

# The Rational Hybrid Monte Carlo Algorithm

Michael A. Clark



Doctor of Philosophy

The University of Edinburgh

2005

# Abstract

This thesis is concerned with the problem of generating gauge configurations for use with Monte Carlo lattice QCD calculations that include the effect of dynamical fermions. Although such effects have been included in calculations for a long time, historically it has been difficult to include the effect of the strange quark because of the square root of the Dirac operator that appears in the action.

The lattice formulation of QCD is discussed, and the various fermion formulations are highlighted. Current popular algorithms used to generate gauge configurations are described, in particular the advantages and disadvantages of each are discussed.

The Rational Hybrid Monte Carlo algorithm (RHMC) is introduced, this uses rational functions to approximate the matrix square root and is an exact algorithm. RHMC is compared with the Polynomial Hybrid Monte Carlo algorithm and the inexact R algorithm for two flavour staggered fermion calculations. The algorithm is found to reproduce published data and to be more efficient than the Polynomial Hybrid Monte Carlo algorithm. With the introduction of multiple time scales for the gauge and fermion parts of the action the efficiency further increases.

As a means to accelerate the Monte Carlo acceptance rate of lattice QCD calculations, the splitting of the fermion determinant into  $n^{th}$  root contributions is described. This is shown to improve the conservation of the Hamiltonian. As the quark mass is decreased this is found to decrease the overall cost of calculation by allowing an increase in the integrating stepsize.

An efficient formulation for applying RHMC to ASQTAD calculations is described, and it is found to be no more expensive than using the conventional R



algorithm formulation.

Full 2+1 quark flavour QCD calculations are undertaken using the domain wall fermion formulation. Results are generated using both RHMC and the R algorithm and comparisons are made on the basis of algorithm efficiency and hadronic observables. With the exception of the stepsize errors present in the R algorithm data, consistency is found between the two algorithms. RHMC is found to allow a much greater integrating stepsize than the R algorithm.

# Declaration

This thesis has been composed by me and contains work done by me as a member of the UKQCD collaboration. Chapters 2 and 3 are mainly background, though some the error analysis is original and was done in conjunction with my supervisor. Subsequent chapters are original work, unless cited otherwise. The discussion at the beginning of chapter 6 regarding multiple pseudofermions is based on conversation with my supervisor.

Some work presented in this thesis has been published in conference proceedings. This can be summarised as follows:

- Chapter 3: *Nucl. Phys. (Proc Suppl.)* **B119** (2003) 1015.
- Chapter 4: *Nucl. Phys. (Proc Suppl.)* **B129** (2004) 850.
- Chapter 6: *Nucl. Phys. (Proc Suppl.)* **B140** (2005) 838.
- Chapters 7 and 8: *Nucl. Phys. (Proc Suppl.)* **B40** (2005) 835.

All of this work was written within the Columbia Physics System which is available at <http://qcdoc.phys.columbia.edu/doxygen/usr/intropage.html>. The original implementation of some of the ASQTAD force code referred to in chapter 7 was written by Zbigniew Sroczynski and Chulwoo Jung. The gauge configurations using the R algorithm in chapter 8 were generated by Chris Maynard and Rob Tweedie, all other data were generated by me. The hadron spectroscopy analysis was done by me using code developed from existing UKQCD code. The autocorrelation code was written by Balint Joó.

Michael Clark

## Acknowledgements

I would like to thank all my colleagues at Edinburgh and in the UKQCD collaboration for any help and support they have given me over the course of my doctorate, in particular my supervisors, Tony Kennedy and Richard Kenway. This thesis would not have been possible without Tony, with whom I have had many illuminating discussions. Thanks to Richard for having faith in me to be part of the QCDOC project at Columbia.

I would like to thank members of the QCDOC team at Columbia University and Brookhaven National Laboratory, who all made me feel very welcome during my stay in New York. A special thanks goes to Norman Christ for taking me on as his student and to Peter Boyle with who I discovered many pubs in New York (and learnt a thing or two about the QCDOC).

Thanks also to my family and friends for putting up with me all these years, especially through the last six months.

Finally, I would like to dedicate this thesis to my wife Vivienne, I'm sure by going through thesis hell together we shall only be closer.

# Contents

1	Introduction	1
2	Background Theory	5
2.1	Lattice quantum chromodynamics . . . . .	5
2.2	Fermions on the lattice . . . . .	6
2.2.1	Naïve Fermions on the Lattice . . . . .	7
2.2.2	Wilson Fermions . . . . .	8
2.2.3	Staggered Fermions . . . . .	9
2.3	Gauge fields on the lattice . . . . .	11
2.4	Improved Fermion Formulations . . . . .	13
2.4.1	ASQTAD Fermions . . . . .	13
2.4.2	Ginsparg-Wilson Fermions . . . . .	15
2.4.3	Domain-Wall Fermions . . . . .	16
2.5	Hadron Spectroscopy . . . . .	18

2.5.1	Correlation Functions . . . . .	18
2.5.2	Extracting the Mass . . . . .	21
2.5.3	Chiral Extrapolations . . . . .	22
2.5.4	Dimensionful Quantities . . . . .	23
2.6	Topological Charge . . . . .	24
2.7	Autocorrelation Length . . . . .	25
<b>3</b>	<b>Algorithms</b>	<b>27</b>
3.1	Generation of gauge configurations . . . . .	27
3.2	Hybrid Monte Carlo . . . . .	29
3.2.1	Molecular Dynamics . . . . .	30
3.2.2	The $\Phi$ Algorithm . . . . .	32
3.2.3	The $\chi$ Algorithm . . . . .	34
3.2.4	Hybrid Monte Carlo . . . . .	34
3.3	The $R_0$ algorithm . . . . .	36
3.4	The R algorithm . . . . .	38
3.5	Polynomial Hybrid Monte Carlo . . . . .	40
3.6	Integration Scheme . . . . .	44
3.7	Sexton-Weingarten Integration . . . . .	46
3.8	Krylov Solvers . . . . .	47



3.8.1	Conjugate Gradients . . . . .	48
3.8.2	Multi-shift Solvers . . . . .	50
3.9	Chronological Inversion . . . . .	52
<b>4</b>	<b>Rational Hybrid Monte Carlo</b>	<b>55</b>
4.1	Rational Approximations . . . . .	56
4.2	Remez Algorithm . . . . .	58
4.3	Partial fraction expansions and multishift solvers . . . . .	60
4.4	Rational Hybrid Monte Carlo . . . . .	61
4.4.1	Pseudofermion Heatbath . . . . .	62
4.4.2	Computation of the Force . . . . .	63
4.4.3	Energy measurement . . . . .	65
4.4.4	Integration Scheme . . . . .	65
4.5	Algorithm Testing . . . . .	66
<b>5</b>	<b>Extensions to the Multi-Shift solver</b>	<b>77</b>
5.1	A Chronological Multi-Shift Solver? . . . . .	78
5.2	The Generalised Multi-Shift Solver . . . . .	80
5.3	The R2 Algorithm . . . . .	81
<b>6</b>	<b>Monte Carlo Acceleration</b>	<b>85</b>

6.1	Hasenbusch's Method and Variants . . . . .	86
6.2	Nroots Acceleration . . . . .	89
6.3	Results . . . . .	90
<b>7</b>	<b>Exact 2+1 Flavour Asqtad Fermions</b>	<b>99</b>
7.1	Algorithm Implementation . . . . .	100
7.1.1	Action Formulation . . . . .	100
7.1.2	Force Term . . . . .	101
7.2	Algorithm Testing . . . . .	103
<b>8</b>	<b>Exact 2+1 Flavour Domain Wall Fermions</b>	<b>105</b>
8.1	Action Formulation . . . . .	105
8.2	Eigenvalue bounds . . . . .	107
8.3	Algorithm Testing . . . . .	110
<b>9</b>	<b>Concluding remarks</b>	<b>133</b>
9.1	Summary . . . . .	133
9.2	Future Work . . . . .	135
<b>A</b>	<b>Staggered Fermion Results</b>	<b>141</b>
<b>B</b>	<b>Nroots Results</b>	<b>145</b>

C Domain Wall Results 153

D QCDOC Implementation 159

D.1 Optimising Matrix Inversion for QCDOC . . . . . 159

D.2 Asqtad RHMC Force Implementation . . . . . 163

# Chapter 1

## Introduction

The verification of quantum chromodynamics (QCD) as the correct theory describing the strong force has been a long and drawn out affair. Due to the non-perturbative nature of the theory, lattice QCD has been seen as the principle tool to achieve this aim. The subject has existed as an ever-evolving field for the past thirty years, and throughout this time there have been many advances in the subject, and as QCD has been explored through such studies, the verification of QCD has become tantalisingly close.

Lattice QCD is characterised in some sense by the immense computational cost required to make predictions about the strong force, whether those predictions be related to particle masses, matrix elements or the prediction of yet unknown particles. For all lattice QCD calculations a statistical mechanics approach is used: namely an ensemble of possible snapshots (so called gauge field configurations) of the QCD vacuum is generated, upon each of these snapshots measurements of the required observable are made, and the mean value of these measurements is a statistical measure of the desired observable. The QCD vacuum consists of a sea of gluons (the force mediating particles) and of

quarks (the particles which are the constituents of baryons and mesons) that are spontaneously created and destroyed according to Heisenberg's energy-time uncertainty principle. There are six quarks present in nature, in increasing mass order they are: the up and down (referred to in this work as the light pair), strange, charm, bottom and top. Since the mass of the quark is inversely proportional to the time it is present before annihilation, the heavier three quarks have a negligible effect on the vacuum compared to the lighter three, and so only these latter three need to be included when generating the ensembles. Lattice QCD calculations which include the effect of these three light quarks are known as 2+1 quark flavour calculations (from the light pair and the single strange). It is the inclusion of the quarks to the QCD vacuum that accounts for the prohibitive cost of QCD calculations.

For years this computational cost of calculation proved to be an insurmountable barrier to precise QCD prediction. As a result, various approximations were necessary: the quenched approximation where the effect of quarks upon the vacuum are ignored resulting in an uncontrolled 10% systematic error, in more recent work the quark effects have been included but using unrealistically massive quarks (the computational cost increases with decreasing quark mass), also popular has been the use of approximate fermion formulations which do not respect all of the symmetries present in nature but do result in a much reduced computational cost.

Since lattice QCD's inception, computers have grown increasingly more powerful allowing more realistic calculations. The work presented here has in part been motivated by the QCDOC supercomputer, which at the time of writing allows a previously unprecedented amount of computer power to be channelled solely at lattice QCD calculations. It is hoped that the results produced using QCDOC shall come closer than ever before to verifying that QCD is the correct theory describing the strong force.



Of equal importance has been the development of algorithms which deal with the generation of the vacuum ensembles. As computers have doubled in performance every couple of years, increasingly more efficient algorithms have been developed which reduce the computational cost requirements for a given calculation. The Hybrid Monte Carlo algorithm (and the various refinements applied to it has been the standard algorithm from which most calculations have been performed. Unfortunately, this algorithm cannot efficiently generate ensembles of configurations with only a single quark flavour present, and so various other algorithms must be used which are either not as efficient (i.e., Polynomial Hybrid Monte Carlo), or have inherent systematic errors present (the R algorithm).

The Rational Hybrid Monte Carlo (RHMC) algorithm presented in this work rectifies this problem, in that it allows an arbitrary number of quark flavours to be included in the background vacuum, has the efficiency of Hybrid Monte Carlo, and is exact, i.e., is free of any systematic errors. This is an extremely general algorithm that can be applied to all fermion formulations currently used in lattice QCD studies.

This work is presented as follows: in chapter 2 the background theory concerning this work is presented, namely an overview of lattice QCD with emphasis on the fermion formulations that are used and how physical calculations are made. The following chapter describes the current algorithms that are used to generate the vacuum ensembles, specifically discussing the advantages and disadvantages of each of these. RHMC is introduced in chapter 4, the method of implementing this algorithm is given and the algorithm is compared against published data. Following this chapter there is a slight diversion off topic looking if it is possible to speed up the matrix inversion calculations applied in RHMC. Chapters 7 and 8 are devoted to applying RHMC to the ASQTAD and domain wall fermion formulations respectively. The conclusions of each of these chapters are summed up in chapter 9 and future avenues of exploration are also discussed.



# Chapter 2

## Background Theory

### 2.1 Lattice quantum chromodynamics

The starting point for QCD calculations is the path integral describing an observable  $\mathcal{O}$  where the vacuum effects of the up, down and strange quarks have been included

$$\langle 0|\mathcal{O}|0\rangle = \frac{1}{Z} \int \mathcal{D}A \mathcal{D}\bar{\psi}_u \mathcal{D}\psi_u \mathcal{D}\bar{\psi}_d \mathcal{D}\psi_d \mathcal{D}\bar{\psi}_s \mathcal{D}\psi_s \mathcal{O} e^{iS_g + iS_u + iS_d + iS_s}, \quad (2.1)$$

where  $S_g = \frac{1}{2} \text{tr} \int d^4x F_{\mu\nu} F^{\mu\nu}$  is the pure gauge action with field strength tensor  $F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu + ig_0[A_\mu, A_\nu]$ , with  $g_0$  the bare coupling. The fermion actions are  $S_f = \int d^4x \bar{\psi}_f (i\mathcal{D} - m_f) \psi_f$  with mass  $m_f$  and  $\mathcal{D}$  is the product of the  $\gamma$  spin matrices and the covariant derivative which ensures a local  $SU(3)$  symmetry. The fields  $\psi$  are Grassman valued because these represent fermionic variables.

A Wick rotation is performed on the time coordinate  $x_0 \rightarrow -ix_4$ , and the path integral thus becomes

$$\langle 0|\mathcal{O}|0\rangle = \frac{1}{Z} \int \mathcal{D}A \mathcal{D}\bar{\psi}_u \mathcal{D}\psi_u \mathcal{D}\bar{\psi}_d \mathcal{D}\psi_d \mathcal{D}\bar{\psi}_s \mathcal{D}\psi_s \mathcal{O} e^{-S_g - S_u - S_d - S_s}, \quad (2.2)$$

where all metrics are measured in Euclidean space. With this formulation, the

action is real and the exponential factor  $e^{-S}$  can be thought of a statistical weight in direct analogy to statistical mechanics.

The path integral is only formally defined, and there is no direct means of evaluating it. The integral has an infinite number of degrees of freedom and is only meaningful once it is defined on a discrete space-time lattice.

The Grassman valued fermion fields can be integrated out using Gaussian integration,

$$\langle 0|\mathcal{O}|0\rangle = \frac{1}{Z} \int \mathcal{D}A \det(\not{D} + m_u) \det(\not{D} + m_d) \det(\not{D} + m_s) \mathcal{O} e^{-S_g}. \quad (2.3)$$

Since  $m_u \approx m_d \equiv m_l$  this can be rewritten as

$$\langle 0|\mathcal{O}|0\rangle = \frac{1}{Z} \int \mathcal{D}A \det(\not{D} + m_l)^2 \det(\not{D} + m_s) \mathcal{O} e^{-S_g}. \quad (2.4)$$

It is from this formulation of 2+1 quark flavour QCD that all of the proceeding work shall follow.

## 2.2 Fermions on the lattice

Historically, placing fermions on the lattice has been fraught with challenges with regards to both formulation and computational cost. The formulaic difficulty with placing fermions on the lattice is due to the so called “fermion doubling problem”. There are many approaches to circumvent this problem, each with their own inherent drawbacks.

### 2.2.1 Naïve Fermions on the Lattice

In free field theory (and Euclidean space) the one flavour fermion contribution to the action is given by

$$S_f = \int dx^4 \bar{\psi}(\not{\partial} + m_f)\psi. \quad (2.5)$$

When formulating on the lattice, the fermionic fields are discretised and placed in a finite box with periodic boundary conditions. The derivative operator is replaced by a symmetric lattice derivative (a symmetric lattice derivative is required for a renormalisable theory, otherwise the forward difference derivative could have been used [1]) and the integral over all space-time is replaced by a summation over the lattice sites. Proceeding naïvely as just described, then the free field lattice fermion action is written

$$S_f^n = \sum_{i,j} \bar{\psi}_{i,\alpha} M_{x\alpha,y\beta}^n \psi_{j,\beta}, \quad (2.6)$$

where

$$M_{i,\alpha;j,\beta}^n = \frac{1}{2a} \sum_{\mu,i} [(\gamma_\mu)_{\alpha\beta} \delta_{j,i+\hat{\mu}} - (\gamma_\mu)_{\alpha\beta} \delta_{j,i-\hat{\mu}}] + m_f \delta_{i,j} \delta_{\alpha,\beta}, \quad (2.7)$$

and  $\alpha$  and  $\beta$  represent the spinor indices. The derivative has been replaced by the central difference approximation. The fermion fields  $\psi$  now live on the lattice sites  $i$ , separated by a distance  $a$  which is the lattice spacing. The presence of a lattice spacing acts as an ultra-violet cutoff in momentum space, and the finite size of the box provides a natural infra-red cutoff. Hence, through discretisation and placing the fields within a box the divergences present in continuum QCD are removed.

For this prescription to be correct, it must reproduce the correct continuum field theory in the limit that the lattice spacing goes to zero and the box size is taken to infinity. The obvious candidate for testing whether this action describes a single Dirac fermion species is the fermion propagator which can be found using



Fourier analysis. It is found that there is a pole in the propagator at each corner of the Brillouin zone. Since each pole in the propagator corresponds to a fermion flavour, this corresponds to a doubling of the number of fermions for each dimension of the lattice theory. Hence in four dimensions, where it was thought there was a single fermion species, there are in fact sixteen degenerate flavours. The cause of the fermion doubling is due to the nature of the lattice derivative used, the central difference derivative has two eigenvectors, one corresponding to the expected solution, and another oscillatory solution which gives rise to the doublers [1].

### 2.2.2 Wilson Fermions

The naïve discretisation of the Dirac operator is merely the simplest one that can be constructed. Any discretisation can be chosen, so long as it has the correct continuum limit. Specifically, terms can be added to the discretised operator which vanish as the continuum limit is approached.

The Wilson fermion prescription is to subtract a second derivative term from the naïve discretised operator [2]. Hence the Wilson action is given by

$$S_f^w = \sum_{i,j} \bar{\psi}_{i,\alpha} M_{i\alpha,j\beta}^w \psi_{j,\beta}, \quad (2.8)$$

where

$$M_{i\alpha,j\beta}^w = -\frac{1}{2a} \sum_{\mu,i} [(r - \gamma_\mu)_{\alpha\beta} \delta_{j,i+\hat{\mu}} + (r + \gamma_\mu)_{\alpha\beta} \delta_{j,i-\hat{\mu}}] + (m + 4r) \delta_{i,j} \delta_{\alpha,\beta}. \quad (2.9)$$

The parameter  $r$  is known as the Wilson parameter, and is typically set equal to unity.

The addition of this second derivative term to the action does not alter the naïve continuum limit since it vanishes linearly with the lattice spacing, hence

the Wilson fermion formulation has  $O(a)$  leading errors. However, it explicitly breaks the chiral symmetry of the formulation rendering it useless for studying physics upon which chiral symmetry is important, e.g. the restoration of chiral symmetry in finite temperature QCD [1]. Indeed it was shown in [3] that it is not possible to construct a lattice fermion theory which possesses all three of chirality, locality and lack of doublers. This is known as the lattice fermion “no go” theorem.

The propagator corresponding to this operator now has a momentum dependent additive mass renormalisation

$$m(p) = m + \frac{2r}{a} \sum_{\mu} \sin^2(p_{\mu}a/2). \quad (2.10)$$

At the corners of the Brillouin zone, where there were poles in the propagator previously, there is a divergence as the lattice spacing is brought to zero. This removes the unwanted poles, and the correct continuum limit describing a single Dirac fermion species is recovered.

When using the Wilson formulation, it is convenient to rescale the action in terms of the  $\kappa$  parameter, where  $\kappa = 1/(2m + 8)$ . Hence the Wilson matrix kernel is written

$$M_{i\alpha,j\beta}^w = \delta_{i,j}\delta_{\alpha,\beta} - \kappa \sum_{\mu,i} [(r - \gamma_{\mu})_{\alpha\beta}\delta_{j,i+\hat{\mu}} + (r + \gamma_{\mu})_{\alpha\beta}\delta_{j,i-\hat{\mu}}]. \quad (2.11)$$

The Wilson kernel is not bounded from below by zero, and can have eigenvalues with magnitude equal to zero.

### 2.2.3 Staggered Fermions

The staggered fermion approach to reducing the number of fermion flavours is much more subtle than the explicit Wilson term applied in the case of Wilson fermions. The actual reason for the fermion doubling is because the lattice

derivative that is applied uses a lattice spacing that is twice the fundamental lattice spacing, which leads to the oscillatory eigenvector solution. If it were somehow possible to double the lattice spacing used for the fermionic degrees of freedom, but keep the spacing the same for the lattice derivative, then the fermion doubling problem would be overcome. This is essentially what the staggered fermion formulation does, double the fermion lattice spacing by spreading the spin degrees of freedom over adjacent lattice sites. This is allowed if the fermion action is diagonalised in spin space, decoupling the spin degrees of freedom [4]. This diagonalisation is done through a local change of variables of the fermion fields which reduces the spin matrices to simple phase factors [1]. In four dimensions this reduces the number of fermion flavours from sixteen to four.

The staggered fermion action is given by

$$S_f^S = \sum_{i,j} \bar{\psi}_i M_{i,j}^S \psi_j, \quad (2.12)$$

where

$$M_{i,j}^S = -\frac{1}{2a} \sum_{\mu,i} \eta_{x,\mu} [\delta_{j,i-\hat{\mu}} - \delta_{j,i+\hat{\mu}}] + m\delta_{i,j}. \quad (2.13)$$

The staggered kernel is bounded from below by the mass parameter, and the momentum dependent mass renormalisation of the Wilson formulation is not present. The staggered formulation does not preserve the chiral symmetry since the concept of left and right handedness is removed with the removal of the spin matrices. However, a remnant  $U(1)$  symmetry remains which can be related to with the original chiral symmetry [1].

The spreading out of the fermion degrees of freedom does have a large drawback. The formulation gives rise to flavour symmetry breaking, an  $O(a^2)$  effect, which gives rise to unphysical interactions. It is this failing and the fact that staggered fermions describe four fermion flavours, not a single flavour, which are the most objectionable aspects of the formulation. This latter problem can



be circumvented if it is believed that taking the fourth root of the staggered determinant leads to theory of a single fermion flavour. While such an operation is permissible in the continuum limit, on the lattice this might lead to a non-local theory which has no continuum limit [5, 6]. Despite these objections, staggered fermions and improvements thereof (see § 2.4.1) are a very popular fermion choice mainly because of the reduced computational cost compared to other formulations.

## 2.3 Gauge fields on the lattice

Compared to fermion fields, the process of discretising gauge fields is a straightforward process free of problems. On the lattice the gauge field is described by link variables which are  $SU(3)$  matrices that live between the sites on which fermion fields are located. The relationship between the gauge link field  $U_{i,\mu}$  and the continuum gauge field  $A$  is given by

$$U_{i,\mu} = \exp(ia g_0 A_\mu(x)). \quad (2.14)$$

The only gauge invariant quantities that can be constructed on the lattice are closed loops of ordered link products. Hence all lattice gauge actions are built from such quantities. The action which was introduced by Wilson [7] is

$$S_g^w[U] = -\frac{\beta}{3} \sum_{i,\mu < \nu} P[U]_{i,\mu\nu} \quad (2.15)$$

where  $P[U]_{i,\mu\nu}$  is the real part of the trace of the path ordered product of links around the  $1 \times 1$  plaquette in the  $\mu, \nu$  plane at point  $i$  and  $\beta \sim 1/g_0^2$ . This is the most basic gauge action that can be constructed, and has leading errors  $O(a^2)$ . The lattice average of the plaquette is a very useful quantity when determining whether two ensembles of data correspond to the same physics, e.g., comparing the ensembles produced by two different algorithms.

The next class of gauge action to consider are those constructed from using the plaquette plus  $1 \times 2$  rectangle. Such actions can be written in the form

$$S_g^{\text{DBW2}}[U] = -\frac{\beta}{3} \left( (1 - 8c_1) \sum_{i,\mu < \nu} P[U]_{i,\mu\nu} + c_1 \sum_{i,\mu \neq \nu} R[U]_{i,\mu\nu} \right), \quad (2.16)$$

where  $R[U]_{i,\mu\nu}$  is the real part of the trace of the ordered product of link matrices along  $1 \times 2$  rectangles in the  $\mu, \nu$  plane. The parameter  $c_1$  can be calculated in a number of ways, though of relevance in this work is the DBW2 [8, 9] gauge action,  $c_1 = -1.4069$ , where the coefficient is calculated non-perturbatively using Swendsen's blocking and the Schwinger-Dyson method.

The Symanzik one loop improved gauge action [10] is the most complicated gauge action covered in this work and is given by

$$S_g^{\text{SymOL}}[U] = -\frac{\beta}{3} \left( c_0 \sum_{i,\mu < \nu} P[U]_{i,\mu\nu} + c_1 \sum_{i,\mu \neq \nu} R[U]_{i,\mu\nu} + c_2 \sum_{i,\mu < \nu < \sigma} C[U]_{i,\mu\nu\sigma} \right), \quad (2.17)$$

where  $C[U]_{i,\mu\nu\sigma}$  denotes the real part of the trace of the ordered product of link matrices along the  $\mu, \nu, \sigma, -\mu, -\nu, -\sigma$  path. The coefficients  $c_0, c_1$  and  $c_2$  are computed in tadpole improved one loop perturbation theory, and result in a cancellation of the leading  $O(a^2)$  errors. Of the three gauge actions discussed here, it is the Symanzik gauge action which approaches the continuum limit fastest.

With the addition of the gauge field onto the lattice, the chosen fermion formulation must be promoted from the free field theory to the fully interacting case. This is done through replacing the central difference derivative used in equations (2.7), (2.9) and (2.13) with the lattice covariant derivative, i.e.,

$$\delta_{j,i+\hat{\mu}} - \delta_{j,i-\hat{\mu}} \rightarrow U_{i,\mu} \delta_{j,i+\hat{\mu}} - U_{i,-\mu} \delta_{j,i-\hat{\mu}} \quad (2.18)$$



## 2.4 Improved Fermion Formulations

### 2.4.1 ASQTAD Fermions

Although the leading errors of the staggered formulation are  $O(a^2)$ , the breaking of flavour symmetry is a large effect [11]. Such symmetry breaking results in flavour changing strong interactions which are completely absent from QCD. The dominant flavour changing interaction is due to a one-gluon exchange between quarks, which is an  $O(a^2)$  lattice artifact. The primary process where this flavour changing takes place is when a low energy quark emits a gluon with momentum  $q \approx \pi/a$ . This artifact can be vastly reduced through the suppression of gluons with such momenta, and through a redefinition of the link variables that connect lattice sites this suppression can be achieved. The single link matrix is replaced by a weighted average over some set of link matrix products which connect the points, e.g. a three link path (staple). This can vastly improve the flavour symmetry at a given lattice spacing for suitably chosen weightings.

ASQ (A Squared) fermions correspond to a staggered type operator, where the parallel transport is defined by a sum over one, three, five and seven link products [11]. There is an additional five link term required to cancel errors of  $O(a^2 p^2)$  introduced by this fattening, the Lepage term. With this addition, the operator has no tree level  $O(a^2)$  corrections, hence the name. The Naik term [12] is also included, which is a three link derivative correction. The effect of this extra term is to improve the Lorentz symmetry, however, it has little effect on the flavour breaking [13].

When the ASQ operator is used in conjunction with tadpole improvement (a process whereby some of the  $O(g^n)$  discretisation errors are reduced [14]) the

ASQTAD operator is obtained [11], whose covariant derivative is written

$$D = c_{KS}D_{KS} + c_N D_N + c_3 D_3 + c_5 D_5 + c_7 D_7 + c_L D_L, \quad (2.19)$$

where the coefficients and Dirac operators correspond to the Kogut-Susskind, Naik, 3 staple, 5 staple, 7 staple and Lepage terms respectively. These coefficients are given by

$$c_{KS} = \frac{1}{8} + \frac{3}{8} + \frac{1}{8} \quad (2.20)$$

$$c_3 = \frac{1}{8} \frac{1}{2} u_0^{-2} \quad (2.21)$$

$$c_5 = \frac{1}{8} \frac{1}{8} u_0^{-4} \quad (2.22)$$

$$c_7 = \frac{1}{8} \frac{1}{48} u_0^{-6} \quad (2.23)$$

$$c_L = -\frac{1}{16} u_0^{-4} \quad (2.24)$$

$$c_N = -\frac{1}{24} u_0^{-2}, \quad (2.25)$$

where the factor  $u_0$  is the inverse fourth root of the average plaquette that arises from the tadpole improvement. The ASQTAD fermion action is exactly as written in equation (2.12), but with the derivative operator replaced by the matrix operator which results from equation (2.19). Hence, like the naïve staggered fermion formulation, the ASQTAD matrix is bounded from below by the mass parameter. Although the ASQTAD formulation has vastly improved flavour symmetry, it is still far from perfect, and retains the problem of naïve staggered fermions that the theory describes four fermion flavours not one.

The gauge action used when simulating ASQTAD fermions is generally the one loop improved Symanzik gauge action.

### 2.4.2 Ginsparg-Wilson Fermions

It was shown by Ginsparg and Wilson [15] that chiral symmetry on the lattice together with locality and a lack of doublers is possible, if instead a lattice definition of chiral symmetry is made. The Ginsparg-Wilson relation defines this symmetry, it is

$$\gamma_5 D + D \gamma_5 = a D R \gamma_5 D, \quad (2.26)$$

where  $a$  is the lattice spacing, and  $D$  is the lattice Dirac operator which corresponds to a solution of the Ginsparg-Wilson relation. The Ginsparg-Wilson relation approaches the continuum definition of chiral symmetry as this limit is taken. The relation implies the exact chiral symmetry of the action under the transformation

$$\delta\psi = \gamma_5(1 - aRD)\psi \quad (2.27)$$

$$\delta\bar{\psi} = \bar{\psi}\gamma_5. \quad (2.28)$$

Narayanan and Neuberger [16] derived the gauge covariant solution of the Ginsparg-Wilson relation, the so called “overlap operator”, where it is defined in terms of the hermitian Wilson operator  $H^W = \gamma_5 D^W$  with a negative mass parameter  $\rho$ ,

$$D_{i,\alpha;j,\beta} = \frac{1}{2a} \left( 1 + \gamma_5 \operatorname{sgn}(H_{i,\alpha;j,\beta}^W(\rho)) \right) \quad (2.29)$$

$$= \frac{1}{2a} \left( 1 + \gamma_5 \frac{H_{i,\alpha;j,\beta}^W(\rho)}{\sqrt{(H_{i,\alpha;j,\beta}^W)^2(\rho)}} \right). \quad (2.30)$$

All solutions of the Ginsparg-Wilson relation can be shown to be equivalent to Neuberger’s solution because of unitarity and hermiticity. The evaluation of the overlap operator applied to a source requires the application of the  $\operatorname{sgn}(H^W(\rho))$  operator, this is generally approximated using polynomial or rational approximations. It is the presence of the  $\operatorname{sgn}$  function (or more exactly, the non-commutivity of the  $\gamma_5$  operator and  $H^W$ ) that accounts for the extreme computational expense of overlap fermions.

### 2.4.3 Domain-Wall Fermions

The domain wall fermion prescription is a five dimensional lattice fermion formulation which has a degree of chiral symmetry [17]. A domain wall fermion is essentially a set of  $L_S$ -flavour Wilson fermions with a certain flavour-mixing mass matrix, where  $L_S$  is the extent of the fifth dimension. One of these Wilson fermions corresponds to a light fermion whose left and right handed modes exist on opposite ends of the fifth dimension. The degree of chiral symmetry this fermion possesses is controlled by the overlap between these modes. Since these modes exponentially decay into the fifth dimension, at finite  $L_S$  there is a finite overlap between the modes and the fermion theory will not be fully chiral. At infinite  $L_S$  there is no overlap between the modes and full lattice chiral symmetry is recovered.

The one flavour fermion action is defined by

$$S_f(U) = - \sum_{i,j,s,s'} \bar{\psi}_{i,s} M_{i,s;j,s'}^{\text{DW}} \psi_{(j,s')}, \quad (2.31)$$

where  $i, j$  run over the sites of the four dimensional volume, and  $s$  and  $s'$  run over the fifth dimension. The fermionic matrix is defined by

$$M_{i,s;j,s'}^{\text{DW}} = \delta_{s,s'} D_{i,j}^{\parallel} + \delta_{i,j} D_{s,s'}^{\perp}, \quad (2.32)$$

where

$$D_{i,j}^{\parallel} = \frac{1}{2} \sum_{\mu} \left( (1 + \gamma_{\mu}) U_{i,\mu} \delta_{i+\hat{\mu},j} + (1 - \gamma_{\mu}) U_{i,-\mu}^{\dagger} \delta_{i-\hat{\mu},j} \right) + (m_5 - 4) \delta_{i,j}, \quad (2.33)$$

$$D_{s,s'}^{\perp} = \begin{cases} P_R \delta_{2,s'} - m_f P_L \delta_{L_S,s'} - \delta_{1,s'}, & s = 1 \\ P_R \delta_{s+1,s'} + P_L \delta_{s-1,s'} - \delta_{s,s'}, & 1 < s < L_S \\ -m_f P_R \delta_{1,s'} + P_L \delta_{s-1,s'} - \delta_{L_S,s'}, & s = L_S, \end{cases} \quad (2.34)$$

with  $0 < m_5 < 2$  the so called domain wall height,  $m_f$  is the fermion mass parameter and  $P_{R,L} = \frac{1}{2}(1 \pm \gamma_5)$  are chiral projection operators.



