\rangle = N_4' \left(e^{-m_\pi (t-1/2)} + e^{-m_\pi (T-t+1/2)} \right), \tag{4.57}$$

$$Z_A = N_5'/N_4'. \tag{4.58}$$

$\Delta_\mu A_\mu$ can be computed directly. To compute $\Delta_\mu \mathscr{A}_\mu$ we can exploit the conservation law (4.40),

$$\left\langle \sum_{\boldsymbol{x}} \Delta_\mu \mathscr{A}_\mu(x) P(0) \right\rangle = 2m \left\langle \sum_{\boldsymbol{x}} J_5^a(x) P(0) \right\rangle + 2 \left\langle \sum_{\boldsymbol{x}} J_{5q}^a(x) P(0) \right\rangle. \tag{4.59}$$

There are extra terms in these formulas that generate nonzero values on time slice 0. In general these terms need not be considered since we never start fitting the data from $t = 0$. Notice that the finite difference shifts the center of the curve in (4.57) by 1/2.

4.5 Eigenvalues of the Shamir Kernel $\gamma_5 D_W (2 + D_W)^{-1}$

The spectrum of the Shamir kernel $H_T = \gamma_5 D_W (2 + D_W)^{-1}$ plays an essential role in the Möbius fermion formalism, as discussed in section 4.1. This section discusses some properties of the spectrum of H_T.

By definition H_T^s depends on b_s and c_s,

$$H_T^s = (b_s + c_s) \gamma_5 D_W \left(2 + (b_s - c_s) D_W \right)^{-1}. \tag{4.60}$$

In the following discussion we only consider the special case $b_s - c_s = 1, \forall s$. We also drop the constant factor $b_s + c_s$ from the above definition. We thus obtain a simplified version that is free of parameters

$$H_T = \gamma_5 D_W \left(2 + D_W\right)^{-1}. \tag{4.61}$$

We can use either the Ritz method or the Rayleigh quotient method [28] to compute the largest/smallest eigenvalues of the above matrix.

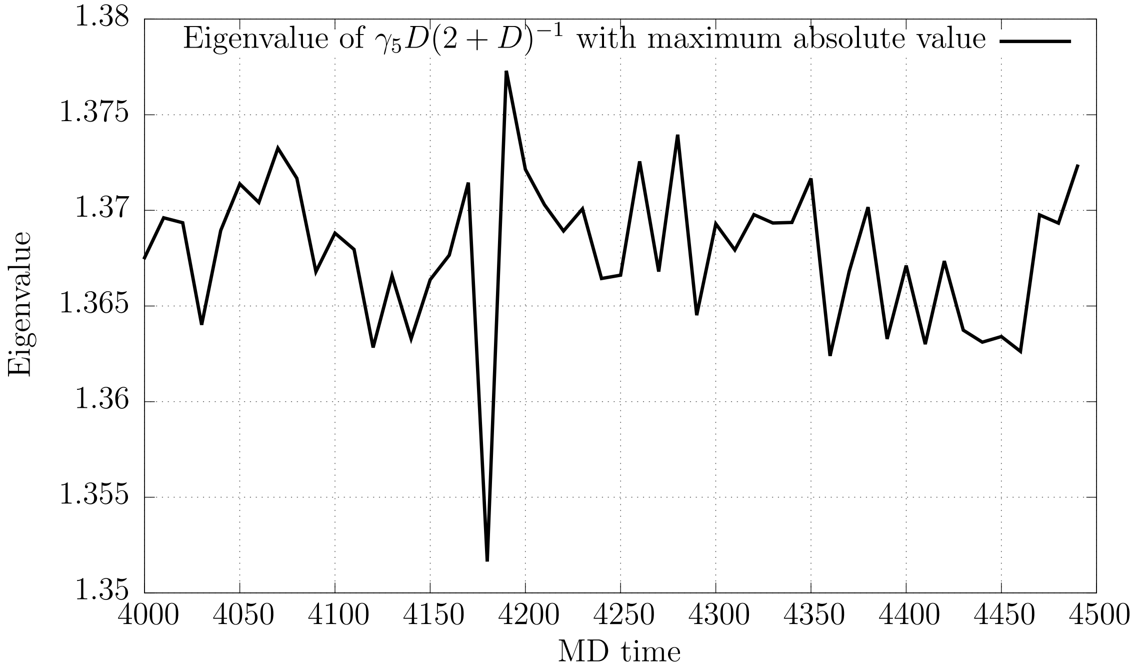


Figure 4.4: Eigenvalue of $\gamma_5 D(2+D)^{-1}$ with maximum absolute value, shown as a function of MD time.

Figure 4.4 shows the maximum eigenvalue measured on the $16^3 \times 32$ (1.8fm, 420MeV) ensemble. A total of 50 configurations were measured. Clearly the largest eigenvalue remains stable around 1.37. The maximum eigenvalue of the H_T kernel increases at finer lattice spacing. Table 4.1 shows a few measurements of the largest eigenvalue of the Shamir kernel with different β values.

We can also compute the analytic upper bound of the spectrum as a reference. The analytic formula for the eigenvalues of $H_T = \gamma_5 D_W (2 + D_W)^{-1}$ in the free field case is as

β	lattice size	approximated largest eigenvalue
1.633	$32^3 \times 8$	1.050(3)
1.671	$32^3 \times 8$	1.085(4)
1.707	$32^3 \times 8$	1.116(4)
1.740	$32^3 \times 8$	1.146(2)
1.801	$32^3 \times 8$	1.200(7)
1.829	$32^3 \times 8$	1.221(7)
2.13	$16^3 \times 32$	1.367(14)
2.25	$32^3 \times 64$	1.500(1)
free field	$32^3 \times 8$	3.742

Table 4.1: Measurement of the maximum eigenvalue of H_T for ensembles with different β. The $32^3 \times 8$ ensembles with various β are listed in appendix B. All ensembles involved use periodic boundary condition for spatial directions and antiperiodic boundary condition for the t direction. The values/errors shown are the averages/standard deviations of the corresponding eigenvalues.

follows

$$\lambda(q) = \pm \frac{\sqrt{\left(4 + M_5 - \sum_\mu \cos q_\mu\right)^2 + \sum_\mu \sin^2 q_\mu}}{\sqrt{\left(4 + M_5 - \sum_\mu \cos q_\mu + 2\right)^2 + \sum_\mu \sin^2 q_\mu}} \tag{4.62}$$

each $\lambda(q)$ has a degeneracy of $2\times$ # of colors (further degeneracy may happen if different q values yield the same eigenvalue)

$$q = \left(\frac{2\pi}{N_0} j_0, \frac{2\pi}{N_1} j_1, \frac{2\pi}{N_2} j_2, \frac{2\pi}{N_3}\left(j_3 + \frac{1}{2}\right)\right). \tag{4.63}$$

N_i is the lattice size in i direction. We assume that we are using antiperiodic boundary condition in t direction and spatial directions have periodic boundary condition. $j_i = 0, 1, \cdots, N_i - 1$.

Chapter 5

Computation Strategies

5.1 Krylov Space Solvers

Solving large linear sparse systems is frequently required in lattice calculations. For example, in the hybrid Monte Carlo algorithm, to evaluate the fermion force term we are required to solve a linear equation for the corresponding preconditioned Dirac operator $\mathcal{M}$. In measurement tasks where fermion propagators are needed, the lattice fermion propagator is $D^{-1}s$, where s is the source for the propagator. The fermion Dirac operator D is a very high dimensional sparse matrix even for a moderately sized lattice. So iterative solvers are the only choices in most lattice applications.

5.1.1 the Conjugate Gradient Method

Consider the following linear system

$$Ax = b. \tag{5.1}$$

Without loss of generality we assume that we start from a zero initial guess $x_0 = 0$. Starting the conjugate gradient algorithm with a nonzero initial guess x_0 is equivalent to solving $Ay = b - Ax_0$ with initial guess $y = 0$. A Krylov space solver approximates the solution x

using the form $P(A)b$, where $P(A)$ is a polynomial of A. A specific Krylov space method may require A to be Hermitian, Hermitian positive definite, or to have no restriction at all.

Denote the true solution as $x^* = A^{-1}b$. A Krylov space method generates a series of approximate solutions x_i, $i = 0, 1, 2, \cdots$, where x_0 is usually the initial guess. To facilitate the discussion, we define two auxiliary sequences e_i, r_i as follows

$$e_i = x_i - x^* \tag{5.2}$$

$$r_i = b - Ax_i = -Ae_i. \tag{5.3}$$

$$\tag{5.4}$$

We also introduce a sequence p_i whose definition is yet to be added. The purpose of p_i is to set the new search direction in step i,

$$x_{i+1} = x_i + \alpha_i p_i. \tag{5.5}$$

By convention we place a number α_i as a coefficient in front of p_i so we can minimize some norm along the search direction p_i.

We define $\mathcal{K}_n(A, b)$ as the nth Krylov space generated by matrix A and vector b

$$\mathcal{K}_n(A, b) = \text{span}\left\{b, Ab, \cdots, A^n b\right\}. \tag{5.6}$$

In other words, any vector in $\mathcal{K}_n(A, b)$ can be written as $P_n(A)b$ where $P_n(y)$ is a polynomial of y up to order n. A Krylov space method finds an approximation to x^* in this space via some criteria, usually minimizing a norm.

5.1.2 Minimizing a Norm

In the conjugate gradient algorithm we search for a vector x_i in $\mathcal{K}_n(A, b)$ that minimizes the following norm *at each step*

$$\|e_i\|_A^2 = e_i^\dagger A e_i. \tag{5.7}$$

Since we use A in the above norm, the conjugate gradient method can only be applied to problems where A is a Hermitian positive definite matrix. The above condition determines α_i,

$$\begin{aligned} \|e_{i+1}\|_A^2 =& \|e_i + \alpha_i p_i\|_A^2 && (5.8)\\ =& \|e_i\|_A^2 + \alpha_i e_i^\dagger A p_i + \alpha_i^\dagger p_i^\dagger A e_i + \alpha_i^\dagger \alpha_i p_i^\dagger A p_i && (5.9)\\ \Longrightarrow& \alpha_i = -\frac{p_i^\dagger A e_i}{p_i^\dagger A p_i}. && (5.10) \end{aligned}$$

Up to this point we have the following partially constructed algorithm,

$$\begin{aligned} x_{i+1} =& x_i + \alpha_i p_i && (5.11)\\ r_{i+1} =& r_i - \alpha_i A p_i && (5.12)\\ \alpha_i =& -\frac{p_i^\dagger A e_i}{p_i^\dagger A p_i} = \frac{p_i^\dagger r_i}{p_i^\dagger A p_i}. && (5.13) \end{aligned}$$

The definition for the sequence p_i is yet to be added.

5.1.3 Finding New Search Directions

A natural way to generate new search directions is to look at the residue

$$p_i = r_i + \sum_{k=0}^{i-1} \beta_{ik} p_k \tag{5.14}$$

since r_i is the only vector independent of all previous search directions in the partially constructed algorithm (5.11, 5.12, 5.13). In infinite precision arithmetic, $r_i = b - Ax_i$. Adding r_i into the Krylov space also increases the dimension of the Krylov space. We perform some projection via parameters β_{ik} to enforce certain orthogonalization conditions. The reason that we don't search directly in the direction defined by r_i, but rather a projected direction p_i is because it avoids moving again in any previous directions. This becomes clear when we enforce orthogonalization conditions on p_i.

The conjugate gradient method further requires all search directions to be A-orthogonal,

$$p_j^\dagger A p_i = 0,\ \forall j \neq i. \tag{5.15}$$

One important consequence of (5.15) is that α_i defined in (5.10) has just the correct size for the final solution. There is no need to move in direction p_i again after step i. To see why this is the case, suppose all A-orthogonal search directions are known in advance and we construct the final solution by minimizing (5.7)

$$\|e\|_A^2 = \|e_0 + \sum_i \alpha_i' p_i\|_A^2. \tag{5.16}$$

We deliberately call the coefficient α_i' to show any possible differences. Minimizing the above norm defines all α_i'

$$\|e\|_A^2 = \|e_0\|_A^2 + \sum_i \alpha_i'^\dagger \alpha_i' \|p_i\|_A^2 + \sum_i \left(\alpha_i' e_0^\dagger A p_i + \alpha_i'^\dagger p_i^\dagger A e_0 \right) \tag{5.17}$$

$$\Longrightarrow \alpha_i' = -\frac{p_i^\dagger A e_0}{\|p_i\|_A^2}. \tag{5.18}$$

So α_i' is equal to α_i *if and only if* (5.15) is met. If the p_i are not A-orthogonal, then moving in new search directions introduces unwanted moves in old directions, spoiling this feature. It is also clear that the matrix used to define the orthogonal search directions (in this case,

A) has to be the same as the matrix that defines the norm. This fact will be used when we generalize the method in section 5.1.5.

Equations (5.11, 5.12, 5.13, 5.14, 5.15) fully define the conjugate gradient algorithm.

5.1.4 Truncated Recurrence Relations

Another important fact in the conjugate gradient algorithm is that its orthogonalization equation (5.14) appears to be truncated. This fact means that β_{ik} is nonzero only when $i - k \leq \epsilon$ where ϵ is a number *independent of* i. The practical implication is that we only need to store ϵ previous search directions for equation (5.14). So the conjugate gradient algorithm operates in a fixed amount of computer memory, independent of the iteration count i.

In fact, we only need to store 1 previous search directions for the conjugate gradient algorithm as $\epsilon = 1$. To see this,

1. By expressing r_j as a linear combination of p_j from equation (5.14) and using the $A-$orthogonal condition imposed on p_j (5.15), it is easy to prove that

   $$r_j^\dagger A p_i = \left(p_j^\dagger - \sum_{k=0}^{j-1} \beta_{jk}^* p_k^\dagger \right) A p_i = 0, \; \forall j < i. \tag{5.19}$$

   In addition we obtain from equation (5.12)

   $$p_j^\dagger r_{i+1} = p_j^\dagger r_i - \alpha_i p_j^\dagger A p_i. \tag{5.20}$$

   So

   $$p_j^\dagger r_i = 0, \; \forall j < i. \tag{5.21}$$

2. By applying r_j to equation (5.12) we obtain in a similar way

$$r_j^\dagger r_i = 0,\ \forall j < i. \tag{5.22}$$

3. We get the following truncated recurrence relation

$$\beta_{ij} = -\frac{p_j^\dagger A r_i}{p_j^\dagger A p_j} = \frac{1}{\alpha_j}\frac{r_{j+1}^\dagger r_i - r_j^\dagger r_i}{p_j^\dagger A p_j} = \begin{cases} r_i^\dagger r_i / r_{i-1}^\dagger r_{i-1} & j = i-1 \\ 0 & j < i-1. \end{cases} \tag{5.23}$$

Putting the various pieces together gives the conjugate gradient algorithm for any Hermitian positive definite matrix A

$$r_0 = p_0 = b \tag{5.24}$$

$$\alpha_i = \frac{r_i^\dagger r_i}{p_i^\dagger A p_i} \tag{5.25}$$

$$x_{i+1} = x_i + \alpha_i p_i \tag{5.26}$$

$$r_{i+1} = r_i - \alpha_i A p_i \tag{5.27}$$

$$\beta_i = \frac{r_{i+1}^\dagger r_{i+1}}{r_i^\dagger r_i} \tag{5.28}$$

$$p_{i+1} = r_{i+1} + \beta_i p_i. \tag{5.29}$$

One additional ingredient missing in the above is the stopping criteria. Although the norm (5.7) decreases at each iteration, it is not possible to calculate this norm unless we already know the solution x^*. Instead we usually evaluate the following norm,

$$\|b - Ax_i\|^2 = \|r_i\|^2. \tag{5.30}$$

Numerically we can evaluate either $\|b - Ax_i\|^2$ literally or evaluate $\|r_i\|^2$. The former ("true residue") is more accurate but requires more work. The latter ("accumulated residue") is more commonly used but in some cases can deviates significantly from the true residue due to the accumulation of round-off errors.

5.1.5 the General Case

We can generalize the above discussion by introducing some free parameters. We restrict our discussion to Krylov space solvers such that they

1. preserve the basic structure of the conjugate gradient method,
2. minimize a norm,
3. have orthogonal search directions (so the method will not move in a direction again), and
4. have truncated recurrence relations.

We first choose α_i to minimize the error according to the following general norm, defined using some Hermitian positive definite matrix B

$$\|e_{i+1}\|_B^2 = \|e_i\|_B^2 + \alpha_i e_i^\dagger B p_i + \alpha_i^\dagger p_i^\dagger B e_i + \alpha_i^\dagger \alpha_i p_i^\dagger B p_i. \tag{5.31}$$

Note that B is not necessarily equal to A. We also require search directions p_i be orthogonal w.r.t. matrix B

$$p_j^\dagger B p_i = 0, \ \forall j \neq i. \tag{5.32}$$

The same matrix B has to be used for both the norm and the orthogonal relations of p_i for the reason discussed previously.

The above equations define the following iterative procedure

$$\alpha_i = -\frac{p_i^\dagger B e_i}{p_i^\dagger B p_i} = \frac{p_i^\dagger B A^{-1} r_i}{p_i^\dagger B p_i} \tag{5.33}$$

$$x_{i+1} = x_i + \alpha_i p_i \tag{5.34}$$

$$r_{i+1} = r_i - \alpha_i A p_i \tag{5.35}$$

$$p_i = D r_i + \sum_{k=0}^{i-1} \beta_{ik} p_k \tag{5.36}$$

$$\beta_{ij} = -\frac{p_j^\dagger B D r_i}{p_j^\dagger B p_j}, \ \forall j < i. \tag{5.37}$$

One extra addition is that we generate a new search direction p_i by applying a matrix D to the residue and then combine the result with previous search directions. Whether equation (5.37) is truncated or not depends on our choice of B and D.

In the simplest case β_{ij} is nonzero only for $j = i - 1$. So in addition to the above, we require

$$p_j^\dagger B D r_i = 0, \ \forall j < i - 1. \tag{5.38}$$

By rewriting equation (5.35) we have

$$0 = p_j^\dagger B D r_i = \frac{1}{\alpha_j^\dagger} \left(r_j^\dagger - r_{j+1}^\dagger \right) A^{-\dagger} B D r_i, \ \forall j < i - 1. \tag{5.39}$$

This equation implies $r_{j-1}^\dagger A^{-\dagger} B D r_i = r_j^\dagger A^{-\dagger} B D r_i$ for all $j < i$. So the condition we imposed is equivalent to

$$r_j^\dagger A^{-\dagger} B D r_i = \lambda_i, \ \forall j < i. \tag{5.40}$$

In other words, we require $r_j^\dagger A^{-\dagger} B D r_i$ be a constant independent of j.

On the other hand, we can easily check

$$r_j^\dagger D^\dagger B p_i = 0, \ \forall j < i. \tag{5.41}$$

This is because Dr_j is a linear combination of $p_k, k = 0, 1, \cdots, j$. Similarly we can prove

$$p_j^\dagger BA^{-1}r_i = 0,\ \forall j < i \tag{5.42}$$

by using (5.35) and the orthogonal condition for p_i.

Using the above equations we can in addition prove the following orthogonality relations

$$r_j^\dagger D^\dagger BA^{-1}r_i = 0,\ \forall j < i. \tag{5.43}$$

So it is sufficient to impose the following relation between the free parameters B and D to ensure a truncated recurrence relation for β_{ij},

$$D^\dagger BA^{-1} = A^{-\dagger}BD. \tag{5.44}$$

If the above condition is met, then we can simplify α_i and $\beta_{i,i-1}$ as

$$\alpha_i = \frac{r_i^\dagger D^\dagger BA^{-1}r_i}{p_i^\dagger Bp_i}, \tag{5.45}$$

$$\beta_{i,i-1} = \frac{r_i^\dagger D^\dagger BA^{-1}r_i}{r_{i-1}^\dagger D^\dagger BA^{-1}r_{i-1}}. \tag{5.46}$$

Apparently the choice of B and D is also subject to practical issues. It is obvious that all A^{-1} must be eliminated for any quantity that is actually maintained during computing.

The following table lists a few examples

method	B	D	restrictions on A
Conjugate Gradient (CG)	A	1	Hermitian positive definite
GCGE	1	$A^\dagger$	no restriction
Conjugate Residue (CR) [29]	$A^\dagger A$	1	Hermitian

Table 5.1: A few Krylov space methods and their specifications.

5.2 Krylov Space Solvers on Lattice Dirac Equations

A typical lattice measurement task spends most of the computing time solving the following Dirac equation

$$Mx = b, \tag{5.47}$$

where M is typically the preconditioned Dirac matrix. In general M is neither Hermitian nor positive definite. Applying the above methods to equation (5.47), we obtain the following variants for each method

1. CG can only solve equations with Hermitian positive definite matrices. The common practice is to multiply $M^\dagger$ to equation (5.47)

   $$M^\dagger Mx = M^\dagger b \tag{5.48}$$

   and set $A = M^\dagger M$.

2. GCGE does not have any restriction on the matrix it solves. So we can simply use $A = M$.

3. CR requires A to be Hermitian. To achieve this we notice that if 4D even-odd preconditioning is used then the plain domain wall operator $\gamma_5 R_5 M$ is Hermitian. So we set $A = M_H = \gamma_5 R_5 M$ and solve the following equation

   $$M_H x = \gamma_5 R_5 b. \tag{5.49}$$

   For general Möbius domain wall fermion actions this can be tricky to do since the Möbius Hermitian Dirac operator involves the inverse of D_-.

5.2.1 Performance Comparison

CR makes use of the Krylov space $\mathcal{K}_n(M_H, r_0)$ while CG/GCGE use $\mathcal{K}_n(M_H^2 = M^\dagger M, r_0)$. CR and CG minimize the same norm since $B = M^\dagger M = M_H^2$. GCGE on the other hand, minimizes $\|e_i\|^2$. Since $\mathcal{K}_n(M_H^2, r_0)$ is a subspace of $\mathcal{K}_{2n}(M_H, r_0)$, CR obtains a solution *strictly* no worse than CG after performing the same number of M_H multiplications in both methods.

Unfortunately for certain Hermitian matrices CR may not perform much better than CG. This usually happens when the spectrum of M_H has certain symmetries [30]. For example, suppose the eigenvalues of M_H appear in positive/negative pairs: $(+\lambda_i, -\lambda_i)$. The corresponding eigenvectors are ϕ_i^+ and ϕ_i^-, respectively. Consider the expansion of the initial residual vector b with respect to the eigenvectors of M_H

$$b = \sum_i \left(b_i^+ \phi_i^+ + b_i^- \phi_i^- \right). \tag{5.50}$$

It is not difficult to prove that if $\|b_i^+\| = \|b_i^-\|$, $\forall i$ then CR produces exactly the same solution as CG after the same number of M_H multiplications. Roughly speaking, this is because CR builds its solution using polynomials of M_H

$$x_k = p_k(M_H) b, \tag{5.51}$$

$p_k(y)$ is an approximation to y^{-1}. However, y^{-1} is an odd function so half of the dimensions of the Krylov space are useless.

To see the problem more clearly, we can compute the kth residue in CR

$$\begin{aligned} \epsilon_k =& \sum_i \|M_H p_k(M_H) b - b\|^2 & (5.52) \\ =& \sum_i \|b_i^+\|^2 \left(1 - \lambda_i p_k(\lambda_i)\right)^2 + \sum_i \|b_i^-\|^2 \left(1 + \lambda_i p_k(-\lambda_i)\right)^2. & (5.53) \end{aligned}$$

Decompose $p_k(y)$ into odd and even parts

$$p_k(y) = yp_o(y^2) + p_e(y^2). \tag{5.54}$$

Then

$$\epsilon_k = \sum_i 2\|b_i^\pm\|^2 \left(\left(1 - \lambda_i^2 p_o(\lambda_i^2)\right)^2 + \lambda_i^2 p_e(\lambda_i^2)^2 \right). \tag{5.55}$$

Apparently the optimal choice (as CR does) is to have $p_e(y) = 0$. So all even powers of M_H does not appear in the polynomials that CR builds. In such cases, CR generates the same sequence of approximate solutions as CG does[1].