The remaining  $L_S - 1$  fermions are massive and must be cancelled out to avoid bulk type infinities in the infinite  $L_S$  limit [17]. To perform this cancellation explicit bosonic fields are introduced, the so called Pauli-Villars fields. There is some flexibility in how the Pauli-Villars fields are defined, for this work the definition used is that given in [18]. The resultant 2+1 flavour QCD path integral originally written in equation (2.4) is now

$$\langle 0 | \mathcal{O} | 0 \rangle = \frac{1}{Z} \int \mathcal{D}U \frac{\det(M_{m_l}^{\text{DW}})^2 \det(M_{m_{\text{pv}}}^{\text{DW}})}{\det(M_{m_{\text{pv}}}^{\text{DW}})^3} \mathcal{O} e^{-S_g}, \quad (2.35)$$

where  $m_{\text{pv}} = 1$ .

To quantify the degree of chiral symmetry breaking, a measure of the overlap between the left and right handed fermion states is required. This is measured by the residual mass  $m_{\text{res}}$ , found through the Ward-Takahashi identity [17], and is given by [19]

$$m_{\text{res}} = \frac{1}{2} \frac{\sum_{\mathbf{x}, \mathbf{y}} \langle J_{5q}(\mathbf{y}, t) \pi(\mathbf{x}, 0) \rangle}{\sum_{\mathbf{x}, \mathbf{y}} \langle \pi(\mathbf{y}, t) \pi(\mathbf{x}, 0) \rangle} \Big|_{t \geq t_{\text{min}}}, \quad (2.36)$$

where  $\langle J_{5q} \pi \rangle$  is the correlator between the pseudoscalar density at the midpoint of the fifth dimension and that of the pion (i.e., pseudoscalar density at the boundaries of the fifth dimension), and  $\langle \pi \pi \rangle$  is the pion-pion correlator.

Like the Wilson case, the finite breaking of chiral symmetry leads to an additive quark mass renormalisation  $m_q = m_f + m_{\text{res}}$ , where  $m_q$  is the true fermion mass, and  $m_f$  is the mass parameter that appears in equation (2.33) [20]. However, the essential difference is now that the magnitude of renormalisation is under full control by the extent of the fifth dimension.

Since the Ginsparg-Wilson relation is known to be the unique solution to the fermion doubling problem which also respects chirality and locality, the domain wall prescription must be a variant of this. Indeed, the domain wall formulation corresponds to a solution of the Ginsparg-Wilson relation using an inverse tanh approximation to the sgn function which is only exact in the limit  $L_S \rightarrow \infty$

where the tanh approximation is exactly the sgn function [21].

When performing quenched analysis using valence domain wall fermions it has been found that the choice of gauge action has a large effect on the residual mass at constant  $L_S$  [19]. In particular, the DBW2 gauge action exhibits the least chiral symmetry breaking compared to using Wilson or Symanzik actions. This analysis has motivated the use of the DBW2 gauge action for dynamical fermion simulations using domain wall fermions.

## 2.5 Hadron Spectroscopy

In order to confirm that QCD is the correct theory of the strong force, observables must be measured that can be compared to their physical values. The hadron mass spectrum is one of the simplest tests that can be performed.

### 2.5.1 Correlation Functions

The masses of hadrons are extracted from Euclidean correlation functions (correlators), which are defined in terms of vacuum expectation values of quantum field operators [22]. Specifically, masses are determined from examining correlators known as two-point functions,

$$\hat{C}(\mathbf{x}, t) = \langle 0 | T \{ O(x) O^\dagger(0) \} | 0 \rangle. \quad (2.37)$$

The operator  $O^\dagger(0)$  creates some state at the origin, which is then annihilated at some other point by the sink operator  $O(x)$ . If one is interested in measuring some specific state, the operator  $O$  is chosen such that it has the same quantum numbers as the desired state. The source operator will create an state of the Hamiltonian that contains linear combinations of all states with these quantum



numbers. This operator's ground state should match the desired state and also have a minimum overlap with any excited states, so that at large Euclidean times the only state present is the desired state.

The functional dependence of the correlator on the mass of the state can be obtained by taking the discrete Fourier transform of the correlator,

$$C(\mathbf{p}, t) = \sum_{\mathbf{x}} \hat{C}(\mathbf{x}, t) e^{-i\mathbf{p} \cdot \mathbf{x}} \quad (2.38)$$

$$= \frac{1}{L^3} \sum_{\mathbf{x}, n, \mathbf{q}} \frac{1}{2E_n(\mathbf{q})} \langle 0 | O(\mathbf{x}) | n, \mathbf{q} \rangle \langle n, \mathbf{q} | O^\dagger(0) | 0 \rangle e^{-i\mathbf{p} \cdot \mathbf{x}} \quad (2.39)$$

$$= \sum_n \frac{1}{2E_n(\mathbf{p})} |\langle 0 | O(0) | n, \mathbf{p} \rangle|^2 e^{-E_n(\mathbf{p})t} \quad (2.40)$$

where in the second line a complete set of states has been inserted (with appropriate normalisation). A mass relation can now be obtained if zero three momentum is considered,  $\mathbf{p} = \mathbf{0}$ . Thus in the limit of large Euclidean time, equation (2.40) is dominated by the ground state, and takes the form

$$\lim_{t \rightarrow \infty} C(\mathbf{0}, t) = A_0 e^{-m_0 t}, \quad (2.41)$$

where  $A_0 = |\langle 0 | O(0) | 0, \mathbf{0} \rangle|^2 / 2m_0$ . When measuring observables, the lattice must be sufficiently large enough to accommodate the observable being studied. In particular the temporal extent of the lattice should be as large as possible to obtain a signal for the desired state, and not be contaminated by other higher excited states. As the temporal extent increases, eventually the ground state's signal will have decayed sufficiently such that its signal will not be measurable. The aim therefore is to locate the window where the ground state is not contaminated by higher excited states, and it is not contaminated by noise from the vacuum.

The above analysis assumes that the temporal length of the lattice  $T$  is infinite, which is obviously not the case in real simulations. At the boundaries of the box the fermion fields have periodic boundary conditions in the spatial extent, and

anti-periodic boundary conditions in the temporal extent [1]. As a result, any state created at a source can also propagate to the sink backwards in time. This backwards propagating state will affect the functional form that the correlator will take, and the general form is now

$$C(\mathbf{0}, t) = \sum_n \left( A_n e^{-m_n t} + \eta B_n e^{-m_n^*(T-t)} \right), \quad (2.42)$$

where the second term is due to backward propagating states with mass  $m_n^*$ . The parameter  $\eta = \pm 1$  depending on how the operator  $O$  transforms under the time reversal operator  $\hat{T} = \gamma_4 \gamma_5$ . For the special case of mesons, the forwards and backwards propagating states have the same mass, and in the limit of infinite statistics the correlator is symmetric about the mid-point in the temporal extent. Assuming that enough statistics have been generated, the meson correlator is folded about the midpoint on top of itself. This increases the information available for mass extraction. The meson ground state correlator is thus given by

$$C_M(\mathbf{0}, t) = A_0 (e^{-m_0 t} + e^{-m_0(T-t)}) \quad (2.43)$$

$$= 2A_0 e^{-\frac{m_0 T}{2}} \cosh \left[ m_0 \left( \frac{T}{2} - t \right) \right]. \quad (2.44)$$

Suitable operators for mesons using Wilson-type fermions (Wilson, Domain Wall, Overlap) are given by

$$O_m(\mathbf{x}, t) = \sum_{\mathbf{x}} \Gamma_{AB}^{(m)} \bar{\psi}_A(\mathbf{x}, t) \psi_B(\mathbf{x}, t). \quad (2.45)$$

The coefficient  $\Gamma_{AB}^m$  is chosen to ensure colour neutrality, and carry the quantum numbers match that of the observable of interest.

For example the desired operator for the pion when using Wilson fermions is given by

$$O_{\pi^+} = \sum_{\mathbf{x}} \bar{d}^a \gamma_5 u^a. \quad (2.46)$$

When a degenerate light pair for the up and down quarks is assumed,  $\pi^+ = \pi^- = \pi^0$ , the meson propagator is given by

$$C_\pi = \sum_{\mathbf{x}, \mathbf{y}} \langle \text{tr } \gamma_5 K_{(\mathbf{x},0);(\mathbf{y},t)}^{-1} \gamma_5 K_{(\mathbf{y},t);(\mathbf{x},0)}^{-1} \rangle, \quad (2.47)$$

where  $K^{-1}$  is the quark propagator, and the trace is over colour and spin. To form the quark propagator, twelve columns of the inverse of the Dirac matrix must be calculated. The explicit matrix inverse is not formed, but rather each required element is calculated by using a point source right hand side and solving the resulting systems of equations (see §3.8 for more details).

### 2.5.2 Extracting the Mass

With the correlation function calculated for the desired state, all now that is required is to extract the mass. To minimise the systematic errors in the correlation function, it is important to ensure that the correlation function only contains the desired state, and not other excited states. When examining correlation functions, it is useful to define the effective mass, for the pion this is given by

$$m_{\text{eff}}^{\text{meson}}(t) = \cosh^{-1} \left[ \frac{C(t-1) + C(t+1)}{2C(t)} \right]. \quad (2.48)$$

As the time increases the effective mass will level off, this plateau region represents the time interval which corresponds to pure ground state information. The plateau region is then used as the fit range from which the ground state mass can be extracted.

The method by which the mass is extracted is through minimisation of the  $\chi^2$  function [23], which is given by

$$\chi^2 = \sum_{t_i, t_j} [f(\mathbf{a}, t_i) - \bar{C}(t_i)] \bar{C}_{\text{OV}}^{-1}(t_i, t_j) [f(\mathbf{a}, t_j) - \bar{C}(t_j)], \quad (2.49)$$

where the covariance matrix is defined by

$$C_{\text{Ov}}(t_i, t_j) = \frac{1}{N(N-1)} \sum_{k=1}^N [C_k(t_i) - \bar{C}(t_i)][C_k(t_j) - \bar{C}(t_j)], \quad (2.50)$$

with  $N$  the number of configurations in the ensemble,  $C_k(t)$  is the  $k^{\text{th}}$  sample from the ensemble,  $\bar{C}(t)$  is the mean of the correlator over the ensemble and  $f(\mathbf{a}, t)$  is the function to which the fit is being performed with parameters  $\mathbf{a}$ . The covariance matrix takes into account any correlations within the correlator, if the data is uncorrelated, then the covariance matrix is set to the identity. When performing correlated fits, a good fit is that which leads to  $\chi^2$  per degree of freedom of around unity, this signifies that the variance of the data has been correctly taken into account. Further details on  $\chi^2$  minimisation can be found in [23].

### 2.5.3 Chiral Extrapolations

Currently all quantities measured on the lattice are extrapolated to the chiral limit with respect to the light quark pair. Such an extrapolation is necessary since current calculations are performed at much heavier quark masses than the physical value, and to a first approximation the chiral limit is the physical value of the light quark masses.

The quark mass dependence with the pion mass is found from lowest order chiral perturbation theory, and is given by [1]

$$m_\pi^2 = A(m_l + m_{\text{res}}), \quad (2.51)$$

and equivalently for the rho meson

$$m_\rho = B + C(m_l + m_{\text{res}}), \quad (2.52)$$

where  $A$ ,  $B$  and  $C$  are coefficients to be determined through  $\chi^2$  minimisation. When performing a chiral extrapolation using a fermion formulation which has



additive quark mass renormalisation, then this must be taken into account when performing the chiral extrapolation, e.g., when using Wilson fermions  $\kappa_c$  must be found, which is the point on the  $\kappa$  axis which corresponds to zero pion mass.

For the case of domain wall fermions the chiral limit is set at  $m_q = -m_{\text{res}}$ , where  $m_{\text{res}}$  is found from equation (2.36). For consistency, the residual mass at the chiral limit must be found. This is slightly less trivial than extrapolating, say, either the pion or the rho, because  $m_{\text{res}}$  appears in both the  $x$  and  $y$  axis of the plot, and so is a self-consistency requirement. A linear extrapolation of the quark mass is used, hence

$$m_{\text{res}}^{\text{chiral}} = \frac{B}{1 + C}, \quad (2.53)$$

where  $B$  is the intercept of the linear extrapolation with the  $y$ -axis and  $C$  is the gradient.

In the case of dynamical simulations, for a unitary theory to be maintained, all spectroscopy must be calculated using  $m_{\text{val}} = m_{\text{dyn}}$ . This requires that multiple gauge field ensembles be generated each with different quark masses. Partial-quenching is where different valence masses are used for spectroscopy calculations, but a single gauge field ensemble is used. This breaks unitarity, but can be useful to obtain first order behaviour.

### 2.5.4 Dimensionful Quantities

The variables on the lattice are dimensionless, and so any observables measured on the lattice are also dimensionless. To convert the dimensionless quantities to dimensionful the lattice spacing  $a$  must be put back in, however, this of course needs to be measured first. The general technique for doing so is to compare the ratio of a dimensionless quantity to a known physical value. The ratio between

these quantities will be the lattice spacing, e.g.,

$$a = \frac{m_{\rho}^{\text{lat}}}{m_{\rho}^{\text{cont}}}. \quad (2.54)$$

In principle any observable can be chosen to set the lattice spacing, and if the correct theory is being simulated, with a sufficient volume to avoid finite size effects, with a weak enough bare coupling strength to keep the discretisation errors under control and the correct quark masses are used, then such lattice spacings should all be equivalent. Usually this is two out of four at best, hence lattice spacings calculated using difference observables are not all equivalent. As a result, different dimensionful quantities for observables are obtained depending on what observable was used to set the lattice spacing. There is not much that can be done to improve this situation, other than to more fully include the effects of QCD in simulations. For the calculations performed in chapter 8 the mass of the rho meson was used to set the lattice spacing.

## 2.6 Topological Charge

In the continuum theory, the Atiyah-Singer theorem holds,

$$Q_{\text{top}} = \text{index}(D), \quad (2.55)$$

where

$$Q_{\text{top}} = \frac{1}{32\pi^2} \int d^4x \epsilon_{\mu\nu\rho\sigma} \text{tr}(F_{\mu\nu} F^{\rho\sigma}) \quad (2.56)$$

is the topological charge of the gauge field and  $\text{index}(D)$  is the difference between the total number of eigenmodes with negative and positive chiralities [24]. Hence this latter quantity is an integer, and therefore so must  $Q_{\text{top}}$ .

On the lattice this equality does not hold exactly, and there is an ambiguity as to how the topological charge is defined. In this work the topological charge was



measured through first smearing the lattice using HYP smearing [25] and then using an  $O(a^2)$  lattice definition of equation (2.56) to construct the quantity [26]. Since this definition has inherent lattice artifacts present it is not expected to result in an integer measurement. However, it is useful when studying an algorithm's performance, because the autocorrelation of the measured topological charge reveals how quickly the algorithm is tunneling between different topological sectors, i.e., how quickly are the long distance physics of the ensembles evolving with simulation time.

## 2.7 Autocorrelation Length

When using a Monte Carlo evolution to produce ensembles of gauge configurations there will be a degree of correlation between ensembles which are close with respect to Monte Carlo time. When calculating observables based on such ensembles, it is crucial that uncorrelated measurements are used, else the errors of these observables shall be incorrectly estimated.

The autocorrelation function  $\rho(t)$  is a measure of how correlated two measurements of the observable  $\mathcal{O}$  are, when separated in Monte Carlo time by a distance  $t$ . It is given by

$$\rho(t) = \frac{1}{N} \sum_{t'=1}^{N-t} (\mathcal{O}(t') - \bar{\mathcal{O}})(\mathcal{O}(t' + t) - \bar{\mathcal{O}}), \quad (2.57)$$

where  $N$  is the number of samples in the data set, and  $\bar{\mathcal{O}}$  is the mean of the observable.

The integrated autocorrelation length,

$$\tau_{\text{int}} = \frac{1}{2} + \frac{1}{\rho(0)} \sum_{t=1}^{t_{\text{max}}} \rho(t), \quad (2.58)$$

where  $\rho(t)$  is the measured autocorrelation at a distance in  $t$  in Monte Carlo time, is the measure which is used to ensure that such data is decorrelated. Plots

of  $\tau_{\text{int}}$  against  $t$  should reveal a plateau, the onset of this plateau corresponds to the Monte Carlo time required to ensure uncorrelated measurements. The autocorrelation length varies for different observables, since certain features of the gauge field evolve faster than others using, e.g., the short distance physics tends to evolve faster than long distance when using Hybrid algorithms.

# Chapter 3

## Algorithms

Since the dominant cost in performing lattice QCD calculations lies in the generation of the gauge field configurations, the choice of algorithm is very important. In this section the various popular algorithms presently used for lattice QCD are discussed.

### 3.1 Generation of gauge configurations

The path integral given in equation (2.4) has a finite number of degrees of freedom since space-time has been discretised, but its direct evaluation is computationally prohibitive, thus a stochastic method must be used. Observe that an integral can be rewritten as

$$\int \mathcal{O}(x) dx = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \mathcal{O}_i + O\left(\sqrt{\frac{C_2}{N}}\right), \quad (3.1)$$

where  $\mathcal{O}_i$  is a randomly chosen sample from the distribution of  $\mathcal{O}(x)$  and  $C_2$  is the variance of  $\mathcal{O}$ . Hence, the evaluation of the path integral becomes a problem of obtaining enough statistics.

The probability distribution for gauge fields  $U$ , where a single quadratic fermion contribution has been integrated out is given by

$$P(U) = \frac{1}{Z} e^{-S_g} \det M(U). \quad (3.2)$$

Thus to perform the path integral calculation stochastically, a large ensemble of gauge field configurations, each of which randomly chosen from this distribution, must be used to calculate desired observables.

The extreme cost in generating gauge configurations lies in the inclusion of the fermion determinant, to evaluate this directly while technically possible, would be computationally prohibitive due to the dimensions of the matrix. In the early days of lattice studies the determinant was simply set equal to unity, a process known as quenching. This corresponds to making the dynamical fermion mass infinite, the physical effect of this is suppress the fermion loops that appear in the vacuum, and leads to a non-unitary theory. There is however, some experimental justification to this procedure, e.g., the OZI rule [27, 28, 29]. When performing quenched calculations the algorithm of choice is generally thought to be the overrelaxation algorithm, which is a local update algorithm [30].

With the advent of more powerful computers, the stochastic inclusion of dynamical quark effects has increasingly become a realistic possibility. This stochastic inclusion could in principle be done using the Grassman valued lattice fermion actions described in § 2.2, but doing so would lead to poor importance sampling since the action would not be positive definite [31]. However, a simple manipulation allows the fermionic determinant to be weighted stochastically using bosonic fields [32]. For a bosonic action, with positive operator  $D$ , the field can be integrated out to leave a determinant factor

$$\int \mathcal{D}\phi^\dagger \mathcal{D}\phi \exp(-\phi^\dagger D \phi) = \frac{1}{\det D}. \quad (3.3)$$



This suggests that the fermionic determinant can be represented by

$$\det M(U) = \frac{1}{\det M^{-1}} \quad (3.4)$$

$$= \int \mathcal{D}\phi^\dagger \mathcal{D}\phi \exp(-\phi^\dagger M^{-1} \phi), \quad (3.5)$$

where  $\phi, \phi^\dagger$  are bosonic-valued fields. This representation is done at the expense of including the inverse of the Dirac matrix in the action. As the fermion mass is decreased, the condition number of the matrix increases, the resulting increase in computational cost of inverting the matrix is what makes lattice QCD extremely computationally challenging. It is actually the matrix kernel  $\mathcal{M} = M^\dagger M$  which is used in such a bosonic fermion representation because it allows the use of heatbath refreshment of the pseudofermion field. It is this requirement that  $\mathcal{M}$  is used, and not  $M$ , that causes the failing of HMC with one fermion flavour (Mdescribes two degenerate fermions).

With both the fermions and gauge field represented as bosonic valued functions in the action, Monte Carlo techniques can now be used to build an ensemble of gauge configurations with the correct distribution and upon which measure observables as if it were any other statistical mechanical system.

## 3.2 Hybrid Monte Carlo

Since its inception, the algorithm of choice for generating dynamical gauge configurations has been the Hybrid Monte Carlo (HMC) algorithm [33]. This algorithm combines many of the techniques used prior to its introduction, momentum and fermion heatbath refreshment, and molecular dynamics (MD) evolution of the gauge fields [34]. The unique difference about HMC was the introduction of a global Metropolis acceptance test at the end of each trajectory which stochastically corrects the systematic error introduced through the finite integrating stepsize used in the MD.



### 3.2.1 Molecular Dynamics

The MD evolution of the gauge field must be done through the introduction of a canonical momentum  $\pi_{j,\mu}$  field which allows the definition of a Hamiltonian. The  $\pi_{j,\mu}$  field is a traceless hermitian matrix and resides in the Lie algebra of  $SU(3)$ , i.e. it is a member of  $su(3)$ . The Hamiltonian is hence defined as

$$H = \frac{1}{2}\pi^2 + S_f + S_g, \quad (3.6)$$

where  $\pi^2 = \sum_{j,\mu} \text{tr} \pi_{j,\mu}^2$  and the pure gauge and fermionic actions take the role of the potential.

To perform the MD integration of the gauge field and its conjugate momentum, expressions are required for  $\dot{U}$  and  $\dot{\pi}$  the derivatives of these variables with respect to the MD time  $\tau$ . Both of these results are derived in [35], the former of these is simply given by  $\dot{U}_{j,\mu} = i\pi_{j,\mu}U_{j,\mu}$ . Since the result of the latter is important for chapter 7, the hand-waving argument described in [36] is reproduced here.

To find an expression for  $\dot{\pi}$ , the constraint of conservation of the Hamiltonian is used,  $\dot{H} \equiv 0$ . In the following derivation, the Wilson gauge action shall be used and a generic pseudofermion action  $S_f = \bar{\phi}(M^\dagger M)^{-1}\phi$  is assumed.

$$\dot{H} = \sum_{j,\mu} \text{tr} \pi_{j,\mu} \dot{\pi} + \dot{S}_g + \dot{S}_f \quad (3.7)$$

$$= \sum_{j,\mu} \text{tr} \pi_{j,\mu} \dot{\pi} + \frac{\beta}{6} \sum_{j,\mu} \text{tr} (\dot{U}_{j,\mu} V_{j,\mu} + V_{j,\mu}^\dagger \dot{U}_{j,\mu}^\dagger) - \bar{\phi}(M^\dagger M)^{-1} \frac{d}{d\tau} (M^\dagger M) (M^\dagger M)^{-1} \phi \quad (3.8)$$

$$= \sum_{j,\mu} \text{tr} \pi_{j,\mu} \dot{\pi} + \frac{\beta}{6} \sum_{j,\mu} \text{tr} (i\pi_{j,\mu} U_{j,\mu} V_{j,\mu} + \text{H.c.}) - X^\dagger \left[ \frac{dM^\dagger}{d\tau} M + M^\dagger \frac{dM}{d\tau} \right] X \quad (3.9)$$

$$= \sum_{j,\mu} \text{tr} \pi_{j,\mu} \dot{\pi} + \frac{\beta}{6} \sum_{j,\mu} \text{tr} (i\pi_{j,\mu} U_{j,\mu} V_{j,\mu} + \text{H.c.}) \quad (3.10)$$

$$- \sum_{j,\mu} \text{tr} \left[ i\pi_{j,\mu} U_{j,\mu} X X^\dagger M^\dagger \frac{dM}{dU_{j,\mu}} + i\pi_{j,\mu} U_{j,\mu} M X X^\dagger \frac{dM^\dagger}{dU_{j,\mu}} + \text{H.c.} \right] \quad (3.11)$$

where in the second line  $V_{j,\mu}$  is the remaining staple of the plaquette, in the

third line the vector  $X = (M^\dagger M)^{-1} \phi$  has been defined and in the last line the identity  $A^\dagger B = \text{tr } B \otimes A^\dagger$ , with  $A$  and  $B$  column matrices, has been used. In this form, the common  $\pi_{j,\mu}$  term can be factored out, and the resulting equation can be written

$$\dot{H} = \sum_{j,\mu} \text{tr} \left\{ \pi_{j,\mu} [\dot{\pi}_{j,\mu} + i(F_{j,\mu} - F_{j,\mu}^\dagger)] \right\}, \quad (3.12)$$

where  $F_{j,\mu}$  is given by

$$F_{j,\mu} = \frac{\beta}{3} U_{j,\mu} V_{j,\mu} - U_{j,\mu} X X^\dagger M^\dagger \frac{dM}{dU_{j,\mu}} + U_{j,\mu} M X X^\dagger \frac{dM^\dagger}{dU_{j,\mu}}. \quad (3.13)$$

Since  $\pi_{j,\mu}$  must remain traceless, equation (3.12) will be satisfied if the quantity in the square brackets is a multiple of the identity matrix, i.e.,

$$\dot{\pi}_{j,\mu} + i(F_{j,\mu} - F_{j,\mu}^\dagger) = cI, \quad (3.14)$$

where the constant  $c$  is determined from the requirement that  $\dot{\pi}_{j,\mu} = 0$  for  $\pi_{j,\mu}$  to always be traceless, hence  $c = \text{tr } \frac{i}{3}(F_{j,\mu} - F_{j,\mu}^\dagger)$ . Thus the derivative of the canonical momentum with respect to the MD time is written

$$i\dot{\pi}_{j,\mu} = [F_{j,\mu}]_{\text{TA}}, \quad (3.15)$$

$$= \left[ U_{j,\mu} \frac{dS}{dU_{j,\mu}} \right]_{\text{TA}}, \quad (3.16)$$

where the subscript indicates the traceless anti-hermitian part of the matrix, and in analogy with classical mechanics  $F_{j,\mu}$  is described as the force acting on the system.

The evolution of the gauge field hence requires that the inverse of the fermion matrix applied to a vector be calculated, i.e., the evaluation of the vector  $X$ , and that the derivative of the fermion matrix with respect to the gauge field is calculated. The latter of these calculations is trivial for all of the fermion formulations discussed in § 2.2 with the notable exception of ASQTAD fermions, where the derivative must be calculated explicitly for every link variable which

appears in the operator, an operation which incurs a considerable computational expense (see § 7.1.2).

With the equations of motion of the gauge field, and its associated momentum determined, the gauge field can be evolved using a numerical integration scheme.

### 3.2.2 The $\Phi$ Algorithm

The  $\Phi$  algorithm [37, 38, 36] is a Hybrid Molecular Dynamics (HMD) algorithm for an integer multiple of the number of flavours described by the fermion kernel  $M^\dagger M$ . It iterates a composite Markov step, which is ergodic and has a fixed point distribution close to the desired one.

The gauge fields  $U$  are evolved for a time  $\tau$  by integrating Hamilton's equations, using a Molecular Dynamics (MD) integration scheme. Each trajectory of the  $\Phi$  algorithm consists of:

- A momentum refreshment heatbath using Gaussian noise ( $P(\pi) \propto e^{-\pi^2/2}$ ).
- A pseudofermion heatbath using Gaussian noise  $\phi \propto M^\dagger(U)\eta$ , where  $P(\eta) \propto e^{-\eta^2/2}$ .
- An MD trajectory consisting of  $\tau/\delta\tau$  steps.

The momentum and fermion heatbaths ensure that their respective variables are Gaussian distributed according to respective actions.

The numerical integration of the MD trajectory requires the introduction of a finite integrating stepsize  $\delta\tau$  and an integration scheme (see § 3.6). Typically a leapfrog integrator is used, that is one where the canonical coordinates and momenta are updated in steps of size  $\delta\tau$  but are kept  $\delta\tau/2$  out of sync through

out the trajectory. For lattice QCD it has been shown that the  $\hat{U}_{\text{QPQ}}$  integrator leads to a marginally better acceptance rate than the  $\hat{U}_{\text{PQP}}$  integrator [39], the difference being between whether the coordinate or the momentum is updated first. Numerical integration schemes do not conserve energy, the leapfrog integrator having  $\delta H = O(\delta\tau^2)$  for any trajectory length, even for trajectories where  $\tau \gg 1$ . Although using a naïve scheme such as the leapfrog integrator can work perfectly adequately, it is not always the optimum solution [40].

For  $\delta\tau > 0$ , the fixed point distribution of the MD step and that of the momentum refreshment heatbath do not coincide. To find the actual equilibrium distribution the fixed point of the full Markov step is required. Let  $\hat{U}(\tau)$  represent the evolution operator for the MD step  $\hat{U}(\tau) : (U, \pi) \mapsto (U'', \pi'')$  and  $e^{-(S+\Delta S)}$  denote the fixed point distribution of the full composite Markov step, where  $\Delta S$  measures the deviation from the desired distribution. This must satisfy the fixed point condition that

$$\begin{aligned}
 e^{-[S(U')+\Delta S(U')]} &= \int dU d\pi e^{-H(U,\pi)-\Delta S(U)} \delta(U' - U'') \\
 &= \int dU'' d\pi'' e^{-(H+\Delta S) \circ \hat{U}^{-1}} \delta(U' - U'') \\
 &= \int dU'' d\pi'' e^{-(H+\Delta S)} e^{-\delta(H+\Delta S)} \delta(U' - U'') \\
 &= e^{-[S(U')+\Delta S(U')]} \int d\pi'' e^{-\frac{1}{2}\pi''^2} e^{-\delta(H+\Delta S)}, \quad (3.17)
 \end{aligned}$$

with  $\delta : \Omega \mapsto \Omega \circ [\hat{U}(\tau) - 1]$  measuring the lack of energy conservation, and assuming reversibility  $\hat{U}^{-1} = F \circ \hat{U} \circ F$  where  $F : (U, \pi) \mapsto (U, -\pi)$ . The resulting condition is

$$\langle e^{-\delta(H+\Delta S)} \rangle_\pi = 1. \quad (3.18)$$

Performing an asymptotic expansion on this condition in powers of  $\delta\tau$ , knowing that  $\delta H = O(\delta\tau^2)$  for any trajectory length  $\tau$ , it is deduced that  $\delta\Delta S \sim O(\delta\tau^2)$ . Thus  $\Delta S \sim O(\delta\tau^2)$ , hence the  $\Phi$  algorithm is accurate to this order when used in combination with a leapfrog integrator. More generally the  $\Phi$  algorithm has



a finite step size error of  $O(\delta\tau^n)$ , where  $n$  is given by the error in the integrator used.

### 3.2.3 The $\chi$ Algorithm

This is very similar to the  $\Phi$  algorithm, except the pseudofermion heatbath is performed before every single MD step as opposed to only before each MD trajectory.

This leads to the expectation that, as for the  $\Phi$  algorithm, the leading error term is  $O(\delta\tau^2)$ . The proof of this follows from that of the  $R$  algorithm given in § 3.4. The fermionic force is given by

$$\frac{dS_f}{dU_{j,\mu}} = - \left\langle \phi^* \mathcal{M}^{-1} \frac{d\mathcal{M}}{dU_{j,\mu}} \mathcal{M}^{-1} \phi \right\rangle_\phi, \quad (3.19)$$

where the average over the pseudofermion  $\phi$  is introduced because the heatbath is performed every MD step.

### 3.2.4 Hybrid Monte Carlo

The  $\Phi$  algorithm illustrated in §3.2.2 has one major drawback, and that is the finite stepsize errors present due to the leapfrog integrator used in the MD evolution resulting in the fixed point of algorithm not matching the desired fixed point. The HMC algorithm stochastically corrects for these errors through the inclusion of a global Metropolis accept/reject step at the end of each trajectory,

$$P_{\text{acc}} = \min(1, e^{-\delta H}), \quad (3.20)$$

where  $\delta H = H_f - H_i$ .

For the Metropolis accept/reject step to be valid the MD integrator must be



reversible

$$\hat{U}^{-1} = F \circ \hat{U} \circ F \quad (3.21)$$

where  $F$  corresponds to a flip of the momenta, and be area preserving,

$$\det \hat{U}_* = \det \left[ \frac{d(U', \pi')}{d(U, \pi)} \right] = 1. \quad (3.22)$$

The simplest integrator which satisfies this requirement is the leapfrog integrator.

A single Hybrid Monte Carlo (HMC) update thus consists of

- Momentum refreshment heatbath using Gausssian noise ( $P(\pi) \propto e^{-\pi^2/2}$ ).
- Fermion refreshment ( $\phi = (\not{D} + m)^\dagger \eta$ , where  $P(\eta) \propto e^{-\eta^2/2}$ ).
- MD trajectory consisting of  $\tau/\delta\tau$  steps.
- Metropolis accept/reject

Although HMC has been tremendously successful, it does have a number of shortcomings in its basic form.

The Hamiltonian is extensive in the volume, i.e., as the volume grows, so does the Hamiltonian. As a result  $\delta H$  is extensive in the volume, and to maintain a constant acceptance rate the integrating stepsize must be reduced as the volume increases.

The cost of the algorithm blows up considerably as the quark mass parameter is reduced, this process is known as “critical slowing down”. This reasons for this are two fold. Firstly, the condition number of the fermion matrix increases as  $1/m_t^2$  which results in an increase in cost to invert the matrix. Secondly, the magnitude of the force originating from the fermionic piece of the action

increases, necessitating a decrease in the stepsize to maintain a constant acceptance rate. The latter of these problems shall be addressed in chapter 6.

HMC also fails when it comes to including a number of fermions that is a non-integer multiple of the fermion multiplet of the formulation. Since the operator  $M^\dagger M$  must be included in the action, instead of just  $M$ , this corresponds to a two flavour theory for Wilson fermions. Historically this has not been a problem since the up and down quarks are treated as being degenerate, and generally any calculations have been performed using mass parameters much heavier than the strange quark, meaning there has been little point in performing 2+1 quark flavour simulations. For staggered fermions, the kernel  $M^\dagger M$  only connects sites of the same parity, this means that through only defining the pseudofermion fields on even (or odd) sites, an extra doubling of the number of flavours can be avoided. The resolution to this problem has traditionally been to use either the inexact R algorithm [36], or more recently a Polynomial Monte Carlo algorithm [41, 42, 43].

### 3.3 The $R_0$ algorithm

The  $R_0$  algorithm [36] does not utilise the pseudofermion approach, however, the resulting fermionic force term is very similar to that obtained in § 3.2.1.

First note that the fermionic determinant can be rewritten

$$\det \mathcal{M}^\alpha = \exp(\alpha \operatorname{tr} \ln \mathcal{M}) \quad (3.23)$$

$$= \exp(-S_f), \quad (3.24)$$

where  $\alpha$  parameterises the number of fermions of the theory. To obtain a theory describing less flavours than that of the fermion multiplet, it is tempting to do so through use of  $\alpha \notin \mathbf{Z}$ . For Wilson type fermions where  $\mathcal{M}$  describes two

flavours of fermions it is a valid approach to set  $\alpha = \frac{1}{2}$ , since  $\mathcal{M}$  is the square of a local operator. For staggered fermions this is not the case, nevertheless through setting  $\alpha = \frac{1}{2}, \frac{1}{4}$  the result is potentially a 2+1 quark flavour theory [44, 5, 6].

The resulting force from this fermionic action is given by

$$\frac{dS_f}{dU_{j,\mu}} = -\alpha \text{tr} \frac{d \ln \mathcal{M}}{dU_{j,\mu}} \quad (3.25)$$

$$= \alpha \text{tr} \left[ \mathcal{M}^{-1} \frac{d\mathcal{M}}{dU_{j,\mu}} \right], \quad (3.26)$$

however, such a fermionic force would require the explicit inverse of the matrix to calculate the trace exactly. The fermionic force is replaced by a noisy estimator for the trace, resulting in a force

$$\frac{dS_f}{dU_{j,\mu}} = -\alpha \text{tr} \left[ (M^\dagger)^{-1} \frac{d\mathcal{M}}{dU_{j,\mu}} M^{-1} \right] \quad (3.27)$$

$$= -\alpha \left\langle \eta^* M^{\dagger-1} \frac{d\mathcal{M}}{dU_{j,\mu}} M^{-1} \eta \right\rangle_\eta \quad (3.28)$$

$$= -\alpha \langle \Sigma' \rangle_\eta, \quad (3.29)$$

where  $\eta$  is a complex noise vector sampled from a Gaussian heatbath with variance 1. (By using this noisy estimator, the conserved Hamiltonian  $H'$  is now written  $H' = H + \Sigma - S + \text{terms specific to integration scheme.}$ ) Defining an auxilliary field  $\chi \equiv \mathcal{M}^\dagger \eta$ , the fermionic force becomes

$$\frac{dS_f}{dU_{j,\mu}} = -\alpha \left\langle \chi^* \mathcal{M}^{-1} \frac{d\mathcal{M}}{dU_{j,\mu}} \mathcal{M}^{-1} \chi \right\rangle_{\chi=M^\dagger \eta}. \quad (3.30)$$

Using such a fermionic force in place of the one used by the  $\Phi$  algorithm, the resulting algorithm is the inexact  $R_0$  algorithm which describes an arbitrary number of fermions per field.

The leading error of the  $R_0$  algorithm can be found by again looking at the equilibrium distribution. To calculate the error in  $\delta H$  the effect of using a noisy

estimator for the fermionic force must be included. For a single noisy MD step using a  $\hat{U}_{\text{QPQ}}$  integrator we have

$$\langle e^{-\delta H} \rangle_\eta = \langle 1 - \delta H + \delta H^2 + O(\delta H^3) \rangle \quad (3.31)$$

$$= 1 - \frac{1}{2} \langle (\Sigma' - S')^2 \rangle_\eta (1 - \pi^2) \delta \tau^2 + O(\delta \tau^3), \quad (3.32)$$

where in moving from the first to the second line, the relation  $\delta H = \delta \Delta H$  has been used. The coefficient of  $(1 - \pi^2) \delta \tau^2$  is proportional to the variance of the estimated force and will only vanish if the force is computed exactly. Since the momentum average is not Gaussian after many leapfrog steps, the leading order term is not cancelled as it would be if only one MD step per trajectory were used. The analogue of equation (3.18) now becomes

$$\langle e^{-\delta(H+\Delta S)} \rangle_{\pi, \eta_1 \dots \eta_N} = \left\langle \langle e^{-\delta H_1} \rangle_{\eta_1} \dots \langle e^{-\delta H_N} \rangle_{\eta_N} e^{-\delta \Delta S} \right\rangle_\pi = 1. \quad (3.33)$$

where  $\delta H = \delta H_1 + \dots + \delta H_N$ . Thus the leading error of the  $R_0$  algorithm is  $\Delta S \sim O(\delta \tau)$ .

It is in principle possible to make the  $R_0$  algorithm exact by including an acceptance test after each MD step [43]. The acceptance test must be done after each MD step, because a new Gaussian noise vector is used at each, rendering the integration scheme irreversible after more than one step. This will greatly increase the autocorrelation of the algorithm. In addition, the action given in equation (3.23) cannot be directly computed, and must instead be estimated using stochastic summation.

### 3.4 The R algorithm

As can be seen from examination of Equations 3.19 and 3.30, the  $\chi$  and  $R_0$  algorithms have very similar fermionic force terms, despite a different derivation.



The difference is that in the former the pseudofermion field is calculated at the beginning of every MD step, and in the latter the auxilliary field is calculated in the middle of each MD step. The  $\chi$  algorithm has  $O(\delta\tau^2)$  errors for  $\alpha = 1$  multiplets, whereas  $R_0$  has errors  $O(\delta\tau)$ . However for  $\alpha = 0$  multiplets (i.e., no fermions) both algorithms are the same and have errors of  $O(\delta\tau^2)$ . If it is assumed that the leading error has a linear dependence on the time the (effective) pseudofermions are evaluated and on the number of multiplets, then by drawing a straight line in MD time between the  $\alpha = 0$   $R_0$  algorithm and  $\alpha = 1$   $\chi$  algorithm points, i.e.,

$$\tau = (1 - \alpha) \frac{\delta\tau}{2}, \quad (3.34)$$

and evaluating the pseudofermions at this point during the trajectory, an  $O(\delta\tau^2)$  algorithm valid for at least  $0 \leq \alpha \leq 1$  fermion multiplets should be obtained. Indeed, this algorithm is known as the  $R$  algorithm [36]. For example, for a theory with two flavours of staggered fermions, this means evaluating the pseudofermion field a quarter way through each MD update. Note that this leads to an algorithm that is neither reversible nor area preserving.

To prove that the  $R$  algorithm does indeed have an  $O(\delta\tau^2)$  leading order error term, again the resulting equilibrium distribution of the algorithm must be found.

$$\begin{aligned} e^{-[S(U') + \Delta S(U')]} &= \int dU d\pi e^{-H(U, \pi) - \Delta S(U)} \delta(U' - U'') \\ &= \int dU'' d\pi'' \det(\hat{U}_*)^{-1} e^{-(H + \Delta S) \circ \hat{U}^{-1}} \delta(U' - U'') \\ &= \int dU'' d\pi'' e^{-(H + \Delta S)} e^{-(\delta + \bar{\delta})(H + \Delta S) - \text{tr} \ln \hat{U}_*} \delta(U' - U'') \\ &= e^{-[S(U') + \Delta S(U')]} \int d\pi'' e^{-\frac{1}{2}\pi''^2} e^{-(\delta + \bar{\delta})(H + \Delta S) - \text{tr} \ln \hat{U}_*}, \quad (3.35) \end{aligned}$$

thus the condition is now

$$\langle e^{-(\delta + \bar{\delta})(H + \Delta S) - \text{tr} \ln \hat{U}_*} \rangle_\pi = 1, \quad (3.36)$$

where  $\delta$  measures lack of energy conservation  $\delta : \Omega \mapsto \Omega \circ F \circ [\hat{U}(\tau) - 1] \circ F$ ,  $\bar{\delta}$  measures lack of reversibility  $\bar{\delta} : \Omega \mapsto \Omega \circ [\hat{U}(\tau)^{-1} - F \circ \hat{U} \circ F]$  and  $\text{tr} \ln \hat{U}_* \equiv \ln \det \frac{\partial(U'', \pi'')}{\partial(U, \pi)}$  measures lack of area preservation. Considering a single step of the  $R$  algorithm, where the auxilliary field  $\chi$  is computed at a time  $\tau = (1 - \gamma)\delta\tau/2$  where  $\gamma$  is some parameter to be determined, and expanding Equation 3.36 in  $\delta\tau$  it is found that

$$\langle e^{-(\delta + \bar{\delta})(H + \Delta S) - \text{tr} \ln \hat{U}_*} \rangle_\eta = 1 - A\delta\tau^2 + O(\delta\tau^3), \quad (3.37)$$

where

$$A = \frac{1}{2}\alpha(\alpha - \gamma)(1 - \pi^2) \text{tr} \left[ \mathcal{M}^{-1} \frac{d\mathcal{M}}{dU_{j,\mu}} \mathcal{M}^{-1} \frac{d\mathcal{M}}{dU_{j,\mu}} \right]. \quad (3.38)$$

If  $\gamma = \alpha$  the leading term cancels, and thus the leading error is  $O(\delta\tau^2)$  for the entire trajectory. Therefore, as claimed the  $R$  algorithm is an  $O(\delta\tau^2)$  algorithm, and thus so is the  $\chi$  algorithm (i.e., which is the  $R$  algorithm with  $\gamma = 1$ ).

Unfortunately since this non-reversible and non-area preserving integrator has been used, a Metropolis accept/reject cannot be included at the end of the MD trajectory. As a result the  $R$  algorithm is an inexact algorithm with errors  $O(\delta\tau^2)$ .

### 3.5 Polynomial Hybrid Monte Carlo

The failing of HMC at an odd number of flavours, and the stepsize errors present in the  $R$  algorithm at finite stepsize motivates another algorithm choice. One such algorithm is the Polynomial Hybrid Monte Carlo algorithm [41, 42, 43] (PHMC). In this algorithm the function of the fermion matrix is replaced by a polynomial approximation of degree  $n$  over the spectral interval of the matrix. Polynomial approximations (and rational approximations) are cheap to evaluate since they do not require explicit diagonalisation of the matrix argument [45].

The approximation which minimises the infinity norm of the magnitude of the difference between the function and the approximation is called the minimax approximation (the name minimax stems from the fact that the maximum error of the approximation is minimised),

$$\|p - f\|_\infty = \min \max_{\lambda^- \leq x \leq \lambda^+} w(x)|p(x) - f(x)|, \quad (3.39)$$

where  $f$  is function which is being approximated and  $\lambda^-$  and  $\lambda^+$  represent the bounds of the approximation interval. The weighting factor  $w(x)$  that appears in the error equation is a positive weighting factor, and defines how the error is measured. It is generally chosen to be  $w(x) \equiv 1/|f(x)|$ , i.e., relative norm, since relative errors are the form of errors one has when performing floating point calculations. For a given degree  $n$ , the optimal approximation is that which satisfies Chebyshev's criterion that the error in the approximation,  $e(x) \equiv w(x)[p(x) - f(x)]$ , attains its maximum value at at least  $n + 2$  points in the interval, and the sign of the error alternates between successive extrema [45]. Chebyshev proved that the polynomial that satisfies this criterion is uniquely defined.

When using a valid polynomial approximation, the fermionic determinant is given by

$$\det \mathcal{M}^\alpha = \int \mathcal{D}\phi^\dagger \mathcal{D}\phi \exp(-\phi^\dagger p(\mathcal{M})\phi) \det[\mathcal{M}^\alpha p(\mathcal{M})], \quad (3.40)$$

where the polynomial  $p(\mathcal{M}) \approx \mathcal{M}^{-\alpha}$  over the spectral range of  $\mathcal{M}$  and the extra determinant weighting factor in the right hand side corrects for errors arising in the approximation. With this substitution for fermionic formulation, PHMC is basically the HMC algorithm as described in § 3.2.

When approximating polynomials are of even degree  $n$ , the roots are non-degenerate and come in complex conjugate pairs. Thus the product repre-

sensation

$$p(x) = c_0 \prod_{k=1}^n (x - z_k) \quad (3.41)$$

can be written

$$p(x) = \hat{p}^\dagger(x) \hat{p}(x). \quad (3.42)$$

Hence the heatbath is given by

$$\phi = (\hat{p}^\dagger(\mathcal{M}))^{-1} \eta, \quad (3.43)$$

where  $\eta$  is a Gaussian noise vector. As the degree of the approximation becomes large, the cost of the heatbath becomes a significant cost of the algorithm, since  $\hat{p}^\dagger(\mathcal{M})^{-1} \eta$  must be evaluated.

When simulating a theory with an integer multiple of the number of flavours described by  $\mathcal{M}$ , then the force is analytic, e.g., when  $\alpha = 1$  the force is given by

$$\frac{dS_f}{dU_{j,\mu}} = -\phi^\dagger \mathcal{M}^{-1} \frac{d\mathcal{M}}{dU_{j,\mu}} \mathcal{M}^{-1} \phi \quad (3.44)$$

$$= -\phi^\dagger p(\mathcal{M}) \frac{d\mathcal{M}}{dU_{j,\mu}} p(\mathcal{M}) \phi. \quad (3.45)$$

However, when  $\alpha$  is non-integer the derivative of the fermion matrix is non-analytic, e.g., for  $\alpha = 1/2$  the force is given by

$$\frac{dS_f}{dU_{j,\mu}} = -\phi^\dagger \mathcal{M}^{-1/2} \frac{d\mathcal{M}^{1/2}}{dU_{j,\mu}} \mathcal{M}^{-1/2} \phi. \quad (3.46)$$

Since there is no analytic expression for  $\frac{d\mathcal{M}^\alpha}{dU_{j,\mu}}$  for non-integer  $\alpha$ <sup>1</sup>, the function must be approximated and the derivative taken of the approximation, as opposed to approximating the exact derivative. The force is thus written

$$\frac{dS_f}{dU_{j,\mu}} = -\phi^\dagger \sum_{k=1}^n \left( \prod_{i=1}^{k-1} (\mathcal{M} - \alpha_i) \right) \frac{d\mathcal{M}}{dU_{j,\mu}} \left( \prod_{j=k+1}^n (\mathcal{M} - \alpha_i) \right) \phi. \quad (3.47)$$

---

<sup>1</sup> Actually there is a formal definition given by  $\frac{d\mathcal{M}^\alpha}{dU_{j,\mu}} = \alpha \int_0^{\alpha-1} d\beta \mathcal{M}^{\beta-\alpha-1} \frac{d\mathcal{M}}{dU_{j,\mu}} \mathcal{M}^{-\beta}$ , however, this clearly is of no use here.



With such a force calculation there is an impracticality due to the memory storage requirements,  $n/2$  vectors must be simultaneously stored to avoid unnecessary and expensive fermion matrix multiplication.

To ensure that the resulting equilibrium distribution is free from any bias due to errors in the approximation, the determinant factor in equation (3.40) must be included either as a weighting factor when any measurements are made or, more commonly, when calculating the acceptance probability, i.e.,

$$P_{\text{acc}} = \min \left( 1, \frac{\det[\mathcal{M}^\alpha(U')p(\mathcal{M}(U'))]}{\det[\mathcal{M}^\alpha(U)p(\mathcal{M}(U))]} \exp(-\delta H) \right). \quad (3.48)$$

Since the determinant ratio cannot be included exactly, a noisy estimator of the ratio must be used [43]. This requires a noisy Metropolis test to accept/reject the trial configuration, e.g., Kennedy-Kuti acceptance test [46]. Unfortunately noisy acceptance tests generally scale worse with volume than the conventional Metropolis accept/reject, though this depends on the quality of the approximation to the determinant ratio.

PHMC has been shown to be a competitive algorithm when compared to the R algorithm for staggered fermions on medium sized volumes [47], but has not been used on large volume calculations most probably because of its poor volume scaling. The increased volume of five dimension fermion formulations, e.g., domain wall, will further increase the cost of the algorithm compared to those with the conventional Metropolis test.

Another disadvantage compared to some hybrid algorithms which use Krylov solvers is that there is no gain to be had from previous solution vectors, i.e. a chronological solver has no meaning. A related problem is that evaluation of the polynomial is potentially sensitive to rounding errors because a large degree is required. When using a Krylov method this problem can be alleviated through a restart of the inversion process, there is no equivalent when evaluating a polynomial expression. Polynomial approximations are similar to using a

Jacobi solver, which are in general much less efficient than Krylov methods. If it is possible to use a Krylov method instead of a polynomial method, e.g. when evaluating  $\mathcal{M}^{-1}$ , then it is nearly always beneficial to do so.

The evaluation of the coefficients that appear in the polynomial approximation is a complicated calculation, and it would not seem to be possible to evaluate the required coefficients for approximations  $n > 1000$  within a reasonable amount of compute time [47].

If the naïve method for calculating the derivative of the ASQTAD operator as used in [48] was used for PHMC (see § 7.1.2), this would lead to a force term calculation that is  $O(n)$  more expensive than HMC where  $n$  is the order of the polynomial and is expected to be  $O(100) - O(1000)$ . It is for this reason that no exact 2 or 1 flavour algorithm has been used for ASQTAD fermions.

### 3.6 Integration Scheme

A general class of integrators that satisfy both reversibility and area preservation are the symmetric symplectic integrators. These are derived as follows [49].

The exact evolution operator for a trajectory of length  $\tau$  is given by

$$\exp\left(\tau \frac{d}{dt}\right) \equiv \exp\left(\tau \left\{ \frac{\partial p}{\partial t} \frac{\partial}{\partial p} + \frac{\partial q}{\partial t} \frac{\partial}{\partial q} \right\}\right) \quad (3.49)$$

$$= \exp\left(\tau \left\{ -\frac{\partial H}{\partial q} \frac{\partial}{\partial p} + \frac{\partial H}{\partial p} \frac{\partial}{\partial q} \right\}\right) \quad (3.50)$$

$$= \exp\left(\tau \left\{ -S'(q) \frac{\partial}{\partial p} + T'(p) \frac{\partial}{\partial q} \right\}\right) \quad (3.51)$$

$$= \exp(\tau \{P + Q\}) \quad (3.52)$$

$$= \exp(\tau H), \quad (3.53)$$

where the Hamiltonian  $H = T + S$ ,  $q$  and  $p$  represent the canonical coordinates

and momenta,  $Q$  and  $P$  represent the evolution operators corresponding to these coordinates, i.e.,

$$e^{\delta\tau Q} : f(q, p) \rightarrow f(q + \delta\tau T'(p), p) \quad (3.54)$$

$$e^{\delta\tau P} : f(q, p) \rightarrow f(q, p - \delta\tau S'(q)) \quad (3.55)$$

That the evolution operator is given by the left hand side can be easily found through Taylor expansion. In the second step the derivatives with respect to time have been replaced using Hamilton's equations of motion.

Since the operator  $\exp(\tau\{Q + P\})$  cannot be applied directly, the best that can be done is to discretise the time, and apply successive applications of  $\exp(\delta\tau Q)$  and  $\exp(\delta\tau P)$ , e.g., the symmetric symplectic  $\hat{U}_{\text{QPQ}}$  integrator is given by

$$\hat{U}_{\text{QPQ}}(\tau) = (e^{\frac{\delta\tau}{2}Q} e^{\delta\tau P} e^{\frac{\delta\tau}{2}Q})^{\tau/\delta\tau}. \quad (3.56)$$

This scheme, and the equivalent  $\hat{U}_{\text{PQP}}$  scheme are known as leapfrog integrators because the coordinates  $q$  and  $p$  are always half a stepsize apart. These are the most commonly used integrators when doing an HMC calculation. By alternating the application of the operators  $Q$  and  $P$  an error is introduced because of non-commutivity. The error in any given integrator is found by considering the Baker-Campbell-Hausdorff formula

$$\exp A \exp B = \exp C, \quad (3.57)$$

where  $C = A + B + \frac{1}{2}[A, B] + \frac{1}{12}[A, [A, B]] + \frac{1}{12}[B, [B, A]] + \text{higher order terms}$ .

Using this formula, it can be seen that the  $\hat{U}_{\text{QPQ}}$  integrator corresponds to

$$\hat{U}_{\text{QPQ}}(\tau) \equiv \left( e^{\frac{\delta\tau}{2}Q} e^{\delta\tau P} e^{\frac{\delta\tau}{2}Q} \right)^{\tau/\delta\tau} \quad (3.58)$$

$$= \left( \exp \left[ (Q + P)\delta\tau - \frac{1}{24}([Q, [Q, P]] + 2[P, [Q, P]])\delta\tau^3 + O(\delta\tau^5) \right] \right)^{\tau/\delta\tau} \quad (3.59)$$

$$= \exp \left[ \tau \left( (Q + P) - \frac{1}{24}([Q, [Q, P]] + 2[P, [Q, P]])\delta\tau^2 + O(\delta\tau^4) \right) \right] \quad (3.60)$$

$$\equiv e^{\tau \hat{H}'_{\text{QPQ}}}. \quad (3.61)$$

It can therefore be seen that the  $\hat{U}_{\text{QPQ}}$  integrator conserves the Hamiltonian  $H'_{\text{QPQ}}$  where

$$H'_{\text{QPQ}} = H + \frac{1}{24} \left\{ -p^2 S'' + 2S'^2 \right\} \delta\tau^2 + O(\delta\tau^4). \quad (3.62)$$

Equivalently for the  $\hat{U}_{\text{PQP}}$  integrator

$$H'_{\text{PQP}} = H + \frac{1}{24} \left\{ 2p^2 S'' - S'^2 \right\} \delta\tau^2 + O(\delta\tau^4). \quad (3.63)$$

Following Camprostrini's wiggle method [50] it is possible to construct integrators that are accurate to any given order in  $\delta\tau^{2n}$  ( $\delta\tau^n$  if non-symmetric symplectic integrators are considered).

### 3.7 Sexton-Weingarten Integration

Since there is freedom to define any integration scheme that satisfies reversibility and area perservation (and has the correct continuum limit), the integration scheme can be tailored to the problem at hand. When the action can be split into two parts, then the Hamiltonian is written as

$$H = T(\pi) + S_1(U) + S_2(U). \quad (3.64)$$

If the following two integrators are constructed

$$\hat{U}_1(\delta\tau) = e^{\frac{\delta\tau}{2} P_1} e^{\delta\tau Q} e^{\frac{\delta\tau}{2} P_1} \quad (3.65)$$

$$\hat{U}_2(\delta\tau) = e^{\delta\tau P_2}, \quad (3.66)$$

where the operators  $P_1$  and  $P_2$  represent the updates to the momenta from the  $S_1$  and  $S_2$  parts of the action, and  $Q$  is the update to the gauge field (see § 3.6 for more details). Using the Baker-Campbell-Hausdorff formula, it can be seen that a reversible integrator for the full Hamiltonian is given by

$$\hat{U}_{\text{PQP}}^{\text{SW}}(\tau) = \hat{U}_2\left(\frac{\delta\tau}{2}\right) \left[ \hat{U}_1\left(\frac{\delta\tau}{\hat{m}}\right) \right]^{\hat{m}} \hat{U}_2\left(\frac{\delta\tau}{2}\right) \quad (3.67)$$



with  $\hat{m} \in \mathbb{N}$ . This compound integrator has effectively introduced two evolution time scales,  $\delta\tau$  and  $\delta\tau/\hat{m}$ . This procedure can be continued recursively, as many time scales can be introduced as desired, so long as the action can be split accordingly. This multiple time scale leapfrog integrator is known as the Sexton-Weingarten integrator [51].

If  $S_1$  is much larger than  $S_2$ , then it will have a dominant contribution to the force: the Sexton-Weingarten integration scheme is ideal for specifically such an action. Clearly such an integration scheme will only be useful when the computational cost of including  $S_2$  is much less than that of  $S_1$  for  $m \gg 1$ . Ideally the fermionic action would take the role of  $S_2$ , and the pure gauge contribution to be  $S_1$  since the cost of pure gauge force is much less than that of the fermionic. Unfortunately as the fermion mass is reduced, the fermionic high frequency fluctuations become dominant, rendering the Sexton-Weingarten scheme less beneficial without some modification of the action [52].

## 3.8 Krylov Solvers

In both the generation of gauge configurations incorporating the effect of dynamical fermions and in propagator calculations, it is required that the inverse of the Dirac operator applied to a source vector is calculated, i.e., solving equations of the form

$$Ax = b, \tag{3.68}$$

where  $A = M^\dagger M$  for dynamical fermion simulations or  $A = M$  for propagator calculations respectively. Due the sparsity and dimensions of the discretised Dirac operator, it is most efficient to use an iterative method rather than an explicit matrix inversion.

When solving a system of equations where the matrix operator is hermitian,

e.g. as in the case of dynamical fermion simulations, generally the conjugate gradient method is used [53]. For non-hermitian operators either the system  $A^\dagger Ax = A^\dagger b$  can be solved, the so called conjugate gradient normal equation method (CG-NE), at the expense of squaring the condition number, or use another method such as BiCGstab [54].

### 3.8.1 Conjugate Gradients

Consider the functional

$$\psi = \frac{1}{2}x^\dagger Ax - x^\dagger b. \quad (3.69)$$

If  $A$  is a hermitian matrix, it can immediately be seen that the minimum of this functional corresponds to the solution of equation 3.68. The residual vector is the negative gradient of the functional at coordinate  $x$  and is given by

$$r = b - Ax. \quad (3.70)$$

In the method steepest descents, an initial guess is made to the system  $x_0$ , and then iterated upon by proceeding down the gradients of the functional, i.e.,  $r$ . Eventually, the global minimum of the functional shall be reached, this corresponding to the required solution. The updates take the form of

$$r_{i+1} = r_i - \alpha_i Ar_i, \quad (3.71)$$

$$x_{i+1} = x_i + \alpha_i r_i, \quad (3.72)$$

with  $r_0 = b - Ax_0$  and where  $\alpha_i = |r_i^\dagger r_i|/|r_i^\dagger Ar_i|$  is chosen so as to minimise the value of the functional, in the direction  $r_i$ .

The problem with this simple minded method is that the gradients are not guaranteed to be orthogonal to each other. Since at each iteration a dimension is removed, any subsequent search directions should ignore any components in that direction. This is the basis for the Conjugate Gradient (CG) algorithm.

When each new residual vector is calculated, the previous search directions are subtracted, and this reduced vector  $p$  is used for the new search direction.

The CG algorithm is given as follows [53].

- $x_0 = \text{initial guess}, r_0 = p_0 = b$
- while  $|r_i| > \text{tol}$ , for  $i = 0, 1, 2 \dots$ 
  - $\alpha_i = \frac{|r_i|^2}{p_i A p_i}$
  - $x_{i+1} = x_i + \alpha_i p_i$
  - $r_{i+1} = r_i - \alpha_i A p_i$
  - $\beta_{i+1} = \frac{|r_{i+1}|^2}{|r_i|^2}$
  - $p_{i+1} = r_{i+1} + \beta_i p_i$

The choice of the coefficient  $\beta_i$  ensures that the direction vectors are  $A$  conjugate to each other (or equivalently that the residual vectors are orthogonal to each other).

The CG algorithm is guaranteed to converge in  $N$  iterations where  $N$  is the dimension of the matrix being inverted, however in practice it will reach a desired tolerance much faster than this. The convergence rate of Krylov solvers is to a first approximation given by the condition number of the matrix, this is defined to be the ratio of the greatest to smallest eigenvalue of the matrix. As the fermion mass is reduced, the condition number increases, which in turn increases the number of iterations required for a Krylov method to converge to a given accuracy.

The CG algorithm (and the like) are known as Krylov solvers because they build up a Krylov subspace  $\mathcal{K} \in \text{span}\{A^i r, i = 0 \dots n - 1\}$ , from which a solution is constructed. Thus the solution can be thought of as a polynomial

in  $A$ . The polynomial constructed through a Krylov method will require fewer matrix-vector products than that constructed through an explicit polynomial approximation, this is because the Krylov polynomial is specific to the current matrix, and in some sense is more optimal than the general polynomial approximation.

### 3.8.2 Multi-shift Solvers

Take the following two equations

$$Ax = b \tag{3.73}$$

and

$$(A + \sigma)x^\sigma = b, \tag{3.74}$$

where  $A$  is some matrix,  $\sigma$  is proportional to the identity,  $b$  is the source vector and  $x$  and  $x^\sigma$  are the solution vectors for two equations respectively. These two equations can be solved separately using a Krylov method, but note that when a zero initial guess is made, then the resulting Krylov subspaces constructed are identical for both, i.e.,  $\mathcal{K}_n(A + \sigma) = \text{span}\{(A + \sigma)^i r_0^\sigma, i = 0 \dots n - 1\} = \text{span}\{A^i r_0, i \dots n - 1\} = \mathcal{K}_n(A)$  if  $r_0 = r_0^\sigma$ . The zero initial guess condition is the only one which will give trivially coincident Krylov subspaces.

Multi-shift (also known as multi-mass) solvers use this coincident Krylov space to calculate matrix inversions simultaneously, without the need for any additional matrix-vector products [55]. The inversion of the lightest system follows that of a regular Krylov method, e.g., the CG method given in §3.8.1. Throughout the inversion, the residual vectors of the shifted and unshifted systems are parallel, and related through the real scalar  $\zeta_i^\sigma$ , where

$$r_i^\sigma = \zeta_i^\sigma r_i. \tag{3.75}$$



In addition, all direction vectors are “A-parallel”, i.e.,

$$(A + \sigma)p_i^\sigma = \zeta_i^\sigma Ap_i. \quad (3.76)$$

The values of  $\zeta_i^\sigma$ ,  $\alpha_i^\sigma$  and  $\beta_i^\sigma$  arise from coupled two term recurrence relations [55], their updates are given by

$$\zeta_{i+1}^\sigma = \frac{\zeta_i^\sigma \zeta_{i-1}^\sigma \alpha_{i-1}^\sigma}{\alpha_i \beta_i (\zeta_{i-1}^\sigma - \zeta_i^\sigma) + \zeta_{i-1}^\sigma \alpha_{i-1}^\sigma (1 - \sigma \alpha_i)} \quad (3.77)$$

$$\alpha_i^\sigma = \alpha_i \frac{\zeta_{i+1}^\sigma}{\zeta_i^\sigma} \quad (3.78)$$

$$\beta_i^\sigma = \beta_i \frac{\zeta_i^\sigma \alpha_{i-1}^\sigma}{\zeta_{i-1}^\sigma \alpha_{i-1}^\sigma}. \quad (3.79)$$

From these relations, the updates to the shifted solution and direction vectors,  $x^\sigma$  and  $p^\sigma$ , can be found without any additional matrix-vector products.

The multi-shift conjugate gradient algorithm is given as follows [55].

- $x_0^\sigma = 0$ ,  $r_0 = p_0^\sigma = b$ ,  $\alpha_{-1} = \zeta_{-1}^\sigma = \zeta_0^\sigma = 1$ ,  $\alpha_0^\sigma = 0$
- while  $|r_i| > tol$ , for  $i = 0, 1, 2, \dots$ 
  - $\alpha_i = \frac{|r_i|^2}{p_i^\sigma Ap_i}$
  - calculate  $\alpha_i^\sigma, \zeta_{i+1}^\sigma$  according to equations (3.78) and (3.77) respectively
  - $x_{i+1}^\sigma = x_i^\sigma + \alpha_i^\sigma p_i^\sigma$
  - $r_{i+1} = r_i - \alpha_i Ap_i$
  - $\beta_{i+1} = \frac{|r_{i+1}|^2}{|r_i|^2}$
  - calculate  $\beta_{i+1}^\sigma$  according to equation (3.79)
  - $p_{i+1}^\sigma = \zeta_i^\sigma r_{i+1} + \beta_i^\sigma p_i^\sigma$

The number of Dirac applications needed is the same as for the zero shift system, since the shifted systems will converge quicker since they are better conditioned.