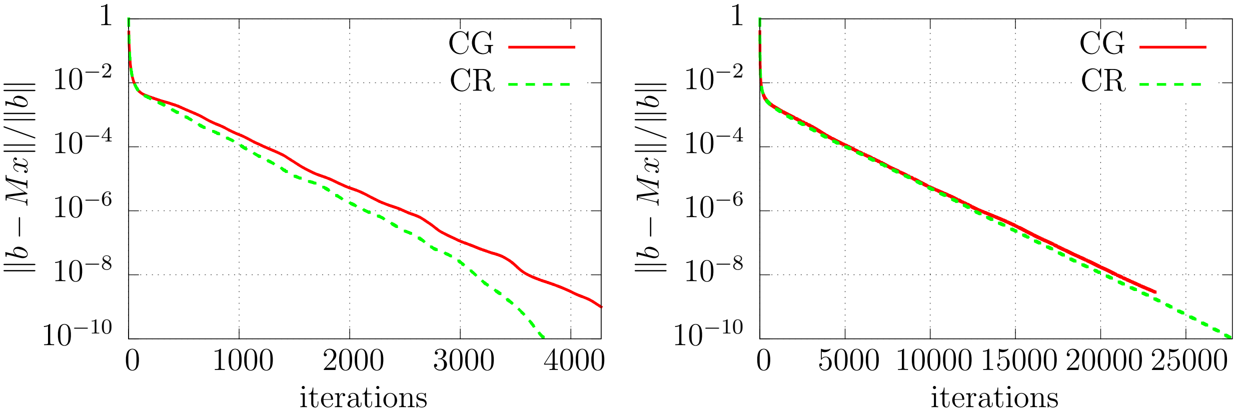


Figure 5.1: Comparison between CG and CR. Left: test on the $16^3 \times 32$ (1.8fm, 420MeV) ensemble. Right: test on the $32^3 \times 64$ (4.6fm, 170MeV) ensemble. For both methods we calculate the true residue, which is the same for these 2 methods. The difference between CR and CG becomes smaller on larger lattices. “Iterations” in this context means the number of multiplications by M.

Figure 5.1 shows the application of CG and CR on the $16^3 \times 32$ (1.8fm, 420MeV) and $32^3 \times 64$ (4.6fm, 170MeV) ensembles. The iterations in the figure should be interpreted as the number of M_H multiplications. Apparently CR is barely better than CG. Moreover, on larger lattices the difference between the 2 methods becomes even smaller. So although

[1] If we use CG we need to multiply the original equation by M_H. For each iteration in CG we multiply an extra M_H^2 to the residue so CG effectively picks up all terms with odd powers of M_H: $M_H^{2k+1}b,\ k \in \mathbb{N}$.

CR deals with the matrix M which has a smaller condition number than $M^\dagger M$, it relies on distributions of eigenvalues of M to manifest this advantage.

5.2.2 The Method of Generalized Minimal Residual

Although truncated recurrence relations are very useful for practical calculations, it puts a very strong restriction on potential methods one can build. Relaxing this requirement results in a very general class of methods, the generalized minimal residual method (GMRES).

Roughly speaking, GMRES(m) computes an orthonormal basis in the Krylov space $\mathcal{K}_n(A, r)$ using the Arnoldi algorithm (a simple Gram-Schmidt orthogonalization on the vectors $A^i r$). Then it finds a vector x in this space that minimizes some norm, e.g.,

$$\|e_i\|_A^2 = \|x_i - x^*\|_A^2. \tag{5.56}$$

We use the Arnoldi algorithm because the naive basis $A^i r,\ i = 0, 1, \cdots, m-1$ can be highly linearly dependent due to numerical instabilities[2]. The algorithm restarts after m steps, where m is a free parameter. The choice of m is entirely due to practical considerations, such as the size of the available computer memory.

GMRES finds the best vector that minimizes a given norm, all truncated methods in previous sections are in fact special cases of this method. For example, if one uses the norm (5.56) in GMRES and A is Hermitian positive definite, then GMRES and CG find exactly the same sequence of approximate solutions[3]. So it is a very useful tool to asses the effectiveness of a given Krylov space.

Tests performed on the $16^3 \times 32$ $(1.8\text{fm}, 420\text{MeV})$ lattice show that for the domain wall fermions, $\mathcal{K}_n(M, b)$ is less efficient than $\mathcal{K}_n(M_H, b)$. This can be seen from figure 5.2. All

[2] Another problem associated with the naive basis is that if we decompose these vectors using eigenvectors of A, then the modes with small eigenvalues vanish after a few applications of A. This is because each component is scaled by λ^i in $A^i r$, where λ is the corresponding eigenvalue.

[3] Of course this is only true if we do not restart GMRES.

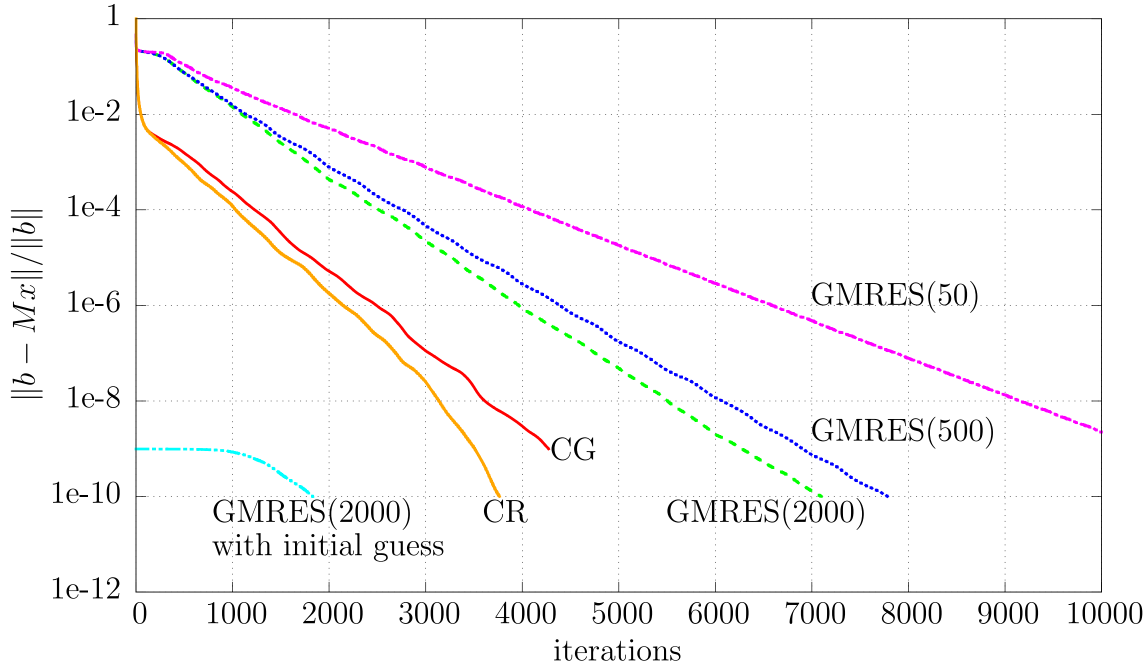


Figure 5.2: Comparison of Krylov space $\mathcal{K}_{2n}(M, b)$ (GMRES), $\mathcal{K}_{2n}(M_H, b)$ (CR), and $\mathcal{K}_n(M^\dagger M, b)$ (CG). All methods minimize the same norm $\|b - Mx\|^2$. "Iterations" in this context means the number of applications of M and $M^\dagger$. The number m in GMRES(m) is the restart frequency as defined above.

GMRES tests in this graph use Krylov space $\mathcal{K}_n(M, b)$. CG uses Krylov space $\mathcal{K}_n(M^\dagger M, b)$ while CR uses $\mathcal{K}_n(M_H, b)$. If we use M_H instead of M in GMRES then it produces a convergence curve the same as the one from CR (not shown in figure 5.2).

A few conclusions can be drawn from figure 5.2 for this particular ensemble,

1. M_H is a better operator to use on the lattice when generating Krylov spaces.

2. The asymptotic convergence rates for CG and GCGE are the same since they use the same Krylov space. From this one may hypothesize that different norms in the solver only effect the initial convergence behavior.

3. The Krylov space $\mathcal{K}_{2n}(M_H, r)$ (as in CR) is just slightly better than $\mathcal{K}_n(M^\dagger M, r)$ (as in CG).

And we can further conclude that among all methods tested, CR on the Hermitian operator M_H behaves the best. The family of CG-like methods performs more or less then same, however. Using GMRES on the Krylov space $\mathcal{K}_{2n}(M, b)$ shows that the solution obtained in this Krylov space is not as good as the Krylov space $\mathcal{K}_n(M^\dagger M, b)$ with respect to the norm

$$\|M(x_i - x^*)\|^2. \tag{5.57}$$

5.3 The Defect Correction Solver

Computers can not perform infinite precision arithmetic, instead one has to express all real numbers truncated to some predefined accuracy in the computer memory. Two common such choices are single precision and double precision representations[4]. Calculations in single precision format require less memory bandwidth and can potentially be faster (since less bits are involved), at the expense of reduced precision.

[4] On graphics processing units one also encounters a half-precision format, we will not discuss it here.

Defect correction solvers combine single and double precision calculations to maximize the efficiency without loss of precision. It involves 2 distinct steps,

1. Solve the Dirac equation using a single precision method up to some precision. Restarts are usually necessary since single precision has only roughly 7 decimal digits.

2. (Restart) Compute from the single precision solution and the source vector the new residue using double precision arithmetic. The algorithm either goes back to step 1 using the new residue as the right hand side vector, or stops if the residue is small enough.

Defect correction solvers show some interesting numerical behavior that is not obvious theoretically. Figure 5.3 presents a few examples showing the convergence curves of the defect correction solver on the $16^3 \times 32$ $(1.8\mathrm{fm}, 420\mathrm{MeV})$ ensemble.

One conclusion can be readily drawn from figure 5.3 is how restarts effect the solver. Restarting more frequently such as every 1000 single precision iterations clearly slows down the convergence rate. A simple explanation is that restarts discard information related to previously explored search directions, so the solver can search in previously visited directions again after a restart. Of course, we are forced to restart at some point because of limited precision of the single precision solver.

Another interesting aspect of the mixed precision solver is that the true residue diverges from the accumulated residue very quickly in the single precision solver. The accumulated residue has more or less a downward slope. While the true residue is actually *increasing* after some number of iterations. However, this does not necessarily indicate the breakdown of the solver. As shown by the figure, restarting the solver can usually make up for the loss in the true residue. In figure 5.3 this is manifested by the rapid drop of both the true residue and the accumulated residue soon after each restart. Restarts eliminate the differences between the true/accumulated residues and the convergence curve more or less

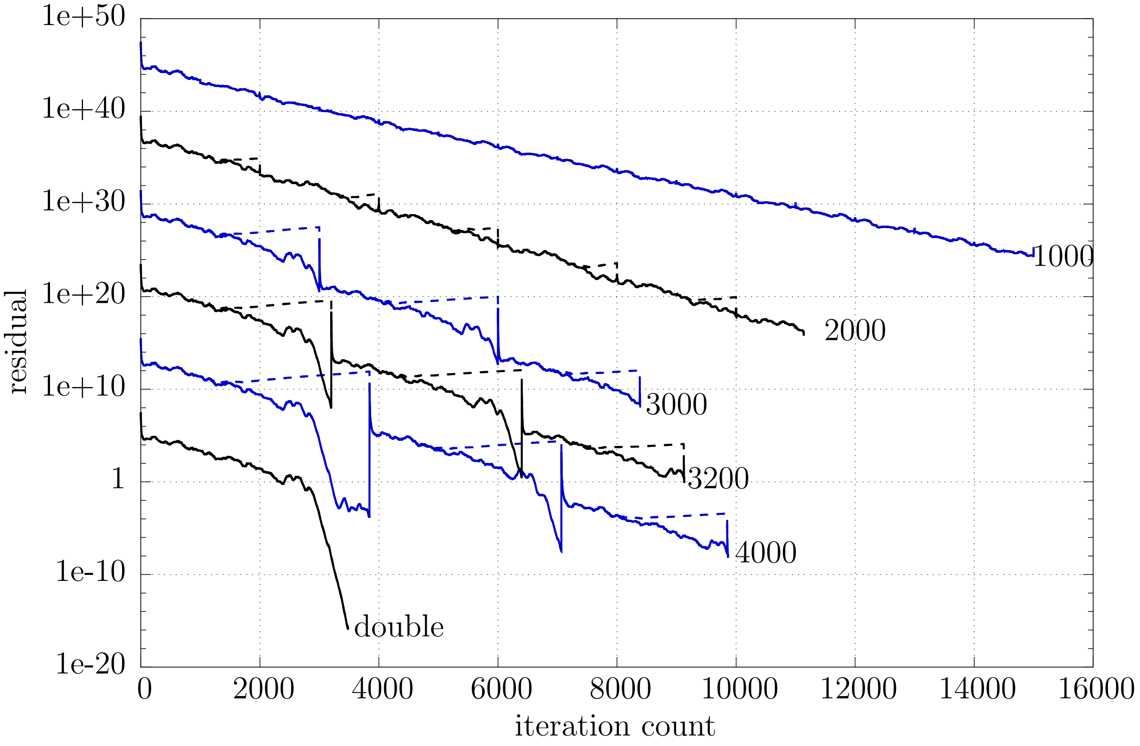


Figure 5.3: Convergence curve of the defect correction solver. Note that each curve is shifted vertically by some amount so multiple restart conditions can be displayed in this graph. The number to the right of each curve is the iteration bound for the single precision solver. The converge curve of a pure double precision solver (labeled as “double”) is shown here as a reference. A restart occurs every time the bound is reached. There are 2 curves for each restart condition. The solid line is the history of the accumulated residue and the dashed line is the true residue.

follows the accumulated residue (not the true residue) after each restart. This effect is more dramatic on larger lattices, as shown in figure 5.4.

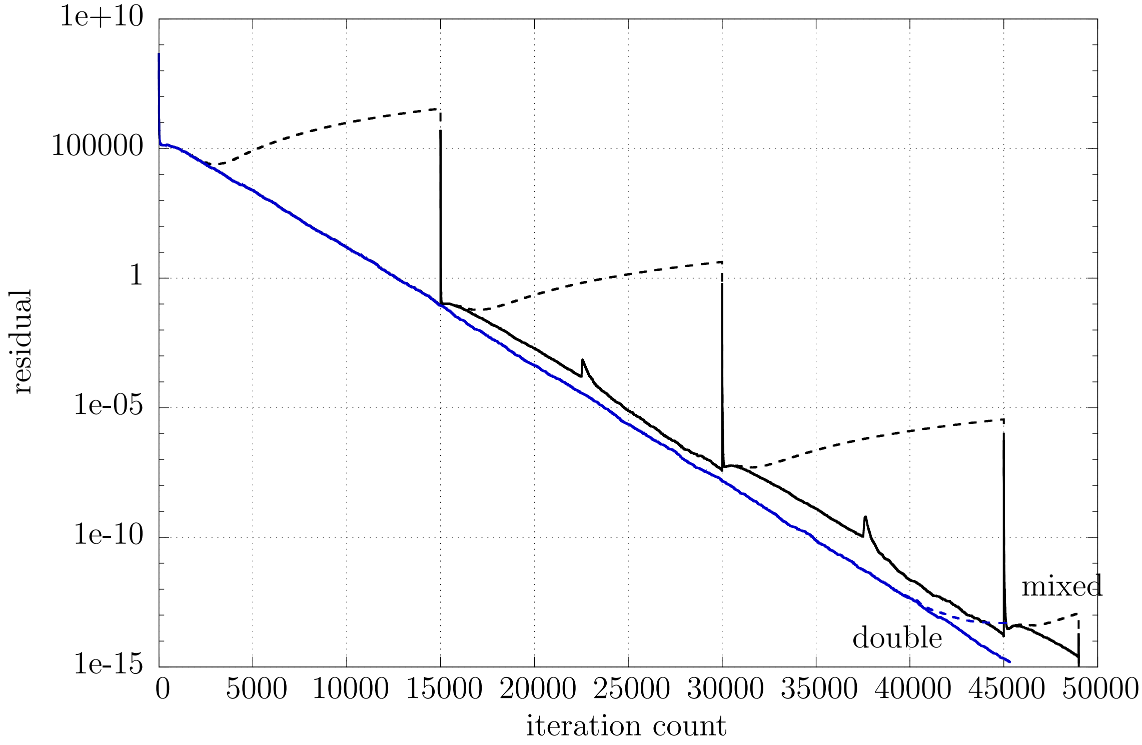


Figure 5.4: Convergence curve of the defect correction solver on the $48^3 \times 96$ (5.5fm, 140MeV) lattice. The single precision part restarts every 15000 steps. As in figure 5.3 the convergence curve of a pure double precision solver (labeled as "double") is shown here as a reference. In both defect correction solver and the double precision solver, the solid line shows the convergence of the accumulated residue and the dashed line shows the convergence of the true residue.

We can proceed even further and use the single precision solver as a preconditioner for the double precision solver. In principle this nested solver converges faster than simple defect-correction solver since it retains information of previous steps just like the conjugate gradient method. The result is shown in figure 5.5.

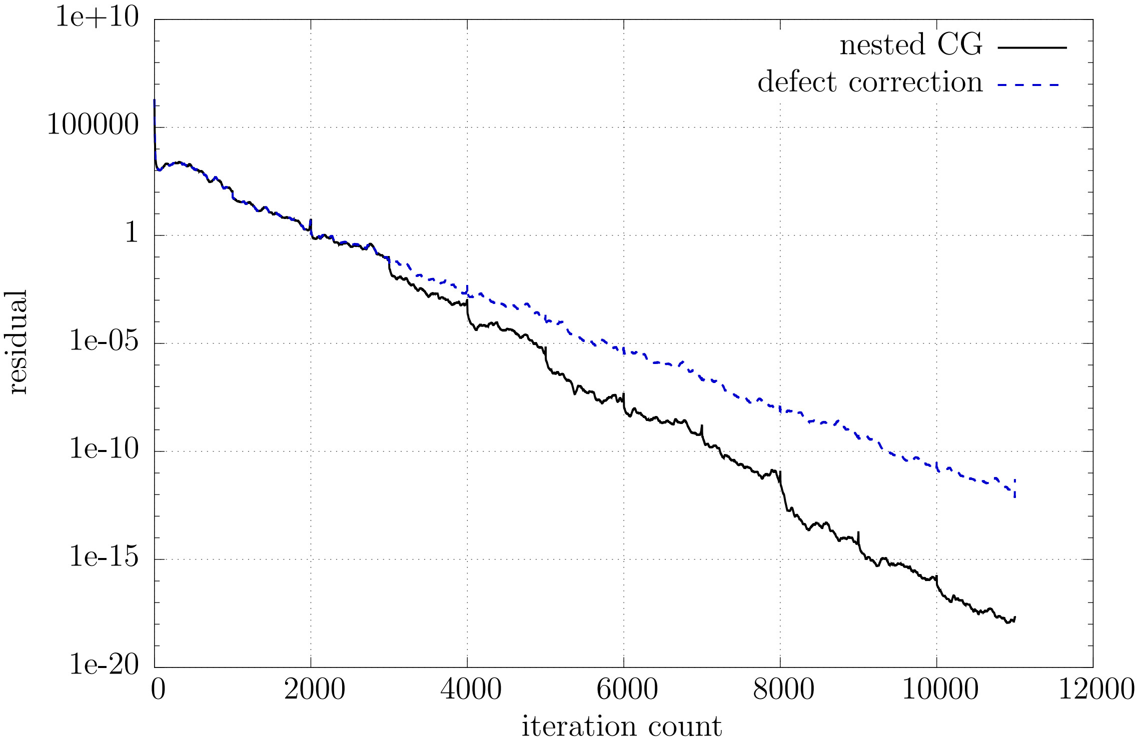


Figure 5.5: Nested CG solver tested on the $16^3 \times 32$ $(1.8\text{fm}, 420\text{MeV})$ lattice with input mass 0.001 in the solver. A single precision solver serves as a preconditioner for the double precision solver in the nested solver. In both nested CG and the defect correction solver single precision CG maximum iteration count is set to 1000.

5.4 Möbius Accelerated Domain Wall Fermion

The Möbius accelerated domain wall fermion approach is a method to produce approximate solutions for the domain wall fermion Dirac equations using the Möbius formalism. It exploits the similarities between Möbius fermions and the Shamir domain wall fermions.

As a transitional technique, Möbius accelerated domain wall fermion solvers were used extensively to find propagators for the Shamir domain wall fermions on the $32^3 \times 64$ $(4.6\text{fm}, 170\text{MeV})$ Shamir domain wall fermions ensemble and gained a factor of 2 speed up. Although we are using Möbius actions directly for the $48^3 \times 96$ $(5.5\text{fm}, 140\text{MeV})$ and $64^3 \times 128$ $(5.5\text{fm}, 140\text{MeV})$ ensembles, it is interesting in its own right. The development of the Möbius accelerated domain wall fermion solvers helps to understand the Möbius formalism itself and we shall describe the method here.

5.4.1 Method Description

The method eventually tries to solve the Dirac equation for *domain wall fermions.* We start by considering the general Dirac equation,

$$D_{DW}(m)x = b. \tag{5.58}$$

Where D_{DW} is the standard domain wall fermion Dirac operator. The goal is to use Möbius fermions to construct an alternative equation that approximates the one above. By solving the alternative equation we obtain an approximation to x, which can cost less than solving equation (5.58) directly to the same accuracy.

The primary tool we use to relate the standard domain wall fermion and the Möbius fermion is the domain wall - overlap transformation[31, 32, 33]. It relates the Möbius Dirac operator to an equivalent 4-dimensional overlap operator. Using this transformation, we first apply $P^{-1}D_{DW}^{-1}(1)$ to both sides of equation (5.58) to transform it into a lower triangular form in the s direction[5]

$$\begin{pmatrix} D_{OV} & & & & \\ S_1 S_2 \cdots S_{L-1}(D_{OV}-1)d & 1 & & & \\ S_2 \cdots S_{L-1}(D_{OV}-1)d & & 1 & & \\ \cdots & & & \cdots & \\ S_{L-1}(D_{OV}-1)d & & & & 1 \end{pmatrix} y = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \cdots \\ c_{L-1} \end{pmatrix}, \tag{5.59}$$

with $y = (y_0, y_1, y_2, \cdots, y_{L-1})^{\mathrm{T}} = P^{-1}x$. L is the size of s direction and

$$c = (c_0, c_1, c_2, \cdots, c_{L-1})^{\mathrm{T}} = P^{-1} D_{DW}(1)^{-1} b \tag{5.60}$$

is the right hand side vector of the new equation. The details, including the definition of symbols $S_0, \cdots, S_{L-1}$ and d, can be found in [31].

After this transformation solving the following 4-dimensional equation requires the most computational work,

$$D_{OV}(m) y_0 = c_0. \tag{5.61}$$

This is the subproblem located on $s = 0$ slice in equation (5.59). The rest of the work involves deriving the entire 5-dimensional solution from y_0. This part can be computed relatively easily, as will be shown below.

We first address the problem of solving the 4-dimensional equation (5.61). We notice that $D_{OV}(m)$ is an approximation to the ideal overlap operator. In parallel we can also obtain a different approximation $D'_{OV}(m)$ from any 5-dimensional Möbius operator $D'_{DW}(m)$. For our purpose we replace $D_{OV}(m)$ by $D'_{OV}(m)$. The solution y'_0 to the following equation

$$D'_{OV}(m) y'_0 = c_0 \tag{5.62}$$

[5] P is a simple matrix that shifts half of the spin components in the fermion vector by one in s direction. Its detailed form can be found in [31].

can be directly used to approximate y_0, and thus reconstruct a 5-dimensional approximated solution to (5.58).

To solve the 4-dimensional Möbius Dirac equation (5.62) we reverse the above process and transform it back into the 5-dimensional form. The detailed procedure can be found in [31]. By using the domain wall - overlap transformation on the 4-dimensional equation (5.62) we get the following 5-dimensional equivalent form

$$D'_{DW}(m)Py' = D'_{DW}(1)Pc' \tag{5.63}$$

with $c' = (c_0, 0, 0, \cdots, 0)$ and y'_0 being the 4-dimensional component of y' on the $s = 0$ hyperplane. By solving equation (5.63) we obtain y'_0 as an approximation to y_0.

To derive the entire 5-dimensional solution y from y'_0, we notice that for a true solution $y = (y_0, y_1, y_2, \cdots, y_{L-1})^{\mathrm{T}}$ the following relation holds,

$$y_k = c_k - S_k S_{k+1} \cdots S_{L-1}(D_{OV} - 1)dy_0,\ k = 1, 2, \cdots, L-1. \tag{5.64}$$

This relation can also be used to construct the guessed solution once y'_0 is obtained,

$$y'_k = c_k - S_k S_{k+1} \cdots S_{L-1}(D_{OV} - 1)dy'_0,\ k = 1, 2, \cdots, L-1, \tag{5.65}$$

where y'_k represents the approximation to y_k. This can be written in matrix form

$$\begin{pmatrix} -D_{OV}y'_0 \\ y'_1 \\ y'_2 \\ \cdots \\ y'_{L-1} \end{pmatrix} = P^{-1}D_{DW}(1)^{-1}D_{DW}(m)P \cdot \begin{pmatrix} -y'_0 \\ c_1 \\ c_2 \\ \cdots \\ c_{L-1} \end{pmatrix} \tag{5.66}$$

Reconstructing the entire 5-dimensional guess in this way requires solving a domain wall fermion equation with the mass equals 1.

The full algorithm involves the following 4 steps,

1. Construct c_0 from the original 5-dimensional source b, using equation (5.60).

2. Replace the 4-dimensional operator in equation (5.61) by a 4-dimensional operator from a Möbius fermion operator with appropriately chosen parameters.

3. Transform the 4-dimensional Möbius Dirac equation back to its 5-dimensional form (5.63) and solve the 5-dimensional equation, thus obtaining the 4-dimensional solution y_0' to equation (5.62).