Multi-shift solvers can be used when performing a partially quenched analysis when the mass parameter is a multiple of the identity matrix. In this case, many propagators can be obtained for approximately the same cost as the lightest mass propagator.

### 3.9 Chronological Inversion

When evolving an MD trajectory, from each step to the next, the gauge field is only changing  $O(\delta\tau)$  which for small  $\delta\tau$  is only a small amount. At each one of these MD steps, the inverse of the Dirac operator applied to a constant right hand side must be evaluated. Hence the function  $\mathcal{M}(\tau)^{-1}\phi$  is a smooth function in time, and it would seem sensible to construct an initial guess for a Krylov method using previous solution information [36]. This indeed proves to be the case, and the optimal method for doing this is to perform a CG minimisation on the history of previous solution vectors, to construct the best initial guess [56]. Since the past solution vectors will have a large overlap, this minimisation is best done by first orthonormalising this vector space. The chronological inverter is thus given by

1. Orthonormalise past solution vectors  $v_k$ .
2. Solve the system  $\sum_{k=1}^n c_k v_l^\dagger A v_k = v_l^\dagger b$  using a direct method.
3. Construct initial guess  $x_0 = \sum_{k=1}^n c_k A v_k$ .
4. Perform CG inversion, with initial guess  $x_0$ .

Typically, it is found that it is best to use up to the last ten solution vectors, presumably, older solution vectors have little relevance to the current problem. If the tolerance on the residual is not small enough there will be an observable

violation of reversibility arising from using a non-zero initial guess. As a result, the residual must be made smaller than one would otherwise expect to maintain an exact algorithm. This reduces the improvement that can be gained from using a chronological solver, but it still generally gains a factor of two [57].

Note that such acceleration cannot be used with the R algorithm, since at every MD step the right hand side changes, which means no improvement can be gained by using previous inversion information.





# Chapter 4

## Rational Hybrid Monte Carlo

In chapter 3 the current popular algorithms for lattice QCD were outlined. There is clearly a need for another choice of algorithm for non-integer  $\alpha$  theories since the R algorithm is inexact with respect to the integrating stepsize, and PHMC is more expensive than HMC (especially for ASQTAD fermions and/or large volumes).

If one is to proceed using a conventional Metropolis acceptance test at the end of a trajectory, the measure of the action at the beginning and at the end of the MD trajectory must be performed without any systematic error, otherwise a weighting factor must be included (equation (3.40)) for the resulting ensembles to have the desired probability distribution. Optimal rational approximations are in general orders of magnitude better than polynomial approximations and, for correctly chosen parameters, allow a computation of the desired function to arbitrary precision with far greater ease than with polynomial approximation. Hence, an algorithm can be constructed which does not require any determinant reweighting. This is the basis for the Rational Hybrid Monte Carlo algorithm (RHMC).

In this section RHMC shall be derived, and the particular implementation used for this work shall be explained. The algorithm shall then be compared to published data, produced using the R algorithm and PHMC using naïve staggered fermions to check the algorithm's correctness and efficiency.

## 4.1 Rational Approximations

Optimal rational approximations are similar to equivalent polynomial approximations, with the obvious difference that a ratio of polynomial functions is used as the approximating function. Chebyshev's criterion states that the error  $e(x)$  takes its maximum absolute value  $\Delta$  at at least  $n + d + 2$  points, where  $n$  and  $d$  are the degrees of the numerator and denominator polynomials respectively, and that the sign of error alternates between successive extrema.

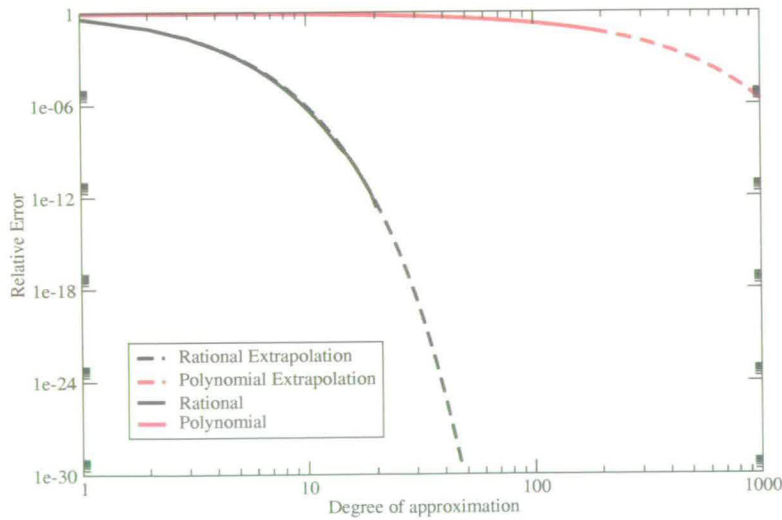


Figure 4.1: A plot comparing the maximum error of the polynomial and rational approximations to the square root function, where the spectral range has been set appropriate for staggered mass  $m = 0.01$ . The extrapolation at high degree is required due to the cost of evaluating such approximations.

The optimal rational approximation satisfies

$$(-1)^i \Delta = w(x_i)(r(x_i) - f(x_i)), \quad (4.1)$$

where  $f(x)$  is the function being approximated,  $r(x) = p(x)/q(x)$  and  $x_i$  are the locations of the extremal errors. For this work the weighting factor was set  $w(x) = 1/|f(x)|$ , i.e., relative error measure.

The advantage of rational approximations is that they converge to the desired function much faster than polynomial approximations as a function of the approximation degree (figure 4.1). All of the work presented here shall assume that the degree of the numerator polynomial, and that of the denominator polynomial are equal<sup>1</sup>. As the quark mass is reduced, the spectral interval increases because the lower bound of the Dirac operator will decrease, while the upper bound remains approximately constant. As a result, the degree of approximation must be increased to maintain a constant maximum error.

Optimal rational approximations to  $x^\alpha$  are calculated using the Remez algorithm [58]. It turns out that the roots of these polynomials are real and non-degenerate. When  $\alpha < 1$  these are always positive, however, when  $\alpha > 1$  they are negative. This has important consequences that shall be touched upon later (§8.3). Figure 4.2 is a plot showing the maximum error as a function of  $\alpha$ . There are discontinuities in the plot at integer multiples of  $\alpha = 1$ , this merely reflects the fact that the optimal rational approximation to  $x$  ( $1/x$ ) is  $x$  ( $1/x$ ) etc.. Obviously the rational approximation to  $x^{-\alpha}$  is equal to that approximation generated by  $x^\alpha$ , but with the numerator and denominator polynomials exchanged.

---

<sup>1</sup>When odd functions are considered, e.g., the sgn function, the degree of these polynomials must differ by an odd number, and by an even number for even functions. Since only positive arguments are considered here, this restriction does not apply.

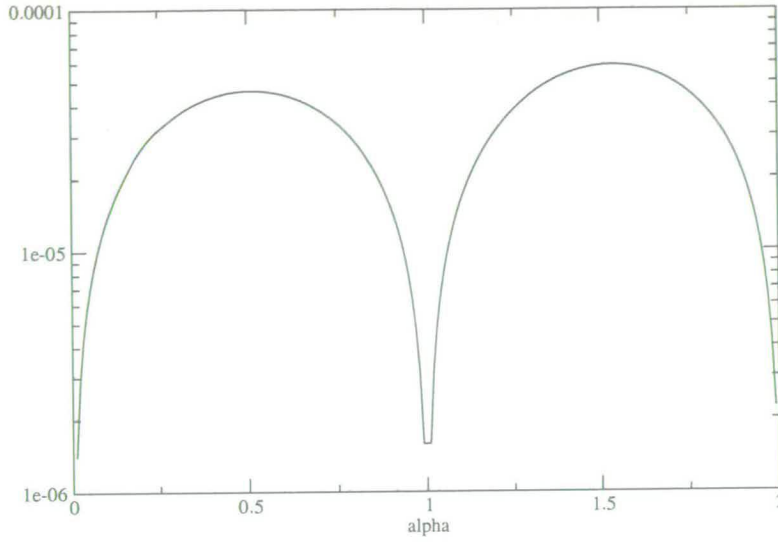


Figure 4.2: Maximum relative error in rational approximation as a function of the approximating power  $\alpha$ ,  $n = 8$ , bounds appropriate for staggered mass  $m = 0.01$ .

## 4.2 Remez Algorithm

The Remez algorithm [58] is the algorithm that is used to generate the coefficients for an optimal rational approximation to a given function over some finite range. This algorithm follows an iterative procedure to form the solution. For the special case of the (inverse) square root function with minimum relative errors, the rational coefficients can be calculated analytically [59].

A coordinate set  $X$  is defined  $\{x_i \in [\lambda^-, \lambda^+], i = 1, \dots, n + d + 2\}$  which are the locations of the maxima of the error function. The Remez algorithm essentially consists of alternating two steps

1. Hold  $X$  fixed, find the new  $r(x)$ .
2. Hold  $r(x)$  fixed, find the new  $X$ .



In doing 1., the rational approximation which satisfies

$$e(x_i) \equiv w(x_i)[f(x_i) - r(x_i)] = (-1)^i \Delta \quad (4.2)$$

must be found. This equation can be written in linear form with respect to the rational coefficients by multiplying both sides of the equation by the polynomial denominator, and writing the two polynomials as  $p(x) = \sum_{j=0}^n x^j p_j$  and  $q(x) = \sum_{j=0}^d x^j q_j$ . Hence the following matrix equation must be solved

$$\sum_{j=0}^d x^j q_j [f(x_i) - (-1)^i \Delta / w(x_i)] - \sum_{j=0}^n x^j p_j = 0 \quad (4.3)$$

$$Mv = 0, \quad (4.4)$$

where the vector  $v = (p_0, \dots, p_n, q_0, \dots, q_d)$ . This is a set of  $n + d + 2$  equations with  $n + d + 2$  degrees of freedom. There are only  $n + d + 1$  degrees of freedom in the vector  $v$  since a rational function is fully described by its roots, poles and an overall scale factor. Thus one of the elements of  $v$  can be set equal to unity without loss of generality. The additional degree of freedom is  $\Delta$ , which appears in the matrix, hence the system of equations cannot be solved directly using matrix inversion. In principle  $\Delta$  can be found directly from the condition that  $\det M = 0$ , which must be true if a solution to equation (4.4) exists. However, finding the unique real solution can be problematic. The most reliable method for solving this matrix problem is to calculate an approximate  $\Delta$  using the previous iterations rational coefficients, and then solve the matrix equation [58]. Since this shall not be an exact solution, it may be required that this procedure is performed iteratively, until self consistency between the input and output  $\Delta$  has been achieved. The initial approximation to use on the first iteration does not strongly affect whether the algorithm converges or not. The algorithm as implemented for this work uses a direct solve to perform the matrix inversion to obtain the rational coefficients, but an iterative Krylov method may be preferable since the previous iteration's solution would form a good initial guess.

The calculation performed in 2. requires that the global maxima of the error function are found. This is generally done through a simple searching method. When approximating a function that has a lot of high frequency information, this process can breakdown and/or require very small steps to be made when doing this searching. Due to this potential instability, the Remez algorithm requires high precision floating point arithmetic, greater than standard double precision, generally 30-50 digit precision is needed, depending on the accuracy and interval required. High precision arithmetic is easily implemented on a computer through the use of the GNU Multiple Precision library [60]. The Remez algorithm can have poor convergence for exceptional functions, this is especially true when a rapidly oscillating function is being approximated. Fortunately for the functions considered here, e.g.,  $f(x) = x^\alpha$ , this is not the case.

Once the rational approximation has been found, all that remains is to find the roots and poles of the rational function, to allow representation in either product form or partial fraction form. The method chosen in this work was the Newton-Raphson root finding method with deflation<sup>2</sup>, where the roots of each of the numerator and denominator polynomials are found separately.

### 4.3 Partial fraction expansions and multishift solvers

A rational function with the Dirac operator  $\mathcal{M}$  as an argument, when written in product representation applied to a source vector, is written

$$r(\mathcal{M})\phi = \alpha_0 \frac{\prod_{k=1}^n (\mathcal{M} + \gamma_k)}{\prod_{k=1}^d (\mathcal{M} + \beta_k)} \phi. \quad (4.5)$$

---

<sup>2</sup>Deflation is where the previous solution is projected out of the function when searching for the next root - this ensures that all roots are found.

In this notation these shifts are always positive, so the condition number is always improved compared to the unshifted case. To evaluate this function would still be very expensive since each denominator monomial has to be evaluated, corresponding to  $d$  Krylov subspace inversion operations.

Any rational function can be written as a polynomial quotient plus the sum of a series of partial fractions. Since it is always assumed that the degree of the numerator and denominator polynomials are equal  $n = d$ , in which case the quotient is just a constant. Hence equation (4.5) can be rewritten

$$r(\mathcal{M})\phi = \left( \alpha_0 + \sum_{k=1}^n \frac{\alpha_k}{\mathcal{M} + \beta_k} \right) \phi. \quad (4.6)$$

To evaluate this function still requires  $d$  inversion operations, but the crucial difference here is that all of the denominator monomials are now acting on a common vector. This allows the use of a multi-shift solver to perform the matrix inversions simultaneously [55].

## 4.4 Rational Hybrid Monte Carlo

In a similar vein to PHMC the fermion determinant is written

$$\det \mathcal{M}^\alpha = \int \mathcal{D}\phi^\dagger \mathcal{D}\phi \exp(-\phi^\dagger r_{\text{mc}}^2(\mathcal{M})\phi) \det[\mathcal{M}^\alpha r_{\text{mc}}^2(\mathcal{M})], \quad (4.7)$$

where  $r_{\text{mc}}(x)$  is an approximation to  $x^{-\alpha/2}$  over the spectral range of the fermion matrix (the reason for the subscript shall become clear in the following). The reason that  $r_{\text{mc}}^2$  approximates the desired function, and not just  $r_{\text{mc}}$ , is to allow a pseudofermion heatbath to be used (see §4.4.1). The approximation  $r_{\text{mc}}$  is chosen such that the maximum relative error is small enough such that the approximation does not induce a systematic bias. In other words the determinant in equation (4.7) can be set equal to unity without fear of bias. This means the conventional Metropolis acceptance test can be used. The question of course



then arises, what is considered a small enough maximum error such that no bias is introduced? Since all calculations are performed using double precision arithmetic, using the unit of lowest precision of double precision arithmetic ( $1 \text{ ulp} \approx 10^{-15}$ ) would seem like a sensible maximum error to require of the approximation. However, since all evaluations of the rational approximation are done using a Krylov solver, the error on the approximation will also be governed by the tolerance of the solver. In most HMC calculations the tolerance of the solver is generally set to  $10^{-8} - 10^{-10}$  when evaluations of the action are required [41, 57]. With this in mind, the maximum error of the approximation  $\Delta_{\text{mc}}$  is always set to be smaller than the tolerance of solver.

Each RHMC trajectory thus consists of repeated applications of the following procedure.

- Momentum refreshment heatbath using Gaussian noise ( $P(\pi) \propto e^{-\pi^2/2}$ ).
- Fermion refreshment ( $\phi = r_{\text{mc}}^{-1}(\mathcal{M})\eta$ , where  $P(\eta) \propto e^{-\eta^2/2}$ ).
- MD trajectory consisting of  $\tau/\delta\tau$  steps.
- Metropolis accept/reject with probability  $P_{\text{acc}} = \min(1, e^{-\delta H})$ .

#### 4.4.1 Pseudofermion Heatbath

As stated in § 4.1, the roots and poles of rational approximations are always real and non-degenerate. Therefore, a simple decomposition of the rational approximation as can be done for the polynomial case (equation (3.42)) cannot be performed. However, if the rational function approximates the square root of the desired function, then the heatbath is trivially given by inverting the rational function,

$$r_{\text{mc}}(\mathcal{M})\phi = \eta. \quad (4.8)$$



where  $\eta$  is again a Gaussian noise vector. The inverse of the rational function is expanded in partial fractions and evaluated using a multi-shift solver, the resulting solutions are then summed together with the constant term to give the required pseudofermion vector. When evaluating the sum of partial fractions, the order of the summation is always from smallest to largest contributions (i.e., from largest shift to smallest shift), since this will minimise the rounding errors incurred. It is empirically found that the summation is extremely stable, and any rounding errors seem extremely mild (all of this work was conducted using double precision arithmetic).

#### 4.4.2 Computation of the Force

When implementing the force, the naïve prescription would be to use the approximation  $r_{\text{mc}}^2(\mathcal{M})$  which appears in the action. However, doing so would lead to a force term of the form

$$\frac{dS_{\text{f}}}{dU_{j,\mu}} = - \sum_{k=1}^n \alpha_k^2 \phi^\dagger (\mathcal{M} + \beta_k)^{-1} \left[ (\mathcal{M} + \beta_k)^{-1} \mathcal{M}' + \mathcal{M}' (\mathcal{M} + \beta_k)^{-1} \right] (\mathcal{M} + \beta_k)^{-1} \phi. \quad (4.9)$$

This results in a force term which requires 2 Krylov inversions per evaluation because of the presence of the double poles, which is double that of both the R and HMC algorithms.

Instead note that it is not required to use  $r_{\text{mc}}^2(\mathcal{M})$  in the pseudofermionic action when calculating the force, in fact any functional form can be used. However, to obtain a non-negligible acceptance rate this functional form must be a good approximation to the above force. Motivated by this, the action can be approximated as

$$S_{\text{f}} = \phi^\dagger r_{\text{mc}}^2(\mathcal{M}) \phi \quad (4.10)$$

$$\approx \phi^\dagger r_{\text{md}}(\mathcal{M}) \phi, \quad (4.11)$$

where  $r_{\text{md}}(\mathcal{M})$  is a rational approximation to  $\mathcal{M}^{-\alpha}$ . Regardless of the error bound  $\Delta_{\text{md}}$  on  $r_{\text{md}}$ , the correct distribution is obtained, any errors being stochastically corrected for by the acceptance test in the same manner that the finite stepsize errors are corrected. This is similar to the algorithm presented in [41], where a looser stopping condition is used for the MD Krylov solves, with a tight one used for MC acceptance evaluation.

For RHMC, the strategy therefore is to use an “exact” approximation  $r_{\text{mc}}(\mathcal{M})$  for the heatbath and Metropolis test, but evolve the gauge fields using a lower order approximation  $r_{\text{md}}(\mathcal{M})$  in the MD. Although the number of Dirac applications is solely dependent on the smallest polar shift in the rational approximation, using a lower degree function is potentially beneficial because the smallest shift increases with decreasing degree, effectively giving a better conditioned matrix to be inverted. There is also a reduction in the amount of linear algebra performed, which is beneficial since linear algebra is the bottleneck with inverter performance (see appendix D). The force is now given by

$$\frac{dS_{\text{f}}}{dU_{j,\mu}} = \phi^\dagger r'_{\text{md}}(\mathcal{M}) \phi \quad (4.12)$$

$$= - \sum_{k=1}^n \bar{\alpha}_k \phi^\dagger (\mathcal{M} + \bar{\beta}_k)^{-1} \frac{d\mathcal{M}}{dU_{j,\mu}} (\mathcal{M} + \bar{\beta}_k)^{-1} \phi. \quad (4.13)$$

As when performing the heatbath, the vectors  $X_k = (\mathcal{M} + \bar{\beta}_k)^{-1} \phi$  are obtained in one step using a multi-shift solver. For each  $k^{\text{th}}$  contribution to the force, the matrix  $\mathcal{M}'$  applied to the  $k^{\text{th}}$  solution vector must be calculated. For simple fermion kernels (e.g., naïve staggered, Wilson) this is a trivial calculation, and so is of negligible cost. For the case of ASQTAD fermions the situation is similar to that as described for the PHMC algorithm (§3.5), where the resulting force calculation cost would render the algorithm infeasible. The resolution to this force calculation cost for ASQTAD fermions is described in chapter 7.

The evaluation of the force requires storage of  $2n + 3$  vectors. However, unlike

the polynomial case, this does not mean any limitation in practice since rational approximations require a much lower degree approximation than polynomials to obtain the same quality of approximation.

#### 4.4.3 Energy measurement

The energy measurement is required in the calculation of the  $e^{-\delta H}$  factor for the Metropolis acceptance test. For the pseudofermionic contribution to the action, this requires the evaluation of the action at the beginning and at the end of the trajectory. The contribution at the start of each trajectory can be evaluated trivially for the cost of a scalar product: observe from equation (4.8)

$$S_f^{\text{initial}} = \phi^\dagger r_{\text{mc}}^2(\mathcal{M}(0))\phi \quad (4.14)$$

$$= \eta^\dagger r_{\text{mc}}^{-1}(\mathcal{M}(0))r_{\text{mc}}(\mathcal{M}(0))r_{\text{mc}}(\mathcal{M}(0))r_{\text{mc}}^{-1}(\mathcal{M}(0))\eta \quad (4.15)$$

$$= \eta^\dagger \eta. \quad (4.16)$$

The evaluation of the contribution at the end of the trajectory requires the evaluation of the norm of the rational approximation applied to the pseudofermion,

$$S_f^{\text{final}} = \phi^\dagger r_{\text{mc}}^2(\mathcal{M}(\tau))\phi. \quad (4.17)$$

This costs an additional inversion operation and a scalar product.

#### 4.4.4 Integration Scheme

For the case of conventional HMC,  $\tau/\delta\tau + 1$  inversions per trajectory are evaluated regardless if a  $\hat{U}_{\text{QPQ}}$  or  $\hat{U}_{\text{PQP}}$  integrator is used. For RHMC the situation is complicated slightly by the extra inversions required for the heatbath and the end of trajectory fermion action evaluation. For HMC using the  $\hat{U}_{\text{PQP}}$  integrator, the final MD inversion is the same inversion as is required for the MC

(modulo solver tolerance), but with RHMC they are different since different approximations are used for MD and MC. Thus if a  $\hat{U}_{\text{PQP}}$  integrator is used, then  $\tau/\delta\tau + 3$  inversions per trajectory will be required. On the other hand,  $\tau/\delta\tau + 2$  inversions are required for a  $\hat{U}_{\text{QPQ}}$  scheme. With this in mind, and the better acceptance rate obtained from such an integrator [39], the optimum choice is a  $\hat{U}_{\text{QPQ}}$  scheme.

In addition a Sexton-Weingarten integration scheme [51] was implemented. For a  $\hat{U}_{\text{QPQ}}$  integrator, the Sexton-Weingarten integration scheme is given by

$$\hat{U}_{\text{QPQ}}^{\text{SW}}(\tau) = \left[ \hat{U}_1 \left( \frac{\delta\tau}{2\hat{m}} \right) \right]^{\hat{m}} \hat{U}_2(\delta\tau) \left[ \hat{U}_1 \left( \frac{\delta\tau}{2\hat{m}} \right) \right]^{\hat{m}} \quad (4.18)$$

where

$$\hat{U}_1(\delta\tau) = e^{\frac{\delta\tau}{2}Q} e^{\delta\tau P_1} e^{\frac{\delta\tau}{2}Q} \quad (4.19)$$

$$\hat{U}_2(\delta\tau) = e^{\delta\tau P_2}. \quad (4.20)$$

The conserved Hamiltonian  $H'$  is found from considering the error in the integrator to occur only in the non-commutivity between  $P_1$  and  $Q$ , and from  $P_2$  and  $Q$  (trivially  $P_1$  and  $P_2$  must commute) to be

$$H' = H + \frac{1}{48\hat{m}} \left\{ -p^2 S_1'' + 2S_1'^2 \right\} \delta\tau^2 + \frac{1}{24} \left\{ -p^2 S_2'' + 2S_2'^2 \right\} \delta\tau^2 + O(\delta\tau^4). \quad (4.21)$$

As to how the action is split between into two pieces is left open depending on the fermion formulation.

## 4.5 Algorithm Testing

All of the initial testing and verification of the RHMC algorithm was done using naïve staggered fermions. The reasoning behind this being that this fermion formulation is the simplest to apply RHMC to, and is the least computationally demanding. The fermion matrix is explicitly bounded from below by the



mass parameter, which means that the required rational approximations can be constructed without complication. In addition, the derivative of the fermion matrix is extremely simple and cheap to calculate. As a result of this, the force written in equation (4.13) comprises a negligible fraction of the algorithm cost, given that the partial fractions have been evaluated, so the algorithm can be used as just described.

Parameters from published data were chosen to perform this comparison [47], the physics parameters being those given in table 4.1. This publication concerned a comparison between the R algorithm and PHMC using two flavours of staggered fermions so is ideal for these purposes.

V	Gauge Action	$\beta$	Fermion action	$N_f$	$m_f$
$16^4$	Wilson	5.70	Staggered	2	0.02

Table 4.1: The physics parameters chosen from [47] for comparison against RHMC

Hence the required action to be used in the calculation is

$$S = S_g^w + \phi^\dagger \mathcal{M}^{-1/2} \phi. \quad (4.22)$$

The possible range of eigenvalues of the staggered kernel is  $[4m^2, 64 + 4m^2]$  (the normalisation was a factor of 4 greater than that given in equation (2.13)), hence the rational approximations were set valid over this range. During this testing, the multiple time scale implementation was also verified, this was done through placing the pure gauge contribution on the short time scale ( $S_g^w = S_1$ ) and the fermion contribution on the long time scale ( $S_f = S_2$ ).

To define a more meaningful measure of algorithmic efficiency to compare these algorithms, the following measure was used

$$\text{Efficiency} = \frac{\langle A \rangle}{N_{mv}}, \quad (4.23)$$

where  $\langle A \rangle$  is the expectation value of the acceptance rate, and  $N_{\text{mv}}$  is the number of matrix-vector products per trajectory. This measure assumes that the dominant cost of hybrid algorithms lies purely in the matrix-vector multiplication. This is not necessarily true, e.g., the force calculation for PHMC may be significant, but as a direct comparison it is suitable for this work.

Algorithm	PHMC	PHMC	PHMC	R
$n$	300	400	500	
$\tau$	1	1	1	1
$\delta\tau$	0.02	0.02	0.02	0.02
$N_{\text{traj}}$	1700	1050	800	1000
$N_{\text{mv}}$	61291(183)	73176(296)	87955(350)	77880(807)
$\langle P \rangle$	0.577099(46)	0.577130(46)	0.577023(43)	0.57726149
$\langle \delta H \rangle$	0.1112(126)	0.1359(147)	0.1497(187)	
$\langle A \rangle$	0.6329(122)	0.7657(168)	0.7675(191)	
$\langle E \rangle \times 10^{-5}$	1.0326(231)	1.0463(274)	0.8726(252)	

Table 4.2: The algorithm parameters and results published in [47].  $n$  is the degree of the polynomial approximation,  $N_{\text{traj}}$  is the number of trajectories obtained,  $\langle P \rangle$  is the average plaquette and  $\langle E \rangle$  is the average efficiency. The errors were found using a jackknife analysis.

The results published from [47] are shown in table 4.2 and results generated using RHMC are shown in table 4.3. The immediate conclusion that can be drawn is that the plaquette distributions between the RHMC and PHMC results are, within errors, consistent with each other, suggesting that RHMC has the correct fixed point. The plaquettes produced by RHMC at the two different stepsizes are compatible with each other, as they must be given the Metropolis test correcting for stepsize errors.

At these parameters RHMC is at least 50% more efficient than the optimal PHMC run ( $n = 400$ ), this efficiency is not due to an increased acceptance rate, the acceptance rate of RHMC being 70% compared to the PHMC's 74%, but rather because of the decreased number of matrix-vector products. This reflects the fact that Krylov methods can obtain a given precision using less matrix-vector multiplication than polynomial methods. What is interesting to note is that  $n = 400$  and  $n = 500$  PHMC results have a higher acceptance rate than that of RHMC.

Algorithm	RHMC	RHMC
$n_{\text{md}}$	10	10
$\Delta_{\text{md}}$	$4.21 \times 10^{-7}$	$4.21 \times 10^{-7}$
$n_{\text{mc}}$	15	15
$\Delta_{\text{mc}}$	$1.34 \times 10^{-10}$	$1.34 \times 10^{-10}$
$\text{res}_{\text{md}}$	$10^{-6}$	$10^{-6}$
$\text{res}_{\text{mc}}$	$10^{-9}$	$10^{-9}$
$\tau$	1	1
$\delta\tau$	0.02	0.04
$N_{\text{traj}}$	2000	2000
$N_{\text{mv}}$	42549(10)	22417(4)
$\langle P \rangle$	0.577110(53)	0.577164(122)
$\langle \delta H \rangle$	0.2974(377)	4.325(230)
$\langle A \rangle$	0.6989(93)	0.1387(104)
$\langle E \rangle$	$1.6153(434) \times 10^{-5}$	$6.106(553) \times 10^{-6}$

Table 4.3: The RHMC algorithmic parameters and some observables measured from the resulting gauge fields. The parameters  $\text{res}_{\text{md}}$  and  $\text{res}_{\text{mc}}$  represent the tolerances used in the multi-shift solver for the MD and heatbath/acceptance solves respectively. The errors were found using a jackknife analysis.

The results shown in table 4.3 were high statistics runs intended to show that RHMC does indeed tend towards to the correct fixed point. In an attempt to ascertain the scaling behaviour of RHMC many shorter runs were carried out. For tabulated details of these results refer to appendix A.

It was deemed important to investigate the scaling behaviour of RHMC with the degree of the MD approximation  $n_{\text{md}}$  since tuning this parameter could severely affect the efficiency of the algorithm. Figure 4.3 is a plot of the maximum error  $\Delta_{\text{md}}$  as a function of  $n_{\text{md}}$ . The exponential reduction of error with increasing degree can clearly be seen.

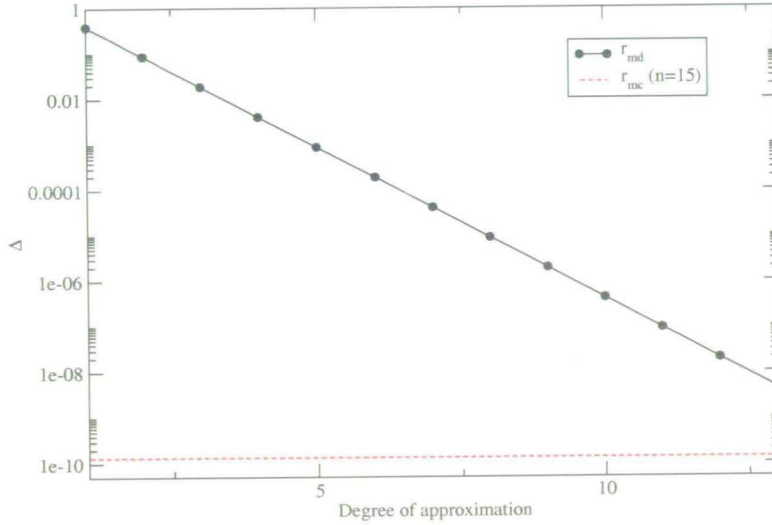


Figure 4.3: A plot showing the dependence of the MD approximation error  $\Delta_{\text{md}}$  (defined in equation (4.2)) with degree  $n_{\text{md}}$ . Also included in the plot is the error of  $r_{\text{mc}}$ . The parameters and results are given in tables 4.1 and A.1.

The variation in the condition number of the fermion matrix as a function of approximation degree is illustrated in figure 4.4. As the degree of approximation is increased from  $n_{\text{md}} = 1$ , there is an initial sharp increase in the number of matrix-vector products which levels off at around  $n_{\text{md}} = 6$ . This might suggest that it could be very advantageous to use a very low degree approximation, however, the acceptance rate obtained when using such small  $n_{\text{md}}$  is negligible



(Figure 4.5). There is a sharp transition between  $n_{\text{md}} = 3$  and  $n_{\text{md}} = 4$  from negligible acceptance to non-negligible. After this transition the acceptance rate slowly increases, reaching a value of 76% at  $n_{\text{md}} = 13$ . This in part explains why the  $n = 400$  and  $n = 500$  PHMC results had a higher acceptance rate than the equivalent RHMC result where  $n_{\text{md}} = 10$  was used, i.e., the error in the MD approximation was causing a reduction in acceptance rate (a similar increase in acceptance rate would be expected if tolerance of the MD solver was tightened). When the resulting efficiency is plotted (Figure 4.6) it can be seen that the two effects almost cancel each other out completely. The conclusion being that when in a regime with a reasonable acceptance rate, there is little that can be tuned to significantly improve the efficiency.

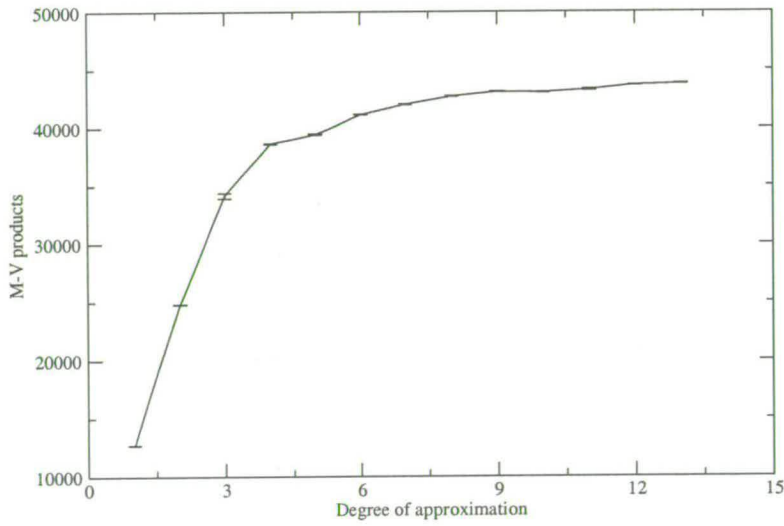


Figure 4.4: A plot showing how  $N_{\text{mv}}$  varies with  $n_{\text{md}}$ , i.e., the plot reveals how the increase in  $n_{\text{md}}$  causes the matrix condition number to increase. The parameters and results are given in tables 4.1 and A.1.

The variation of the plaquette with MD approximation degree is shown in figure 4.7. This plot was initially worrying, because there appears to be a trend that the plaquette decreases with increasing approximation error, suggesting that the errors in the approximation are not corrected by the Metropolis accept/reject

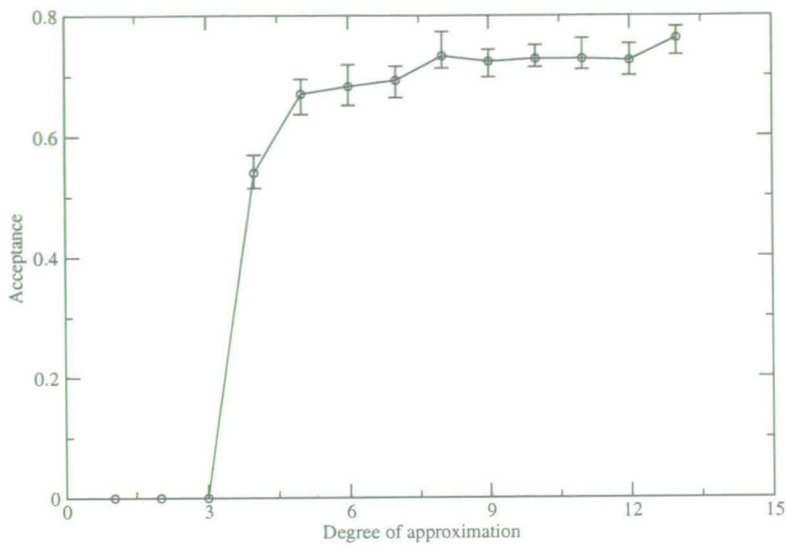


Figure 4.5: A plot showing the variation of the acceptance with  $n_{\text{md}}$ . The parameters and results are given in tables 4.1 and A.1.

test. This is not the case, the apparent trend in plaquette is merely a reflection of poor statistics, the mean plaquette is not properly estimated. To verify this is the case a long run with  $N_{\text{traj}} = 2000$  using  $n_{\text{md}} = 5$  was performed, this point is included on the plot verifying that the approximation errors are corrected for.

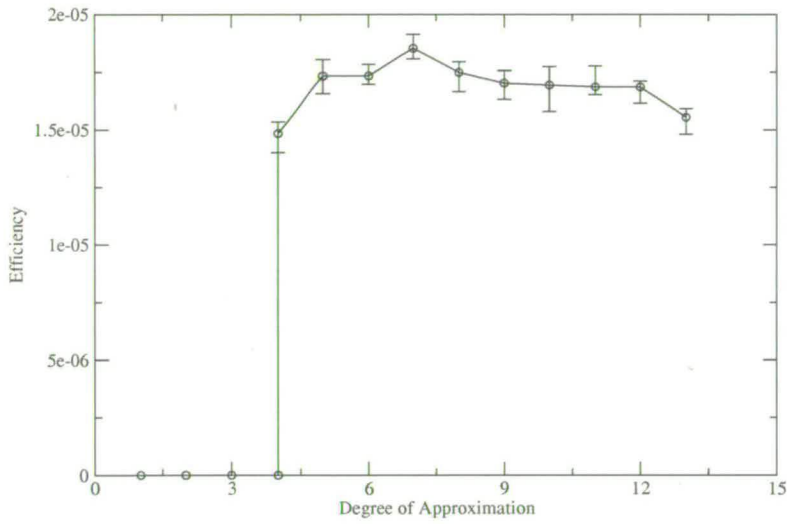


Figure 4.6: A plot showing the variation of the efficiency with  $n_{\text{md}}$ . The parameters and results are given in tables 4.1 and A.1.

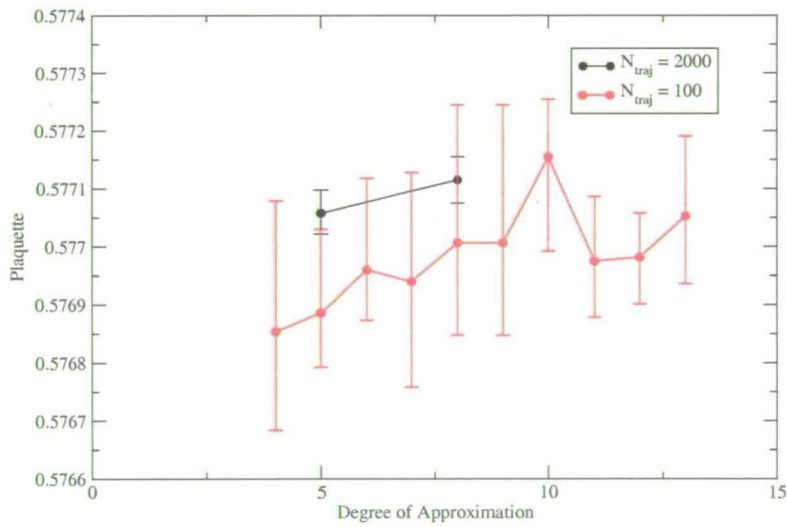


Figure 4.7: A plot showing the dependence of the RHMC plaquette when varying  $n_{\text{md}}$ . When the statistics are low (red line) there is an apparent bias in the data which favours a lower plaquette. As the statistics increase, this apparent bias is removed (black line). The parameters and results are given in tables 4.1 and A.1.

It is obviously also important to check the scaling of observables with stepsize. Figure 4.8 is a plot revealing the stepsize independence of the algorithm. Since the blue data points are only short runs at 100 trajectories each, the error bars are large, however, it can be seen that there is consistency with the  $\delta\tau = 0.02, 0.04, 2000$  trajectory runs. The R algorithm  $\delta\tau$  scaling results from [47] are included on the plot. It is clear that RHMC is able to use an integrating stepsize greater than the R algorithm because of the stepsize errors present in the latter. For consistency between the exact result of RHMC and PHMC the R algorithm stepsize must be set to  $\delta\tau = 0.005$ . When compared to the optimum RHMC stepsize, this corresponds to at least a factor of two (four if multiple time steps are used) increase in matrix-vector products.

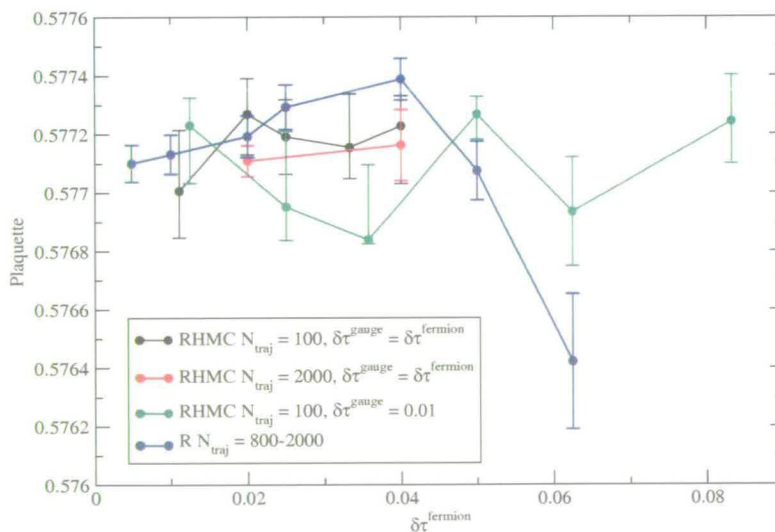


Figure 4.8: A plot showing the independence of the RHMC plaquette with stepsize for the runs with and without multiple length time scales. The stepsize scaling of the R algorithm comes from [47]. The parameters and results are given in tables 4.1 and A.2.

The variation of  $\delta H$  with stepsize is shown in figure 4.9. Included on the plot are also results from using the multiple time scale Sexton-Weingarten integrator, where the gauge integrating stepsize has been kept constant at  $\delta\tau \approx 0.01$ . The



advantage of using multiple time scales is very clear in this case, the breakdown in the integrator does not occur until a much greater integrating stepsize is used. This behaviour of course translates into better acceptance rate scaling with stepsize (figure 4.10). Figure 4.11 reveals that the result of using the multiple length time scales is to increase the peak efficiency of the algorithm by more than a factor of two. The conclusion is that for these parameters chosen, the gauge force is still the dominant contribution, and hence the use of multiple length time scales is extremely beneficial.

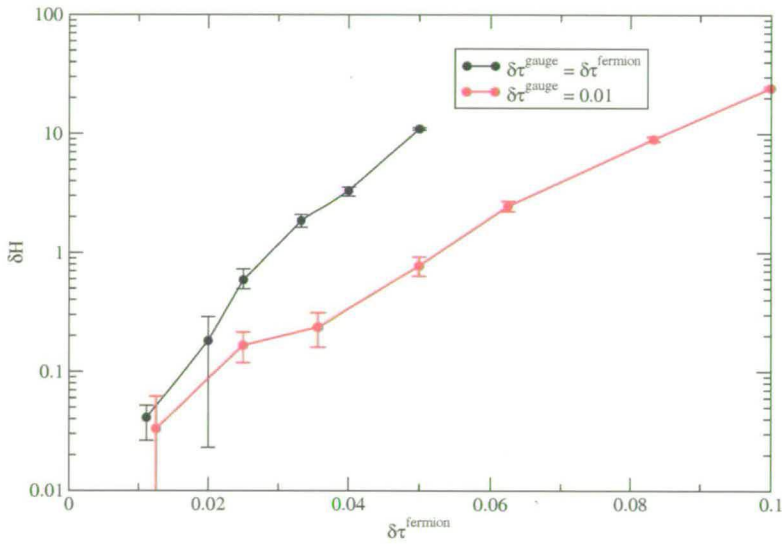


Figure 4.9: A plot showing the dependence of the RHMC  $\delta H$  with stepsize, with and without multiple length time scales. The parameters and results are given in tables 4.1 and A.2.

This comparison against published data thus proves that RHMC is a valid and efficient algorithm with which to perform arbitrary flavour dynamical fermion QCD calculations using staggered fermions. At these parameters RHMC has been shown to be significantly more efficient than PHMC. When compared against the R algorithm, RHMC has been demonstrated to be much more efficient than the R algorithm because it allows the use of a far greater stepsize.

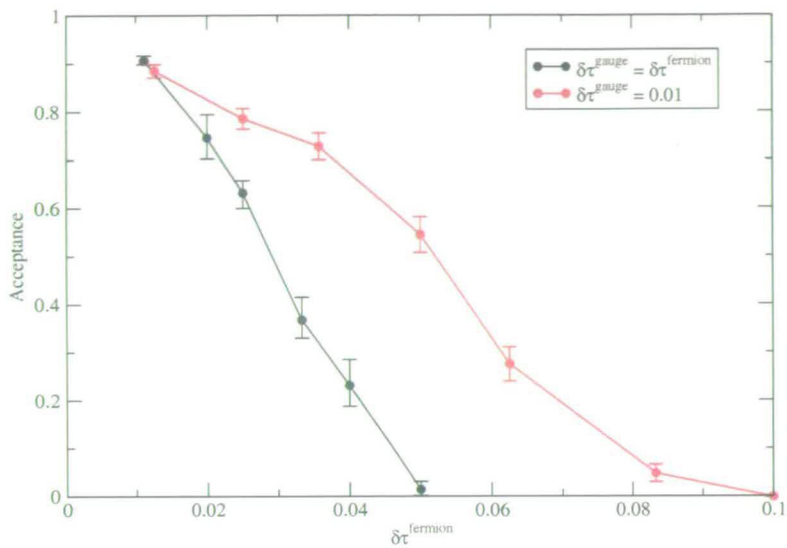


Figure 4.10: A plot showing the dependence of the RHMC acceptance with stepsize with and without multiple length time scales. The parameters and results are given in tables 4.1 and A.2.

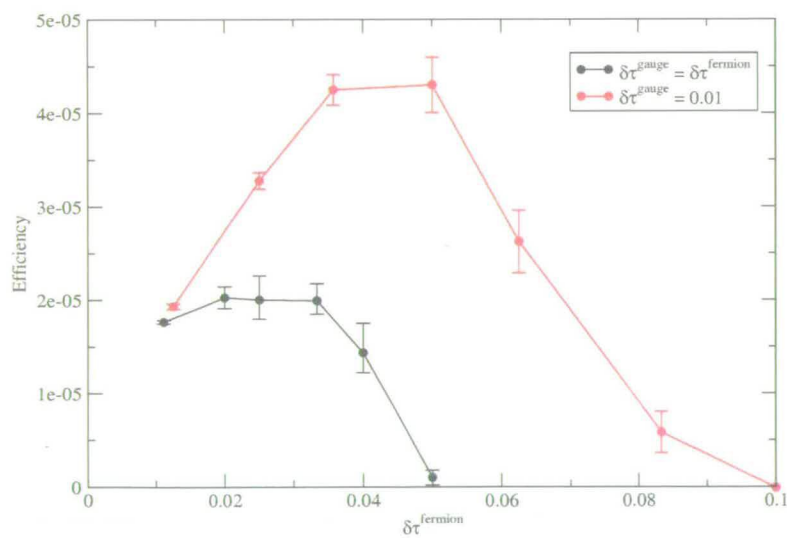


Figure 4.11: A plot showing the dependence of the RHMC efficiency with stepsize with and without multiple length time scales. The parameters and results are given in tables 4.1 and A.2.

# Chapter 5

## Extensions to the Multi-Shift solver

The multi-shift solver as given by Jegerlehner [55] is restricted compared to more conventional solvers in the sense that a zero initial guess must be used for the algorithm to work correctly. The reason for this can easily be seen by realising that the Krylov subspace of the initial residuals must coincide exactly. Consider the case where a multi-shift solver is used to solve the following two matrix equations,

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

$$(A + \sigma)x^\sigma = b. \tag{5.2}$$

where the shift  $\sigma$  is a multiple of the identity. The initial residuals of these systems are given by

$$r = b - Ax_0, \tag{5.3}$$

$$r^\sigma = b - (A + \sigma)x_0^\sigma \tag{5.4}$$

respectively, where  $x_0$  and  $x_0^\sigma$  are the respective initial guesses. It can be seen instantly that these residuals will coincide if a zero initial guess is used, and

hence the regular multi-shift solver can be used. Is it possible to start with an initial guess for either/or both of these systems, and still be able to use the multi-shift solver? If possible this could open up the use of a chronological inverter [56] in RHMC, giving another potential benefit over the R algorithm, where an initial guess cannot be used because a new fermion vector is created every MD step. Being able to start a multi-shift solve with an initial guess would also have important consequences for overlap fermion calculations, where a rational approximation to the  $\text{sgn}$  function is evaluated using a multi-shift solver.

## 5.1 A Chronological Multi-Shift Solver?

For the following the analysis, the following parameterisation shall be useful, where the initial guess constructed satisfies the matrix equation

$$A'x_0 = b, \quad (5.5)$$

where

$$A' = \sum_{k=1}^n c_k A(U_k) \quad (5.6)$$

$$= A + \delta A. \quad (5.7)$$

The matrix  $\delta A$  cannot be represented by a Dirac operator with a gauge field argument since in general the lattice Dirac operator  $M^\dagger M$  will not be a linear function of the gauge fields.

From the requirement that the initial residuals be parallel, from equations 5.3 and 5.4 it is clear that the following condition must be met

$$Ax_0 = (A + \sigma)x_0^\sigma, \quad (5.8)$$

which can in itself only be directly solved by an explicit matrix inversion.



The initial guess must be optimal for the unshifted system, since this will take the greatest number of iterations to solve. Using the optimal chronological guess will give a residual

$$r = b - Ax_0 \quad (5.9)$$

$$= b - A(A')^{-1}b \quad (5.10)$$

$$\approx \delta AA^{-1}b, \quad (5.11)$$

where it has been assumed that  $A \gg \delta A$ , which must be true else the initial guess would not be useful. When doing the shifted guess there are a number of possible routes that can be taken. If the same coefficients  $c_k$  are used for the shifted history vectors, the effective matrix equation that the initial guess will satisfy is

$$(A' + \sigma)x_0^\sigma = b. \quad (5.12)$$

Hence the initial residual will be

$$r^\sigma = b - (A + \sigma)(A' + \sigma)^{-1}b \quad (5.13)$$

$$\approx \delta A(A + \sigma)^{-1}b. \quad (5.14)$$

An alternative approach for the shifted guess is to use the same initial guess as used for the unshifted case. The shifted residual is then given by

$$r^\sigma = b - (A + \sigma)x_0 \quad (5.15)$$

$$= r - \sigma(A + \delta A)^{-1}b. \quad (5.16)$$

Unfortunately, neither of the shifted residuals in equations (5.14) and (5.16) are equal to the unshifted residual in equation (5.11). The conclusion to this investigation therefore is that the use of an initial guess cannot currently be used to reduce the number of matrix-vector products over the course of an RHMC trajectory.

## 5.2 The Generalised Multi-Shift Solver

Take the following two systems

$$Ax = b, \quad (5.17)$$

$$(A + \sigma)x^\sigma = b^\sigma, \quad (5.18)$$

where  $b^\sigma$  is independent of  $b$ . Their residuals are given by

$$r = b - Ax_0, \quad (5.19)$$

$$r^\sigma = b^\sigma - (A + \sigma)x_0^\sigma \quad (5.20)$$

respectively. Trivially the vectors  $r$  and  $r^\sigma$  can be made to coincide by the requirement that

$$x_0^\sigma = x_0 \quad (5.21)$$

and

$$b = b^\sigma - \sigma x_0^\sigma. \quad (5.22)$$

This suggests that given the systems of equations in equations (5.17) and (5.18), then the solutions  $x$  and  $x^\sigma$  can be found simultaneously through choosing the initial guess such that

$$x_0 = \frac{1}{\sigma}(b^\sigma - b). \quad (5.23)$$

Hence the multi-shift solver can be seen to solve systems of the following general form

$$(A + \sigma)x_0^\sigma = b^\sigma \quad (5.24)$$

$$= b + \sigma x_0. \quad (5.25)$$

If a zero initial guess is used, then the conventional multi-shift solver with the same right hand side for all shifts  $\sigma$  is obtained. With a non-zero initial guess, then a  $\sigma$  dependent right hand side is the result. Only two independent right

hand sides can be chosen, since any further right hand sides are dependent on the choice of the initial guess which is determined by the first two right hand sides. However, the question is now, is this more generalised form actually any use?

## 5.3 The R2 Algorithm

One potential use for this form of the multi-shift solver is for the R algorithm when doing 2+1 quark flavours of staggered fermions. At each MD step two matrix inverses must be evaluated, one for the light pair and one for the strange quark.

$$(-\not{D}^2 + 4m_1^2)X_l = \phi_l \quad (5.26)$$

and

$$(-\not{D}^2 + 4m_s^2)X_s = \phi_s. \quad (5.27)$$

Using an initial guess of  $X_0 = (m_s^2 - m_l^2)^{-1}(\phi_s - \phi_l)$ , the generalised multi-shift solver can be used to solve both of these systems simultaneously. In such a situation the strange quark inversion would be completed before the light pair, so by using this inversion process the strange quark inversion is obtained for free. This is particularly well suited to the R algorithm because there is no means of making any intelligent initial guess to the Krylov solver, so one guess is as good as any another. The case when the R algorithm uses this generalised multi-shift solver to reduce the number of matrix-vector operations shall from now on be referred to as the R2 algorithm.

Figure 5.1 is a plot of the total number of conjugate gradient iterations per trajectory for the R and R2 algorithm using naïve staggered quarks on a  $4^4$  lattice. As can be seen, for these parameters there is a saving of around 30% in the number of iterations per trajectory. However, as a more physical pa-

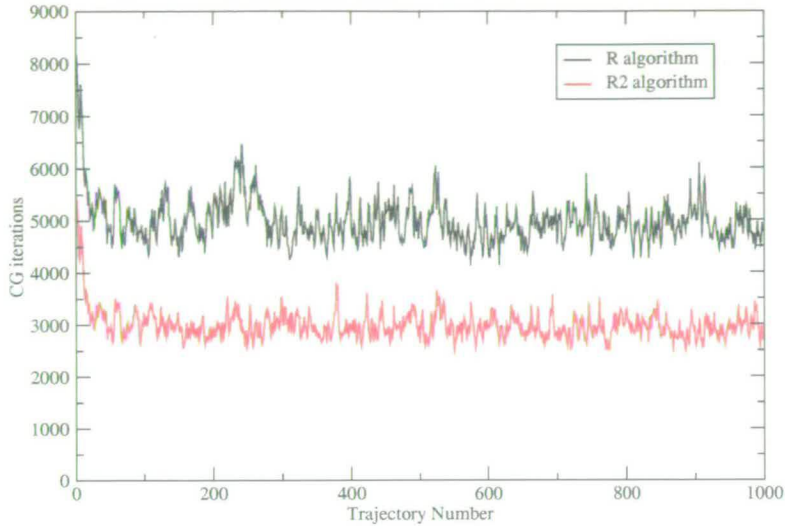


Figure 5.1: CG iterations per trajectory for  $N_f = 2+1$ ,  $4^4$  lattice, Wilson gauge action,  $\beta = 5.6$ ,  $m_l = 0.05$ ,  $m_s = 0.2$ ,  $\tau = 1.0$ ,  $\delta\tau = 0.04$ .

parameter regime is approached, i.e., a lighter light fermion pair, this gain would be reduced. Figure 5.2 is a plot of the plaquette history for the same evolution. It can be seen that after around 100 trajectories the plaquettes diverge between the algorithms (figure 5.3). This is to be expected because the solutions produced will be slightly different between the algorithms because of the different initial guess. Although the difference will be small per inversion, i.e., of magnitude no greater than the tolerance of the residual to which the inverter converges, the difference gradually increases as the lattice is evolved until there is no correlation between the two evolved gauge fields. A similar divergence can be witnessed when comparing the same code running on different architectures because of the variation in floating point precision. Hence both plaquette histories are equally valid, as is suggested by the fact that the mean value of the histories are compatible. The loss of correlation can be seen in figure 5.3, which is a plot of the difference between the plaquettes obtained using the two algorithms.

This technique is of no use when doing 2+1 flavour staggered fermions simula-



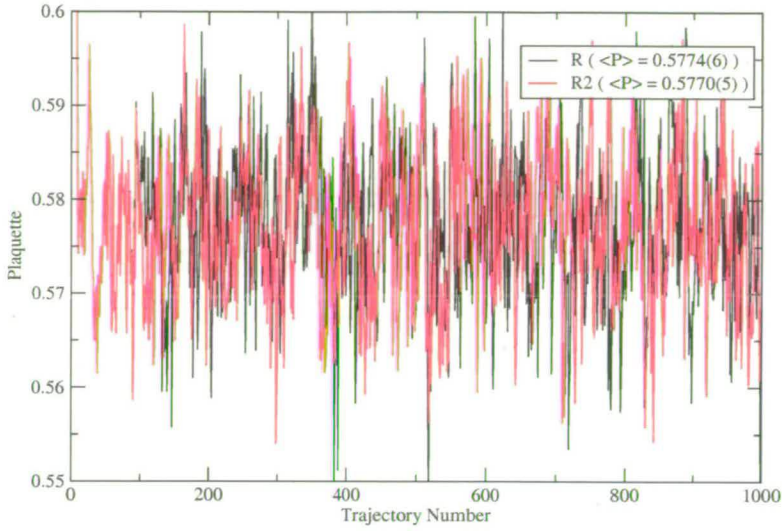


Figure 5.2: Plaquequette history for  $N_f = 2 + 1$ ,  $4^4$  lattice, Wilson gauge action,  $\beta = 5.6$ ,  $m_l = 0.05$ ,  $m_s = 0.2$ ,  $\tau = 1.0$ ,  $\delta\tau = 0.04$ .

tions using RHMC, since here the conventional multi-shift solver is already being used for each field's inversion. It is also equally useless for doing 2+1 flavour simulations using domain wall fermions with the R algorithm, the failing being due to the mass term not being a multiple of the identity matrix.

One possible use that the generalised multi-shift solver might have is with overlap calculations. When performing dynamical overlap calculations, it may be beneficial to use the Hasenbusch trick [61, 62, 63, 64] to accelerate the fermionic dynamics. In such a circumstance it would be required that two inversions per MD step would be required (see equation (6.2)), where the right hand sides are different, and respective matrices being different by a constant shift, i.e., exactly the scenario where the generalised multi-shift solver can be put to use.

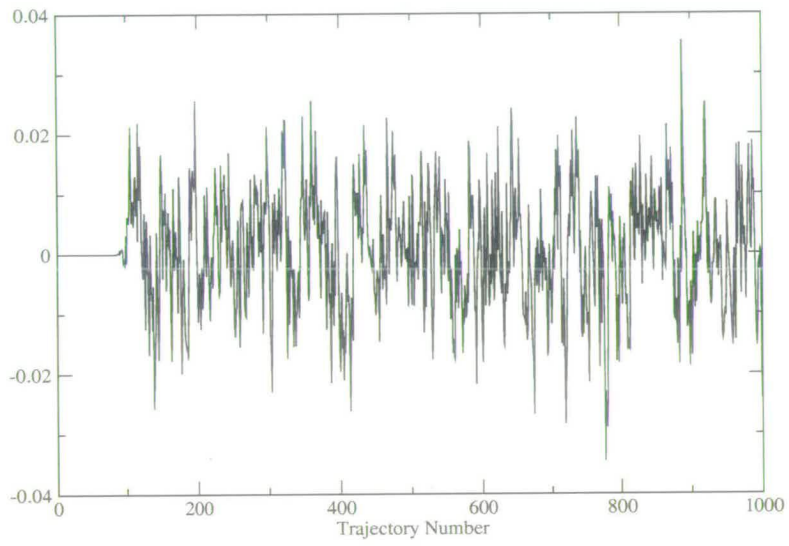


Figure 5.3: Difference in plaquette  $P_R - P_{R2}$ ,  $N_f = 2 + 1$ ,  $4^4$  lattice, Wilson gauge action,  $\beta = 5.6$ ,  $m_l = 0.05$ ,  $m_s = 0.2$ ,  $\tau = 1.0$ ,  $\delta\tau = 0.04$ .

## Chapter 6

# Monte Carlo Acceleration

The biggest failing of HMC [33] is the rapid blow up in cost of the algorithm as the fermion mass is decreased. This reason for this is twofold: firstly because of the increase in condition number of the fermion matrix, Krylov solvers take many more iterations to converge within some tolerance. Secondly, in order to keep the HMC acceptance rate  $P_{\text{acc}}$  constant the MD integration step size  $\delta\tau$  has to be reduced, and for MD trajectories of length  $\tau = 1$  (or more precisely proportional to the correlation length of the system [65, 66], which is  $O(1)$  for present-day computations) this corresponds directly to an increased number of MD integration steps and hence cost. This section addresses the latter of these problems.

It took some time for the underlying cause of this dependence of step size on fermion mass to be understood: the phenomenon showed itself most clearly in the breakdown of reversibility of symmetric symplectic integrators in finite precision floating-point arithmetic. For pure gauge systems, or systems with heavy dynamical fermions, the violation of reversibility was traced to the exponential amplification of rounding errors due to the chaotic nature of the underlying

equations of motion [67, 68, 69]. For light dynamical fermions another cause of such exponential amplification of rounding errors becomes predominant, namely the inherent instability of the symplectic integration scheme itself [70]. This instability is also seen to be directly responsible for the exponential decrease of HMC acceptance with integration step size above some critical value.

The instability is easily understood from a simple free field model [68] in which the fermionic contribution to the gauge action is replaced by a set of modes with highest “effective frequency”  $\omega$ . In this model, when  $\omega \delta\tau$  exceeds some critical value the symplectic integrator ceases to even approximately conserve energy, and the energy starts to diverge exponentially. In realistic interacting quantum field theories, such as QCD, it is most likely that this instability occurs for a few isolated light fermion modes, whose frequency is well separated from the bulk of the modes.

## 6.1 Hasenbusch’s Method and Variants

There has been much progress in recent years in reducing the severity of this problem, by reducing the highest “effective frequency” of the fermionic modes, or equivalently of decreasing the magnitude of the fermionic contribution to the force acting on the gauge fields. The basic idea of Hasenbusch is to split the fermionic action into two parts, and to introduce separate pseudofermion fields for each part [61, 62, 63, 64]

$$\det M^\dagger M = \det (M'^\dagger M') \det ((M'^\dagger)^{-1} (M^\dagger M) M'^{-1}) \quad (6.1)$$

$$= \int \mathcal{D}\phi^\dagger \mathcal{D}\phi \mathcal{D}\phi'^\dagger \mathcal{D}\phi' \exp(-\phi^\dagger M' (M^\dagger M)^{-1} M'^\dagger \phi - \phi'^\dagger (M'^\dagger M')^{-1} \phi'). \quad (6.2)$$

Now two pseudofermion fields  $\phi, \phi'$  are used to approximate the fermionic determinant. If the mass parameter of the “dummy” Dirac matrix  $M'$  is substantially



larger than the desired mass, then the algorithmic cost of the second term in the action will be small compared to the first. The cost of the first term in the action will be unchanged compared to the standard formulation except for an additional inverse required for the heatbath evaluation. This reformulation of the fermionic determinant improves the acceptance rate significantly, and the net effect is an overall reduction in algorithmic cost.

There are several ways of understanding why introducing multiple pseudofermion fields might reduce the fermionic force. The first is to observe that the force due to the fermion kernel  $\mathcal{M}^{-1}$  is dominated by the smallest eigenvalues of  $\mathcal{M}$ . The condition number  $\mathcal{K}(\mathcal{M})$  is the ratio of the largest eigenvalue to the smallest eigenvalue, and to a first approximation controls the rate of convergence of iterative Krylov space solvers [71]. As the largest eigenvalue remains approximately constant as the fermion mass  $m_q$  is decreased, and the smallest eigenvalue is typically of the order  $m_q^\alpha$  where  $\alpha$  is 1 or 2, it is expected that  $\mathcal{K}(\mathcal{M}) \propto m_q^{-\alpha}$ . The condition number can be thought of as a gross measure of the magnitude of the fermion force. The reformulation of the fermionic determinant therefore improves the acceptance rate because the condition number of the effective matrices that appear in the action,  $\mathcal{M}'\mathcal{M}^{-1}$  and  $\mathcal{M}'^{-1}$ , are significantly lower than in the standard formulation with just  $\mathcal{M}^{-1}$ . Unfortunately, although the matrix  $\mathcal{M}'\mathcal{M}^{-1}$  is better conditioned than just the matrix  $\mathcal{M}^{-1}$ , there is no way to directly evaluate this matrix directly using a Krylov method to take advantage of the improved condition number.

Another way of understanding the need for multiple pseudofermion fields is to recall that in conventional HMC the fermion determinant is represented as a pseudofermion Gaussian functional integral,  $\det \mathcal{M} \propto \int d\phi d\phi^\dagger \exp(-\phi^\dagger \mathcal{M}^{-1} \phi)$ , and a single equilibrium pseudofermion configuration using a Gaussian heatbath is used [32]. It should be expected, therefore, that the variance of this stochastic estimate of the fermion determinant will lead to statistical fluctuations in the

fermionic force: in other words the pseudofermionic force may be larger than the exact fermionic force, which is the functional derivative  $d \operatorname{tr} \ln \mathcal{M}(U)/dU$  with respect to the gauge field  $U$ . This means that the pseudofermionic force may trigger the instability in the symplectic integrator even when the exact fermionic force would not.

An obvious way of ameliorating this effect is to use  $n > 1$  pseudofermion fields to sample the functional integral representing the fermion determinant, which is exactly what the Hasenbusch trick is doing<sup>1</sup>.

This process can be extended to an arbitrary number of fields, with increasingly larger masses for each extra field, leading to potentially larger improvements [72]. Unfortunately this requires fine tuning of the mass parameters, and there is no *a priori* method for calculating these. Since each additional field increases the parameter space by one dimension, empirically calculating the optimum mass parameters becomes a time consuming and (computationally) costly exercise. When properly tuned, the use of multiple psuedo-fermions is found to give an acceleration of 2-6 over the standard single pseudo-fermion formulation.