4. Use y_0' as an approximation to the 4-dimensional domain wall Dirac equation (5.61), and reconstruct the 5-dimensional approximated solution using (5.66).

The above process requires solving two domain wall fermion equations (5.60) and (5.66) with quark mass equal to 1 (the Pauli-Villars equations) and also the Möbius equation (5.63). The gain over the direct method comes from the fact that the Möbius equation has a smaller size in the s direction. Once an approximate solution y_0' is produced in this way, we apply the standard conjugate gradient algorithm to the original problem (5.58), using y_0' as the initial guess. This step converges quickly and guarantees that the final solution has the desired accuracy.

It is clear that the approximated solution y_0' must be sufficiently close to the true solution y_0 to be beneficial. This leads to the problem of choosing the best parameters.

5.4.2 Choosing the Optimal Möbius Parameters

We found that the choice of b_i and c_i values strongly effects the quality of the Möbius approximation y_0' in (5.62). In what follows we will consider only the polar decomposition for the Möbius Dirac operator, such that b_i and c_i are constants throughout the 5th dimension and we will simply refer to them as b and c instead.

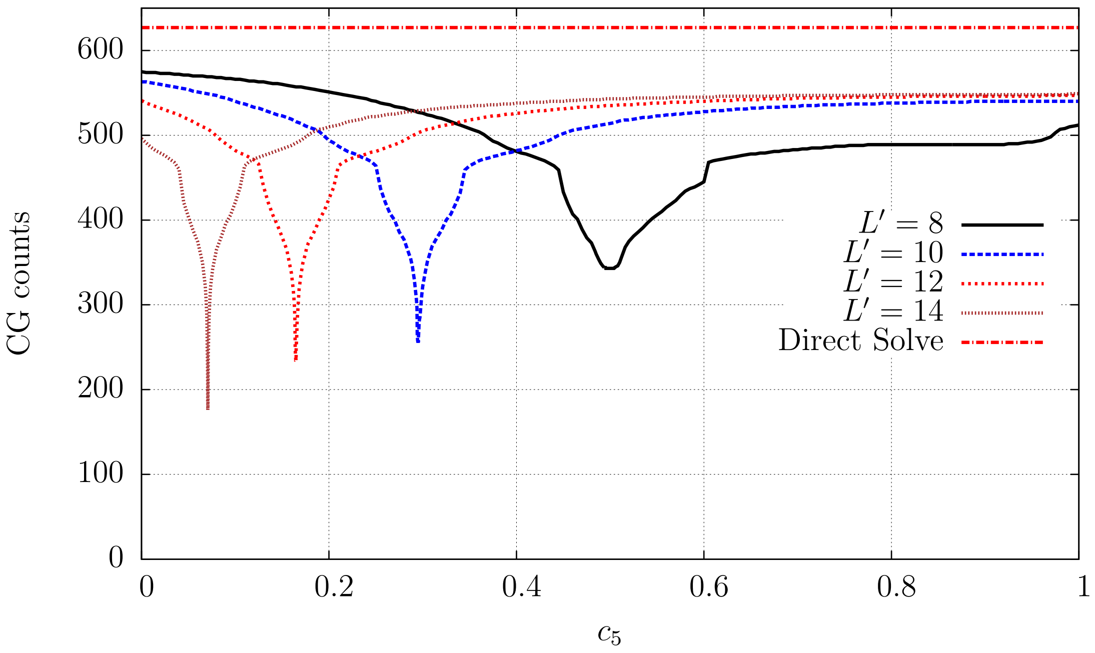


Figure 5.6: Iteration count of the conjugate gradient method started from a Möbius approximated solution, plotted as a function of c. The residual is set to 1e-8. $L' = 8, 10, 12, 14$ Möbius lattice are used to approximate $L = 16$ DWF. In all cases $b = c+1$. Optimal c values are obtained from a separate nonlinear search. The optimal values are: $L' = 8$, $c = 0.4996$. $L' = 10$, $c = 0.2949$. $L' = 12$, $c = 0.1647$. $L' = 14$, $c = 0.0708$.

Let L be the lattice size in s direction for DWF, L' be the corresponding value for the Möbius Dirac operator. For standard DWF the corresponding overlap operator is

$$D_{\text{OV,DWF}}(m) = \frac{1}{2}\left(1 + m + (1-m)\gamma_5 \frac{(x+1)^L - (x-1)^L}{(x+1)^L + (x-1)^L}\right). \tag{5.67}$$

with the kernel x equal to

$$x = \gamma_5 D_w \frac{1}{2 + D_w}. \tag{5.68}$$

the same approximation in Möbius is

$$D_{\mathrm{OV,Möbius}}(m) = \frac{1}{2}\left(1 + m + (1-m)\gamma_5 \frac{(\lambda x' + 1)^{L'} - (\lambda x' - 1)^{L'}}{(\lambda x' + 1)^{L'} + (\lambda x' + 1)^{L'}}\right). \tag{5.69}$$

with $\lambda = b + c$. The kernel x' is equal to

$$x' = \gamma_5 D_w \frac{1}{2 + (b-c)D_w}. \tag{5.70}$$

It is thus desirable to keep $b - c$ equal to 1 so that $x = x'$. Also, matching up to the first order term in the Taylor expansions of the two functions (5.67, 5.69) results the requirement $\lambda = L/L'$.

Figure 5.6 presents a scan of c values and the quality of the approximated solution y'. The quality of the approximated solution is measured by the number of the conjugate gradient iterations required to achieve a fixed stopping condition, starting from the Möbius initial guess. It is clear that $b + c = L/L'$ proves to be a good approximation. However, the iteration count depends sharply on c near the optimal point. As a consequence fine tuning of b and c is almost always required.

5.4.3 Defect Correction and Performance in Production

The Möbius approximation to the DWF equation has an intrinsic accuracy. Figure 5.6 shows that a direct method such as a Krylov space solver is required to solve the equation to the designated accuracy after adopting the Möbius initial guess. As a consequence the Möbius equation (5.63) need not be solved very accurately.

Figure 5.7 shows the quality of the Möbius approximation as a function of the accuracy used when solving equation (5.63). We use the conjugate gradient method to solve the equation (5.58) to an accuracy of 10^{-10}. The quality of the approximation is measured by the number of the conjugate gradient iterations to solve equation (5.58), starting from the

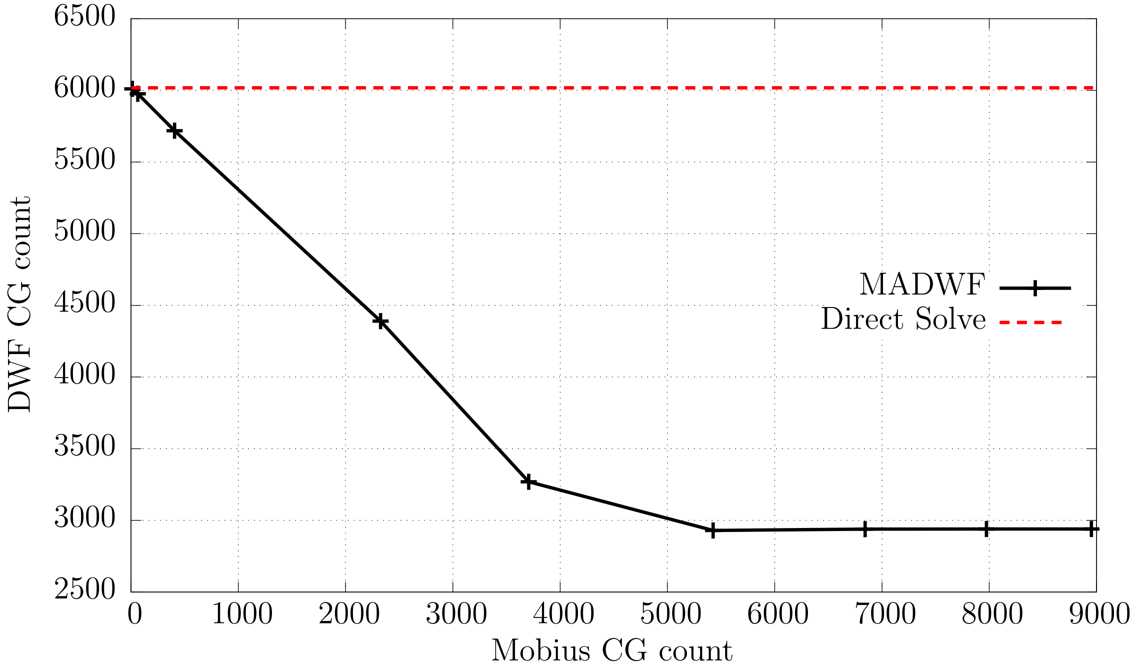


Figure 5.7: Iteration count from solving the DWF equation shown as a function of the iteration count from solving the Möbius equation. Both equations use the conjugate gradient method. The domain wall fermion action has $L = 32$ and the Möbius action has $L' = 16$, The parameters for the Möbius action are $b = 1.5$ and $c = 0.5$.

Möbius initial guess. The accuracy used when solving equation (5.63) is measured by the number of iterations when applying the conjugate gradient method to the Möbius equation. A turning point can be clearly seen on the graph. Before the turning point the number of the conjugate gradient iterations when solving equation (5.58) decreases linearly as the accuracy used to solve equation (5.63) improves. After the tuning point, the quality of the Möbius approximation does not improve further. So in practice the Möbius problem needs only to be solved up to or less than this accuracy. Higher accuracy can be achieved by using a simple defect-correction procedure, just like the mixed precision solver described earlier.

We present a production use of this method with the defect correction procedure on the $32^3 \times 64$ $(4.6\text{fm}, 170\text{MeV})$ ensemble. This lattice uses the Iwasaki+DSDR gauge action. We compare it with the direct conjugate gradient method in table 5.2. The corresponding Möbius approximation uses $L' = 12$. To make the comparison clear we count the number of 4-dimensional Wilson Dirac operator applications (denoted by "Op. count" in the table), since standard DWF and Möbius fermion use different 5th dimension sizes. The table shows that the Möbius accelerated DWF (MADWF) reduces the number of 4-dimensional Wilson Dirac operator applications from 3.6e5 to 2.0e5, by a factor of 1.8. The MADWF solver uses 42% of wall clock time, when compared with the direct method. Note that while some gain is from code optimization issues, most is due to the fact that MADWF requires many fewer 4-dimensional Wilson Dirac operations.

	Direct method	Möbius Accelerated DWF			
Operation	-	Möbius equation(*3)	Pauli-Villars(*6)	others	total
Op. Count	11290*32=3.6e5	4.6e3*12	1.0e2*32	-	2.0e5
time(s)	2672	285	25	125	1138

Table 5.2: Cost comparison of the MADWF solver with the direct conjugate gradient method. $L = 32$, $L' = 12$, with $b = 1.841556$, $c = 0.841556$. The stopping condition is set to 10^{-10}. Note that there are 3 Defect-correction steps, each includes 2 Pauli-Villars unit mass inversions. The factor 32 and 12 in the "Op. Count" row are due to different 5th dimension sizes.

5.4.4 Parameter Usability

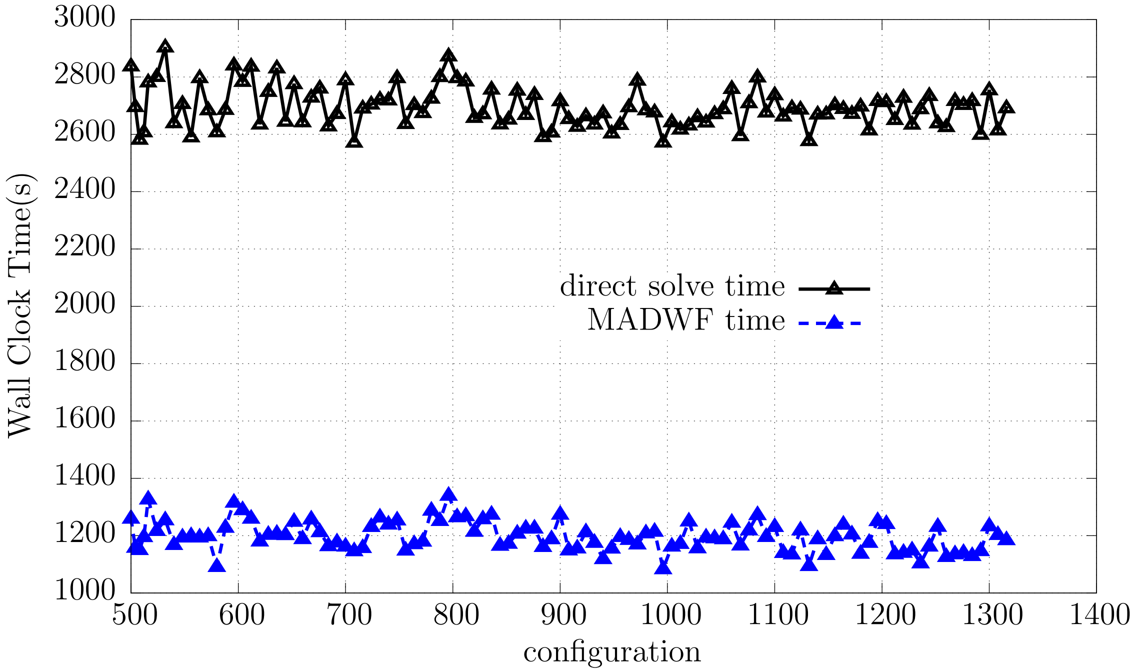


Figure 5.8: Parameter usability test: The black and blue triangles show the comparison of the wall clock time of the solvers. Both solve to 1e-10 for all configurations. A wall source is used.

It is of course impracticable to tune the Möbius parameters L', b and c for each configuration used. However, we found in practice that a set of well tuned parameters can be used for any configuration from the same ensemble.

We present in figure 5.8 the plot of wall clock time of the direct method and the MADWF solver for 100 configurations from the same calculation as used in table 5.2. The Möbius parameters are kept the same throughout the calculation. This includes using the same L', b, c, restart count and stopping condition settings. It is clear that both the total wall clock time of the Möbius accelerated solver show little variation across the 100 configurations tested.

5.5 The EigCG Algorithm

The eigCG algorithm is a method to accelerate Krylov space solvers when the same large sparse matrix needs to be solved for multiple right hand side vectors (in the language of lattice calculations, sources). In lattice applications it is a very common practice to solve multiple source vectors using the same Dirac operator. For example, we must solve 12 Dirac equations to obtain a quark propagator, since quarks have 4 spins and 3 colors. In addition, we may need/want to solve more than one quark propagator on a single lattice configuration, yielding even more right hand side vectors.

Obviously if we want to accelerate this process, some information must be shared when solving different right hand side vectors. Different views on what is to be shared lead to different algorithms. In lattice applications we have found that sharing the eigenvectors with near zero eigenvalues is a very good strategy. The eigCG algorithm [34] does exactly this.

The eigCG algorithm builds an explicitly restarted Lanczos algorithm on top of the conjugate gradient method. It does not interfere with the original conjugate gradient algorithm, nor does it require additional matrix-vector multiplications. The idea is to reuse the search directions generated by the conjugate gradient algorithm to compute the necessary Lanczos vectors. Since the Lanczos algorithm is numerically unstable, it also restarts itself when a certain number of eigenvector/eigenvalue pairs are obtained. Once we obtain the near zero part of the spectrum of the sparse matrix, we can use them to deflate the search space of the Krylov space solver, thus accelerating the solver.

The details of the algorithm can be found in [34]. Here we present figures showing its applications on our current lattices. Figure 5.9 shows the convergence history of the conjugate gradient algorithm on our $64^3 \times 128$ $(5.5\text{fm}, 140\text{MeV})$ ensemble with physical quark masses. This calculation is very expensive in its original form because the preconditioned Dirac operator $\mathcal{M}$ has a very large condition number due to a very small quark mass. To speed up

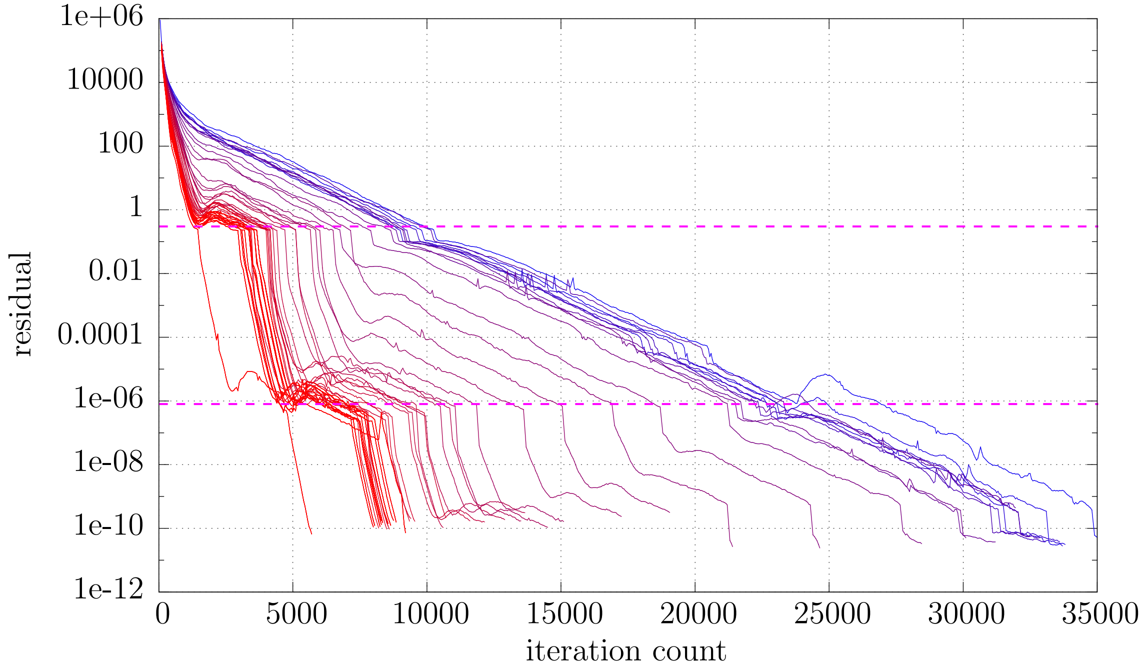


Figure 5.9: Convergence history of eigCG on ensemble $64^3 \times 128$ (5.5fm, 140MeV). Each curve corresponds to a conjugate gradient solve. Blue curves are earlier solves and red ones occur latter. The eigCG algorithm accumulates low mode information while we continue to solve Dirac equations, thus reducing the iteration count of latter conjugate gradient applications. 2 magenta lines are placed in the figure to show the locations where we restart the mixed precision solver.

the calculation we build an eigCG algorithm on top of the conjugate gradient solver. The figure plots the residue of the conjugate gradient solver (5.30) as a function of the iteration count. Each line tracks a conjugate gradient application. We use different colors to label different solves. Among them blue curves are earlier solves while latter solves are gradually tinted towards red. The eigCG algorithm accumulates the low mode information of the Dirac operator and passes this information from earlier applications of CG to latter applications of CG, thus reducing the cost as we solve more Dirac equations.

The solver in figure 5.9 is in fact a mixed precision conjugate gradient solver. 2 horizontal magenta lines are added to the figure showing the approximate location where we restart the solver. Most of the computation, including the entire eigCG algorithm operates in single precision arithmetic. We only use double precision arithmetic to compute the defect (a.k.a. the new right hand side) at each restart. For this reason and also the nature of eigCG, the low modes accumulated are inexact. Consequently, starting from each restart the conjugate gradient method initially converges rapidly since the condition number is improved by low modes deflation. After a certain point the conjugate gradient method “feels” the effect of the inexact low modes and the convergence rate is reduced.

It can also be seen in the figure that restarts cure this problem. Initially and at each restart, we deflate the CG using the low mode information available at that point. The convergence rate immediately following each restart is always better than the slow convergence rate of the original CG.

5.6 All to All Propagators

Due to the difficulty of inverting the entire Dirac operator D, conventionally we compute the fermion propagators by placing a source b on the lattice and compute $D^{-1}b$. Clearly this approach is asymmetric in its handling of source and sink - there is only one source but any sink r can be easily obtained by forming the inner product $r^\dagger D^{-1}b$.

Unlike the conventional source-sink approach, the all to all propagator intends to approximate the exact Dirac propagator

$$S(x, y) = D^{-1}(x, y) \tag{5.71}$$

for all possible combinations of x and y, hence the name "all to all" propagators. If we arrange $S(x, y)$ such that x and y are the row and column indices then we get the matrix D^{-1}. The all to all propagator tries to construct an approximation to *the entire matrix* D^{-1}.

Suppose we have N_{h} 4 dimensional random vectors w_i $(i = 0, 1, \cdots, N_{\mathrm{h}} - 1)$ such that

$$\frac{1}{N_{\mathrm{h}}} \sum_{i=0}^{N_{\mathrm{h}}-1} w_i w_i^{\dagger} \approx I, \tag{5.72}$$

i.e., they form a stochastic approximation of the identity matrix I. This can be done in a variety of ways. For example, each element of w_i can be drawn independently from random distributions such as $\mathbb{Z}_2$, $\mathbb{Z}_4$ or $U(1)$. Using the above approximation, we have

$$D^{-1} \approx \frac{1}{N_{\mathrm{h}}} \sum_{i=0}^{N_{\mathrm{h}}-1} \left(D^{-1} w_i\right) w_i^{\dagger}. \tag{5.73}$$

Define $v_i = D^{-1} w_i / N_{\mathrm{h}}$, $(i = 0, 1, \cdots, N_{\mathrm{h}} - 1)$. We can then approximate D^{-1} in the following way

$$D^{-1} \approx \sum_{i=0}^{N_{\mathrm{h}}-1} v_i w_i^{\dagger}. \tag{5.74}$$

The vectors v_i and w_i can be used whenever a fermion propagator is needed. For example, a pion propagator can be approximated as

$$C_\pi(x,y) = \mathrm{Tr}\left(D^{-1}(x,y)\gamma_5 D^{-1}(y,x)\gamma_5\right) \tag{5.75}$$

$$\approx \sum_{i,j=0}^{N_\mathrm{h}-1} \mathrm{Tr}\left(v_i(x)w_i^\dagger(y)\gamma_5 v_j(y)w_j^\dagger(x)\gamma_5\right) \tag{5.76}$$

$$= \sum_{i,j=0}^{N_\mathrm{h}-1} w_i^\dagger(y)\gamma_5 v_j(y)\cdot w_j^\dagger(x)\gamma_5 v_i(x). \tag{5.77}$$

Now it is easy to sum over x and y spatially,

$$\sum_{x,y} C(x,t;y,t') = \sum_{ij}\left(\sum_y w_i^\dagger(y,t')\gamma_5 v_j(y,t')\right)\left(\sum_x w_j^\dagger(x,t)\gamma_5 v_i(x,t)\right). \tag{5.78}$$

For convenience we define the *meson field* matrix at time slice t to be Π^t

$$\Pi_{ij}^t = \sum_x w_i^\dagger(x,t)\gamma_5 v_j(x,t), \tag{5.79}$$

so the pion correlator takes the following form

$$\sum_{x,y} C(x,t;y,t') = \sum_{ij}\Pi_{ij}^{t'}\Pi_{ji}^t = \mathrm{Tr}\left(\Pi^{t'}\Pi^t\right). \tag{5.80}$$

To improve the approximation (5.73) we usually separate N_l low modes of D and only approximate the rest using random matrices. Suppose h_i and λ_i are the ith normalized low mode of D and the corresponding eigenvalue, $i = 0, 1, \cdots, N_\mathrm{l}-1$. These eigenvectors and eigenvalues can be solved using methods such as the Lanczos algorithm. Once solved, we can construct the low modes part of D^{-1} exactly and use the random vectors to approximate only the rest part. In such a case, v_i is computed using the following formula

$$v_i = \left(D^{-1} - \sum_{i=0}^{N_\mathrm{l}-1}\frac{h_i h_i^\dagger}{\lambda_i}\right) w_i, \tag{5.81}$$

and D^{-1} is approximated in the following way

$$D^{-1} \approx \sum_{i=0}^{N_\mathrm{l}-1} \frac{h_i h_i^\dagger}{\lambda_i} + \sum_{i=0}^{N_\mathrm{h}-1} v_i w_i^\dagger. \tag{5.82}$$

Further complications can occur with the above formalism. For example, computing the low end spectrum of D can be slow and usually its even odd preconditioned form (2.29) is preferred. In such cases, there are additional matrices that must be taken care of when constructing vectors in the all to all formalism.

5.6.1 Example: $\pi\pi$ scattering

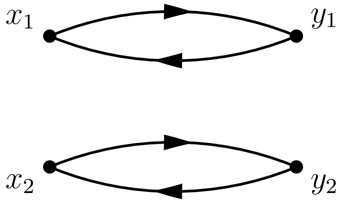


Figure 5.10: One of the two “D” diagrams in the two pion scattering process. Each dot has a γ_5 matrix associated with it.