Multiple pseudofermion fields can also accelerate the performance of inexact algorithms that do not include a Metropolis acceptance test, however, how one quantifies the cost reduction is a non-trivial problem. Multiple-pseudofermion fields were used in [44] as a means of algorithm testing, and it clear from these results that multiple-pseudofermion fields reduce the stepsize dependence of observables.

---

<sup>1</sup>Even in the infinite pseudofermion limit, the pseudofermionic force will not equal the exact fermionic force because each pseudofermion is held fixed throughout the trajectory.

## 6.2 Nroots Acceleration

Another method of introducing multiple pseudofermions is to write

$$\det \mathcal{M} = [\det \mathcal{M}^{1/n}]^n \quad (6.3)$$

$$\propto \prod_{j=1}^n d\phi_j d\phi_j^\dagger \exp \left( -\phi_j^\dagger \mathcal{M}^{-1/n} \phi_j \right), \quad (6.4)$$

that is, introducing  $n$  pseudofermion fields  $\phi_j$  each with kernel  $\mathcal{M}^{-1/n}$ . This has the obvious advantage over Hasenbusch's trick that there are no parameters to tune. Fortunately the “technology” for dealing with non-local actions of the form  $\mathcal{M}^{-1/n}$  is exactly what RHMC is designed for. The usual pseudofermion action  $\phi^\dagger \mathcal{M}(U)^{-1} \phi$  is thus replaced with a sum over  $n$  pseudofermions  $\sum_{i=1}^n \phi_i^\dagger \mathcal{M}(U)^{-1/n} \phi_i$ , which are dealt with using RHMC.

Extending the previous condition number argument, the force is expected to be of order  $n\mathcal{K}(\mathcal{M})^{1/n}\delta\tau$  (or very optimistically perhaps  $\sqrt{n}\mathcal{K}(\mathcal{M})^{1/n}\delta\tau$  if the different pseudofermion contributions were to add up incoherently), which is small compared to  $\mathcal{K}(\mathcal{M})\delta\tau$  for large  $\mathcal{K}(\mathcal{M})$ . The maximum force must be kept fixed so as to avoid the instability in the integrator, so the integration step size may be increased to  $\delta\tau'$  such that

$$n\mathcal{K}(\mathcal{M})^{1/n}\delta\tau' = \mathcal{K}(\mathcal{M})\delta\tau. \quad (6.5)$$

The cost of an RHMC trajectory using  $n$  pseudofermions is proportional to  $n\frac{\tau}{\delta\tau'}$ , thus the cost relative to a single pseudofermion formulation can be estimated by the ratio

$$\text{Cost} = n \frac{\tau/\delta\tau'}{\tau/\delta\tau} \quad (6.6)$$

$$= n^2 \mathcal{K}(\mathcal{M})^{\frac{1}{n}-1}. \quad (6.7)$$

The minimum of this function is

$$n_{\text{opt}} = \frac{1}{2} \ln \mathcal{K}(\mathcal{M}), \quad (6.8)$$



and should correspond to the optimum number of pseudofermions <sup>2</sup>.

### 6.3 Results

The parameters that were used to test RHMC in § 4.5 were again used for this analysis (see table 4.1 for a recap). For this investigation, the multiple time step integrator was used, and the gauge integrating stepsize was kept constant at  $\delta\tau \approx 0.01$ . Apart from increasing the overall efficiency of the simulation, this also ensures that the acceptance rate behaviour is dominated by the fermion stepsize, so a true measure of the effectiveness of nroots can be obtained.

The staggered formulation is such that the condition number of the fermion matrix is analytically known, given the fermion mass. With this information, a prediction can be made about the optimal number of pseudofermions that should be used. For a staggered mass parameter of  $m = 0.02$ , the resulting condition number should be

$$\mathcal{K}(\mathcal{M}) = \frac{64 + 4m^2}{4m^2}. \quad (6.9)$$

In reality this upper bound is far too large because the upper bound is the infinite volume result, and an upper bound of around 20 is actually measured. This equates to an  $n_{\text{opt}} = 4.7$  as given by equation (6.8).

The tabulated results from this section can be seen in appendix B. Figure 6.1 is a plot of the variation of  $\delta H$  with stepsize for  $n = 1, 2, 3, 4$  pseudofermions. The benefit of using nroots can clearly be observed. At small  $\delta\tau$  there is a decrease in  $\delta H$  when using multiple pseudofermion fields compared to using only a single field, however, there appears to be no gain at all in using  $n > 2$ . As  $\delta\tau$  is increased, the differences between using the different numbers of fields becomes

---

<sup>2</sup>Strictly speaking  $n_{\text{opt}} \in \mathbf{Z}$ , so it must be either  $\lfloor \frac{1}{2} \ln \mathcal{K}(\mathcal{M}) \rfloor$  or  $\lceil \frac{1}{2} \ln \mathcal{K}(\mathcal{M}) \rceil$ .



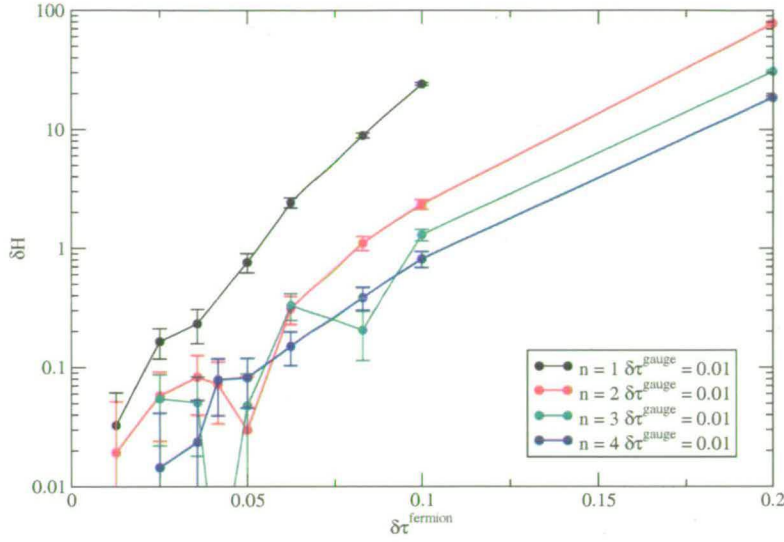


Figure 6.1: The variation of  $\delta H$  with stepsize for  $n = 1, 2, 3, 4$ ,  $m = 0.02$  (see tables B.1 - B.4).

more apparent, and there is clear improvement as  $n$  is increased. This improved conservation of  $\delta H$  is of course reflected in improved acceptance rates, as can be seen in figure 6.2. What can also be noted is that the improvement factor in moving from  $n = 1$  to  $n = 2$  seems to be far greater than the improvement in moving  $n = 2$  to  $n = 3$  or  $n = 3$  to  $n = 4$ . This is to be expected if the magnitude of the force is described by  $n\mathcal{K}(\mathcal{M})^{1/n}\delta\tau$ .

Although the acceptance rates may have improved significantly through the use of nroots, this does not necessarily mean that the use of multiple pseudo-fermions is more efficient. Figure 6.3 is a plot of the efficiency as defined in equation (4.23). Although the stepsize can be increased to maintain a constant acceptance rate, the increase is not enough to warrant the use of nroots. At these parameters, it is therefore concluded that although nroots allows the stepsize to be increased maintaining a constant acceptance rate, the increase does not result in a net gain. However, the expectation is that nroots will be especially beneficial as the condition number of fermion matrix is increased.

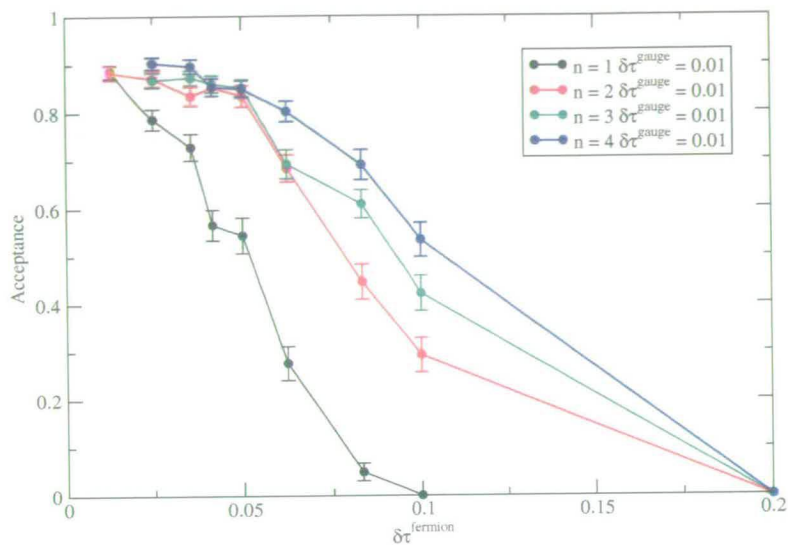


Figure 6.2: The variation of acceptance with stepsize for  $n = 1, 2, 3, 4$ ,  $m = 0.02$  (see tables B.1 - B.4).

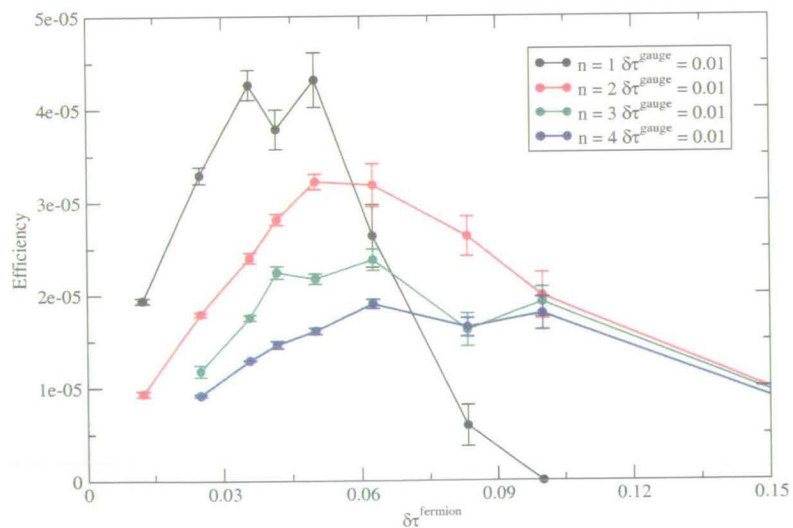


Figure 6.3: The variation of efficiency with stepsize for  $n = 1, 2, 3, 4$ ,  $m = 0.02$  (see tables B.1 - B.4).

As a second test of nroots acceleration, the same parameters were used again, with the exception that now the quark mass was reduced by a factor of two,  $m = 0.01$ . Figure 6.4 is a plot of  $\delta H$  for  $n = 1, 2, 3$  pseudofermion fields. The  $n = 1$  data is interesting, as it appears to reveal a breakdown in the integrator at  $\delta\tau = 0.09$ , where  $\delta H$  blows up by about a factor  $10^3$ . This is in agreement with [70], where the breakdown in the integrator can be explained by an exponential amplification of errors. There is no such breakdown, on the scale plotted for  $n = 2, 3$ . Again at small  $\delta\tau$  there seems to be little to differentiate  $n = 2$  and  $n = 3$ , and as  $\delta\tau$  is increased, the gain that  $n = 3$  has over  $n = 2$  is small. The plot of the resulting acceptance rate behaviour is shown in figure 6.5, where the superior scaling of nroots can be seen. Comparing this plot and that of figure 6.2, it can be seen that nroots shows more improvement relative to the single pseudofermion case for the light mass case. Unfortunately, even with this decrease in the quark mass, a single pseudofermion field is still the most efficient formulation (see figure 6.6). The ratio now between the most efficient  $n = 1$  case ( $\delta\tau = 0.0303$ ) and the most efficient  $n = 2$  case ( $\delta\tau = 0.0769$ ) is now much closer to unity. This is suggestive that a further decrease in the quark mass will result in nroots actually being more efficient than the use of a single pseudofermion field. What is interesting to note is that the  $n = 1$  formulation seems to have a much more sharply peaked efficiency distribution, where both the  $n = 2$  and  $n = 3$  formulations are much broader.

Finally the same experiments were carried out but this time using a quark mass parameter  $m = 0.005$  in an attempt to ascertain that nroots can be beneficial. Figure 6.7 is a plot of the stepsize variation of  $\delta H$  from this test. Again there is a breakdown in the integrator for the  $n = 1$  case, which is not reproduced for  $n = 2, 3$ . In fact, the breakdown appears not to occur at all when using nroots, however, this is most likely because too small a  $\delta\tau$  has been used to observe this breakdown. From figure 6.9 it is evident that the use of nroots

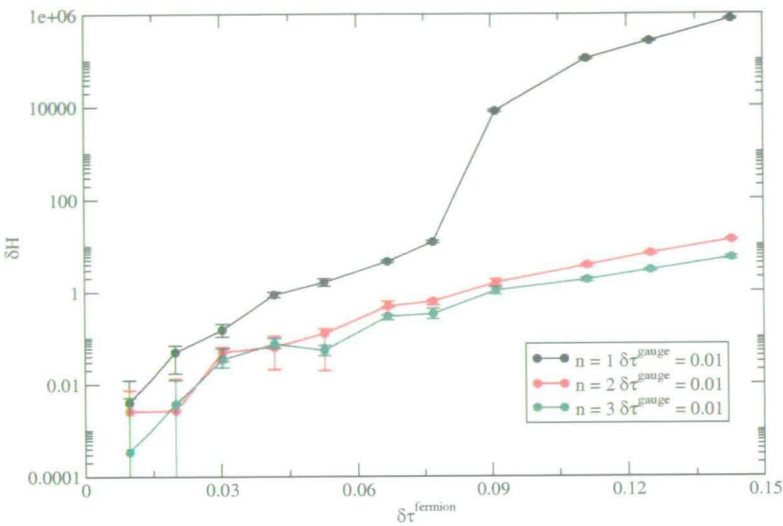


Figure 6.4: The variation of  $\delta H$  with stepsize for  $n = 1, 2, 3$ ,  $m = 0.01$  (see tables B.5 - B.7).

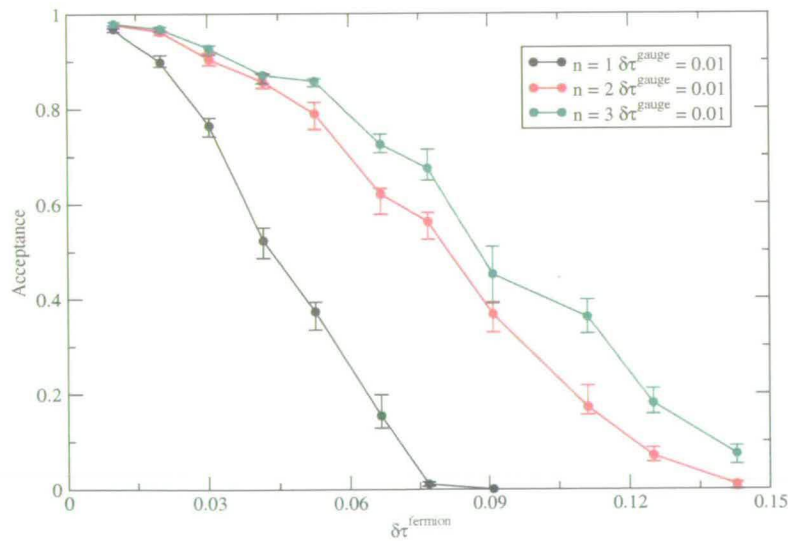


Figure 6.5: The variation of acceptance with stepsize for  $n = 1, 2, 3$ ,  $m = 0.01$  (see tables B.5 - B.7).



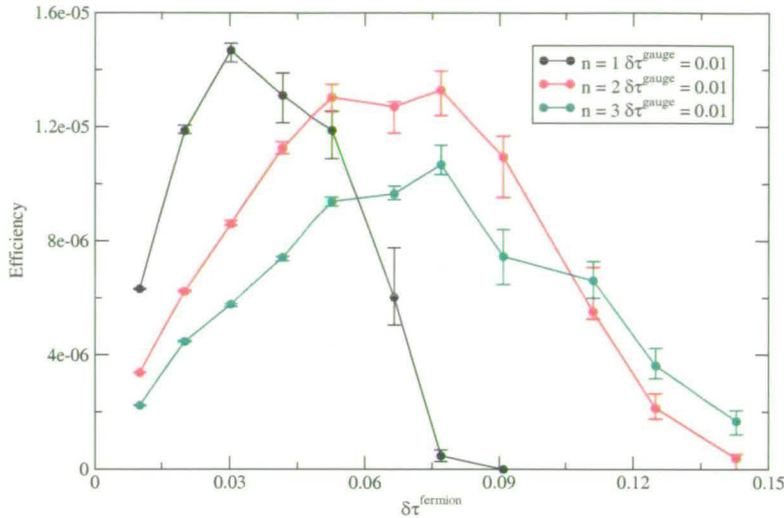


Figure 6.6: The variation of efficiency with stepsize for  $n = 1, 2, 3$ ,  $m = 0.01$  (see tables B.5 - B.7).

is beneficial. Using 2 pseudofermion fields is 13% more efficient than using a single field, when comparing optimum efficiency points.

In all three of the tests conducted here, the optimal number of pseudofermions predicted by equation (6.8) (4.7, 5.4, 6.1) has been differed significantly from the actual value (1, 1, and 2 respectively). The reason for this is probably due to the presumption in equation (6.8) that the fermionic force magnitude can be approximated by the condition number  $\mathcal{K}(\mathcal{M})$ . This approximation is likely too crude, and the force is better described by taking into account the bulk distribution of eigenvalues, not just the lowest one. For Wilson fermions, where the lowest eigenvalue is unbounded from below, associating the force magnitude with the condition number is likely to be a better approximation than for the staggered fermion example here.

This section has demonstrated an alternative method to the Hasenbusch trick for accelerating fermionic dynamics. This method is only beneficial when an extremely light staggered quark mass is used.

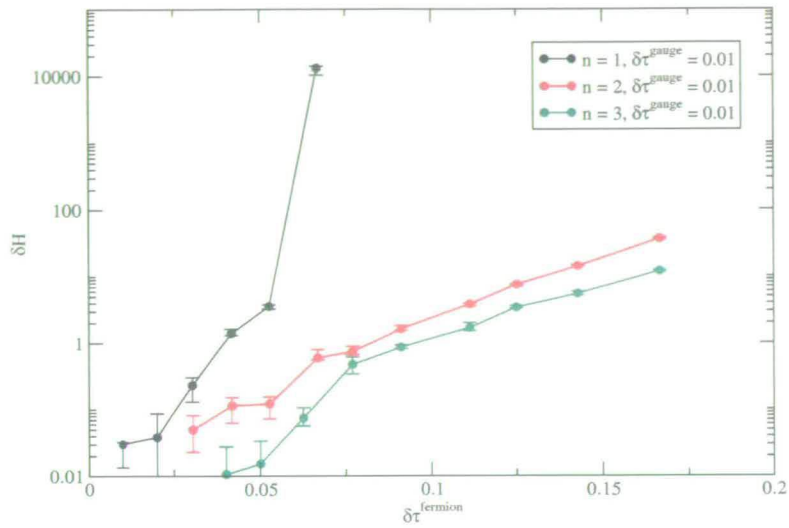


Figure 6.7: The variation of  $\delta H$  with stepsize for  $n = 1, 2, 3$ ,  $m = 0.005$  (see tables B.8 - B.10).

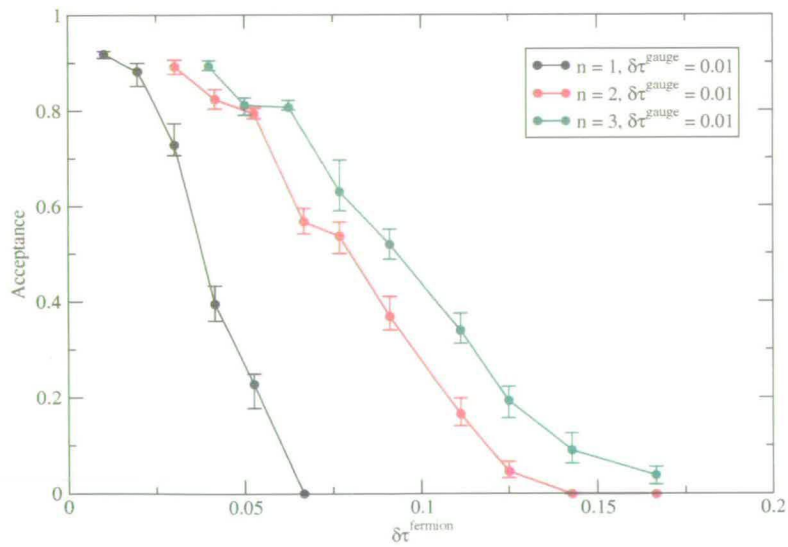


Figure 6.8: The variation of acceptance with stepsize for  $n = 1, 2, 3$ ,  $m = 0.005$  (see tables B.8 - B.10).

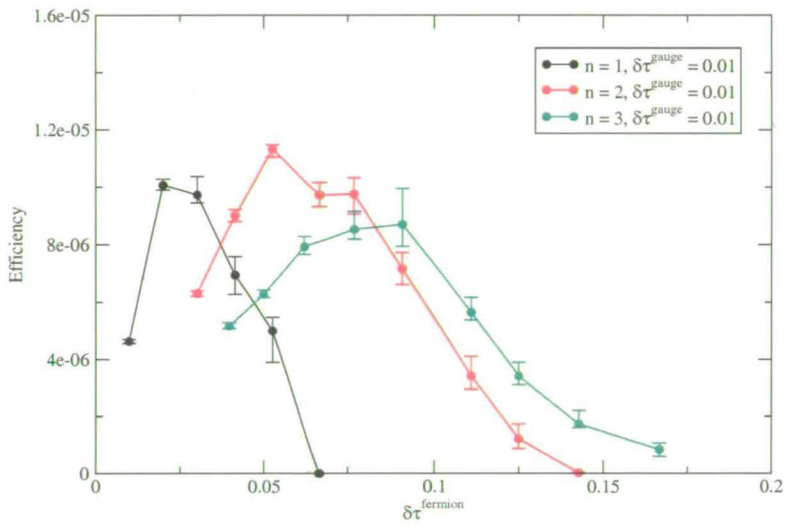


Figure 6.9: The variation of efficiency with stepsize for  $n = 1, 2, 3$ ,  $m = 0.005$  (see tables B.8 - B.10).





# Chapter 7

## Exact 2+1 Flavour Asqtad Fermions

Currently, ASQTAD fermions are an extremely popular choice of fermion theory, mainly because of their comparative cheapness compared to chiral fermion formulations. All large scale ASQTAD calculations to date have been performed using the R algorithm [36, 44], and so a possible question mark lies over this data, as it is unclear as to how much the finite stepsize error is affecting these results. The method advocated in [48] is to choose the stepsize as  $\delta\tau = \frac{2}{3}m_1$ , with  $m_1$  the light quark mass. The stepsize error of the R algorithm has been shown to affect the location of the QCD chiral phase transition, and it has been observed that this stepsize error is large when using the “two thirds” rule [73]. Indeed, the results in § 4.5 demonstrated that a stepsize of  $m_1/4$  was required to obtain an answer compatible with the exact result. A question mark remains over the anomalously long autocorrelation length of the topological charge [26], it has been postulated that this behaviour could be due to the finite stepsize

error [74]<sup>1</sup>. It is therefore imperative that ASQTAD calculations be performed using an exact algorithm to verify that the generated results do not have large systematic errors present. RHMC is the obvious candidate to achieve such an objective.

## 7.1 Algorithm Implementation

### 7.1.1 Action Formulation

The fermionic action is given as follows

$$S_f = \phi_l^\dagger \mathcal{M}_{m_l}^{-1/2} \phi_l + \phi_s^\dagger \mathcal{M}_{m_s}^{-1/4} \phi_s \quad (7.1)$$

$$= \phi_l^\dagger r_l^2 \phi_l + \phi_s^\dagger r_s^2 \phi_s \quad (7.2)$$

Thus two different approximations for the 2 and 1 flavour contributions to the action are required. To achieve an equal maximum error in the approximations, the degree of the approximation used for the light pair will be greater than that of the strange quark because of the difference in condition number.

Note that it is actually possible to formulate the 2+1 ASQTAD quark flavour action using only a single fermion field. Observe

$$\det(\mathcal{M}_{m_l}^{1/2}) \det(\mathcal{M}_{m_s}^{1/4}) = \det(\mathcal{M}_{m_l}^{1/2} \mathcal{M}_{m_s}^{1/4}) \quad (7.3)$$

$$= \det(\mathcal{M}_{m_l}^{1/2} (\mathcal{M}_{m_l} + \delta m^2)^{1/4}) \quad (7.4)$$

$$= \int \mathcal{D}\phi^\dagger \mathcal{D}\phi \exp(-\phi^\dagger (\mathcal{M}_{m_l}^3 + \delta m^2 \mathcal{M}_{m_l}^2)^{-1/4} \phi) \quad (7.5)$$

$$= \int \mathcal{D}\phi^\dagger \mathcal{D}\phi \exp(-\phi^\dagger r^2 (\mathcal{M}_{m_l}) \phi), \quad (7.6)$$

where the parameter  $\delta m^2 = m_s^2 - m_l^2$ . Hence, like the R2 algorithm, a 2+1 quark flavour ASQTAD calculation can be obtained with the cost of only one matrix

---

<sup>1</sup>Essentially, this is because at a stepsize  $\delta\tau \approx m$ , the induced stepsize error is predicted to be large for the long distance physics, but leave the short distance physics intact.

inversion per MD step. Unfortunately, reformulating the action to only a single field is expected to increase the fermionic noise, and would be equivalent to an “inverse Hasenbusch trick” which would reduce the acceptance rate. Since it was found in the chapter 6 that an extremely light mass was required to make the use of nroots worthwhile, this single fermion field formulation may prove to be beneficial for certain parameters.

### 7.1.2 Force Term

The use of an exact algorithm, e.g., PHMC, for performing 2+1 quark flavour ASQTAD simulations has not previously been possible because of the extreme expense of the resulting force term. The ASQTAD operator is given by equation (2.19), i.e., it is composed by the sum of the ordered product of 1, 3, 5 and 7 link matrices. As was shown in § 3.2.1, the derivative of the fermion action with respect to the gauge field must be found to allow the integration of the equations of motion. For ASQTAD fermions, with a rational approximation used to represent the fermion kernel, this requires

$$\frac{dS_f}{dU_{j,\mu}} = \phi^\dagger \frac{dr(\mathcal{M})}{dU_{j,\mu}} \phi \quad (7.7)$$

$$= - \sum_{k=1}^n \alpha_k \phi^\dagger (\mathcal{M} + \beta_k)^{-1} \frac{d\mathcal{M}}{dU_{j,\mu}} (\mathcal{M} + \beta_k)^{-1} \phi \quad (7.8)$$

$$= - \sum_{k=1}^n X_k^\dagger \frac{d\mathcal{M}}{dU_{j,\mu}} X_k, \quad (7.9)$$

$$= - \sum_{k=1}^n \text{tr} \left( M X_k X_k^\dagger \frac{dM^\dagger}{dU_{j,\mu}} + \frac{dM}{dU_{j,\mu}} X_k X_k^\dagger M^\dagger \right), \quad (7.10)$$

where  $X_k = \sqrt{\alpha_k} (\mathcal{M} + \beta_k)^{-1} \phi$ . The exact derivative of the complicated fermion matrix must be calculated explicitly.

The technique used to evaluate the kernel derivative for the R algorithm is to evaluate it applied to the solution vector  $X$  and take the outer product. This

requires operations of the form  $U \dots U X X^\dagger$ , and involves using matrix-vector operations as opposed to matrix-matrix. The resulting number of operations that must be performed to conduct this calculation scale with the volume  $V$  as  $\text{flops}^R = 196,920V$  (this turns out to be similar in computational cost to the conjugate gradient inversion unless a quark mass less than  $m_s/3$  is used [48]).

If this method were applied to RHMC the derivative would have to be recalculated for each solution vector in the partial fraction expansion, leading to an algorithm cost which would be  $O(n)$  times more expensive than the R algorithm. The calculation of the force must be therefore be reformulated to remove/reduce this expense.

The solution of course lies in how the derivative is calculated. When there are just one or two pseudofermion fields  $X_k$ , it is most efficient to use the R algorithm method. The alternative method is to calculate the link matrix products explicitly, and then perform the matrix-vector product and take the outer product. Since the only shift dependence occurs in the solution vector, the link matrix multiplication has to be performed only once, and then each solution vector is multiplied by the resulting matrix, and the outer product taken, i.e.,

$$\sum_{k=1}^n (U \dots U X_k)(X_k^\dagger) = U \dots U \left( \sum_{k=1}^n X_k X_k^\dagger \right). \quad (7.11)$$

The drawback to this approach is that the bulk of the calculation is spent performing the more expensive matrix-matrix operations and not matrix-vector. The total cost of using this method to calculate the derivative scales with the volume  $V$  and the approximation degree  $n$  as  $\text{flops}^{\text{RHMC}} = (782,424 + 720n)V$ . This would imply an approximate fourfold overhead in using RHMC as compared to the R algorithm for the force calculation. However, since all calculations are only concerned with using 2+1 quark flavours, and the only mass dependence in the derivative appears in the vectors, the calculation of the derivative for the



light pair and the strange can be done using the same link product multiplication, reducing the effective overhead by about a factor of two. Also, for the 2+1 case, the R algorithm requires that the ASQTAD operator be constructed three times throughout a single MD step (once for each heatbath, and once for the inversion), compared to only once for inversion with RHMC. Taking these extra two constructions of the Dirac operator into account, and assuming that all other costs are equal (gauge force, fermion matrix inversion), it can be seen that the R algorithm actually has slightly greater flops count of 828,288V. The conclusion therefore is that RHMC can be used with ASQTAD fermions without any large computational detriment. The relative performance of the algorithms thus is purely governed by the acceptance rate of RHMC and how much systematic error in observables can be deemed acceptable in the results generated using the R algorithm.

The RHMC method of calculating the derivative of the ASQTAD operator could also be applied to the PHMC algorithm. With  $n$  expected to be  $O(1000)$  for the light quark mass, this would lead to a force calculation only about a factor of two more expensive than the R algorithm.

## 7.2 Algorithm Testing

It was originally intended that this work include a large scale comparison between 2+1 quark flavour results obtained using the R and RHMC algorithms, in an attempt to address the questions raised at the beginning of this section, namely whether the stepsize errors in current R algorithm data is significant or not. This would have comprised a comparison of physics observables including hadronic observables and topological charge, comparing not just the mean quantities observed, but also comparing the autocorrelation length of these ob-

servables. The parameters that were to be chosen for this comparison can be seen in table 7.1. Such a large volume, with small mass would also have been ideal for testing the performance of nroots acceleration, or conversely whether a more efficient formulation would be obtained by using a single pseudofermion field to simulate the 2+1 quark flavours.

$V$	Gauge Action	$\beta$	$m_l$	$m_s$
$24^3.64$	One loop Symanzik	6.76	0.0078	0.039

Table 7.1: The physical parameter set chosen for ASQTAD algorithm comparison

Unfortunately, due to reasons outwith the author's control, the implementation of the ASQTAD Dirac operator on the QCDOC was found to have problems, resulting in a machine up time of around one hour. Considering that for the desired parameters, the ensemble generation would be a process lasting weeks (with continuous uptime), it was regrettably decided to abandon this study.

## Chapter 8

# Exact 2+1 Flavour Domain Wall Fermions

With the increasing computer power available currently, all future lattice QCD calculations shall without doubt be based on fermion formulations which obey the Ginsparg-Wilson relation. The standard domain wall approach is currently the most popular of these approaches, and shall be the basis of a large bulk of the initial QCDOC calculations.

Before the commencement of this work, 2+1 quark flavour simulations had not been performed with the domain wall formulation. To date the only large scale calculations are those concerning two flavour simulations [57].

### 8.1 Action Formulation

When simulating two flavours of domain wall fermions, it is known that the optimal fermion action is that whereby the kernel is written as the ratio of the Pauli-Villars and light pair contributions, and simulated using a single pseudo-

fermion field [57], i.e.,

$$S_f = \phi_1^\dagger M_{m_{pv}} (M_{m_l}^\dagger M_{m_l})^{-1} M_{m_{pv}}^\dagger \phi_1 \quad (8.1)$$

With this formulation the heavy modes are cancelled out exactly<sup>1</sup>, not stochastically as they would be if the action had been split into pseudofermionic and bosonic pieces. Thus the force is less noisy, and a larger integrating stepsize can be used than would be otherwise.