There is an additional subtlety related to the arrangement of the random vector w_i when forming contractions. Take the calculation of the D diagram in $\pi\pi$ scattering as an example. Suppose we calculate the correlation function of the D diagram that has 2 pions going from x_1, x_2 to y_1, y_2 (all are 4D coordinates)

$$D(x_1, x_2; y_1, y_2) = \mathrm{Tr}\left[\gamma_5 D^{-1}(y_1, x_1)\gamma_5 D^{-1}(x_1, y_1)\right] \mathrm{Tr}\left[\gamma_5 D^{-1}(y_2, x_2)\gamma_5 D^{-1}(x_2, y_2)\right] \tag{5.83}$$

(here we omit an alternative diagram that has y_1 and y_2 exchanged). Using the all to all propagator, we insert a set of random vectors at each point, each set is meant to approximate the identity matrix as in (5.72),

$$\begin{aligned}&\sum_{ijkl}\mathrm{Tr}\left[\gamma_5 D^{-1}(y_1,x_1')w_i(x_1')w_i^\dagger(x_1)\gamma_5 D^{-1}(x_1,y_1')w_j(y_1')w_j^\dagger(y_1)\right]\\&\quad\mathrm{Tr}\left[\gamma_5 D^{-1}(y_2,x_2')w_k(x_2')w_k^\dagger(x_2)\gamma_5 D^{-1}(x_2,y_2')w_l(y_2')w_l^\dagger(y_2)\right].\end{aligned} \tag{5.84}$$

If

$$\begin{aligned}&\sum_{ijkl} w_i(x_1')w_i^\dagger(x_1)w_j(y_1')w_j^\dagger(y_1)w_k(x_2')w_k^\dagger(x_2)w_l(y_2')w_l^\dagger(y_2)\\&\approx\delta(x_1'-x_1)\delta(y_1'-y_1)\delta(x_2'-x_2)\delta(y_2'-y_2),\end{aligned} \tag{5.85}$$

then equation (5.84) would be a correct stochastic approximation of $D(x_1,x_2;y_1,y_2)$. Unfortunately equation (5.85) does not hold. In fact

$$\begin{aligned}&\sum_{ijkl} w_i(x_1')w_i^\dagger(x_1)w_j(y_1')w_j^\dagger(y_1)w_k(x_2')w_k^\dagger(x_2)w_l(y_2')w_l^\dagger(y_2)\\&\approx\delta(x_1'-x_1)\delta(y_1'-y_1)\delta(x_2'-x_2)\delta(y_2'-y_2)\\&+\delta_{ij}\delta_{kl}\delta(x_1'-y_1)\delta(y_1'-x_1)\delta(x_2'-y_2)\delta(y_2'-x_2)\\&+\delta_{ik}\delta_{jl}\delta(x_1'-x_2)\delta(x_2'-x_1)\delta(y_1'-y_2)\delta(y_2'-y_1)\\&+\delta_{il}\delta_{jk}\delta(x_1'-y_2)\delta(y_2'-x_1)\delta(y_1'-x_2)\delta(x_2'-y_1).\end{aligned} \tag{5.86}$$

Although the additional terms in (5.86) are in general suppressed because of the extra factor like $\delta_{ij}\delta_{kl}$, they can not be simply neglected for practical calculations. To cure the problem, we either guarantee that it is never possible to have i, j, k and l equal each other, or that it is impossible to have $x_1' = y_1$(among other similar equations). This implies that in the summation we avoid using the same random source for 2 points that are possibly at the same spacetime location.

5.6.2 Using Extended Sources

The major advantage of the all to all propagators is that we can use a variety of extended sources. Suppose we use the following very general operator to create a pion with total momentum $p - p'$

$$\mathcal{O}^\dagger(t;p,p') = \sum_{xy} \bar{l}(x,t) f(x-y)\gamma_5 l(y,t) e^{ipx - ip'y}, \tag{5.87}$$

$$\mathcal{O}(t;p,p') = \sum_{xy} \bar{l}(y,t) f(x-y)\gamma_5 l(x,t) e^{ip'y - ipx}, \tag{5.88}$$

where the source $f(x-y)$ is symmetric

$$f(x-y) = f(y-x). \tag{5.89}$$

A pion correlator using the above interpolating operator looks like the following

$$\begin{aligned}&\left\langle \mathcal{O}(t_2;p_2,p_2')\mathcal{O}^\dagger(t_1;p_1,p_1')\right\rangle \\ =&\sum_{xyx'y'} \mathrm{Tr}\left(f(x'-y')\gamma_5 e^{-ip_2x'} D^{-1}(x',t_2;x,t_1) e^{ip_1x} \cdot f(x-y)\gamma_5 e^{-ip_1'y} D^{-1}(y,t_1;y',t_2) e^{ip_2'y'} \right).\end{aligned} \tag{5.90}$$

Using all to all propagators, the central problem is to calculate the meson field

$$\Pi_{ij}^{tpp'} = \sum_{xy} w_i^\dagger(x,t) f(x-y)\gamma_5 e^{ipx - ip'y} v_j(y,t). \tag{5.91}$$

$\Pi_{ij}^{tpp'}$ can be easily computed using Fourier transform. This is because the source $f(x-y)$ depends only on $x-y$, not the individual coordinates x or y. Suppose the Fourier transform of $f(x)$ is $\widetilde{f}(p)$, then

$$\Pi_{ij}^{tpp'} =\frac{1}{V}\sum_{xyq} w_i^\dagger(x,t)\widetilde{f}(q)e^{iq(x-y)}\gamma_5 v_j(y,t)e^{ipx-ip'y} \tag{5.92}$$

$$=\frac{1}{V}\sum_{q} \widetilde{w}_i^\dagger(q+p,t)\widetilde{f}(q)\gamma_5\widetilde{v}_j(q+p',t). \tag{5.93}$$

The formula for the final pion correlator remains unchanged,

$$\left\langle \mathcal{O}(t_2;p_2,p_2')\mathcal{O}^\dagger(t_1;p_1,p_1')\right\rangle = \sum_{ij}\Pi_{ij}^{t_1p_1p_1'}\Pi_{ji}^{t_2p_2'p_2}. \tag{5.94}$$

5.7 All Mode Averaging

All mode averaging is an error reduction technique for lattice calculations [35]. Similar to the Hasenbusch mass splitting, the use of the all mode averaging method is characterized by dividing the calculation into the following two parts:

- a part that is easy to calculate but has significant impact on the size of the overall error, and
- a part that is difficult to calculate but has relatively small contribution to the overall errors.

By calculating the easy to calculate part more frequently we achieve higher accuracy without incurring excessive overhead.

Suppose the desired lattice observable is $\mathcal{O}$. Without loss of generality we assume it to be a function of the gauge field $U_\mu(x)$

$$\mathcal{O} = \mathcal{O}\left[U_\mu(x)\right]. \tag{5.95}$$

The idea of all mode averaging involves finding an approximation of $\mathcal{O}$ and rewrite it as the sum of the following two parts

$$\mathcal{O} = \mathcal{O}' + \Delta\mathcal{O}. \tag{5.96}$$

The intention is that $\mathcal{O}'$ is easy to calculate and will be improved by multiple measurements. $\Delta\mathcal{O}$ is more computationally demanding but does not contribute significantly to the overall error. So it will be computed less frequently.

Since $\mathcal{O}'$ will be calculated more frequently one must ensure that the computed $\Delta\mathcal{O}$ is an unbiased estimation for a $\Delta\mathcal{O}$ associated with *any* $\mathcal{O}'$, should this $\Delta\mathcal{O}$ be calculated. To achieve this we can find a transformation group G such that both the lattice action and the *ensemble average* of $\mathcal{O}$ and $\mathcal{O}'$ are invariant,

$$\begin{aligned} \langle \mathcal{O}\left[U_\mu(x)\right] \rangle &= \left\langle \mathcal{O}\left[U_\mu^g(x)\right] \right\rangle, \\ \langle \mathcal{O}'\left[U_\mu(x)\right] \rangle &= \left\langle \mathcal{O}'\left[U_\mu^g(x)\right] \right\rangle. \end{aligned} \tag{5.97}$$

Instead of measuring (5.96), we measures the following quantity to have a better estimation of $\mathcal{O}$

$$\mathcal{O}_{AMA} = \frac{1}{N}\sum_{g\in G} \mathcal{O}' \circ g + \Delta\mathcal{O}, \tag{5.98}$$

where N is the total number of group elements used. The sum can be performed either on the entire transformation group or its subset. We discuss two examples of the all mode averaging method in the following.

5.7.1 Low Mode Averaging

Low mode averaging constructs $\mathcal{O}'$ by replacing all (or some) instances of D^{-1} with its low modes counterpart

$$D^{-1} \approx \sum_i \frac{1}{\lambda_i} e_i e_i^\dagger. \tag{5.99}$$

Where $(\lambda_i, e_i), i = 1, 2, \cdots$ are the low eigenvalue/eigenvector pairs of the Dirac operator D. The low mode reconstructed parts $\mathcal{O}'$ is then shifted to various time slices.

For example, suppose we want to calculate the pion correlator

$$\mathcal{O} = \mathrm{Tr}\left(D^{-1}(x,0)D^{-\dagger}(x,0)\right). \tag{5.100}$$

The 2 corresponding parts using low mode averaging would be

$$\mathcal{O}' = \mathrm{Tr}\left(\sum_i \frac{1}{\lambda_i} e_i(x) e_i^\dagger(0) \cdot \sum_j \frac{1}{\lambda_j} e_j(0) e_j^\dagger(x)\right) \tag{5.101}$$

$$\Delta\mathcal{O} = \mathcal{O} - \mathcal{O}'. \tag{5.102}$$

The low modes part is then translated and measured multiple times on the lattice.

5.7.2 Inexact Propagator Approximation

The other common cases of all mode averaging involves using inexact (less accurate) propagators to construct $\mathcal{O}'$ and the inexact approximations are translated into various lattice locations. In this case the transformation group G is the translation group on the lattice. Suppose the exact propagators are $S_F\left[U_\mu(x)\right]$ and the inexact propagators are $S'_F\left[U_\mu(x)\right]$, then

$$\mathcal{O} = \mathcal{O}\left[S_F\left[U_\mu(x)\right]\right] \tag{5.103}$$

$$\mathcal{O}' = \mathcal{O}\left[S'_F\left[U_\mu(x)\right]\right] \tag{5.104}$$

$$\Delta\mathcal{O} = \mathcal{O}\left[S_F\left[U_\mu(x)\right]\right] - \mathcal{O}\left[S'_F\left[U_\mu(x)\right]\right]. \tag{5.105}$$

$\Delta\mathcal{O}$ is expensive to calculate since it involves exact propagators. $\mathcal{O}'$ is simpler to compute since it can be calculated via the conjugate gradient algorithm with a larger stopping condition. $\mathcal{O}'$ is then measured multiple times to improve the statistics. The overall scheme thus involves the following,

- $\Delta\mathcal{O}$ is expensive to calculate and it is computed once (or a few times) by placing the source of the exact/inexact propagators on some specific time slice,
- $\mathcal{O}'$ is less computationally demanding. We can reduce the error by measuring it multiple times. This is done via translating the source of the inexact propagators to multiple time slices and using the average of the measured $\mathcal{O}'$ as an estimation of this part.

A subtle problem with such a scheme is how the inexact propagators are obtained. We usually solve light quarks as inexact propagators since this is the expensive part of most calculations. Practically one always uses low modes projection when solving these inexact propagators. Low modes serve two purposes in the calculation

1. The Krylov space solvers are much faster when the low spectrum of the Dirac operator are projected out.
2. Practically we find that $\mathcal{O}'$ approximates $\mathcal{O}$ better when low mode deflation is included in the calculation, if other conditions are kept the same (mainly the stopping condition)[6].

For this reason, all mode averaging using inexact propagators is always combined with some low modes projection technique. The low modes of the Dirac operator can be obtained using methods such as the Lanczos algorithm or its variants such as the eigCG algorithm.

Another subtle problem arises because of the even odd preconditioning. Even odd preconditioning distinguishes between even sites and odd sites, this may violate equation (5.97). For example wall sources on even t slices and odd t slices are multiplied by very different

[6] This is probably because CG has higher tolerance on errors associated with low modes, since errors are weighted according to the eigenvalues. By supplying the sloppy CG with low modes we explicitly remove this part of error. Thus the overall error in the solution is reduced even with the same residual.

matrices when the lattice Dirac equation is transformed into the even odd preconditioned form.[7] In such cases an estimation of $\Delta\mathcal{O}$ made with the wall source on an even time slice can be a biased estimation of the same quantity with the wall source on an odd time slice.

A simple method to fix this problem is to shift the location of where $\Delta\mathcal{O}$ is calculated by a random number in t direction. The problem is solved if t takes even or odd values with equal probabilities. The ensemble average of the all mode averaging observable $\mathcal{O}_{AMA}$ in this scenario is an unbiased estimation of the original observable, though bias can exist for each configuration.

[7] In fact they are multiplied by the same 4 dimensional matrix. But the result will be very different since the two sources have their nonzero parts on even/odd time slices respectively.

Chapter 6

$Kl3$ Calculation on the Lattice

The interest of computing the process $K \longrightarrow \pi l \nu$ stems from the interest in determining the CKM matrix element V_{us}. Combined with experimental data, lattice calculations of this process can be used to determine the CKM matrix element V_{us} to very high accuracy. The decay width of this process can be expressed in terms of a few physical parameters, including the Fermi coupling constant G_F, the CKM matrix element V_{us}, and so on [2]. There are two unknown factors among all these parameters,

$$\Gamma_{Kl3} \propto |V_{us}|^2 |f^{+}_{K\pi}(0)|^2, \tag{6.1}$$

where $f^{+}_{K\pi}(q^2)$ is the form factor of the QCD process $K \longrightarrow \pi$ via a vector current vertex,

$$\langle \pi(p_\pi) \,|\, \bar{s}\gamma_\mu u \,|\, K(p_K) \rangle = (p_\pi + p_K)_\mu f^{+}_{K\pi}(q^2) + (p_K - p_\pi)_\mu f^{-}_{K\pi}(q^2), \tag{6.2}$$

where

$$q = p_K - p_\pi. \tag{6.3}$$

V_{us} is the target that can be calculated once the total decay width Γ_{Kl3} and $f^{+}_{K\pi}(0)$ are known. The decay width Γ_{Kl3} is determined from experimental data. On the other hand, the form factor at zero momentum transfer $f^+_{K\pi}(0)$ is a suitable target for lattice calculations.

There are numerous previous attempts on the lattice to calculate the form factor $f^+_{K\pi}(0)$, using various lattice sizes and parameter settings. The combined fit has lead to a good result for $f^+_{K\pi}(0)$ and thus V_{us} [1]. Although a significant success, there are two shortcomings in these calculations.

1. The previous calculations are performed using unphysical parameters. Due to the enormous cost of simulating pions at physical mass, past calculations are done by first computing the form factor with heavier, unphysical quarks and then extrapolating to the physical point via chiral perturbation theory. An obvious disadvantage of this approach is that the size of the error grows rapidly when approaching the physical point due to the behavior of the "chiral log".

2. Most past calculations are performed not directly at the kinematic point $q^2 = 0$, but rather with $q^2 > 0$ or $q^2 < 0$. The value of $f^+_{K\pi}(q^2)$ with $q^2 = 0$ is calculated via interpolation. This is mainly due to the difficulty of injecting arbitrary momenta to the quarks on the lattice. In fact, this is a difficulty for many lattice calculations that requires particles carrying momenta. In our calculation on the $48^3\times 96$ (5.5fm, 140MeV) and $64^3 \times 128$ (5.5fm, 140MeV) ensembles, we use the twisted boundary condition to add arbitrary momenta to the quarks.

With the introduction of the Möbius fermions, the forecasted force gradient integrator, the eigCG algorithm and the all mode averaging method, we are now able to perform the calculation directly using physical parameters. This is possible also because of the size of the lattice. The $48^3 \times 96$ (5.5fm, 140MeV) and $64^3 \times 128$ (5.5fm, 140MeV) ensembles both have boxes large enough to accommodate this calculation with negligible finite volume effect. In addition, with the help of twisted boundary conditions we also eliminate the need to interpolate to the zero momentum transfer point $q^2 = 0$.

6.1 Kinematics of the $Kl3$ calculation

The $Kl3$ weak decay width at tree level can be computed in a straightforward manner. Taking the process

$$K^0(p_K) \longrightarrow \pi^-(p_\pi)e^+(p_e)\nu_e(p_\nu) \tag{6.4}$$

as an example. We start from the following effective Lagrangian [36]

$$\mathcal{L}_{\text{eff}} = -\frac{G_F}{\sqrt{2}}V_{us}^*\overline{s}\gamma_\mu(1-\gamma_5)u \cdot \overline{\nu}_e\gamma^\mu(1-\gamma_5)e. \tag{6.5}$$

By using the strong interaction matrix element (6.2) we can write down the decay amplitude

$$i\mathcal{M} = -\frac{G_F}{\sqrt{2}}V_{us}^*\overline{u}(p_\nu)\gamma^\mu(1-\gamma_5)v(p_e) \cdot \left((p_\pi + p_K)_\mu f_{K\pi}^+(q^2) + (p_K - p_\pi)_\mu f_{K\pi}^-(q^2)\right). \tag{6.6}$$

The tree level decay width is thus eqaul to

$$\Gamma_{Ke3} = \frac{G_F^2|V_{us}|^2m_K^5}{128\pi^3}\left|f_{K\pi}^+(0)\right|^2 I_{Kl}. \tag{6.7}$$

Where I_{Kl} is the phase space integral that counts the contribution from all final kinematic configurations. When performing this phase space integral we express $f_{K\pi}^+(q^2)$ in terms of $f_{K\pi}^+(0)$ and kinematic parameters [37]. So there is an overall dependence of $f_{K\pi}^+(0)$ in the formula.

At the current level of accuracy, various other effects must be included. Accordingly, the above formula must be changed to

$$\Gamma_{K^0\longrightarrow\pi^-e^+\nu_e} = \frac{G_F^2|V_{us}|^2m_K^5}{128\pi^3}\left|f_{K\pi}^+(0)\right|^2 I_{Kl}S_{EW}\left(1 + 2\Delta_{SU(2)} + 2\Delta_{EM}\right), \tag{6.8}$$

where Δ_{EM}, $\Delta_{SU(2)}$ and S_{EW} are QED, isospin breaking and short distance electroweak interaction effects. The details of these corrections can be found in [2]. These quantities are also computed to reasonable accuracy so they do not pose significant problems to the problem of determining V_{us}.

It is therefore the form factor $f^+_{K\pi}(q^2)$ at zero momentum transfer that we are interested in. Computing $f^+_{K\pi}(0)$ requires setting up kaon/pion momenta such that

$$q^2 = (p_K - p_\pi)_\mu (p_K - p_\pi)^\mu = 0. \tag{6.9}$$

Suppose $p^\mu_K = (E_K, \boldsymbol{p}_K)$ and $p^\mu_\pi = (E_\pi, \boldsymbol{p}_\pi)$, we then have

$$\sqrt{m_K^2 + \boldsymbol{p}_K^2} - \sqrt{m_\pi^2 + \boldsymbol{p}_\pi^2} = \pm \left| \boldsymbol{p}_K - \boldsymbol{p}_\pi \right| . \tag{6.10}$$

$q^2 = 0$ can be achieved by setting either $\boldsymbol{p}_K$ or $\boldsymbol{p}_\pi$ or both to non zero values.

If we allow $\boldsymbol{p}_\pi \neq 0$, then $q^2 = 0$ requires the size of $\boldsymbol{p}_\pi$ to be

$$|\boldsymbol{p}_\pi| = \frac{m_K^2 - m_\pi^2}{2m_K}. \tag{6.11}$$

Alternatively, if we use non zero $\boldsymbol{p}_K$, then $q^2 = 0$ yields the following condition

$$|\boldsymbol{p}_K| = \frac{m_K^2 - m_\pi^2}{2m_\pi}. \tag{6.12}$$

Using twisted boundary conditions, both of the above settings can be achieved. So we are free to choose the best combination. In addition, it is also possible to have nonzero momenta on both particles. But this setting requires inverting an extra set of quark propagators, so we are not using it.

Testing data on the $32^3 \times 64$ $(4.6\text{fm}, 170\text{MeV})$ ensemble show that the signal to noise ratio is usually better with smaller momenta, so we choose to set $\boldsymbol{p}_\pi \neq 0$ and use (6.11). This will be seen explicitly in the actual data in chapter 7.

6.1.1 Twisted Boundary Conditions

A prominent problem with many lattice calculations is how to add momenta of suitable sizes to the particles on the lattice. Typically the lattice formalism allows direct use of a very restrictive set of momenta.

The allowed momenta are quantized on the lattice as the lattice is a box with periodic or antiperiodic boundary conditions. Suppose periodic boundary conditions are applied to the 3 spatial directions. If the numbers of spatial points of the lattice are L_x, L_y and L_z for the 3 directions, then the allowed lattice momenta will be

$$\boldsymbol{p} = \frac{2\pi}{L_x}k_x\hat{e}_x + \frac{2\pi}{L_y}k_y\hat{e}_y + \frac{2\pi}{L_z}k_z\hat{e}_z,\ 0 \leq k_{x,y,z} < L_{x,y,z},\ k_{x,y,z} \in \mathbb{Z}. \tag{6.13}$$

Even the smallest of the above momenta is typically large using the standard of the pion or kaon masses. For example, on the $48^3 \times 96$ $(5.5\text{fm}, 140\text{MeV})$ lattice both the pion and kaon have physical masses. This lattice has the inverse lattice spacing $a^{-1} = 1.741(23)\text{GeV}$. So the *smallest* lattice momentum is

$$p_{\min} = \frac{2\pi}{48a} \approx 228\text{MeV}. \tag{6.14}$$

This momentum is larger than the mass of a pion ($m_\pi \approx 140\text{MeV}$).

Consequently, the momenta imposed by (6.11, 6.12) are in general not integer multiples of $2\pi/L$. So some techniques are required to generate quarks on the lattice with the desired momentum. [1] We use the twisted boundary conditions to inject arbitrary momentum to the quarks [38].

[1] Interestingly, on both $48^3 \times 96$ $(5.5\text{fm}, 140\text{MeV})$ and $64^3 \times 128$ $(5.5\text{fm}, 140\text{MeV})$ ensembles one unit of lattice momentum is very close to the momentum required by equation (6.11). With $m_K = 498\text{MeV}$ and $m_\pi = 140\text{MeV}$ we have

$$\frac{m_K^2 - m_\pi^2}{2m_K} = 229\text{MeV}, \tag{6.15}$$

very close to the value shown in equation (6.14). Similarly the $64^3 \times 128$ lattice has inverse lattice spacing

Twisted boundary conditions exploit symmetries possessed by the lattice action. In the usual lattice formulation we apply periodic boundary conditions so the lattice fermion fields are single valued. This ensures that the corresponding actions and also all observables are single valued. However, even with multivalued field we can still have single valued action. Consider the following boundary condition

$$\psi(x + L_i\widehat{e}_i) = V_i\psi(x),\ i = x, y, z. \tag{6.17}$$

If V_i is a symmetry of the action then any observable will be single valued,

$$S(V_i\psi(x)) = S(\psi(x)). \tag{6.18}$$

Though V_i can potentially change the field. This is because

$$\begin{aligned}
\langle \mathcal{O}(\psi)\rangle =& \frac{1}{Z}\int \mathcal{D}\psi\ e^{-S(\psi)}\mathcal{O}(\psi) && (6.19)\\
=& \frac{1}{Z}\int \mathcal{D}\psi\ e^{-S(V\psi)}\mathcal{O}(V\psi) && (6.20)\\
=& \frac{1}{Z}\int \mathcal{D}\psi\ e^{-S(\psi)}\mathcal{O}(V\psi) && (6.21)\\
=& \langle \mathcal{O}(V\psi)\rangle . && (6.22)
\end{aligned}$$

equal to 2.302(35)GeV, so the smallest lattice momentum on $64^3 \times 128$ lattice is

$$p_{\text{min}} = \frac{2\pi}{64a} \approx 226\text{MeV}. \tag{6.16}$$

This is also very close to the momentum required by the pion.

One may thus argue that there is no need to use twisted boundary conditions. There is an advantage with this approach. Twisted boundary condition changes the underlying fermion Dirac operator, thus we need to collect a new set of eigenvalues/eigenvectors for use with eigCG. This is not a problem if lattice momentum is used instead. This fact was not well appreciated at the beginning of the calculation, and can be a potential oversight of the calculation.

However, there is also an advantage with twisted boundary conditions. Using twisted boundary condition we can (and we do) distribute the momentum uniformly to all 3 spatial dimensions. Potentially this can reduce the noise associated with the excited pions.

Twisted boundary condition sets V_i to a phase factor

$$\psi(x + L_i\widehat{e}_i) = e^{i\theta_i}\psi(x), \tag{6.23}$$

θ_i is the *twisting angle* associated with direction i. The twist can be applied in 3 spatial directions. Consequently there are 3 twisting angles.