With this in mind, the optimal 2+1 flavour fermion action should then be given by

$$S_f = \phi_1^\dagger M_{m_{pv}} (M_{m_l}^\dagger M_{m_l})^{-1} M_{m_{pv}}^\dagger \phi_1 + \phi_s^\dagger \sqrt{M_{m_{pv}} (M_{m_s}^\dagger M_{m_s})^{-1} M_{m_{pv}}^\dagger} \phi_s. \quad (8.2)$$

The Pauli-Villars contributions have been combined directly with their respective fermionic partners. The first term in the action is the standard two flavour contribution to the vacuum and the second term is the strange quark contribution. The most naïve implementation would be to replace the fermion matrix in the second term by a rational approximation to the square root over the bounds of this matrix ratio. If this were the case, the strange quark contribution would be

$$S_s = \phi_s^\dagger r(M_{m_{pv}} (M_{m_s}^\dagger M_{m_s})^{-1} M_{m_{pv}}^\dagger) \phi_s. \quad (8.3)$$

This formulation would work, but because of the matrix inverse within the argument of the rational function, whenever the rational function needed to be evaluated each iteration of the multi-shift solver would require a matrix inversion. This renders the formulation extremely expensive, and so an alternative must be found.

An alternative is therefore to separate the Pauli-Villars and fermion contributions into separate fields. The resulting strange quark action is

$$S_s = \phi_{pv}^\dagger \mathcal{M}_{m_{pv}}^{1/2} \phi_{pv} + \phi_s^\dagger \mathcal{M}_{m_s}^{-1/2} \phi_s. \quad (8.4)$$

---

<sup>1</sup>Or at least as good as the tolerance used in the solver used to evaluate such an action.



This form of the action is of course done at the cost of including an extra field in the action, which will require an extra inversion for evaluating the force and the action. However, since the Pauli-Villars field is so heavy compared to the light fermion pair, this extra cost is expected to be negligible. By separating the kernels there is an expected increase in noise in the system because the heavy mode cancellation is no longer exact, which would be expected to have a detrimental effect on the acceptance rate.

In this work, the fermionic action that was actually used was

$$S_f = \sum_{k=1}^n \bar{\phi}_l \mathcal{M}_{m_l}^{-1/n} \phi_l + \phi_{pv}^\dagger \mathcal{M}_{m_{pv}} \phi_{pv} + \phi_{pv}^\dagger \mathcal{M}_{m_{pv}}^{1/2} \phi_{pv} + \phi_s^\dagger \mathcal{M}_{m_s}^{-1/2} \phi_s. \quad (8.5)$$

For  $n = 1$  this formulation will be inherently more noisy than that given in equation (8.1), as  $n$  is increased the noise on the light fermionic contribution will be reduced giving an improved acceptance rate. The two flavour Pauli-Villars contribution will still be noisy, and is a large contribution to the action so it is expected that this will affect the acceptance rate. This can be addressed by splitting the action so that the pure gauge and two flavour Pauli-Villars contributions are updated using the short time scale, and the remainder placed on the longer one, i.e.,

$$S_1 = S_g + \phi_{pv}^\dagger \mathcal{M}_{m_{pv}} \phi_{pv} \quad (8.6)$$

$$S_2 = \sum_{k=1}^n \bar{\phi}_l \mathcal{M}_{m_l}^{-1/n} \phi_l + \phi_{pv}^\dagger \mathcal{M}_{m_{pv}}^{1/2} \phi_{pv} + \phi_s^\dagger \mathcal{M}_{m_s}^{-1/2} \phi_s. \quad (8.7)$$

## 8.2 Eigenvalue bounds

An added complication exists when doing domain wall simulations that is not present in staggered type formulations, that being what to use as the lower spectral bound for the approximation. For the staggered case this was easy because the mass is a multiple of the identity matrix, hence acts as a trivial

lower bound. The domain wall mass parameter does not act as a lower bound in such a transparent manner. In addition, it is the red-black preconditioned fermion matrix which is generally used [75], this further complicates what the bounds on the matrix should be. For positive  $m_f$  there does, however, exist a lower bound in the the infinite  $L_S$  limit [17]. The upper bound is essentially independent of the fermion mass and so presents no problem.

Since there is no *a priori* method for calculating the bounds of the fermion matrix, the next best thing that can be done it to perform an empirical study to see if such a bound exists. For a range of  $\beta$  values, the lower bound behaviour as a function of the quark mass was studied in the quenched theory using the Wilson gauge action. Apart from being much cheaper computationally, the quenched theory does not stochastically suppress zero modes as is the case for full QCD since the fermionic determinant does not appear, so for a given quark mass a better measure of the lower bound can be obtained.

$V$	$L_S$	Gauge Action	$\beta$
$16^3.32$	8	Wilson	5.2
$16^3.32$	8	Wilson	5.8
$16^3.32$	8	Wilson	6.4
$16^3.32$	8	Wilson	7.2

Table 8.1: The physical parameter sets chosen for quenched eigenvalue study of the domain wall matrix.

Figure 8.1 is a plot from this quenched analysis. Except for the  $\beta = 5.2$  data, around the quark mass values  $m_f = 0.01 - 0.1$  there seems to a large degree of consistency in the value of the lower bound. This is as expected if lowest bound is be governed purely by the mass. As the quark mass is brought lighter, there is a plateau in the lowest eigenvalue. This is interpreted as originating from the residual mass, giving an additional lower bound in the eigenspectrum.

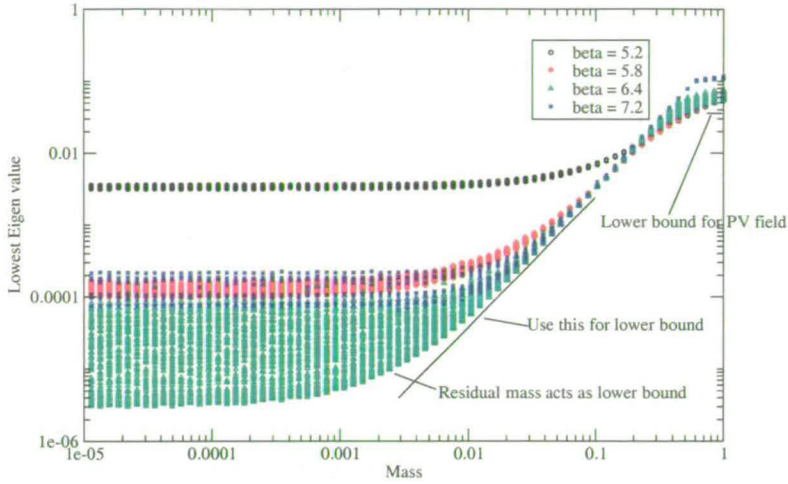


Figure 8.1: Lowest eigenvalue of preconditioned domain wall kernel  $M^\dagger M$  as a function of valence quark mass  $N_f = 0$ ,  $16^3.32.8$  lattice.

The  $\beta = 5.2$  data represents an extremely coarse lattice spacing, and thus has a much larger residual mass, giving rise to the plateau effect early on in the quark mass decrease. As the mass parameter approaches the Pauli-Villars mass there is a change in the behaviour of the lowest bound, it begins to level out. This is interpreted as an effect induced by the change in boundary conditions at  $m_t = 1$ .

The method utilising RHMC was therefore to use a speculative bound on the approximation, this speculative bound is either taken from the plot shown in figure 8.1, or taken from an evolution using the R algorithm, before “switching on” RHMC. The appropriate rational approximation can then be constructed, and RHMC used to evolve the gauge fields accordingly as described in chapter 4. Since the lower bound is taken from a quenched analysis, and from a relatively small  $L_S$ , the lower bound can only be taken as a rough estimate. On one hand the lower bound could be significantly greater because there should be zero mode suppression because of the dynamical fermions and the fact that the DBW2 gauge action is used. However, when  $L_S$  is increased the residual mass



will decrease reducing the lower bound. When using R algorithm generated lattices as a guide to what the lower bound should be, because of the finite stepsize errors, the lower bound could be higher than that obtained using an exact algorithm (see [43] for an example). Therefore, the only sensible path that can be taken is to track the eigenvalue bounds as the simulation evolves to check if there is a crossing of the eigenvalues over the bounds of validity of the rational approximations, and to check if a tightening of the bounds is possible to enable the reduction of the approximation degree.

### 8.3 Algorithm Testing

As a benchmark calculation, to test the correctness and efficiency of the RHMC algorithm formulation, the algorithm was directly compared to the R algorithm. Two different sets of parameters were chosen, with the only difference being the mass parameter of the light fermion pair. Upon each ensemble produced a variety of physical observables were measured to compare the algorithms, in addition both algorithms' ensembles were used to obtain dynamical fermion mass extrapolations to the chiral limit.

The algorithmic parameters for each of the runs are shown in table 8.2. For the degenerate three flavour run, one pseudofermion field could in principle have been used, using an approximation to the kernel  $\frac{3}{2}$ . However, as well as being a more noisy formulation, there is an inherent problem when using such an approximation kernel. In § 4.1 it was noted that when the parameter  $\alpha$  is greater than one, the lowest roots are negative. This has the implication that when the rational function is evaluated, the number of conjugate gradient iterations will blow up, vastly increasing the cost of the algorithm. Indeed, with such a large negative shift, the matrix may cease to be positive definite, and an



alternative Krylov method such as BiCGstab would be required. The ensemble labelled data set 5 was produced to obtain a measure of the stepsize scaling of the R algorithm, and no measurements were made on this other than basic algorithmic quantities and the average plaquette.

Data set	1	2	3	4	5
Algorithm	R	RHMC	R	RHMC	R
$m_l$	0.04	0.04	0.02	0.02	0.02
$\tau$	0.5	0.5	0.5	0.5	0.5
$\delta\tau$	0.01	0.02	0.01	0.0185	0.005
$N_{\text{fer}}$	1 $\{\frac{3}{2}\}$	3 $\{\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\}$	2 $\{1, \frac{1}{2}\}$	3 $\{\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\}$	$\{1, \frac{1}{2}\}$
$N_{\text{bos}}$	1 $\{\frac{3}{2}\}$	2 $\{1, \frac{1}{2}\}$	2 $\{1, \frac{1}{2}\}$	2 $\{1, \frac{1}{2}\}$	2 $\{1, \frac{1}{2}\}$
$n_{\text{md}}^{\text{fer}}$		$\{9, 9, 9\}$		$\{10, 10, 9\}$	
$\Delta_{\text{md}}^{\text{fer}}$		$\{3.18 \times 10^{-7}\}$		$\{1.59 \times 10^{-7}, 3.18 \times 10^{-7}\}$	
$n_{\text{md}}^{\text{bos}}$		$\{-, 5\}$		$\{-, 5\}$	
$\Delta_{\text{md}}^{\text{bos}}$		$1.01 \times 10^{-6}$		$1.01 \times 10^{-6}$	
$n_{\text{mc}}^{\text{fer}}$		$\{14, 14, 14\}$		$\{15, 15, 14\}$	
$\Delta_{\text{mc}}^{\text{fer}}$		$\{4.08 \times 10^{-11}\}$		$\{3.34 \times 10^{-11}, 4.08 \times 10^{-11}\}$	
$n_{\text{mc}}^{\text{bos}}$		$\{-, 8\}$		$\{-, 8\}$	
$\Delta_{\text{mc}}^{\text{bos}}$		$8.77 \times 10^{-11}$		$8.77 \times 10^{-11}$	
$\text{res}_{\text{md}}$	$10^{-6}$	$10^{-6}$	$10^{-6}$	$10^{-6}$	$10^{-6}$
$\text{res}_{\text{mc}}$		$10^{-10}$		$10^{-10}$	
$\hat{m}$	1	2	1	2	1
$N_{\text{traj}}$	1565	2395	2500	5000	1650

Table 8.2: The parameters for the data sets chosen for the domain wall algorithm comparison ( $V = 16^3.32$ ,  $L_S = 8$ , DBW2 gauge action,  $\beta = 0.72$ ). The parameters  $N_{\text{fer}}$  and  $N_{\text{bos}}$  represent the number of fields used to represent the fermionic and bosonic contributions respectively.

Figures 8.2 and 8.3 are plots of the lower bounds of the fermion matrix for the three relevant masses to these results. It can be seen that in all three cases the smallest eigenvalue is well behaved in that it does not fluctuate significantly, which allows the use of a constant range for the rational approximation. On both figures the associated bound used for the rational approximation is also included, it is clear that a sufficient margin has been given, such if the lower bound were to fluctuate, the rational approximation would still be valid. In fact the bounds used for this test run are probably too conservative, and some fine tuning could be done to raise the lower bound on the approximation which perhaps would allow the degree of the approximations to be reduced and/or lead to a slightly better-conditioned matrix.

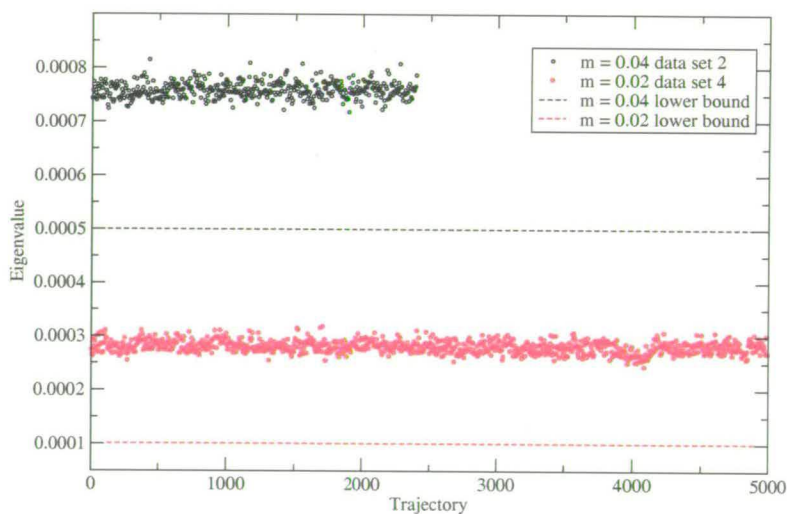


Figure 8.2: The lower bound on the eigenvalues with  $m_f = 0.02$  and  $m_f = 0.04$  (each data set is defined in table 8.2).

The equivalent plot for the upper eigenvalues is included in figure 8.4. Here there is no mass dependence on the eigenvalue, unlike the lower bound. This of course is exactly as expected, since the mass dependent parameter which appears in the matrix is small compared to the greatest eigenvalue for all masses considered here.

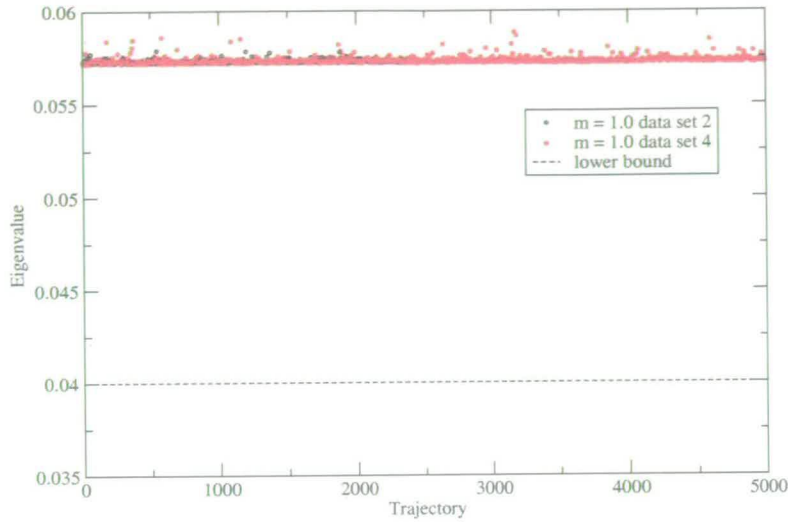


Figure 8.3: The lower bound on the eigenvalues with  $m_f = 1.0$  (each data set is defined in table 8.2).

Some of the measurements from the generated lattices are shown in table 8.3. From both data sets, it is clear that RHMC has plaquette measurements consistent with the R algorithm to within 0.1%. Figure 8.5 is a plot comparing the average plaquettes of the first two data sets as function of jackknife bin size. There is strong evidence that RHMC is producing ensembles with the correct statistical weight, but given that the averages are separated by around  $12\sigma$  it is also apparent that the R algorithm has a significant  $\delta\tau$  error at this stepsize. Figure 8.6 is the same plot, but with the lighter mass data sets. As the stepsize is halved from  $\delta\tau = 0.01$  to  $\delta\tau = 0.005$  there is a clear shift towards the RHMC result, however, even at  $\delta\tau = 0.005$  there is still a  $4\sigma$  deviation. The extrapolated stepsize dependence of the plaquette is shown on figure 8.7. For consistency between the algorithms, it is estimated that the R algorithm must be run using a stepsize  $\delta\tau < 0.002 \approx \frac{m_l}{10}$ .

Unexpectedly, the acceptance rate of data set 2 is less than that of data set 4. The stepsize in data set 4 had been reduced slightly, because of the expectation that the acceptance rate would decrease with the mass. This result here suggests



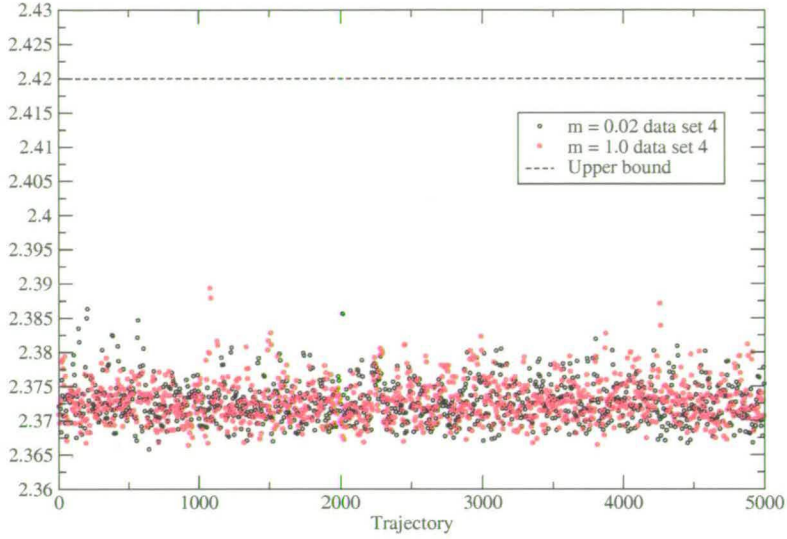


Figure 8.4: The upper bound on the eigenvalues with  $m_f = 1.0, 0.02$  (each data set is defined in table 8.2).

that with this prescription, namely using three pseudofermion fields, the fermion force is insensitive to the mass. This of course would be expected to change as the mass is further decreased. The relative cost of the algorithms is shown in terms of matrix-vector products per trajectory, RHMC at  $\delta\tau = 0.02$  is more expensive than R at  $\delta\tau = 0.01$  by 48% and 13% respectively. The degenerate three flavour run is significantly more expensive because the R algorithm is using 1 pseudofermion field to simulate the fermions, whereas RHMC is using three. It may have been more efficient to use two pseudofermion fields each with kernel  $\frac{3}{4}$ , presumably with a lower stepsize, but three fields were chosen for a direct comparison of the two RHMC data sets. The efficiency measure used in previous sections is of no use here because the concept of acceptance cannot be applied to the R algorithm data sets.

Data set	1	2	3	4	5
$\langle P \rangle$	0.608120(15)	0.607788(10)	0.608293(11)	0.608094(8)	0.608168(30)
$\langle \delta H \rangle$		0.3977(170)		0.3107(115)	
$\langle A \rangle$		0.6550(63)		0.6934(43)	
$N_{\text{mv}}$	13449(4)	20300(3)	32782(12)	36265(8)	65345(15)

Table 8.3: Various observables from all data sets (each data set is defined in table 8.2). The errors were found using a jackknife analysis.

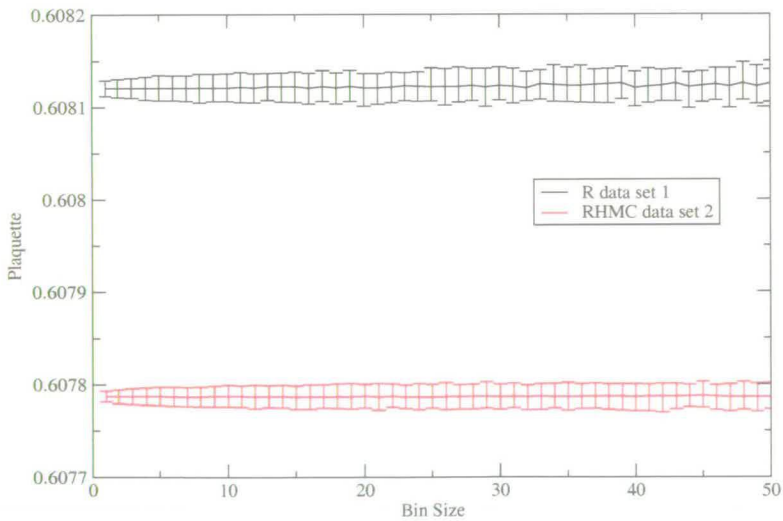


Figure 8.5: Jackknife error of the plaquette for data sets 1 and 2 (each data set is defined in table 8.2).

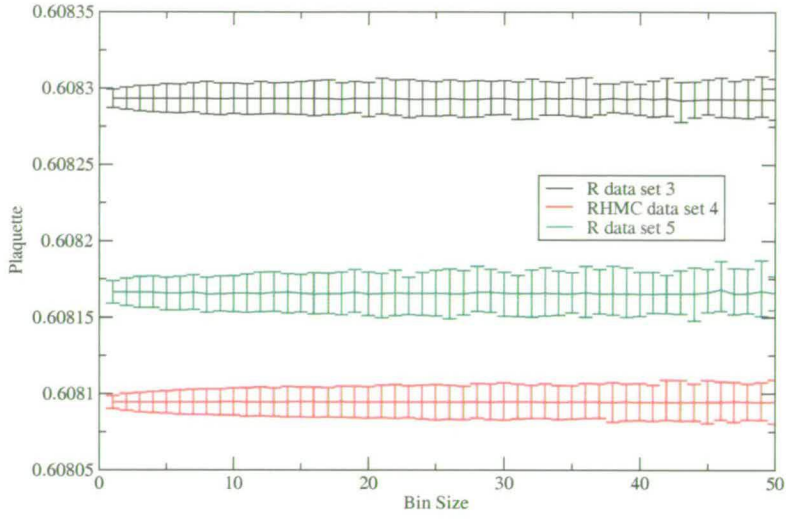


Figure 8.6: Jackknife error of the plaquette for data sets 3 and 4 (each data set is defined in table 8.2).

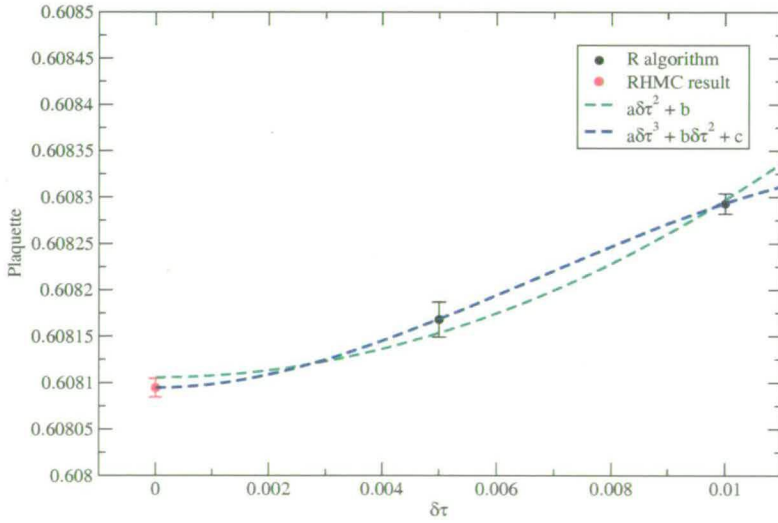


Figure 8.7: The extrapolated stepsize dependence of the average plaquette using the R algorithm,  $m_l = 0.02$ . The quadratic fit uses only the two R algorithm points and the cubic fit uses the RHMC result to constrain the fit.

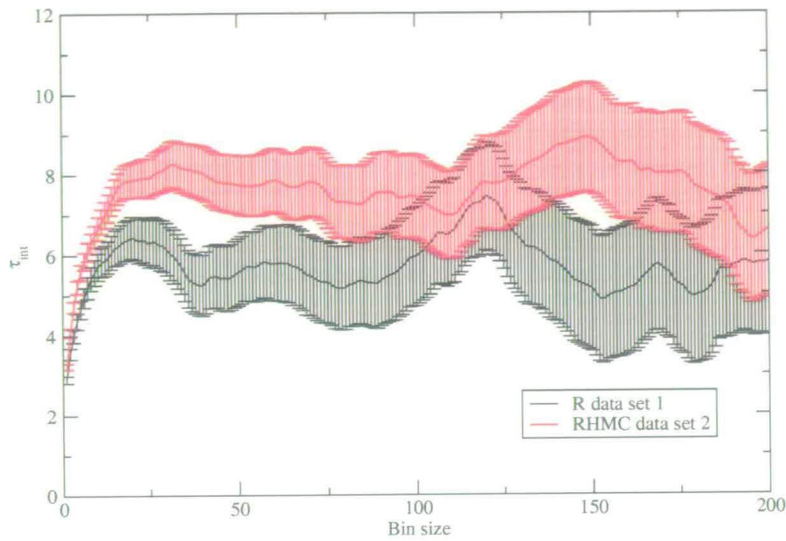


Figure 8.8: Integrated autocorrelation length of the plaquette for data sets 1 and 2 (each data set is defined in table 8.2).

The integrated autocorrelation length of the plaquette for data sets 1 and 2 is shown in figure 8.8 (see also table 8.4). From this evidence, RHMC appears to have an autocorrelation length that is comparable to the R algorithm. This is an extremely good result in RHMC's favour given that the acceptance rate is only 65.5%. On the other hand, from figure 8.9, which is the same data obtained from data sets 3 and 4, RHMC has an integrated autocorrelation time that is nearly three times that of the R algorithm. Even with the acceptance rate of 69% factored in, this is a surprising result, especially so given the similarity observed with data sets 1 and 2. The reason for this large difference in autocorrelation was found to be due to the differing statistics between the data sets. If the number of measurements present in data set 4 are truncated to 2500, i.e., the size of data set 3, then the integrated autocorrelation length decreases significantly (see table 8.4), such that within error bars it is compatible with the R algorithm result. The integrated autocorrelation is therefore underestimated on data set 3 (and perhaps 1 and 2 also), and more statistics are required to enable a fair comparison.



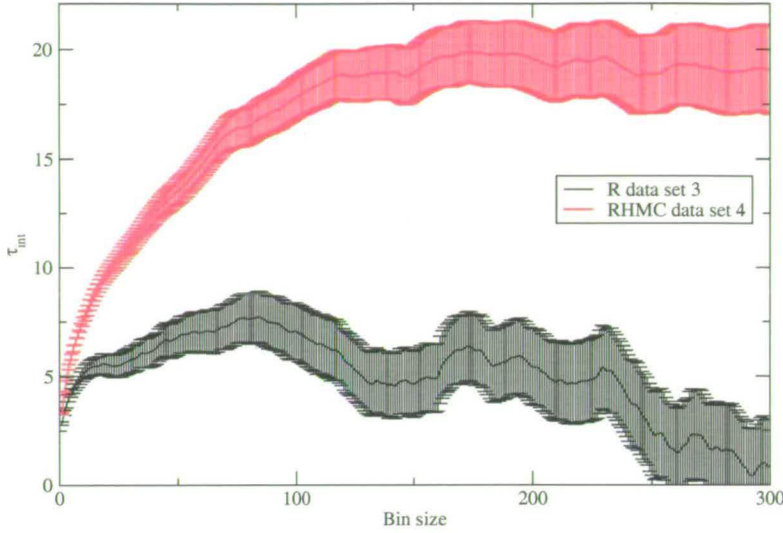


Figure 8.9: Integrated autocorrelation length of the plaquette for data sets 3 and 4 (each data set is defined in table 8.2).

Figures 8.10 and 8.11 are plots of the integrated autocorrelation lengths from data sets 1 and 2 and 3 and 4 respectively taken from the 13<sup>th</sup> timeslice of the pion correlator. The behaviour that was observed with the plaquette is clearly reproduced here, namely that the degenerate 3 flavour data have very similar integrated autocorrelation lengths, and the 2+1 data sets have an RHMC integrated autocorrelation time about three times that of the R algorithm. Again when data set 4 is truncated, the integrated autocorrelation length decreases, approaching a similar value to the result from data set 3.

On all of the ensembles the time history of the topological charge was measured. Figures 8.12 and 8.13 are such plots. These plots appear very similar to the topological charge history of 2 flavour domain wall ensembles published in [57], this of course would be expected, since the addition of the strange quark should not adversely affect the tunnelling rate of the topology.

Data set	$\tau_{\text{int}}^P$	$\tau_{\text{int}}^\pi$
1	7.43 (1.35)	10.97 (2.18)
2	8.92 (1.36)	10.81 (1.63)
3	7.74 (1.17)	9.55 (1.21)
4	19.87 (1.41)	27.95 (3.31)
4 (truncated)	9.13 (1.24)	11.23 (2.24)

Table 8.4: The measured integrated autocorrelation lengths parameters for the plaquette and the 13<sup>th</sup> timeslice of the pion correlator (each data set is defined in table 8.2). The errors were found using a jackknife analysis.

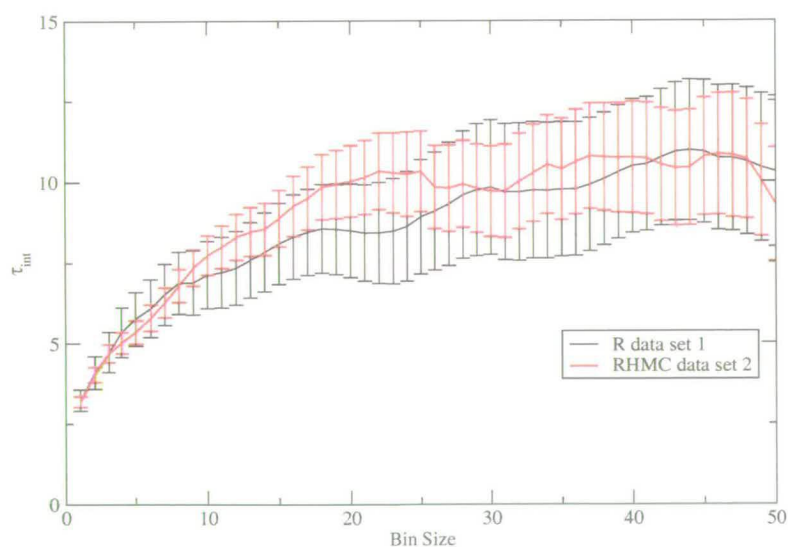


Figure 8.10: Integrated autocorrelation time of the pion, timeslice 13 for data sets 1 and 2 (each data set is defined in table 8.2).

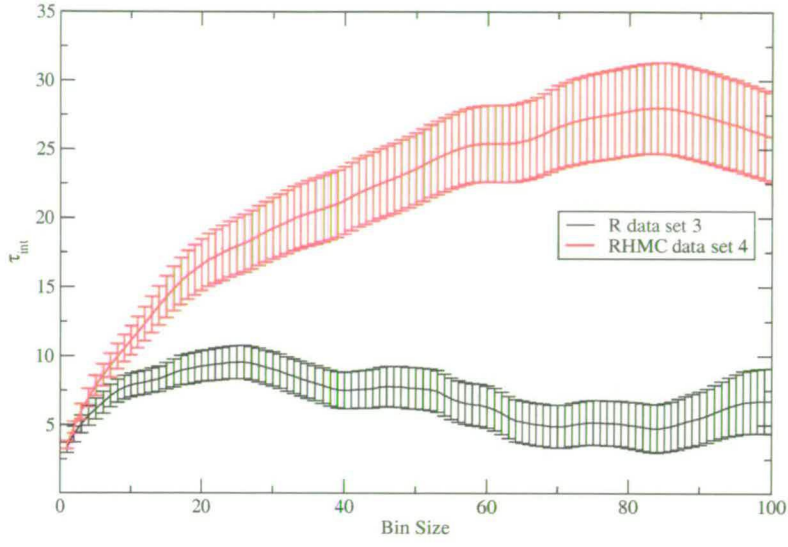


Figure 8.11: Integrated autocorrelation time of the pion, timeslice 13 for data sets 3 and 4 (each data set is defined in table 8.2).