To see why the twisted boundary conditions produce quarks with momentum, we redefine the fermion field by

$$\psi(x) = R(x)\psi'(x), \tag{6.24}$$

where

$$R(x) = \exp\left(i \sum_{i=x,y,z} \frac{\theta_i}{L_i} x_i\right). \tag{6.25}$$

In other words, we uniformly distribute the twists across the entire lattice so the phase is gradually changed from 0 to the required twist imposed in the boundary condition. The new field $\psi'(x)$ satisfies periodic boundary conditions

$$\psi'(x + L_i\widehat{e}_i) = \psi'(x), \tag{6.26}$$

Since the required transformation factor V_i is carried by $R(x)$. The fermion Lagrangian can be written using $\psi'(x)$ as

$$\mathcal{L} = \overline{\psi'}(x)\left(\not{D} + m + \left(R(x)^\dagger \not{\partial} R(x)\right)\right)\psi'(x) \tag{6.27}$$

$$= \overline{\psi'}(x)\left(\not{D} + m + i\not{P}\right)\psi'(x), \tag{6.28}$$

where

$$P_i = \frac{\theta_i}{L_i},\ i = x, y, z. \tag{6.29}$$

The net effect of the twisted boundary conditions is thus to add momentum P_i to the associated quark.

6.1.2 Periodic + Antiperiodic Propagators

Using $P \pm A$ propagators can double the effective size of the lattice in one or more directions, thus suppressing any "around the world" effect. The following discussion uses the temporal direction as an example.

Let the original size in t direction be T. Suppose we have a source vector s and the corresponding solution is ψ_{bc}

$$D_{bc}\psi_{bc} = s \tag{6.30}$$

where bc denotes the boundary condition in t direction, either periodic (P) or antiperiodic (A).

We extend the lattice in t direction by simply duplicating the gauge field. It is clear that the solutions ψ_P and ψ_A satisfy the following equations on the extended lattice with overall *periodic* boundary condition,

$$D' \begin{pmatrix} \psi_P \\ \psi_P \end{pmatrix} = \begin{pmatrix} s \\ s \end{pmatrix}, \; D' \begin{pmatrix} \psi_A \\ \psi_A \end{pmatrix} = \begin{pmatrix} s \\ -s \end{pmatrix}. \tag{6.31}$$

The difference in the way s is extended manifests the difference between ψ_P and ψ_A. We can average the 2 equations in (6.31) to eliminate one of the 2 sources, thus double the lattice extent in t direction,

$$\begin{aligned} D' \begin{pmatrix} (\psi_P + \psi_A)/2 \\ (\psi_P - \psi_A)/2 \end{pmatrix} &= \begin{pmatrix} s \\ 0 \end{pmatrix} \\ D' \begin{pmatrix} (\psi_P - \psi_A)/2 \\ (\psi_P + \psi_A)/2 \end{pmatrix} &= \begin{pmatrix} 0 \\ s \end{pmatrix}. \end{aligned} \tag{6.32}$$

In this way, we form a full propagator on a lattice with t direction size $2T$, the source is also allowed to locate anywhere in $[0, 2T]$.

The method to form a propagator on the doubled lattice is thus as follows

- For any source location $t_0 \in [0, 2T)$, we compute 2 propagators on the original lattice with source located at $t_0 \mod T$. One has periodic boundary condition in t direction and the other has antiperiodic boundary condition. Let the 2 propagators be $D_P^{-1}(t, t_0)$ and $D_A^{-1}(t, t_0)$.

- The full propagator evaluated at source location $t_0 \in [0, 2T)$ and sink location $t \in [0, 2T)$ is equal to

$$\frac{1}{2}\left(D_P^{-1}(t', t_0') + \epsilon\left(T - \frac{1}{2} - t\right)\epsilon\left(T - \frac{1}{2} - t_0\right)D_A^{-1}(t', t_0')\right), \qquad (6.33)$$

  where $t' = t \mod T$ and $t_0' = t_0 \mod T$, $\epsilon(x)$ is the sign function.

It is easy to see that this method can in principle be used to extend the lattice size by any integer factor k. Simply solve the Dirac equation using k different boundary conditions. For each boundary condition we associate a phase factor $\omega^i, i = 0, 1, \cdots, k-1$ to the fermion fields on the boundary, where ω is a primitive kth root of unity. The propagators on the extended lattice can be obtained by linear combinations (essentially a DFT) of these solutions in a way similar to (6.32).

6.2 The Vector Current Renormalization Factor Z_V

The point operator that appears in the QCD $K \longrightarrow \pi$ matrix element (6.2) has the form of a vector current. This operator receives a multiplicative renormalization factor as discussed in section 4.4.

Like the axial current, in principle the vector current renormalization constant can be computed using meson like contractions. In the case of axial current a pion interpolating operator is usually used to couple to the current operator since pion is the lightest physical particle on the lattice and this setup has very good statistics. However one must use a different particle for vector currents since the pion is a pseudoscalar particle and does not produce a signal when couples to the vector current. The choice for the vector current is usually the vector ρ meson,

$$Z_V = \frac{\langle \mathcal{V}_\mu(x) \cdot \overline{q}(0)\gamma_\mu q(0)\rangle}{\langle V_\mu(x) \cdot \overline{q}(0)\gamma_\mu q(0)\rangle}. \tag{6.34}$$

The signal to noise ratio of this quantity decays rapidly as the vector current moves away from the ρ source, since ρ is heavy on the lattice.

A better but more expensive method to compute the vector current renormalization factor involves using the zero momentum $\pi \longrightarrow \pi$ matrix element

$$\sum_{\boldsymbol{x}} \langle \pi(t_\pi) V_\mu(\boldsymbol{x}, t) \pi(0)\rangle = \frac{|Z_\pi|^2}{2m_\pi} \frac{1}{Z_V},\ 0 < t < t_\pi. \tag{6.35}$$

Where both the initial and final pions are static on the lattice. We use wall source propagators to generate both the initial and final pions. The setup is shown in the right panel of figure 6.1. The apparent advantage is that this matrix element maintains good signal to noise ratio even when the 2 pions are separated far away. This is because a zero momentum pion is the lightest particle on the lattice, consequently the signal to noise ratio does not degrade when the separation between the 2 pions becomes larger. This correlation function is more expensive to calculate, requiring light quark propagators computed at both $t = 0$ and $t = t_\pi$. However, due to the vastly better signal to noise ratio, it is the preferred method we use in the $Kl3$ calculation on the $48^3 \times 96$ $(5.5\text{fm}, 140\text{MeV})$ and $64^3 \times 128$ $(5.5\text{fm}, 140\text{MeV})$ ensembles.

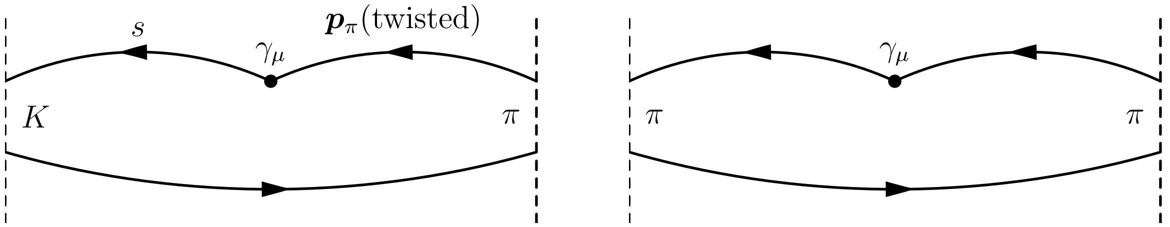


Figure 6.1: Left: $Kl3$ matrix element with twisted pion. Right: Computing the vector current renormalization factor Z_V using the π to π matrix element.

6.3 Detailed Setup of the Contractions

6.3.1 Meson Correlators

Meson correlators take the following form

$$C(t,t_0) = \langle 0 \,|\, \overline{q}'(t)\Gamma_{\text{snk}} q(t) \cdot \overline{q}(t_0)\Gamma_{\text{src}} q'(t_0) \,|\, 0 \rangle \,. \tag{6.36}$$

Where we place the source of both quarks q and q' at t_0. The sink can be a wall sink or point sink at varying time slices t. Γ_{snk} and Γ_{src} are 2 gamma matrices.

6.3.2 $Kl3$ Correlators

$Kl3$ contractions take the following form

$$C(t_\pi, t, t_K) = \langle 0 \,|\, \overline{q_3}(t_\pi)\gamma_5 q_2(t_\pi)\; \overline{q_1}(t)\Gamma q_3(t)\; \overline{q_2}(t_K)\gamma_5 q_1(t_K) \,|\, 0 \rangle \,. \tag{6.37}$$

We place the sources of both quark lines q_2 and q_3 at the same time slice t_π so they can be combined with strange quark propagator q_1 at an arbitrary time slice t_K to produce a $Kl3$ correlator. Γ is one of the 4 gamma matrices γ_μ ($\mu = 0, 1, 2, 3$). We twist the quark q_3 so we have the desired kinematic condition (6.9). We distribute the momentum (twist) in q_3 evenly into all spatial directions to fully make use of the contraction data.

6.3.3 B_K Contractions

B_K contraction takes the following form,

$$C(t_a, t, t_b) = \left\langle 0 \,\middle|\, \bar{l}(t_a)\gamma_5 s(t_a)\ \mathscr{O}_{VV+AA}(t)\ \bar{l}(t_b)\gamma_5 s(t_b) \,\middle|\, 0 \right\rangle, \tag{6.38}$$

where $\mathscr{O}_{VV+AA}(t)$ is equal to

$$\mathscr{O}_{VV+AA}(t) = \overline{s}(t)\gamma_\mu(1-\gamma_5)d(t) \cdot\ \overline{s}(t)\gamma^\mu(1-\gamma_5)d(t). \tag{6.39}$$

The 2 possible contractions can be seen in figure 6.2.

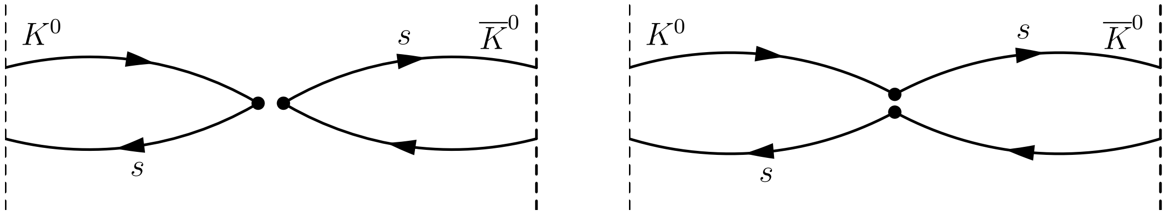


Figure 6.2: B_K diagrams. Each black dot represents an $\overline{s}\gamma_\mu(1-\gamma_5)d$ operator. The dashed lines represent the interpolating operator for kaons, where we put wall sources for the corresponding d and s quarks. There is an extra sign difference between the 2 diagrams since they have different number of fermion loops.

6.4 Fitting the Data

The physical amplitudes as well as meson mass parameters are extracted via fitting the computed correlators. To be specific, fitting is done on the ensemble averaged correlators $\langle C(t)\rangle$ via

$$\chi^2(\lambda) = \sum_{ts} \left(f(t;\lambda) - \langle C(t)\rangle\right) V^{-1}(t,s) \left(f(s;\lambda) - \langle C(s)\rangle\right). \tag{6.40}$$

Where λ represents all free parameters in the fitting function $f(t;\lambda)$. $V(t,s)$ is the covariance matrix. By minimizing the above function we obtain an estimation of λ.

In practice equation (6.40) poses a difficulty when the data sample is not large enough, which is frequently the case in lattice calculations. The covariance matrix V can be highly singular, causing difficulties when trying to compute its inverse V^{-1}. Phenomenologically this is because giving reliable estimation for *every element* of V is difficult to do. V has $\mathcal{O}(N^2)$ elements if we take samples from N time slices, this can easily outnumber the available measurements - too much information is required to fully describe V and there is simply not enough data.

We may cure the problem by simply setting all non-diagonal elements of V to zero, thus turning the procedure to uncorrelated least square minimization. There are two problems associated with this approach

1. Since all correlation information is lost, the resultant "χ^2 value" is no longer a good estimation of the true χ^2. Thus we have to resort to heuristics to ensure that the fitting model is appropriate.

2. Also because there is no correlation information, highly correlated data points are not properly weighted. Thus they can have more influence on the final value of λ than they should, yielding degraded results.

However, due to its robustness this is the method of choice in our work. We use the jackknife resampling method to estimate the error of any fit quantity λ.

6.4.1 Fitting Functions for $Kl3$ Data

The final result $f^+_{K\pi}(0)$ is obtained by combined fitting of the 2 point meson correlators $\pi \longrightarrow \pi$, $K \longrightarrow K$, 3 point $Kl3$ and $\pi \longrightarrow \pi$ correlators. For the meson correlators, we combine both the wall source point sink (WP) results and the wall source wall sink (WW) results to get an improved estimation of the meson masses.

The following functions are used to fit all the correlators,

1. For wall source point sink meson correlators,

$$f(t) = C_i \left(e^{-m_i t} \pm e^{-m_i (T-t)} \right) . \tag{6.41}$$

   Where C_i is not used elsewhere.

2. For wall source wall sink meson correlators,

$$f(t) = \frac{Z_i^2}{2E_i} \left(e^{-m_i t} \pm e^{-m_i (T-t)} \right) . \tag{6.42}$$

   Where Z_i is also used as the normalization factor in certain 3 point $Kl3$ or $\langle \pi \,|\, V_\mu(x) \,|\, \pi \rangle$ correlators. In both cases the sign between the 2 exponential terms is determined by the underlying quantity computed.

3. For the 3 point $Kl3$ or $\langle \pi \,|\, V_\mu(x) \,|\, \pi \rangle$ correlators,

$$f(t) = \frac{Z_i Z_f}{4 Z_V m_i m_f} \left(f_{if}^{+}(q^2)(p_i + p_f)_\mu + f_{if}^{-}(q^2)(p_i - p_f)_\mu \right) e^{-m_i t - m_f (T-t)} . \tag{6.43}$$

   Where i and f are the initial and final states of the process. This fitting function does not count any effect due to around the world meson propagators.

   For the $\pi \longrightarrow \pi$ process we simply have $f_{\pi\pi}^{+}(0) = 1$. For the $Kl3$ matrix element there is a nonzero signal in t direction or any direction with nonzero spatial momentum. We combine all possible directions with nonzero signals to improve the overall statistics.

We choose to start fitting 10 time slices away from the kaon or pion sources, as suggested by the effective mass plots 7.6 and 7.7. For the choice of the $K \longrightarrow \pi$ separations, we combine all the data with separations in $[20, 32]$ to increase the statistics. The combined fit of the above gives results for m_K, m_π, Z_V, $f_{K\pi}^{+}(q^2)$ and $f_{K\pi}^{-}(q^2)$.

Chapter 7

Simulation Results

We discussed various techniques used in the generation of the $48^3 \times 96$ $(5.5\mathrm{fm}, 140\mathrm{MeV})$ and $64^3 \times 128$ $(5.5\mathrm{fm}, 140\mathrm{MeV})$ ensembles and the measurement of the $Kl3$ decay amplitudes in previous chapters. In this chapter, we present the results calculated on the $32^3 \times 64$ $(4.6\mathrm{fm}, 170\mathrm{MeV})$ and $48^3 \times 96$ $(5.5\mathrm{fm}, 140\mathrm{MeV})$ ensembles. We also made substantial effort to apply the techniques to the $64^3 \times 128$ $(5.5\mathrm{fm}, 140\mathrm{MeV})$ ensemble. Since the $48^3 \times 96$ and $64^3 \times 128$ lattices are so large we performed an exploratory calculation on the $32^3 \times 64$ ensemble which has a physical kaon mass and 170MeV pions. It served as a tool to develop suitable computation strategies for the larger ensembles. The $48^3 \times 96$ and $64^3 \times 128$ calculations have all parameters set to physical values, including important quantities such as the pion and kaon masses. Evolution parameters for these ensembles can be found in table B.1 in appendix B. Parameters related to our measurement can be found in table 7.1. We discuss results on the $32^3 \times 64$ ensemble in section 7.1. The results led to some important decisions when the strategy was applied to the $48^3 \times 96$ and $64^3 \times 128$ ensembles. Results on the $48^3 \times 96$ are presented in section 7.2. We also discuss some ongoing work on the $64^3 \times 128$ in section 7.3.

Ensemble	$32^3 \times 64$ DWF+ID	$48^3 \times 96$ DWF+I	$64^3 \times 128$ DWF+I
β	1.75	2.13	2.25
a^{-1}(GeV)	1.370(8)	1.741(23)	2.302(35)
m_π	0.1250(2)	0.08054(46)	0.0584
m_K	0.3594(5)	0.28855(60)	0.214
m_π(MeV)	171	140	134
m_K(MeV)	492	502	493
p_π	0.1580	0.1336	0.0993
p_K	0.4542	0.4766	0.365

Table 7.1: Parameters for the $Kl3$ measurements on the $32^3 \times 64$, $48^3 \times 96$ and $64^3 \times 128$ ensembles. p_π and p_K are twisted momenta needed in the $Kl3$ calculation computed from equations (6.11, 6.12).

7.1 The $32^3 \times 64$ $(4.6\text{fm}, 170\text{MeV})$ Ensemble

The $32^3 \times 64$ $(4.6\text{fm}, 170\text{MeV})$ ensemble was generated using the $\beta = 1.75$ Iwasaki gauge action and 2+1 flavor Shamir domain wall action with $L_s = 32$, together with the Dislocation Suppressing Determinant Ratio (DSDR) [22, 39]. This combination of actions leads to a coarse lattice with inverse lattice spacing equal to $a^{-1} = 1.370(8)$GeV. The dynamic pion mass is set at 170MeV, close to the physical value. Detailed parameters can be found in table B.1. Since calculations on this ensemble were intended to be a test for larger ensembles, we used the less expensive Möbius fermions for valence quarks in the measurement. The Möbius domain wall action has $L_s = 16$, $c_5 = 0.5$, $b_5 = 1.5$ and otherwise identical to the domain wall action. As shown in section 5.4.2, such Möbius action is a very good approximation to the $L_s = 32$ domain wall action used in the ensemble generation. We used 31 configurations each separated by 40 molecular dynamics time units, assuming that 40 time units is a large enough separation so there was no apparent correlation between configurations. Since this is an exploratory calculation aimed at determining the effectiveness of various techniques, we focus our discussion on comparing the results/timings rather than the physical consequences.

We computed Coulomb gauge fixed wall source propagators for both the light quarks and the strange quarks. In addition, we also calculated twisted light quark and strange quark propagators according to the twist settings (6.11, 6.12). We computed all these propagators using periodic + antiperiodic boundary conditions in t direction to suppress any around the world effect.

We computed all the above propagators on all possible 64 time slices to improve the signal to noise ratio, a decision that was justified by the results. A total number of $64 * 12 = 768$ Dirac equations were solved for each specific quark/twist/boundary condition combination. This was an excellent target to apply the eigCG algorithm. To this end we applied the eigCG algorithm to the single precision part of the solver for Dirac equations associated with light quarks. We generate a total of 600 low modes using eigCG. As can be seen below, this yielded a substantial speed up in the calculation.

To explore the effectiveness of the all mode averaging, we also computed inexact propagators on 7 configurations to compare its results from the regular correlators.

7.1.1 The Effect of Temporal Translation Averaging

A major goal of this project is to determine the $Kl3$ form factor $f^{+}_{K\pi}(0)$ on the physical $48^3 \times 96$ and $64^3 \times 128$ lattices. Since eigCG can reduce the cost of the Krylov space solver, we have strong motivation to reuse the eigCG low modes as much as possible. So we computed propagators on all time slices to determine if this is worth the effort.

We computed the $Kl3$ contraction (6.37) for all possible combinations of (t_π, t, t_K) since propagators on all time slices were available. The ensemble average of this matrix element is the same if we translate all operators in (6.37) by the same amount in t direction. So we can average the data with different temporal translations to improve the signal to noise ratio. Computing propagators on all time slices is a good strategy only when the signal to noise ratio is indeed improved.

We can thus analyze the effect of temporal translation averaging by averaging different number of temporal translations. The result is shown in figure 7.1. The graph shows the fitting results of $f^+_{K\pi}(0)$ and $f^-_{K\pi}(0)$ with different kaon-pion separations (i.e., with data having different $t_K - t_\pi$ values). Three curves are shown in each figure, each with time translational averaging on 64, 32 and 16 time slices. Apparently smaller errors can be achieved by adding more time translations to the data set, showing that for quantities such as $f^+_{K\pi}(0)$ and $f^-_{K\pi}(0)$ computing on all time slices is a good strategy.

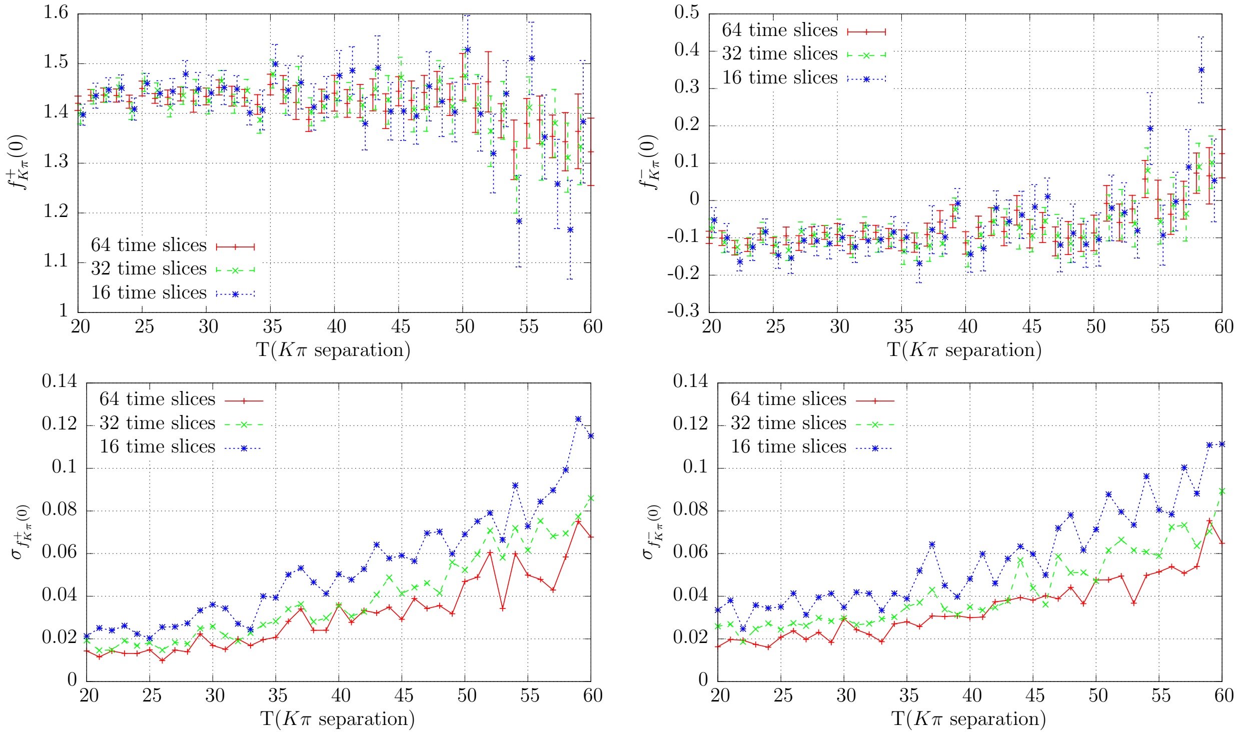


Figure 7.1: Top: plot of $f^\pm_{K\pi}(0)$ results by fitting the data with different $K - \pi$ separations. Bottom: Standard deviation of $f^\pm_{K\pi}(0)$ plotted separately. The results averaged with 16, 32 and 64 temporal translations are shown as blue/green/red points.

7.1.2 The Effect of Twisted Kaon and Twisted Pion

The condition that $f^+_{K\pi}(q^2)$ must be calculated at $q^2 = 0$ imposes equation (6.10) on possible momenta we can add to the kaon and the pion. In this calculation we compare the results from twisting only the pion (6.11) and results from twisting only the kaon (6.12).

The pion/kaon correlator and effective mass plot are shown in figure 7.2 and 7.3. We computed both wall source wall sink (WW) and wall source point sink (WP) correlators. One important observation is that results with a twisted kaon appears to be much noisier than results with a twisted pion. This is because the zero momentum transfer condition demands a much larger momentum if the kaon is made to carry the required momentum. Equation (6.11, 6.11) suggest that the momentum carried by the twisted kaon is m_K/m_π times larger than the momentum carried by the twisted pion, should a single twist be used.

The $K \longrightarrow \pi$ data also agree with this observation. Figure 7.4 shows various $K \longrightarrow \pi$ correlators with twisted pion or twisted kaon, where we can obtain a reasonable signal with $K - \pi$ separation as large as 60 if we twist only the pion. If we twist the kaon there is virtually no signal if the operator is more than 22 time slices away from the kaon source.

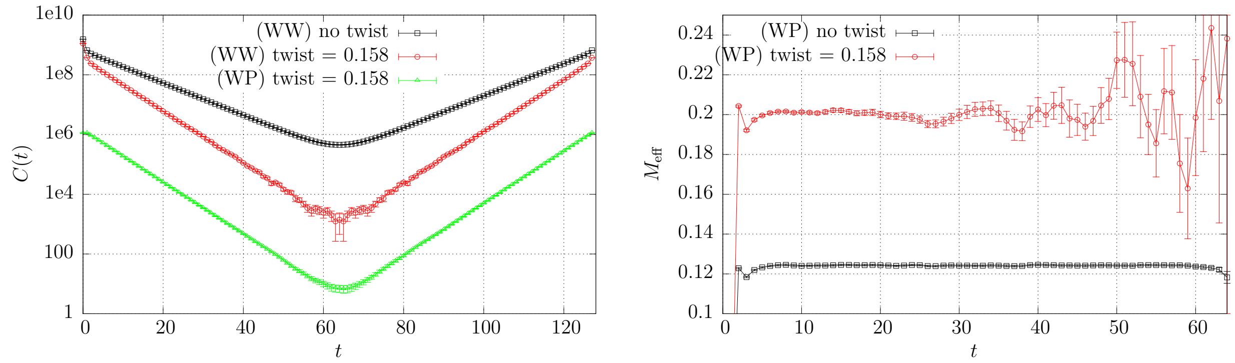


Figure 7.2: Left: pion correlation function. From top to bottom: zero momentum pion (WW), twisted pion (WW) and twisted pion (WP). Right: pion effective mass plot. From top to bottom: twisted pion (WP) and zero momentum pion (WP).

Table 7.2 shows the fitting results from correlators with only twisted pion or twisted kaon, or from the combination of both. Clearly data with only twisted kaon have much larger errors. The error in $f^{\pm}_{K\pi}(0)$ when fitting the combined data set is almost equal to the case where only twisted pion is used. Since the $48^3 \times 96$ and $64^3 \times 128$ have similar m_K/m_π ratios, we decided to *not* calculate $Kl3$ correlators with twisted kaons on these large ensembles to save some time.

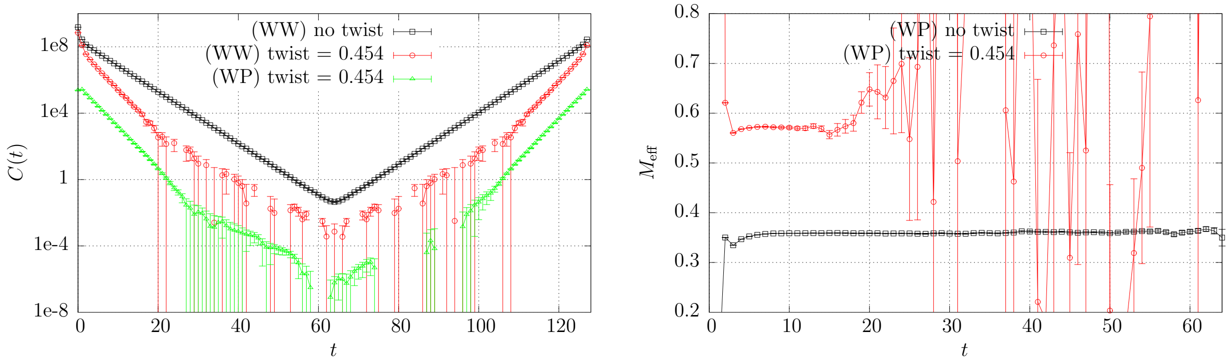


Figure 7.3: Left: kaon correlation function. From top to bottom: zero momentum kaon (WW), twisted kaon (WW) and twisted kaon (WP). Right: kaon effective mass plot. From top to bottom: twisted kaon (WP) and zero momentum kaon (WP). Note that the kaon has a different twist compared to the pion.

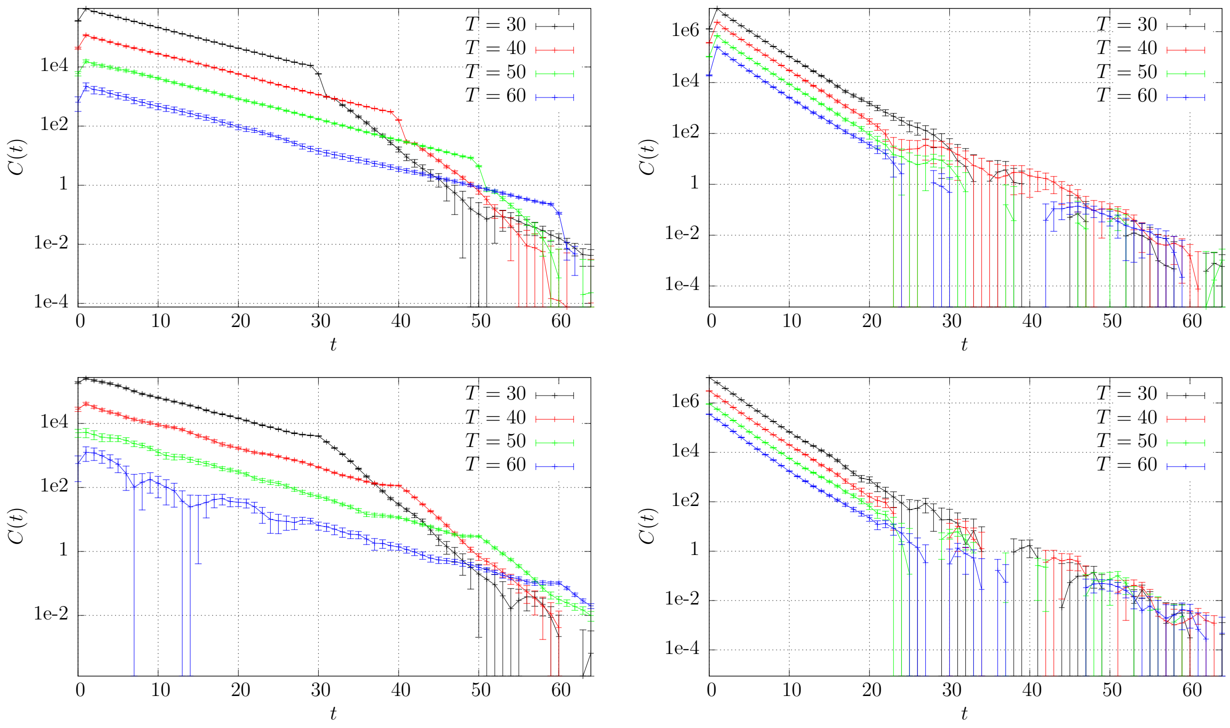


Figure 7.4: $Kl3$ contraction $\langle \pi^-(T+\tau) \,|\, \overline{s}(t+\tau)\gamma_{3,0}u(t+\tau) \,|\, K^0(\tau)\rangle$. Averaged over 64 possible τ values. Top left: operator $\overline{s}\gamma_3 u$ with twisted pion. Top right: operator $\overline{s}\gamma_3 u$ with twisted kaon. Bottom left: operator $\overline{s}\gamma_0 u$ with twisted pion. Bottom right: operator $\overline{s}\gamma_0 u$ with twisted kaon.

data used	m_π	m_K	$f^+_{K\pi}(0)$	$f^-_{K\pi}(0)$
twisted pion	0.1241(4)	0.3588(5)	1.424(7)	-0.151(16)
twisted kaon	0.1241(4)	0.3588(5)	1.329(50)	0.069(53)
both	0.1240(4)	0.3587(5)	1.428(7)	-0.122(11)

Table 7.2: Fitting Results from 31 configurations. Showing the effect of kaon twist and pion twist. $K-\pi$ separations between 20 and 30 (inclusive) are used. The values shown here for $f^+_{K\pi}(0)$ and $f^-_{K\pi}(0)$ are not normalized by the Z_V factor.

7.2 The $48^3 \times 96$ $(5.5\mathrm{fm}, 140\mathrm{MeV})$ Ensemble

In this section we present the results on the $48^3\times 96$ $(5.5\mathrm{fm}, 140\mathrm{MeV})$ DWF+I ensemble. The ensemble was generated using $\beta = 2.13$ Iwasaki gauge action and $L_s = 24$, $c_5 = 0.5$, $b_5 = 1.5$ Möbius domain wall fermion action. We used the forecasted force gradient integrator and the rational action for the strange quark. The detailed setup of the nested integration scheme is shown in table 3.3. The inverse lattice spacing was estimated to be $a^{-1} = 1.741(23)$GeV. Other parameters can be found in table B.1 and 7.1.

We use gauge fixed wall sources for all quark propagators. We also employ the all mode averaging technique by calculating inexact light quark propagators to reduce the cost. To be specific, the following propagators are computed:

1. Inexact light quark propagators are computed on all 96 time slices, including both zero momentum quark and light quark with properly twisted momentum. These propagators are obtained using mixed precision conjugate gradient method. The stopping criteria for these propagators is set to 10^{-4}, based on equation (5.30). We use the eigCG algorithm to obtain 600 single precision low eigenvalues to both speed up the conjugate gradient algorithm and improve the quality of the inexact light quark propagators[1].

2. Exact light quark propagators are computed on time slices 0, 76, 72, 68, 64, 60 and 56, subject to a random shift as discussed before. For the same reason we calculate

[1] The choice of the number 600 is entirely due to the limitation on the amount of available memory. Later experiment on larger machines shows that an additional factor of 2 speed up in terms of rack hours may be obtained if we use more low modes.

propagators for both the zero momentum light quark and the twisted light quark. These accurate propagators serve to compute the correction term "$\Delta\mathcal{O}$" for the all mode averaging technique.

3. Exact strange quark propagators on all 96 time slices. The cost to compute strange quark is low so there is no need to apply the all mode averaging technique to it.

Once we compute the above propagators, we can then compute the following correlators

1. Inexact correlators using inexact light quark propagators and exact strange quark propagators, and

2. Exact correlators using exact light quark propagators and exact strange quark propagators.

The inexact correlators are the easy to calculate part ("$\mathcal{O}'$") in the all mode averaging method and the differences between the exact/inexact correlators are the expensive part ("$\Delta\mathcal{O}$"). The above procedure produces 7 estimations for $\Delta\mathcal{O}$ and 96 estimations for $\mathcal{O}'$. These 2 parts are averaged separately and then combined together to produce the final results, according to (5.98). Each configuration that we measure on is separated by 20 molecular dynamics time units.

Table 7.3 shows the running time of various parts of the calculation on a 1024 node BG/Q rack using only antiperiodic propagators.

item	total time (hours)
Coulomb gauge fixing	4
contractions	3
light quark propagator EigCG setup[a]	29.5
exact light quark propagator (1e-8)[b]	18.7
inexact light quark propagator (1e-4)[c]	64
exact strange quark propagator (1e-8)[d]	8
total	127

[a] There are 3 EigCG instances, 1 for untwisted light quark and 2 additional setups for the twisted quark in $Kl3$ and $K \longrightarrow \pi\pi$. Within each setup we solve a 4D volume source twice to get 600 low modes.

[b] 7 locations for each light quark. A total of 21 propagators are solved using EigCG low modes.

[c] 1 untwisted + 2 twisted quarks, each set solves 96 propagators. Iteration count after deflation fluctuates within the range 1100 ∼ 1600.

[d] Untwisted, 96 propagators. There was a set of twisted s quark propagators for the $Kl3$ calculation but it was dropped due to large noise.

Table 7.3: Running time of the measurements on ensemble $48^3 \times 96$ $(5.5\text{fm}, 140\text{MeV})$ on a 1K BG/Q partition.

7.2.1 The Pseudoscalar Decay Constants f_π and f_K

Quantities related to two point correlators such as m_π, m_K, f_π and f_K can all be described by the general meson correlator (6.36). To be more specific we use the following notation to describe a general 2 point correlator

$$C(t; q, q'; \Gamma_{\text{snk}}, \Gamma_{\text{src}}; type) = \langle 0 \,|\, \overline{q}'(t)\Gamma_{\text{snk}} q(t) \cdot \overline{q}(t_0)\Gamma_{\text{src}} q'(t_0) \,|\, 0 \rangle \,. \tag{7.1}$$

Where *type* can be either WP (wall source point sink) or WW (wall source wall sink). For point sink we also sum over all spatial sink locations on a given time slice t. q and q' can be either l (up/down quark) or s (strange quark).

We combine a few of the above correlators to determine the meson masses m_π, m_K and pseudoscalar decay constants f_π and f_K. For example, the pion mass can be extracted from

$$C(t; l, l; \gamma_5, \gamma_5, WP/WW) = \frac{|Z_\pi^{WP/WW}|^2}{2m_\pi} \left(e^{-m_\pi t} + e^{-m_\pi (T-t)} \right) . \tag{7.2}$$

f_π can be obtained from

$$C(t;l,l;\gamma_3\gamma_5,\gamma_5,WP) = \frac{Z_\pi^{WW} f_\pi \sqrt{V}}{2Z_A}\left(e^{-m_\pi t} - e^{-m_\pi (T-t)}\right), \tag{7.3}$$

where V is the size of the source (48^3). m_K and f_K can be obtained in a similar manner. The axial current renormalization constant Z_A is obtained through procedure described in section 4.4. In figure 7.6 and 7.7 we provide effective mass plots of the pion, kaon and omega baryon correlators. Based on these effective mass plots, we leave out 10 time slices for the pion and kaon and 13 time slices for the omega baryon to eliminate effects from any excited states.

The fitting results for m_π, m_K, f_π, f_K and Z_A can be found in table 7.4. We also provide fitting results for the mass of the omega baryon to extract information about the lattice spacing. The method to measure the omega baryon mass is described in [40]. Using our estimation of $a^{-1} = 1.741(23)$GeV, we have the following preliminary results

$$f_\pi = 132.2(3)_{\text{stat}}(17)_{a^{-1}}\text{MeV} \tag{7.4}$$

$$f_K = 157.5(2)_{\text{stat}}(21)_{a^{-1}}\text{MeV} \tag{7.5}$$

$$f_K/f_\pi = 1.1914(21)_{\text{stat}}. \tag{7.6}$$

Figure 7.5 compares the value f_K/f_π from this work with a few previous calculations [41, 42, 43, 44, 45, 46, 47, 48]. As a reference, the figure also highlights the 2011 lattice averaging result from the FLAG working group [2]. The 2011 FLAG average of various lattice calculations for f_K/f_π is [41]

$$f_K/f_\pi = 1.193(5). \tag{7.7}$$

As shown in figure 7.5, the statistical error for our f_K/f_π value is already very small from

[2] FLAG stands for Flavianet (or Flavor) Lattice Averaging Group [49]. They provide estimates of a few important low energy constants such as f_K, f_π by averaging results from various lattice groups.

	AMA	Exact
m_π	0.08056(17)	0.08064(21)
m_K	0.28845(25)	0.28889(38)
f_π	0.07594(16)	0.07622(28)
f_K	0.09047(12)	0.09052(42)
f_K/f_π	1.1914(21)	1.1876(57)
Z_A	0.71184(13)	0.71278(63)
m_Ω	-	0.9649(50)
m_{res}	0.000620(5)	0.000619(5)

Table 7.4: Lattice values for m_π, m_K, f_π, f_K, Z_A, m_Ω and m_{res}. All quantities other than m_Ω are measured from 26 configurations, the value m_Ω is obtained from 18 configurations. For the "AMA" column full $\sum \mathcal{O}' + \Delta\mathcal{O}$ data are used, so there are 96 estimations of $\mathcal{O}'$ and 7 estimations of $\Delta\mathcal{O}$. In the "exact" column only contractions computed from exact propagators are used, so in this case there are 7 estimations per configuration.

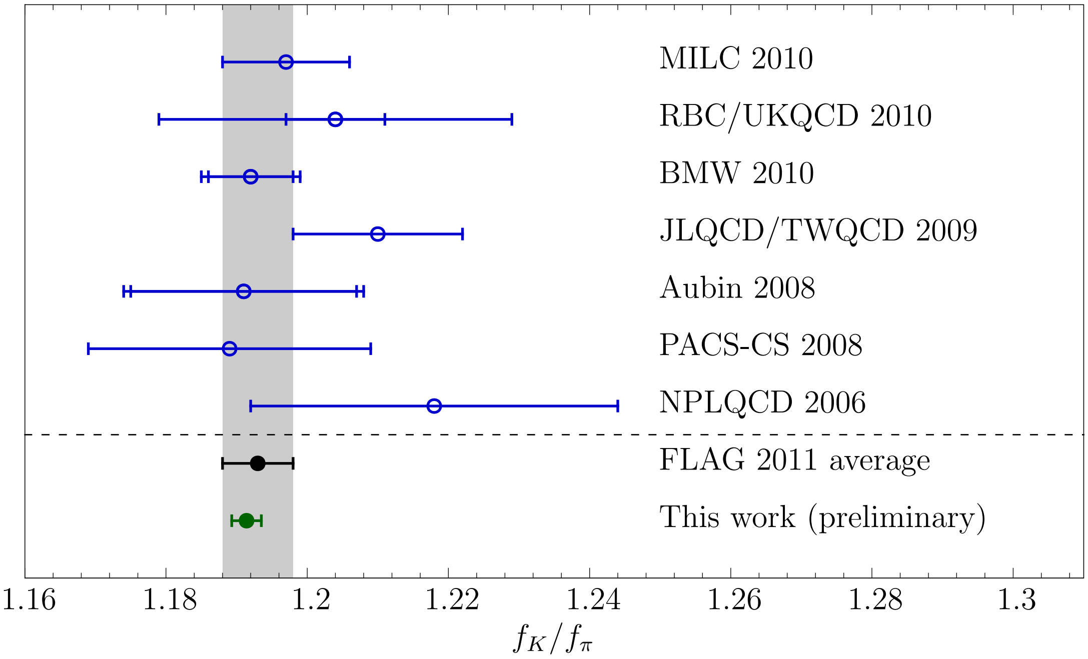


Figure 7.5: Comparison of lattice f_K/f_π results. The black point is the latest average from the FLAG group. The green point is from this work.

26 configurations. We further notice that the error due to the chiral extrapolation is largely absent here, since both the $48^3 \times 96$ ensemble and the measurement are performed at very close to physical pion and kaon masses. We expect a few percent correction to the pion/kaon mass, the errors introduced by such percent level chiral corrections will be very small.

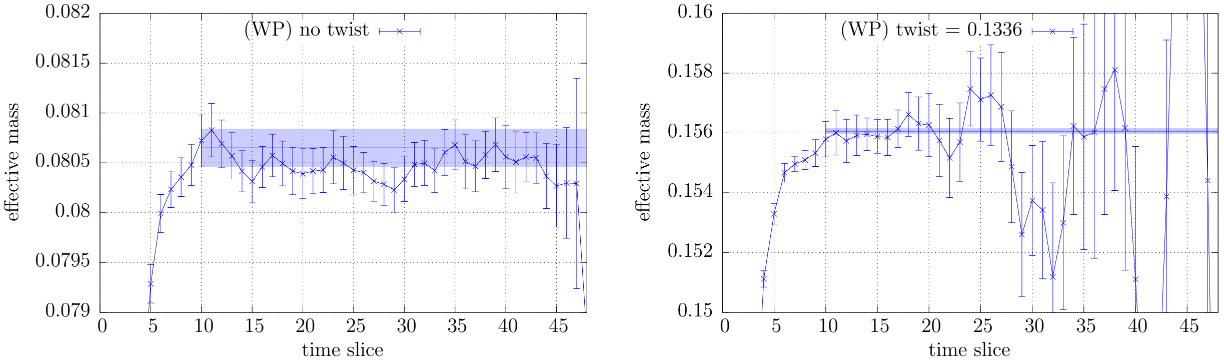


Figure 7.6: Pion effective mass plot. Left: static pion with no twist. Right: twisted pion with $p = 0.1336$. We use the points that are at least 10 time slices away from the source when fitting the masses.

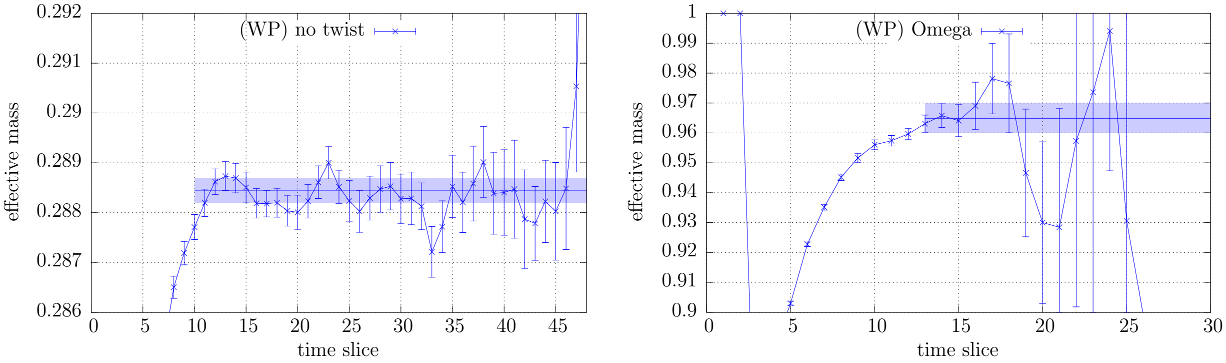


Figure 7.7: Kaon and omega baryon effective mass plot. For kaons we start fitting from time slice 10. For omega baryons we start fitting from time slice 13.

7.2.2 The $Kl3$ Form Factors $f^{\pm}_{K\pi}(0)$

We extend the notation in (6.37) to indicate the type of quarks involved in the contraction

$$C(t_\pi, t, t_K; \Gamma, q_1, q_2, q_3) = \langle 0 \,|\, \overline{q_3}(t_\pi)\gamma_5 q_2(t_\pi) \cdot \overline{q_1}(t)\Gamma q_3(t) \cdot \overline{q_2}(t_K)\gamma_5 q_1(t_K) \,|\, 0 \rangle \,. \tag{7.8}$$

Where q_i can be l (light quark), l' (twisted light quark) and s (strange quark). We combine the following 3 point correlators to produce the fitting results for $f^{\pm}_{K\pi}(0)$

$$\textcircled{1} = C(t_\pi, t, t_K; \gamma_i, s, l, l'),\ i = x, y, z. \tag{7.9}$$

$$\textcircled{2} = C(t_\pi, t, t_K; \gamma_t, s, l, l'), \tag{7.10}$$

$$\textcircled{3} = C(t_\pi, t, t_K; \gamma_t, l, l, l). \tag{7.11}$$

Where $\textcircled{1}$ and $\textcircled{2}$ are spatial/temporal components of the $K \longrightarrow \pi$ matrix element (cf. equation 6.2), $\textcircled{3}$ is the temporal component of the $\pi \longrightarrow \pi$ process. The purpose of $\textcircled{3}$ is mainly to determine Z_V. We use all 3 spatial directions for the $K \longrightarrow \pi$ matrix element since the momentum is distributed uniformly in all spatial directions. We setup the kinematics at zero momentum transfer in $\textcircled{1}$ and $\textcircled{2}$ by twisting the light quark q_3. As discussed above, contractions with twisted strange quark q_1 have poor signal to noise ratio so we are not calculating them.

Using the fitting functions (6.41, 6.42, 6.43) we get the fitting results shown in table 7.5. The relevant graphs are shown in figure 7.11.

$K-\pi$ sep	AMA?	$f^{+}_{K\pi}(0)$	$f^{-}_{K\pi}(0)$	Z_V
20:24	AMA	0.9672(45)	-0.1327(123)	0.7123(13)
20:28	AMA	0.9602(52)	-0.1254(97)	0.7089(17)
20:32	AMA	0.9639(49)	-0.1318(96)	0.7093(16)
24:28	AMA	0.9598(59)	-0.1230(112)	0.7087(18)
24:32	AMA	0.9646(52)	-0.1322(106)	0.7092(17)
20:24	exact	1.0018(253)	-0.1206(320)	0.7315(150)
20:28	exact	0.9552(227)	-0.0850(205)	0.7016(157)
20:32	exact	0.9537(246)	-0.1004(215)	0.6971(162)

Table 7.5: Fitting Results from 26 configurations. Data points that are at least 10 slices away from the sources are used. The meson sector also includes the wall source point sink (WP) contractions. The column "$K-\pi$ sep" shows what data are used when fitting $f^{\pm}_{K\pi}(0)$. As an example, 20:32 means that all data with the kaon and pion separation in the range $[20, 32]$ are used. The column "AMA?" indicates whether we are using the full all mode averaged data ("AMA") or just the exact contractions ("exact").

One important fact that can be seen in table 7.5 is that AMA results are much more accurate than results from just the exact contractions. This is because AMA has 96 estimations of the inexact part $\mathcal{O}'$ and 7 estimations of the correction part $\Delta\mathcal{O}$, while the results from exact contractions have much less averaging effects from temporal translations. This fact also proves that measuring on all 96 time slices as well as all mode averaging are very useful strategies.

The data in table 7.5 show that we are able to measure $f^+_{K\pi}$ at zero momentum transfer to high accuracy using this combined scheme. The relative error for $f^+_{K\pi}(0)$ is estimated to be at 0.5% with 26 configurations. Figure 7.8 compares this calculation with a few previous works. The red points are estimated using chiral perturbation theory [50, 51, 52, 53]. And the blue points are from various lattice calculations [54, 55, 56, 57, 58]. As more statistics are accumulated the data point from this work will become better.

V. Cirigliano *et al.* [2] report that $|V_{us}|f^+_{K\pi}(0) = 0.2163(5)$ using experimental data from [59]. Using our current best estimation $f^+_{K\pi}(0) = 0.9639(49)$ we obtain the following preliminary value for V_{us}

$$|V_{us}| = 0.2244(5)_{\text{exp}}(11)_{\text{lat}}. \tag{7.12}$$

The lattice error shown here is purely statistical. As discussed before, we expect a small correction due to the small difference between the simulated pion mass and the physical pion mass. Such percent level chiral corrections will introduce a very small systematic error. The discretization error can be eliminated once the results on the $64^3 \times 128$ ensemble are obtained. For now the error is still dominated by $f^+_{K\pi}(0)$, but we believe that as more statistics are obtained the lattice error will be of comparable size compared with experimental error on the combined quantity $|V_{us}|f^+_{K\pi}(0)$.

Figure 7.9 compares $f^+_{K\pi}(0)$ obtained by this work and previous paper [1, 57]. As a comparison, chiral extrapolation on the 4 unphysical UKQCD data points is also presented

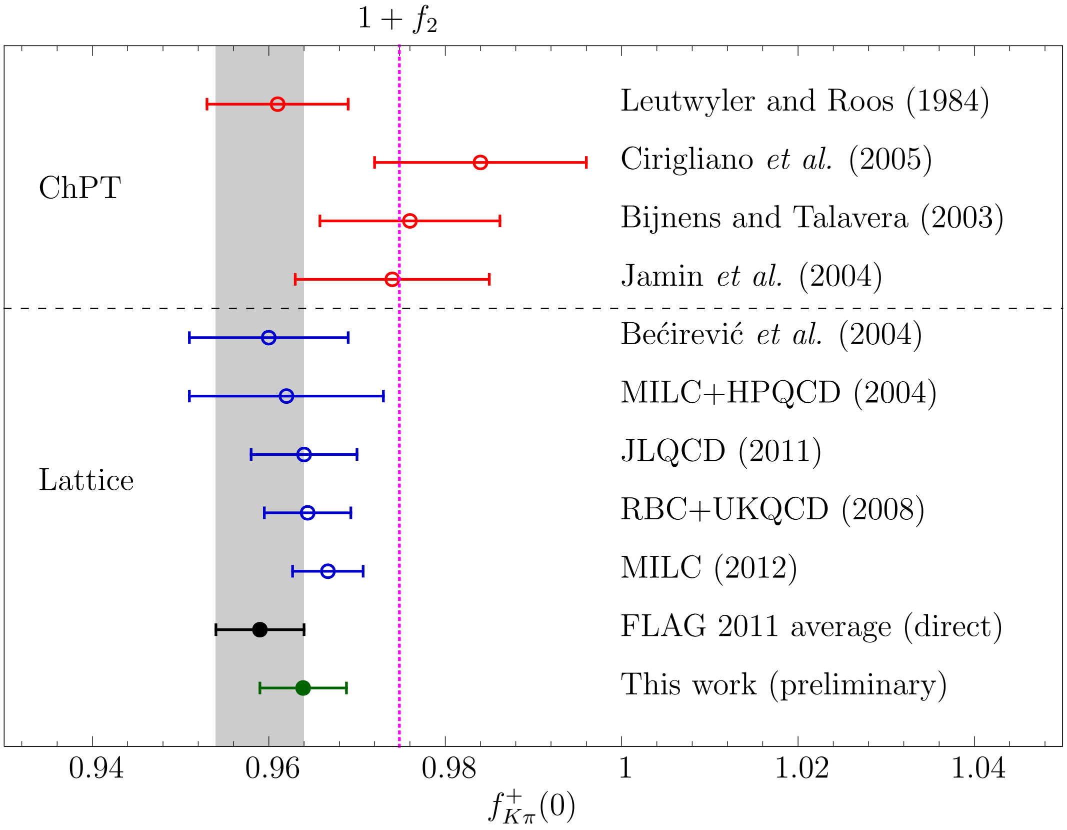


Figure 7.8: Comparison of $f^+_{K\pi}(0)$ with other works. Red points are from chiral perturbation theory, blue points are from 2+1 flavor lattice calculations. The black data point comes from 2011 lattice average performed by the FLAG group. This data point averages "direct" lattice calculations (i.e., calculations that do not use CKM unitarity). The green point is the preliminary result in this work.

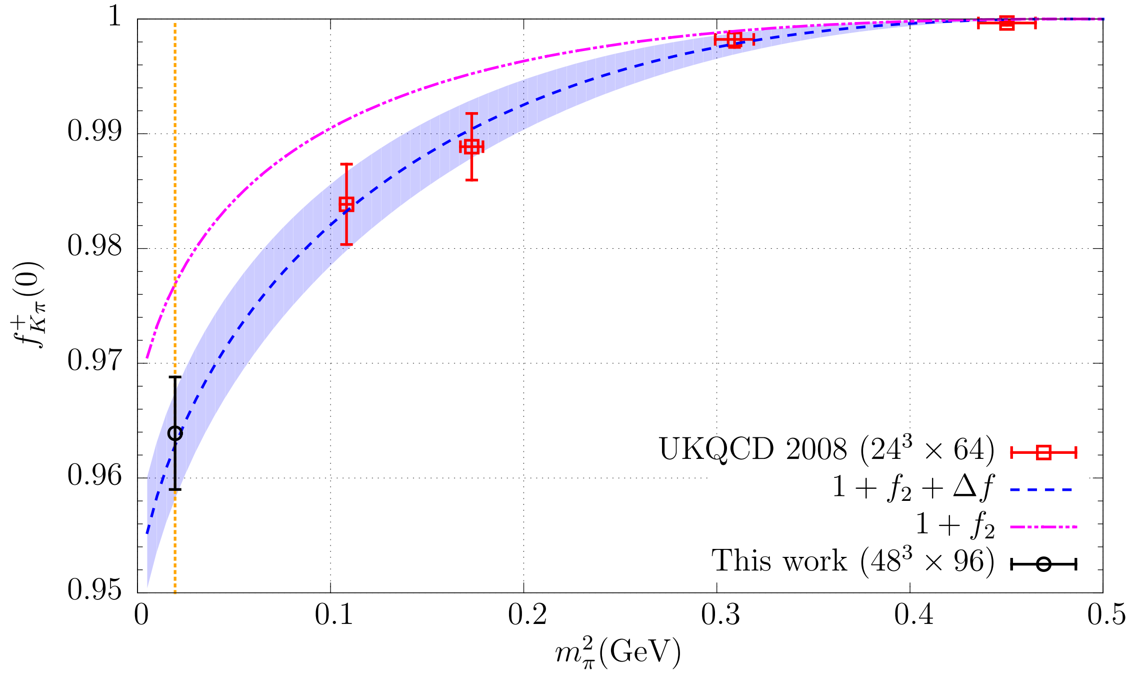


Figure 7.9: Compare with earlier $Kl3$ data at unphysical pion masses. Chiral extrapolation is also shown. The $24^3 \times 64$, $N_f = 2 + 1$ data are obtained from [1]. The ChPT fitting *does not* use the data point presented in this work.

in the figure. The chiral extrapolation makes use of the following fitting functions [37, 56, 57]

$$f^+_{K\pi}(0) = 1 + f_2 + \Delta f, \tag{7.13}$$

$$f_2 = \frac{3}{2}\left(H_{K\pi} + H_{K\eta}\right), \tag{7.14}$$

$$H_{PQ} = -\frac{1}{128\pi^2 F_0^2}\left(m_P^2 + m_Q^2\right)\left(1 + \frac{2m_P^2 m_Q^2}{m_Q^4 - m_P^4}\log\frac{m_P^2}{m_Q^2}\right), \tag{7.15}$$

$$\Delta f = \left(m_K^2 - m_\pi^2\right)^2\left(A_0 + A_1\left(m_K^2 + m_\pi^2\right)\right), \tag{7.16}$$

$$m_\eta^2 = \frac{1}{3}\left(4m_K^2 - m_\pi^2\right). \tag{7.17}$$

To use chiral extrapolation to obtain the physical form factor from unphysical values, we fit the above formula and determine A_0 and A_1 from lattice data, using unphysical m_π and m_K values. Then we evaluate physical $f^+_{K\pi}(0)$ by substituting physical pion and kaon masses into the fitting function.

There are clear advantages to perform the computation directly at physical point. First of all, the above chiral perturbation theory formulas do not include contributions from all orders. Equation (7.16) should be treated as an ansatz and can not model all higher order contributions accurately. So we introduce systematic errors in this extrapolation. In addition, the extrapolation itself also introduces extra errors because of the behavior of the fitting function. Δf grows when approaching physical m_π and m_K, so the size of the error in the physical form factor is larger than those in the unphysical ones.

Figure 7.10 shows various constraints on the values of $|V_{ud}|$ and $|V_{us}|$. The unitary constraint is the line

$$|V_{ud}|^2 + |V_{us}|^2 + |V_{ub}|^2 = 1, \tag{7.18}$$

where the contribution from $|V_{ub}|^2$ is very small ($|V_{ub}| = 0.00411(^{+27}_{-28})$ from [60]). The constraint on $|V_{ud}|$ comes from nuclear β transitions [61]. The constraint on the $|V_{us}|/|V_{ud}|$ ratio comes from [62]. For $|V_{us}|$ we use the preliminary value shown in equation (7.12). By collecting more data we believe the error on $|V_{us}|$ will be greatly reduced.

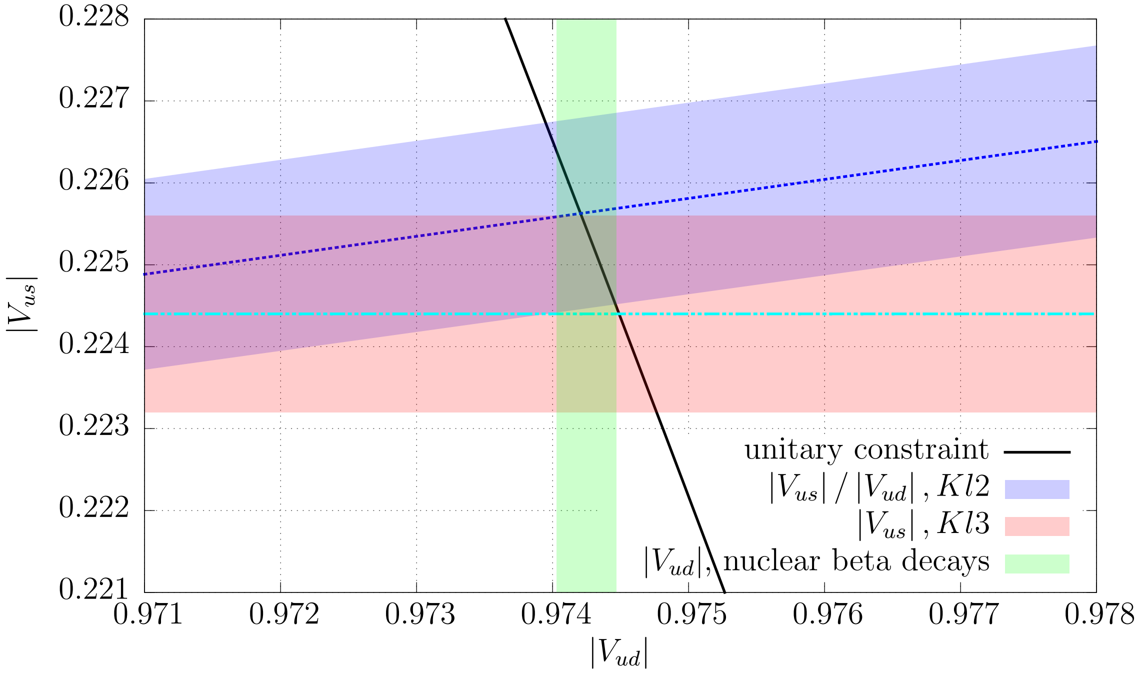


Figure 7.10: Various constraints on $|V_{ud}|$ and $|V_{us}|$.

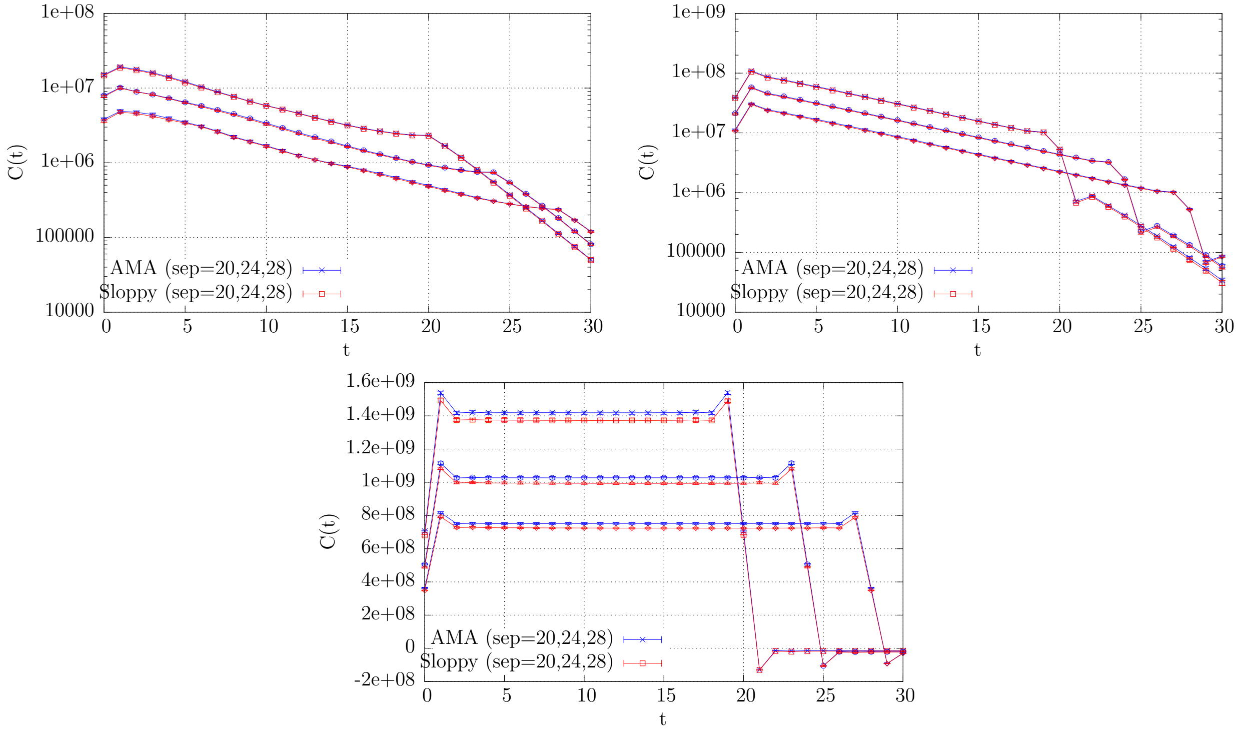


Figure 7.11: Inexact and AMA Kl3 correlators on the $48^3 \times 96$ lattice. Top left: γ_x, top right: γ_t, bottom: $\pi - \pi$ 3 point function for Z_V measurement.

7.2.3 The Neutral Kaon Mixing Matrix Element B_K

On the lattice we use the following ratio to define the raw lattice value for B_K, similar to [63]

$$B_K^{\mathrm{lat}} = \frac{3\left\langle K^0(t_a)\,\middle|\,\mathcal{O}_{VV+AA}(t)\,\middle|\,\overline{K}^0(t_b)\right\rangle}{8\left\langle K^0(t_a)\,\middle|\,A_3(t)\right\rangle\left\langle A_3(t)\,\middle|\,\overline{K}^0(t_b)\right\rangle}. \tag{7.19}$$

Where $\left\langle A_3(t)\,\middle|\,\overline{K}^0(t_b)\right\rangle$ in the denominator are the following wall source point sink correlators using our general notation in equation (7.1)

$$\left\langle A_3(t)\,\middle|\,\overline{K}^0(t_b)\right\rangle = C(t; d, s; \gamma_5, \gamma_3\gamma_5; WP). \tag{7.20}$$

Due to the sum over all possible sink locations only the temporal component of the axial current contributes to a nonzero signal. We use wall source for all quarks in the above formulas, the walls are placed at t_a and t_b. The fitting results of B_K^{lat} is presented in table 7.6.

$K^0 - \overline{K}^0$ sep	AMA?	B_K^{lat}
20:4:24	AMA	0.5836(11)
20:4:28	AMA	0.5844(12)
20:4:32	AMA	0.5839(12)
20:4:24	exact	0.5712(109)
20:4:28	exact	0.5870(110)
20:4:32	exact	0.5845(116)

Table 7.6: B_K fitting results from 25 (AMA)/26 (exact) configurations. Data points that are at least 10 slices away from the sources are used. The column "$K^0 - \overline{K}^0$ sep" shows the separation of the 2 kaons used in the fit. Due to the arrangement of the exact light quark propagators, results are only available for some $K^0 - \overline{K}^0$ separations. In the above table, 4 different separations are used: 20, 24, 28 and 32. As an example, the symbol 20:4:28 means that the separations 20, 24 and 28 are used.

To transform the lattice result B_K^{lat} into the conventional result in the $\overline{\mathrm{MS}}$ scheme, we need to compute the renormalization factors for the lattice operators. Since dimensional regularization can not be implemented on the lattice, the continuum results are obtained using the following two steps,

1. The lattice results are transformed to some intermediate renormalization scheme such as momentum subtraction schemes used by [63]. Since this intermediate renormalization scheme serves as a bridge to transform the lattice results to the $\overline{\mathrm{MS}}$ scheme, we must be able to implement it on the lattice.

2. Perform a perturbative calculation to transform from the intermediate renormalization scheme to the $\overline{\mathrm{MS}}$ scheme. This step is independent of lattice calculations.

Previous analysis on the $32^3 \times 64 \times 16$, $\beta = 2.25$, $a^{-1} = 2.28(3)\mathrm{GeV}$ lattice and the $24^3 \times 64 \times 16$, $\beta = 2.13$, $a^{-1} = 1.73(3)\mathrm{GeV}$ lattice showed that the continuum B_K under the $\overline{\mathrm{MS}}$ scheme at 2GeV and 3GeV are [63]

$$B_K^{\overline{\mathrm{MS}}}(2\mathrm{GeV}) = 0.549(5)_{\mathrm{stat}}(15)_{\chi}(2)_{\mathrm{FV}}(21)_{\mathrm{NPR}}, \tag{7.21}$$

$$B_K^{\overline{\mathrm{MS}}}(3\mathrm{GeV}) = 0.529(5)_{\mathrm{stat}}(15)_{\chi}(2)_{\mathrm{FV}}(11)_{\mathrm{NPR}}. \tag{7.22}$$

Where the 4 errors are

- "stat" are statistical errors from lattice calculations. These errors have the same nature as the errors shown in table 7.6.

- "χ" are errors from the chiral extrapolation.

- "FV" are errors due to finite volume effects on the lattice.

- "NPR" are errors associated with the calculation of the non-perturbative renormalization factors.

With the results computed on the $48^3 \times 96$ shown in table 7.6, the errors associated with the chiral extrapolation and the finite volume effects are largely eliminated. The "NPR" errors remain unchanged. Using the combined renormalization factor from [63] which transforms the above bare lattice value to the $\overline{\rm MS}$ scheme, we have the following preliminary results

$$B_K^{\overline{\rm MS}}(2{\rm GeV}) = 0.5473(11)_{\rm stat}(209)_{\rm NPR}, \tag{7.23}$$

$$B_K^{\overline{\rm MS}}(3{\rm GeV}) = 0.5371(11)_{\rm stat}(111)_{\rm NPR}. \tag{7.24}$$

Where the "NPR" errors are scaled directly from the previous analysis in [63]. We should emphasize that almost all errors in the above results are from factors associated with non-perturbative renormalization. The statistical errors are reduced to a very low level. In addition, the large errors from chiral extrapolations are also absent since our $48^3\times 96$ ensemble is simulated with physical pion and kaon masses. To improve these results further, we need to improve our estimations on the renormalization factors. Figure 7.12 compares this preliminary result at 2GeV with a few recent other calculations [41, 64, 63, 65, 66].

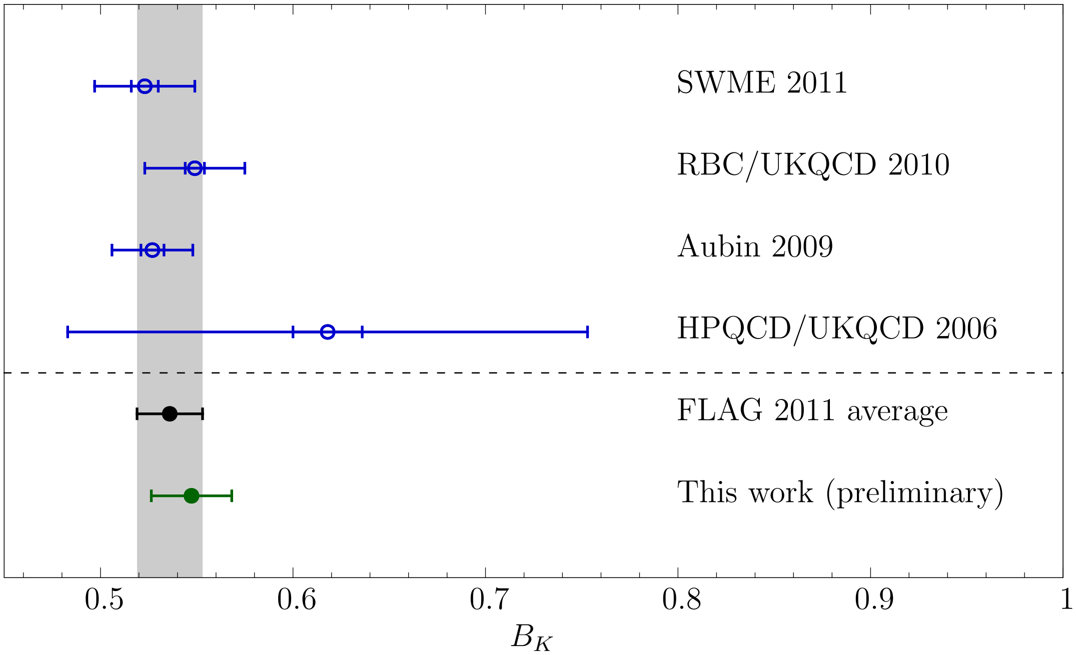


Figure 7.12: Comparison of $B_K^{\overline{\rm MS}}$ at 2GeV. The black point is the latest average from the FLAG group. The green point is from this work.

7.2.4 The $I = 2$ $K \longrightarrow \pi\pi$ Decay Amplitude

We briefly mention the $I = 2$ $K \longrightarrow \pi\pi$ calculation performed on the $48^3 \times 96$ (5.5fm, 140MeV) ensemble in this section. Previous work on this subject was performed on the $32^3 \times 64 \times 16$, $\beta = 2.25$ and $24^3 \times 64 \times 16$, $\beta = 2.13$ ensembles [67], reporting the following results

$$\Re A_2 = 1.436(63)_{\text{stat}}(258)_{\text{syst}} \times 10^{-8}\text{GeV}, \tag{7.25}$$

$$\Im A_2 = -6.83(51)_{\text{stat}}(130)_{\text{syst}} \times 10^{-13}\text{GeV}. \tag{7.26}$$

The calculation on the $48^3 \times 96$ ensemble aims at reducing lattice related uncertainties. We expect that with the all mode averaging technique the statistical error will be greatly reduced. In addition, various lattice artifacts will also be reduced due to the physical kinematics and the large size of the lattice.

A major advantage of combining this calculation with previous calculations is that we can reuse the intermediate results such as the expensive quark propagators, thus reducing the cost greatly. The $I = 2$ $K \longrightarrow \pi\pi$ calculation requires computing two sets of light quark propagators and one set of strange quark propagators. One set of light quark propagators has a built-in twist which must be calculated separately. However, the other set of light quark propagators and the strange quark propagators are the same as the ones used by the $Kl3$ calculation. By sharing these propagators across these calculations we can reduce the cost of the $I = 2$ $K \longrightarrow \pi\pi$ calculation by more than 50%.

The details of this calculation are not discussed here as the focus of this work is the $Kl3$ calculation. Nevertheless we shall report our preliminary analysis on the quantity A_2 based on 19 configurations,

$$\Re A_2 = 1.431(33)_{\text{stat}} \times 10^{-8}\text{GeV}, \tag{7.27}$$

$$\Im A_2 = -6.53(15)_{\text{stat}} \times 10^{-13}\text{GeV}. \tag{7.28}$$

The systematic errors of these results are not analyzed yet so they are not provided in the above results. At the current stage, we expect the systematic errors to be much smaller than that reported in [67] but they will be at least 7% for $\Re A_2$ and 8% for $\Im A_2$. This is because the systematic errors in $\Re A_2$ and $\Im A_2$ will be at least equal to the systematic errors in the Wilson coefficients used to compute them, and the errors in these Wilson coefficients are estimated to be 7% for $\Re A_2$ and 8% for $\Im A_2$ in [67].

7.3 The $64^3 \times 128$ $(5.5\mathrm{fm}, 140\mathrm{MeV})$ DWF+I Ensemble

The $64^3 \times 128$ $(5.5\mathrm{fm}, 140\mathrm{MeV})$ ensemble was generated using $\beta = 2.25$ Iwasaki gauge action and 2+1 flavor Möbius fermions. This yields a lattice spacing of 2.302(35)GeV. Its physical volume is thus equal to $(5.49\mathrm{fm})^3$. Like the $48^3 \times 96$ ensemble, we simulate with physical pion and kaon masses so there are no large chiral extrapolations for quantities measured on it. We use the forecasted force gradient integrator in the hybrid Monte Carlo algorithm to generate this ensemble. This lattice has similar physical volume but has a finer lattice spacing compared to the $48^3 \times 96$ ensemble. By combining the results from the two ensembles we can extrapolate to the continuum point $a = 0$, thus eliminating any discretization error.

As of this writing, the $64^3 \times 128$ measurement strategy is yet to be finalized. Early experiment performed on the Mira supercomputer proved that the strategy we used on the $48^3 \times 96$ ensemble can also be applied to this ensemble with minor modifications. The measurement is to be performed on a 32K node BG/Q partition, which has a much larger memory pool compared to the 1K BG/Q rack for the $48^3 \times 96$ measurements. So we increase the number of low modes to be collected in eigCG from 600 to 1000 $\sim$ 1500, which reduces the running time substantially. Further results on this ensemble will be reported by our future work.

Chapter 8

Conclusions and Discussions

In this work, we developed new integrators for use with the hybrid Monte Carlo algorithm. We also explored the Möbius domain wall fermion action, which is an improvement over the original Shamir domain wall fermion action. The development allows us to simulate large lattices such as the $48^3 \times 96$ $(5.5\text{fm}, 140\text{MeV})$, $a^{-1} = 1.741(23)\text{GeV}$ lattice and the $64^3 \times 128$ $(5.5\text{fm}, 140\text{MeV})$, $a^{-1} = 2.302(35)\text{GeV}$ lattice. We are able to simulate directly at physical quark masses on these large lattices. This helps to eliminate the extrapolation to the physical point, which is a great advantage over many previous works since the need of a chiral extrapolation is almost eliminated.

By combining various measurement techniques such as the eigCG algorithm and the all mode averaging technique, we performed a precision lattice measurement of the semileptonic kaon decay $K \longrightarrow \pi l \nu$ directly at zero momentum transfer. Once finished, the resulting form factors $f^{\pm}_{K\pi}(0)$ permit precise determination of the CKM matrix element V_{us}. This projects also provide precise determination of a few other low energy QCD constants such as the pseudoscalar decay constants f_K and f_π, and the neutral kaon mixing constant B_K to very high statistical precision at no extra cost.

This project also includes the $I = 2$ $K \longrightarrow \pi\pi$ calculation. By reusing intermediate results we are able to perform the $I = 2$ $K \longrightarrow \pi\pi$ calculation with less than 50% of computing resources compared to a separate calculation.

As of this writing, the project is still not finished. Most notably, the results on the $64^3 \times 128$ ensemble are missing. These results are important to perform a continuum extrapolation. However, we shall mention that the $64^3 \times 128$ measurement is almost ready to start, using the same techniques mentioned in this work and minor improvements.

Bibliography

[1] P.A. Boyle et al. Determining the K(l3) form factors directly at zero momentum transfer. *PoS*, LAT2009:248, 2009.

[2] Vincenzo Cirigliano, Gerhard Ecker, Helmut Neufeld, Antonio Pich, and Jorge Portoles. Kaon Decays in the Standard Model. *Rev.Mod.Phys.*, 84:399, 2012.

[3] Holger Bech Nielsen and M. Ninomiya. Absence of Neutrinos on a Lattice. 1. Proof by Homotopy Theory. *Nucl.Phys.*, B185:20, 1981.

[4] H. Rothe. Lattice Gauge Theories - An Introduction. 2005.

[5] Y. Iwasaki and T. Yoshie. RENORMALIZATION GROUP IMPROVED ACTION FOR SU(3) LATTICE GAUGE THEORY AND THE STRING TENSION. *Phys.Lett.*, B143:449, 1984.

[6] Y. Iwasaki. Renormalization Group Analysis of Lattice Theories and Improved Lattice Action: Two-Dimensional Nonlinear O(N) Sigma Model. *Nucl.Phys.*, B258:141–156, 1985.

[7] Kenneth G. Wilson. Quarks and Strings on a Lattice. 1975.

[8] David B. Kaplan. A Method for simulating chiral fermions on the lattice. *Phys.Lett.*, B288:342–347, 1992.

[9] P. Vranas. Domain wall fermions and Monte Carlo simulations of vector theories. *Nucl.Phys.Proc.Suppl.*, 53:278–282, 1997.

[10] E. Y. Remez. General Computational Methods of Chebyshev Approximation. *US Atomic Energy Commission*, 1962.

[11] Steven A. Gottlieb, W. Liu, D. Toussaint, R.L. Renken, and R.L. Sugar. Hybrid Molecular Dynamics Algorithms for the Numerical Simulation of Quantum Chromodynamics. *Phys.Rev.*, D35:2531–2542, 1987.

[12] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J.Chem.Phys.*, 21:1087–1092, 1953.

[13] M. Fecko. Differential Geometry and Lie Groups for Physicists. 2011.

[14] Alberto Ramos. Playing with the kinetic term in the HMC. *PoS*, LATTICE2012:193, 2012.

[15] Massimo Campostrini and Paolo Rossi. A COMPARISON OF NUMERICAL ALGORITHMS FOR DYNAMICAL FERMIONS. *Nucl.Phys.*, B329:753, 1990.

[16] A.D. Kennedy, M.A. Clark, and P.J. Silva. Force Gradient Integrators. *PoS*, LAT2009:021, 2009.

[17] Siu A. Chin and Donald W. Kidwell. Higher-order force gradient symplectic algorithms. *Phys.Rev.*, E62:8746–8752, 2000.

[18] R.C. Brower, T. Ivanenko, A.R. Levi, and K.N. Orginos. Chronological inversion method for the Dirac matrix in hybrid Monte Carlo. *Nucl.Phys.*, B484:353–374, 1997.

[19] Mike J. Peardon and James Sexton. Multiple molecular dynamics time scales in hybrid Monte Carlo fermion simulations. *Nucl.Phys.Proc.Suppl.*, 119:985–987, 2003.

[20] Martin Hasenbusch. Speeding up the hybrid Monte Carlo algorithm for dynamical fermions. *Phys.Lett.*, B519:177–182, 2001.

[21] C. Urbach, K. Jansen, A. Shindler, and U. Wenger. HMC algorithm with multiple time scale integration and mass preconditioning. *Comput.Phys.Commun.*, 174:87–98, 2006.

[22] Dwight Renfrew, Thomas Blum, Norman Christ, Robert Mawhinney, and Pavlos Vranas. Controlling Residual Chiral Symmetry Breaking in Domain Wall Fermion Simulations. *PoS*, LATTICE2008:048, 2008.

[23] A. Bazavov et al. The chiral transition and $U(1)_A$ symmetry restoration from lattice QCD using Domain Wall Fermions. *Phys.Rev.*, D86:094503, 2012.

[24] Ting-Wai Chiu. Optimal domain wall fermions. *Phys.Rev.Lett.*, 90:071601, 2003.

[25] Richard C. Brower, Hartmut Neff, and Kostas Orginos. Mobius fermions: Improved domain wall chiral fermions. *Nucl.Phys.Proc.Suppl.*, 140:686–688, 2005.

[26] Richard C. Brower, Harmut Neff, and Kostas Orginos. The Móbius Domain Wall Fermion Algorithm. 2012.

[27] Vadim Furman and Yigal Shamir. Axial symmetries in lattice QCD with Kaplan fermions. *Nucl.Phys.*, B439:54–78, 1995.

[28] M. Geradin. The computational efficiency of a new minimization algorithm for eigenvalue analysis. *Journal of Sound Vibration*, 19:319–331, December 1971.

[29] Y. Saad and M.H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3):856–869, 1986.

[30] Roland Freund, Gene H. Golub, and Noel M. Nachtigal. Iterative solution of linear systems. *Acta Numerica*, 1:57–100, 1992.

[31] R.C. Brower, H. Neff, and K. Orginos. Mobius fermions. *Nucl.Phys.Proc.Suppl.*, 153:191–198, 2006.

[32] Artan Borici. Computational methods for the fermion determinant and the link between overlap and domain wall fermions. pages 25–39, 2004.

[33] Alban Allkoci and Artan Borici. Reducing the beta-shift in domain wall fermion simulations. *PoS*, LAT2005:099, 2006.

[34] Andreas Stathopoulos and Kostas Orginos. Computing and deflating eigenvalues while solving multiple right hand side linear systems in quantum chromodynamics. *SIAM J.Sci.Comput.*, 32:439–462, 2010.

[35] Thomas Blum, Taku Izubuchi, and Eigo Shintani. A new class of variance reduction techniques using lattice symmetries. 2012.

[36] Gerhard Buchalla, Andrzej J. Buras, and Markus E. Lautenbacher. Weak decays beyond leading logarithms. *Rev.Mod.Phys.*, 68:1125–1144, 1996.

[37] V. Cirigliano, H. Neufeld, and H. Pichl. K(e3) decays and CKM unitarity. *Eur.Phys.J.*, C35:53–65, 2004.

[38] C.T. Sachrajda and G. Villadoro. Twisted boundary conditions in lattice simulations. *Phys.Lett.*, B609:73–85, 2005.

[39] R. Arthur et al. Domain Wall QCD with Near-Physical Pions. 2012.

[40] Kiyoshi Sasaki and Shoichi Sasaki. Excited baryon spectroscopy from lattice QCD: Finite size effect and hyperfine mass splitting. *Phys.Rev.*, D72:034502, 2005.

[41] Gilberto Colangelo, Stephan Durr, Andreas Juttner, Laurent Lellouch, Heinrich Leutwyler, et al. Review of lattice results concerning low energy particle physics. *Eur.Phys.J.*, C71:1695, 2011.

[42] A. Bazavov et al. Results for light pseudoscalar mesons. *PoS*, LATTICE2010:074, 2010.

[43] Y. Aoki et al. Continuum Limit Physics from 2+1 Flavor Domain Wall QCD. *Phys.Rev.*, D83:074508, 2011.

[44] S. Durr, Z. Fodor, C. Hoelbling, S.D. Katz, S. Krieg, et al. The ratio FK/Fpi in QCD. *Phys.Rev.*, D81:054507, 2010.

[45] J. Noaki et al. Chiral properties of light mesons with N(f) = 2+1 overlap fermions. *PoS*, LAT2009:096, 2009.

[46] Christopher Aubin, Jack Laiho, and Ruth S. Van de Water. Light pseudoscalar meson masses and decay constants from mixed action lattice QCD. *PoS*, LATTICE2008:105, 2008.

[47] Yoshinobu Kuramashi. PACS-CS results for 2+1 flavor lattice QCD simulation on and off the physical point. *PoS*, LATTICE2008:018, 2008.

[48] S.R. Beane, P.F. Bedaque, K. Orginos, and M.J. Savage. f(K)/f(pi) in Full QCD with Domain Wall Valence Quarks. *Phys.Rev.*, D75:094501, 2007.

[49] Gilberto Colangelo. FLAG phase 2: Status and prospects. *PoS*, LATTICE2012:021, 2012.

[50] H. Leutwyler and M. Roos. Determination of the Elements V(us) and V(ud) of the Kobayashi-Maskawa Matrix. *Z.Phys.*, C25:91, 1984.

[51] V. Cirigliano, G. Ecker, M. Eidemuller, Roland Kaiser, A. Pich, et al. The $\langle SPP \rangle$ Green function and SU(3) breaking in K(l3) decays. *JHEP*, 0504:006, 2005.

[52] Johan Bijnens and P. Talavera. K(l3) decays in chiral perturbation theory. *Nucl.Phys.*, B669:341–362, 2003.

[53] Matthias Jamin, Jose Antonio Oller, and Antonio Pich. Order p^6 chiral couplings from the scalar $K\pi$ form-factor. *JHEP*, 0402:047, 2004.

[54] D. Becirevic, G. Isidori, V. Lubicz, G. Martinelli, F. Mescia, et al. The $K \to \pi$ vector form-factor at zero momentum transfer on the lattice. *Nucl.Phys.*, B705:339–362, 2005.

[55] Masataka Okamoto. Full CKM matrix with lattice QCD. 2004.

[56] T. Kaneko et al. Kaon semileptonic form factors in QCD with exact chiral symmetry. *PoS*, LATTICE2011:284, 2011.

[57] P.A. Boyle, A. Juttner, R.D. Kenway, C.T. Sachrajda, S. Sasaki, et al. K(l3) semileptonic form-factor from 2+1 flavour lattice QCD. *Phys.Rev.Lett.*, 100:141601, 2008.

[58] A. Bazavov, C. Bernard, C.M. Bouchard, C. DeTar, Daping Du, et al. Kaon semileptonic vector form factor and determination of $|V_{us}|$ using staggered fermions. 2012.

[59] F. Ambrosino et al. Precision Measurement of KS Meson Lifetime with the KLOE detector. *Eur.Phys.J.*, C71:1604, 2011.

[60] Mario Antonelli, David Mark Asner, Daniel Adams Bauer, Thomas G. Becher, M. Beneke, et al. Flavor Physics in the Quark Sector. *Phys.Rept.*, 494:197–414, 2010.

[61] J.C. Hardy and I.S. Towner. Superallowed 0+ → 0+ nuclear beta decays: A New survey with precision tests of the conserved vector current hypothesis and the standard model. *Phys.Rev.*, C79:055502, 2009.

[62] Vincenzo Cirigliano and Helmut Neufeld. A note on isospin violation in Pl2(gamma) decays. *Phys.Lett.*, B700:7–10, 2011.

[63] Y. Aoki, R. Arthur, T. Blum, P.A. Boyle, D. Brommel, et al. Continuum Limit of B_K from 2+1 Flavor Domain Wall QCD. *Phys.Rev.*, D84:014503, 2011.

[64] Jangho Kim, Chulwoo Jung, Hyung-Jin Kim, Weonjong Lee, and Stephen R. Sharpe. Finite volume effects in B_K with improved staggered fermions. *Phys.Rev.*, D83:117501, 2011.

[65] Ying-He Zhao, Jia-Sheng Huang, M.L.N. Ashby, G.G. Fazio, and S. Miyazaki. The Deep Optical Imaging of the Extended Groth Strip. *Res.Astron.Astrophys.*, 9:1061–1077, 2009.

[66] Elvira Gamiz et al. Unquenched determination of the kaon parameter B(K) from improved staggered fermions. *Phys.Rev.*, D73:114502, 2006.

[67] T. Blum, P.A. Boyle, N.H. Christ, N. Garron, E. Goode, et al. The $K \to (\pi\pi)_{I=2}$ Decay Amplitude from Lattice QCD. *Phys.Rev.Lett.*, 108:141601, 2012.

[68] T. Blum, P.A. Boyle, N.H. Christ, N. Garron, E. Goode, et al. K to $\pi\pi$ Decay amplitudes from Lattice QCD. *Phys.Rev.*, D84:114503, 2011.

[69] R. Arthur et al. Domain Wall QCD with Near-Physical Pions. 2012.

Appendix A

Force Implementation

A.1 Fermion forces

Let's take a quotient fermion action as an example,

$$S(m_f, m_b) = \phi^\dagger \mathcal{M}_b \frac{1}{\mathcal{M}_f^\dagger \mathcal{M}_f} \mathcal{M}_b^\dagger \phi, \tag{A.1}$$

where $\mathcal{M}_b$ and $\mathcal{M}_f$ are shorthands for $\mathcal{M}(m_b)$ and $\mathcal{M}(m_f)$. The force is proportional to

$$e_i(S) = \phi^\dagger e_i\left(\mathcal{M}_b\right) \frac{1}{\mathcal{M}_f^\dagger \mathcal{M}_f} \mathcal{M}_b^\dagger \phi - \phi^\dagger \mathcal{M}_b \frac{1}{\mathcal{M}_f^\dagger \mathcal{M}_f} \mathcal{M}_f^\dagger e_i\left(\mathcal{M}_f\right) \frac{1}{\mathcal{M}_f^\dagger \mathcal{M}_f} \mathcal{M}_b^\dagger \phi + \text{h.c.}. \tag{A.2}$$

Only the following Dirac equation needs to be solved to evaluate the above force

$$\mathcal{M}_f^\dagger \mathcal{M}_f x = \mathcal{M}_b^\dagger \phi. \tag{A.3}$$

Further more, we can implement the following general fermion force to calculate all individual force terms in (A.2)

$$F_i = c\phi_1^\dagger e_i(\mathcal{M})\phi_2 + \text{h.c.}, \tag{A.4}$$

where ϕ_1 and ϕ_2 are 2 different vectors and c is a real scaling factor for convenience. This form can also be used to calculate all force terms generated by the rational action (2.39).

Since $\mathcal{M} = M_{oo} - M_{oe}M_{ee}^{-1}M_{eo}$ and $e_i(M_{oo}) = e_i(M_{ee}) = 0$,

$$F_i = -c\phi_1^\dagger e_i\left(M_{oe}\right) M_{ee}^{-1} M_{eo}\phi_2 - c\phi_1^\dagger M_{oe}M_{ee}^{-1}e_i\left(M_{eo}\right)\phi_2 + \text{h.c.}. \tag{A.5}$$

M_{eo} takes the following form for a general Möbius action

$$M_{eo}(x,y) = -\frac{1}{2}D_{eo}(x,y)B_{oo}, \tag{A.6}$$

where B_{oo} is another constant matrix that depends only on the Möbius parameters. D_{eo} is the only gauge dependent quantity,

$$D_{eo}(x,y) = \sum_\mu \left((1-\gamma_\mu)U_\mu(x)\delta_{x+\mu,y} + (1+\gamma_\mu)U_\mu^\dagger(x-\mu)\delta_{x-\mu,y}\right). \tag{A.7}$$

So the force F_i can be written as

$$F_i = \frac{1}{2}c\phi_1^\dagger e_i\left(D_{oe}\right) B_{ee}M_{ee}^{-1}M_{eo}\phi_2 + \frac{1}{2}c\phi_1^\dagger M_{oe}M_{ee}^{-1}e_i\left(D_{eo}\right)B_{oo}\phi_2 + \text{h.c.}. \tag{A.8}$$

For simplicity we define two vectors v_1 and v_2 which have both even and odd parts

$$v_1 = (v_{1o}, v_{1e}) = \left(\phi_1, M_{ee}^{-\dagger}M_{eo}^\dagger\phi_1\right) \tag{A.9}$$

$$v_2 = (v_{2o}, v_{2e}) = \left(B_{oo}\phi_2, B_{ee}M_{ee}^{-1}M_{eo}\phi_2\right). \tag{A.10}$$

Using v_1 and v_2 we have

$$F_i = \frac{1}{2}cv_1^\dagger e_i(D)v_2 + \text{h.c.}. \tag{A.11}$$

And $e_i(D)$ is straightforward to evaluate

$$e_i(D) = \sum_\mu \big((1-\gamma_\mu) T_i U_\mu(x) \delta_{x+\mu,y} - (1+\gamma_\mu) U_\mu^\dagger T_i(x-\mu) \delta_{x-\mu,y} \big) \,. \tag{A.12}$$

A.2 Gauge forces

Taking the Wilson gauge action (2.1) as an example,

$$e_i \left(S_G^W(U) \right) = -\frac{1}{3} \beta \sum_{x,\mu<\nu} \mathrm{ReTr} \left(e_i U_P(x,\mu,\nu) \right) . \tag{A.13}$$

For a specific derivative $e_i = e^a_{\mu x}$ only plaquettes with the corresponding link $U_\mu(x)$ have nonzero derivatives. So

$$e^a_{\mu x} \left(S_G^W(U) \right) = -\frac{1}{3} \beta \sum_\nu \mathrm{ReTr} \big(e_i \left(U_\mu(x) \right) \cdot U_\nu(x+\mu) U_\mu^\dagger(x+\nu) U_\nu^\dagger(x) \big) \,. \tag{A.14}$$

Appendix B

Lattice Ensembles Used in This Work

In this appendix we summarize various ensembles discussed in the text. References are also given so the reader can find additional details.

All of the following ensembles are generated using 2+1 flavor dynamic fermions. The u and d quarks are treated as degenerated flavors and s quark is added as the third flavor. All heavy quarks are ignored since with the current lattice spacing it is not possible to count the effect accurately.

We use the Iwasaki gauge action in all of these ensembles, which was discussed in chapter 2. Some ensembles also include the Dislocation Suppressing Determinant Ratio (DSDR) [22] to suppress the tunneling of topological charge.

B.1 Zero Temperature Ensembles

The main ensembles related to this work are the $48^3 \times 96$ and $64^3 \times 128$ ensembles. Two smaller ensembles $16^3 \times 32$ and $32^3 \times 64$ [68, 69] are also used throughout this work, mainly for the purpose of determining the effectiveness of the algorithms. Their basic properties are listed in table B.1. We label these ensembles using the notation `lattice_size(physical_size, pion_mass)` throughout the text. Where `lattice_size` is the number of lattice points in spatial and temporal directions, `physical_size` is the physical size of the spatial dimensions in femtometer (fm) and `pion_mass` is the dynamic pion mass in MeV. For example, the 4 ensembles listed in table B.1 are referred as $16^3 \times 32$ (1.8fm, 420MeV), $32^3 \times 64$ (4.6fm, 170MeV), $48^3 \times 96$ (5.5fm, 140MeV) and $64^3 \times 128$ (5.5fm, 140MeV), respectively.

size	$16^3 \times 32$[a]	$32^3 \times 64$[b]	$48^3 \times 96$[c]	$64^3 \times 128$ [c]
a^{-1}(GeV)	1.741(23)	1.370(8)	1.741(23)	2.302(35)
physical volume	$(1.81\text{fm})^3$	$(4.61\text{fm})^3$	$(5.44\text{fm})^3$	$(5.49\text{fm})^3$
Gauge action	Iwasaki	Iwasaki + DSDR	Iwasaki	Iwasaki
β	2.13	1.75	2.13	2.25
Fermion action	DWF	DWF	Möbius	Möbius
Möbius b	1	1	1.5	1.5
Möbius c	0	0	0.5	0.5
L_s	16	32	24	12
m_π	0.24373(47)	0.1250(2)	0.08054(46)	0.0584
m_K		0.3594(5)	0.28855(60)	0.214
m_π(MeV)	424	171	140	134
m_K(MeV)		492	502	493
m_l(input)	0.01	0.001	7.8e-4	6.78e-4
m_s(input)	0.032	0.045	0.0362	0.02661
$m_{\text{res}}(\times 10^{-4})$	30.8(4)	18.510(43)	6.19(6)	2.93(8)

[a] Data for this ensemble mainly come from [68].
[b] Data for this ensemble mainly come from [69].
[c] These ensembles are described by this work.

Table B.1: Zero temperature ensembles used in this work.

B.2 Finite Temperature Ensembles

The $32^3 \times 8$ finite temperature ensembles are generated using the Iwasaki gauge action with the Dislocation Suppressing Determinant Ratio (DSDR) [22]. The input light and strange quark masses are set to values such that all ensembles have pion and kaon masses at physical values. All ensembles listed here are simulated with 2+1 flavor Möbius domain wall fermions.

T (MeV)	β	L_s	c	m_l	m_s	$m_{\text{res}}^{(\text{esti.})}$
139	1.633	24	1.5	0.00022	0.05960	0.00211
149	1.671	16	1.5	0.00034	0.05538	0.00164
154	1.689	16	1.5	0.00075	0.05376	0.00124
159	1.707	16	1.5	0.00112	0.05230	0.00088
164	1.725	16	1.5	0.00120	0.05045	0.00061
168	1.740	16	1.2	0.00126	0.04907	0.00058
177	1.771	16	1.0	0.00132	0.04614	0.00043
186	1.801	16	1.0	0.00133	0.04345	0.00031
195	1.829	16	0.9	0.00131	0.04122	0.00014

Table B.2: Finite temperature ensembles related to this work. The size of the lattice is $32^3 \times 8$ for all listed ensembles. All ensembles use 2+1 flavor Möbius domain wall fermions with uniform b_i and c_i (so they are the same on different s slices). The c values are listed in the table, while b is always equal to $c + 1$.