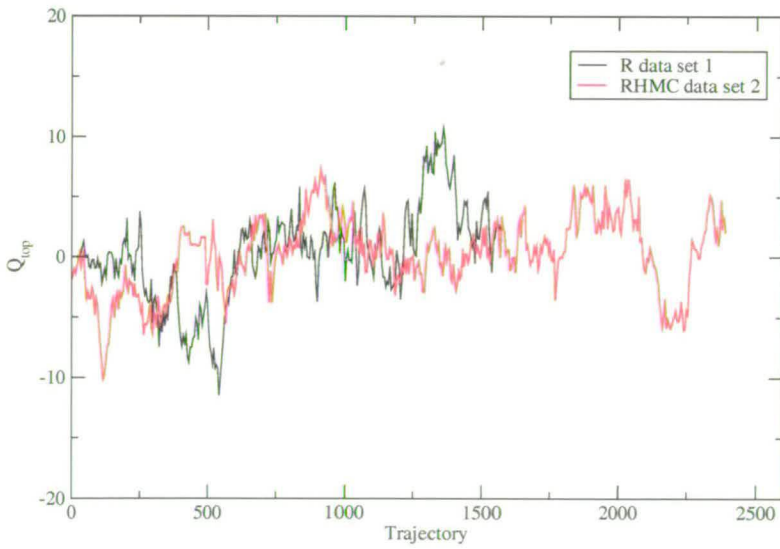


Figure 8.12: A plot of the topological charge history from data sets 1 and 2 (each data set is defined in table 8.2).

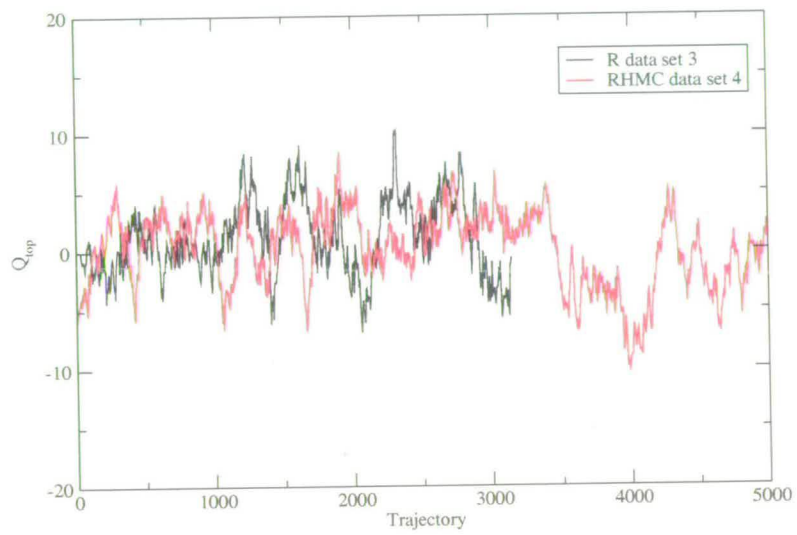


Figure 8.13: A plot of the topological charge history from data sets 3 and 4 (each data set is defined in table 8.2).



The remainder of this section concentrates on the hadronic quantities that have been extracted from the generated ensembles. In this section only the graphs are included, for a full breakdown of the extracted numbers see appendix C.

Something that was noticed when looking at the raw data coming from observables measured was that the R algorithm data appears to be more noisy than that of the RHMC data. An example of this observation is shown in figure 8.14, where the time history of the pion from the 13<sup>th</sup> timeslice is compared for data sets 1 and 2. It may be a subjective statement, but the pion correlator produced by the R algorithm appears to be more noisy than that of RHMC. There are five clear spikes in the R algorithm data, which do not occur in the RHMC data set. Presumably these spikes correspond to configurations that would otherwise be rejected by the RHMC algorithm (because they would correspond to large changes in the Hamiltonian resulting in a negligible acceptance rate), but are not suppressed by the R algorithm. This spiky behaviour appears on both data sets 1 and 3, on all the timeslices looked at, and it is also present on the rho correlator measurements. It would be interesting to see if these spikes are suppressed by reducing the integrating stepsize.

As an illustrative example of what was described in § 2.5.2, an effective mass plot is shown in figure 8.15. This plot is that of the pion correlator, and the black line represents the value of the extracted mass using the fitting range shown on the x axis. The error bars are included in the plot, but are too small to see at this scale.

Figure 8.16 is a plot of the valence mass extrapolation of  $m_\pi$  taken from the degenerate 3 flavour data sets. For each of the valence masses there is a general agreement in the extracted masses. At the chiral limit, the extracted value of  $m_{\text{res}}$  deviates between R and RHMC outwith the quoted error bars. However, this is probably not statistically significant given that this result comes from

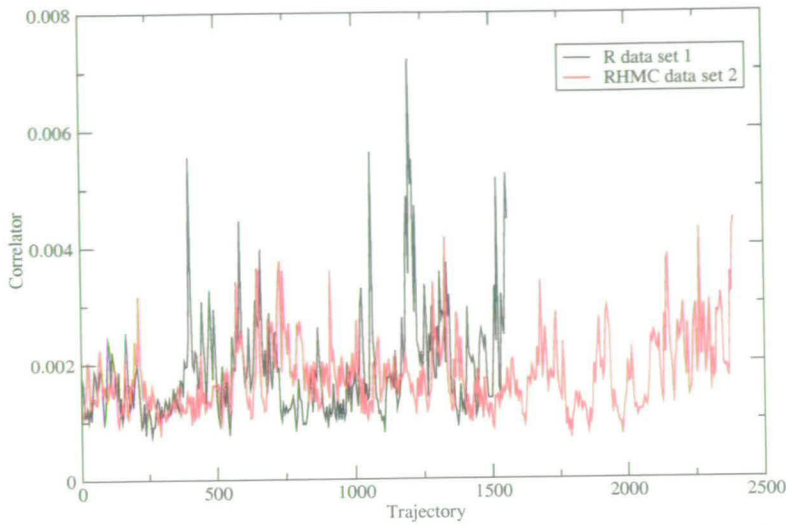


Figure 8.14: The time history of the pion correlator, timeslice 13 for data sets 1 and 2 (each data set is defined in table 8.2).

the extrapolation of three data points. The equivalent plot from the 2+1 ensembles is shown in figure 8.17. Here, at all three mass values the value of  $m_\pi$  is significantly smaller for RHMC than for the R algorithm. This is also true of the extracted value of  $m_{\text{res}}$ , where a  $4 - 5\sigma$  deviation is observed.

The valence mass extrapolations of  $m_{\text{res}}$  are shown in figures 8.18 and 8.19. For both extracted mass values there is consistency between the algorithms, though the trend that RHMC gives a smaller mass value is carried over from the pion results. When calculating the chiral value of  $m_{\text{res}}$ , the consistency requirement described in §2.5.3 was used. Exactly the same behaviour can be seen in the  $m_\rho$  valence extrapolations, shown in figures 8.20 and 8.21. The  $m_\rho$  values are consistent within errors, with the trend of smaller RHMC masses. The noisy nature of the rho compared to the pion can be observed from the larger error bars compared to those from the pion and  $m_{\text{res}}$ . The extracted values of lattice spacing are consistent within errors, where the lattice spacing is measured at  $m_{\text{val}} = -m_{\text{res}}$ .

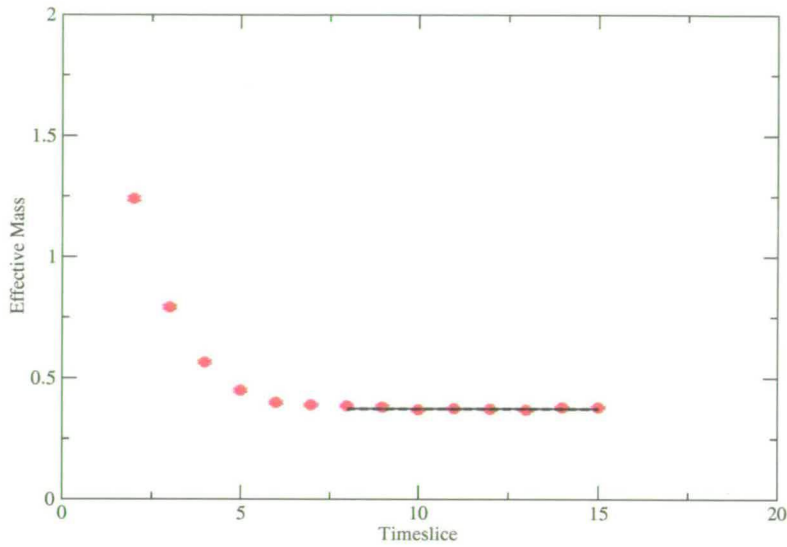


Figure 8.15: Effective mass plot for  $m_\pi$ , data set 4,  $m_{\text{val}} = 0.02$  (each data set is defined in table 8.2).

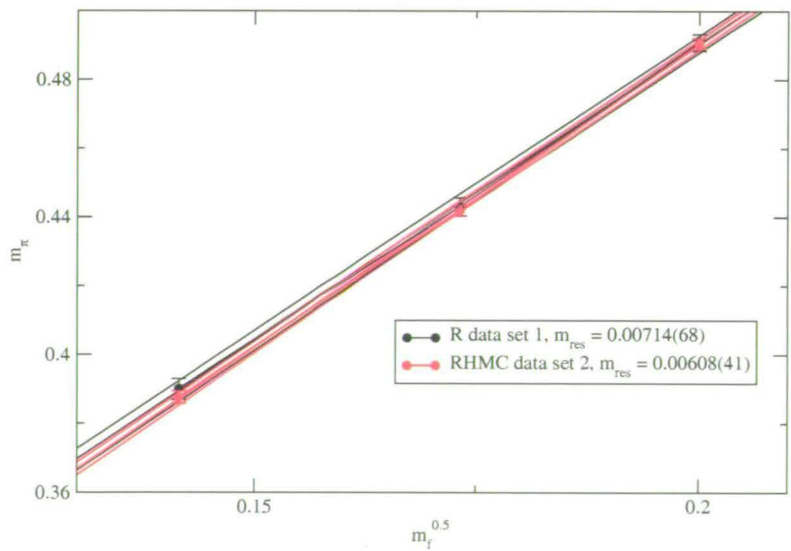


Figure 8.16: Valence mass extrapolations of  $m_\pi$  from data sets 1 and 2. (each data set is defined in table 8.2, data is tabulated in tables C.1 and C.2).

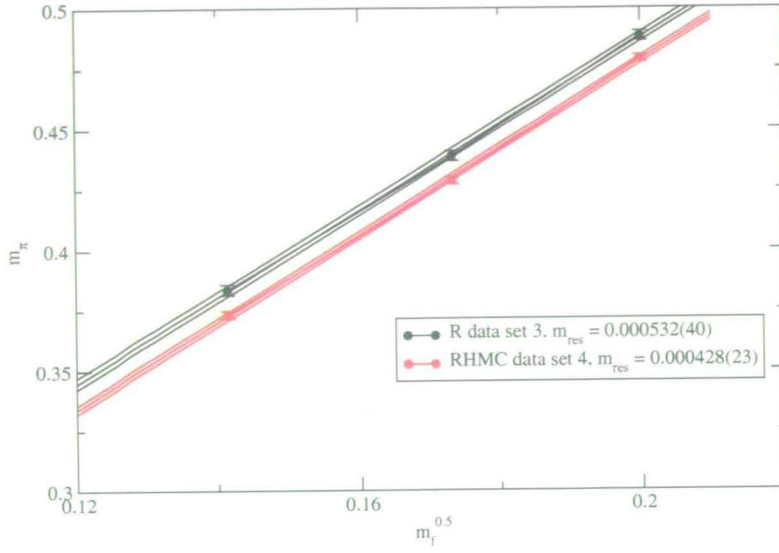


Figure 8.17: Valence mass extrapolations of  $m_\pi$  from data sets 3 and 4 (each data set is defined in table 8.2, data is tabulated in tables C.3 and C.4).

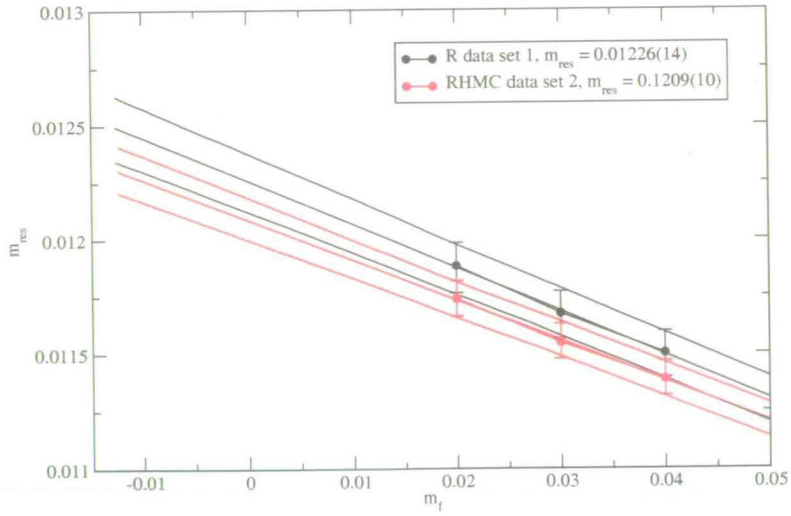


Figure 8.18: Valence mass extrapolations of  $m_{\text{res}}$  from data sets 1 and 2 (each data set is defined in table 8.2, data is tabulated in tables C.5 and C.6,  $m_{\text{res}}$  is defined from equation (2.36)).



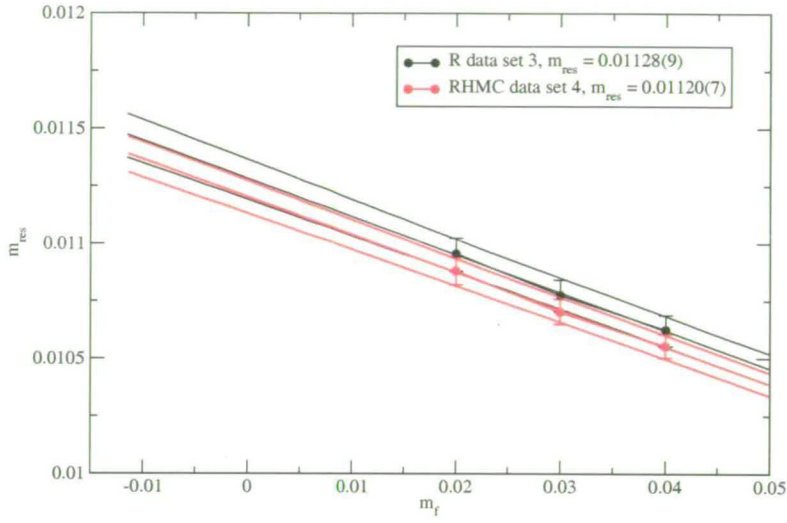


Figure 8.19: Valence mass extrapolations of  $m_{\text{res}}$  from data sets 3 and 4 (each data set is defined in table 8.2, data is tabulated in tables C.7 and C.8,  $m_{\text{res}}$  is defined from equation (2.36)).

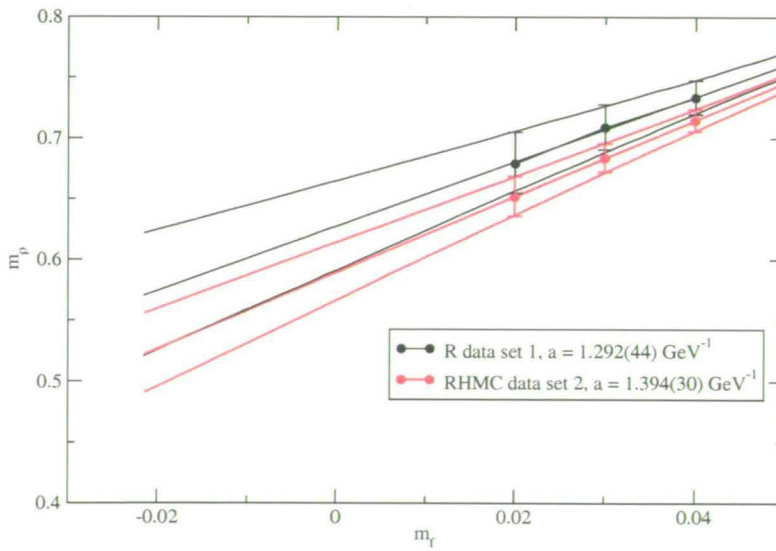


Figure 8.20: Valence mass extrapolations of  $m_{\rho}$  from data sets 1 and 2 (each data set is defined in table 8.2, data is tabulated in tables C.9 and C.10):

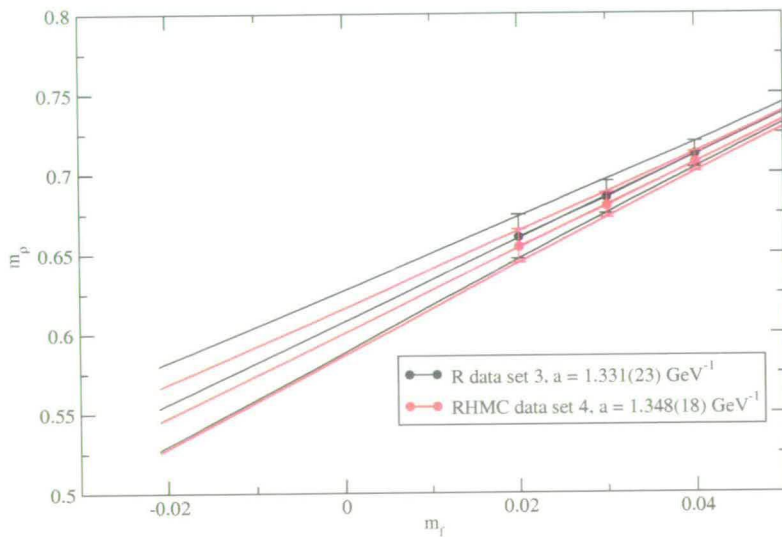


Figure 8.21: Valence mass extrapolations of  $m_\rho$  from data sets 3 and 4 (each data set is defined in table 8.2, data is tabulated in tables C.11 and C.12).

The ensembles were analysed together to obtain dynamical mass extrapolations of the above quantities discussed. This analysis is shown in figures 8.22, 8.23 and 8.24. In all three plots the above trend that the RHMC algorithm leads to smaller masses can be observed. An interesting observation from figure 8.23, is that the value of  $m_{\text{res}}$  appears to decrease with decreasing quark mass. This contrasts directly with the valence extrapolations, figures 8.19 and 8.18, which show  $m_{\text{res}}$  increasing with decreasing quark mass. This is perhaps an example of where a unitary extrapolation is required to obtain the correct physics behaviour. This change in slope direction is not found for either measurements of  $m_{\pi}$  or  $m_{\text{res}}$ .

The observables obtained from these extrapolations may be trusted more than the valence extrapolations above from a physics perspective, but because only two data points are used, the extrapolations are merely a case of putting a straight line through the points, and so have to be taken with a large pinch of salt. This is a good example of the difficulty in performing chiral extrapolations. The valence analysis may be very cheap to perform, and it is possible to obtain data for an arbitrary number of masses, but there is a risk of getting the physics wrong, i.e., as was demonstrated with  $m_{\text{res}}$ . The dynamical analysis, while certain to be correct, is extremely expensive to perform if a reasonable number of masses are to be included in the extrapolations.

From the results presented in this section, the implementation of RHMC for domain wall fermion simulations appears to be validated. While most quantities are, within errors, consistent with the R algorithm results, there is a consistent trend that the RHMC masses are smaller. It is tempting to interpret this as being a signal for the finite stepsize error of the R algorithm, although how to prove this conjecture is beyond the scope of this work.

This chapter can therefore be concluded by saying that RHMC has now been

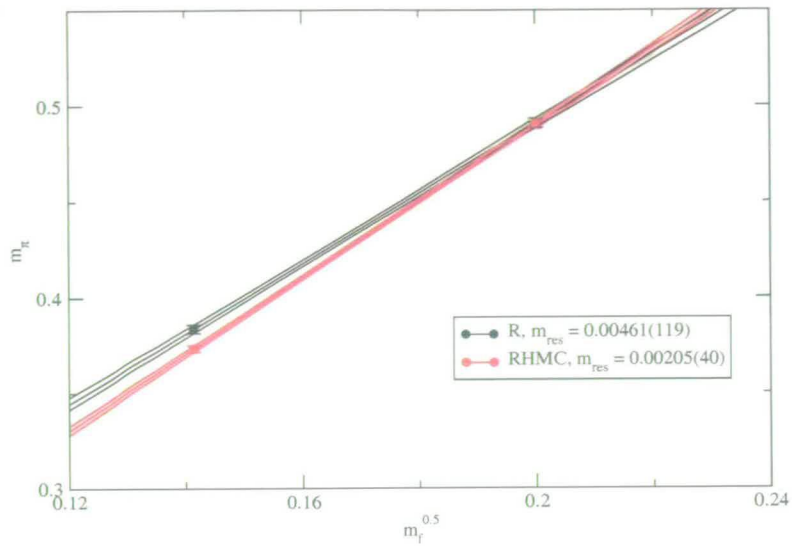


Figure 8.22: Dynamical mass extrapolations of  $m_\pi$ , (each data set is defined in table 8.2, data is tabulated in tables C.13 and C.14).

shown to work on large lattices, using an improved fermion formulation. When compared against the R algorithm, to obtain a consistent measurement of the plaquette, the R algorithm must use an integrating stepsize 10 times less than that of RHMC. This behaviour must also be true for more complicated observables such as the hadronic quantities measured in this chapter.



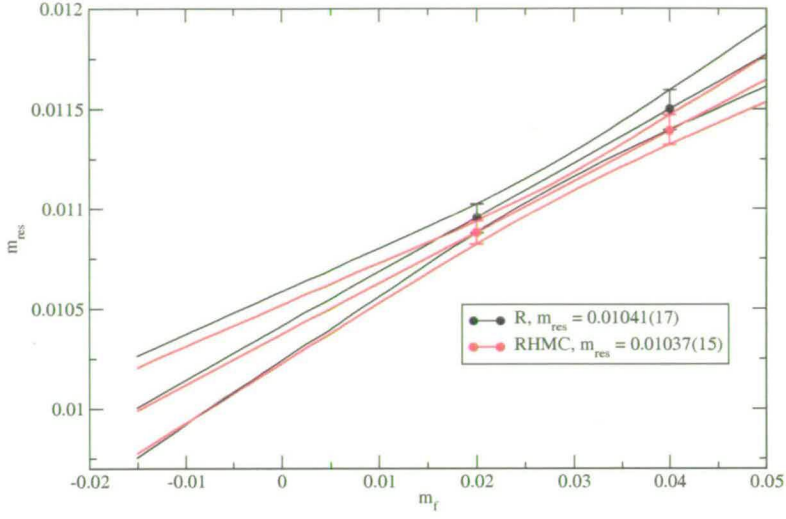


Figure 8.23: Dynamical mass extrapolations of  $m_{\text{res}}$ , (each data set is defined in table 8.2, data is tabulated in tables C.15 and C.16).

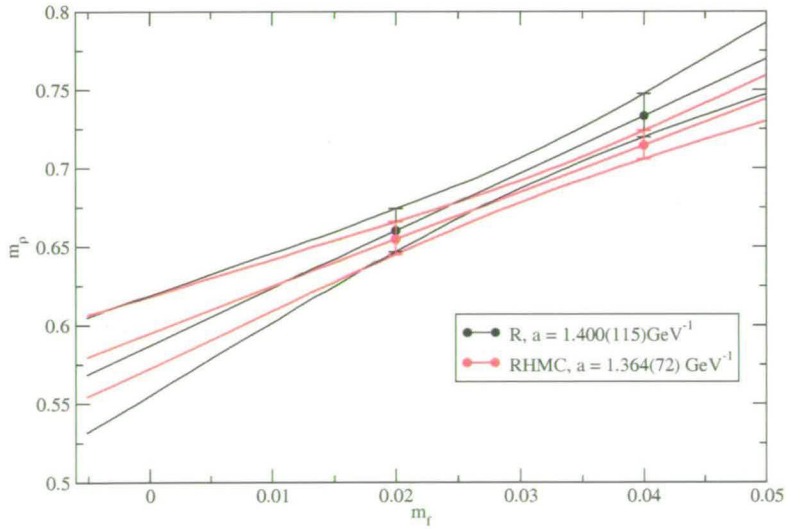


Figure 8.24: Dynamical mass extrapolations of  $m_{\rho}$ , (each data set is defined in table 8.2, data is tabulated in tables C.17 and C.18).



# Chapter 9

## Concluding remarks

### 9.1 Summary

The aim of this work has been to introduce the Rational Hybrid Monte Carlo algorithm, and to apply it to the problem of generating gauge field configurations that represent full QCD, i.e., including both the light pair and the strange quarks into the vacuum.

In chapter 4 the algorithm was introduced, the motivation for the algorithm being the desire to have an algorithm which has the efficiency of conventional HMC but also have the ability to simulate fewer flavours than are described by the fermion kernel. This latter property is a requirement for all fermion formulations, to enable 2+1 quark flavour calculations to be performed. The reason behind the efficiency of RHMC lies in the high accuracy of rational approximations, and the fact that rational functions with the Dirac matrix as an argument can be evaluated using a multi-shift solver. When compared against published data performed by Polynomial Hybrid Monte Carlo and the R algorithm, RHMC was shown to be both correct, and significantly more efficient.

An investigation was made in chapter 5, to determine whether it is possible to use an initial guess based on previous solutions in the multi-shift solver to speed up the evaluation of rational functions. Unfortunately, this investigation did not reveal a method to do this, but as a by-product of this, a generalisation of the multi-shift solver was found which could be applied to the R algorithm to reduce the number of conjugate gradient iterations when doing 2+1 calculations.

A resolution to the critical slowing down in acceptance rate with decreasing quark mass was presented in chapter 6. The reason that the Hasenbusch method produces better acceptance rates was given: namely that increasing the number of pseudofermion fields used to represent the fermion determinant reduces the noise on the measure, that manifests an improved acceptance rate. An alternative method to produce the same effect was introduced, the nroots method. This method has an advantage over the Hasenbusch trick because it uses RHMC, so can be applied to any number and type of fermions. At the range of parameters tested, there was only an overall gain to be had when using the lightest quark mass parameter. As the quark mass is reduced and the volume increased, the expectation is that the improvement gained by the nroots acceleration will increase considerably.

RHMC was applied to the ASQTAD fermion formulation in chapter 7. Careful consideration was given to the calculation of the kernel derivative, and for the case of 2+1 flavour calculations, it was found that there is no overhead in using RHMC compared to using the R algorithm for ASQTAD calculations.

The results in chapter 8 show RHMC can be perfectly well adapted for use with chiral fermion formulations. On the statistics used in this study there is evidence to suggest that the R algorithm and RHMC have compatible autocorrelation lengths. On the parameters explored here, the R algorithm must be run at a stepsize around ten times less than that of RHMC to obtain consistency.



## 9.2 Future Work

From this work there are a lot of unanswered questions that shall be the focus of future work.

The most unsatisfactory aspect of this work is the inability to have been able to perform a full comparison of RHMC with the R algorithm using ASQTAD fermions. This fermion formulation shall constitute a large fraction of calculations done by UKQCD, so it is extremely important to validate current data, and to check which algorithm is the most efficient. It is expected that RHMC will allow increases in the stepsize over the two thirds rule employed for R algorithm calculations, which should lead to a considerable decrease in the cost to generate such ensembles.

After the work presented in this thesis was completed, a comparison between the R algorithm and RHMC was performed using naïve staggered fermions near the deconfinement transition the author. It was found that the finite stepsize errors present in the R algorithm can adversely affect the location of the transition point, i.e., the value of  $\beta_c$  the critical value of the coupling. In order to obtain consistent results between the two algorithms, the stepsize used for the R algorithm must be around twenty times smaller than the RHMC stepsize. This is very significant since it demonstrates the danger in using an inexact algorithm. This study shall be presented in a future publication.

The use of a chronological inverter with the multi-shift solver is still obviously desirable. Further research in this area shall be performed in an attempt to find a resolution.

It will of course be sensible to perform a full cost comparison of the nroots algorithm presented here against the method advocated by Hasenbusch [62] to see which method is superior. Although this work demonstrated that nroots

seems to be only giving very small benefits, this was for the staggered fermion formulation, which suffers much more mildly from critical slowing down in acceptance rate compared to Wilson fermions, where the Hasenbusch trick has been so successful.

Further refinement of the nroots acceleration shall be investigated as there are potentially more gains that can be made. There are two possible avenues for refinement. The first of these is to explore whether calculating the  $n$  different fermionic contributions to the force at different times in each MD step could be beneficial. Take for example, the case where  $n = 2$  pseudofermion fields are used. The current nroots procedure would have been to calculate the contribution to the force at the mid point of each MD step. It could be argued that if one fermion contribution were calculated at the mid-point and half of the other at each end of the MD step, a smoother evolution of gauge fields would be obtained. This would presumably lead to an improved acceptance rate with no overhead, i.e., an integrator of the following form

$$\hat{U}(\tau) = \left( e^{\delta\tau(P_G+P_2)/2} e^{\delta\tau Q/2} e^{\delta\tau P_1} e^{\delta\tau Q/2} e^{\delta\tau(P_G+P_2)/2} \right)^{\tau/dt}, \quad (9.1)$$

where  $P_G$  represents the momentum update due to the pure gauge, and  $P_1$  and  $P_2$  due to the fermionic contributions. In doing this it would probably be required that the  $P_2$  contribution is calculated at the end points to ensure it can be glued with the  $P_2$  updates in adjacent MD steps to remove the extra inversion otherwise incurred.

The second avenue that might be worth exploring is the use of techniques developed for propagator calculations, where there is a single fermion matrix with multiple right hand sides to be solved against. This is effectively the nroots problem, where  $n$  right hand sides must be solved at some point in MD time. Currently this is done through sequential multi-shift solves. One alternative may be to use deflation methods, where on the first inversion the eigenvectors

of the fermion matrix are projected out, these are then used to accelerate the convergence of the subsequent solves [76]. This procedure can be extended to shifted systems as well [77], so could be applied to nroots. The alternative to deflation methods when solving for multiple right hand sides is to use the block algorithm approach [78], where the solution vectors and right hand sides are arranged as matrices whose columns are these vectors, and all vector linear algebra becomes matrix algebra. This accelerates the inversion process because of off-diagonal terms which occur because of an overlap between the Krylov subspaces of the righthand sides. Block algorithms have not been applied to the case of shifted systems, it may be worth investigating whether it is possible.

Another improvement that can be made is in the generation of the rational approximations. A method to accelerate its convergence is desired since this can be a slow process when undertaken on a QCDOC. For the case of finding the  $n^{th}$  root approximation, the analytic form of the function and its derivatives are known, so in principle this knowledge can be used to accelerate the search process. Since the  $n^{th}$  root is a smooth function the Newton-Raphson search method should work without difficulty for finding the maxima (and zeros) of the error function. The linear equation solving, currently done using a direct method, could be accelerated using a Krylov method, and since the previous solution vector could be used as an initial guess the method would be expected to converge rapidly. With such changes to the Remez algorithm implementation, it may be possible to relax the requirement to perform the calculation using high precision arithmetic to merely double precision, this step alone would lead to a large improvement in the execution time. Since the Remez algorithm can also be used to find pure polynomial approximations, such an implementation might be very useful for large degree approximations, where it can take days to calculate the approximation [47].

The autocorrelation length of the RHMC algorithm should be investigated fur-



ther. What may be interesting to explore is whether the use of ultra-violet preconditioning [79] can be used to further improve the RHMC algorithm. The method is to factor out the ultra-violet, i.e., short distance, contributions to the fermionic determinant, these factored out pieces correspond to shifts in the gauge coupling [80]. This technique has successfully been applied to the multi-boson algorithm, and has been shown to significantly reduce the plaquette auto-correlation time [81] as well as allowing a reduction in the degree of polynomial approximation. The reason for the former is that the short distance physics are now included exactly, and not stochastically through auxillary fields. The correlation length of topological charge, and other long distance physics, is unchanged by this filtering.

The difficulties in producing reliable chiral extrapolations was highlighted in § 8.3. A good improvement to this situation could possibly be obtained through including the difference between the fermion determinant with mass parameter  $m_{\text{dyn}}$  and the determinant with mass parameter  $m_{\text{val}}$  into the observable being measured, i.e. for the simplest case of staggered fermions,

$$\det(-\not{D}^2 + m_{\text{val}}^2) = \det(-\not{D}^2 + m_{\text{dyn}}^2) \det(1 + \Delta) \quad (9.2)$$

$$= \det(-\not{D}^2 + m_{\text{dyn}}^2) \exp[\text{tr} \ln(1 + \Delta)], \quad (9.3)$$

where  $m_{\text{val}}^2 = m_{\text{dyn}}^2 + \delta m^2$  and  $\Delta = \frac{\delta m^2}{-\not{D}^2 + m_{\text{dyn}}^2}$ . What this is saying is that if one requires a measurement of an observable with a mass  $m_{\text{val}}$ , then an ensemble with dynamical mass  $m_{\text{dyn}}$  can be used if the  $\exp[\text{tr} \ln(1 + \Delta)]$  factor is included in the observable measured. The  $\exp[\text{tr} \ln(1 + \Delta)]$  could be calculated through stochastic summation on the Taylor expansion of the exponential and logarithm functions, the latter of which is valid for  $|\Delta| < 1$ . This reweighting method is similar to that in [82], however, it has never been applied to the dynamical fermion mass in such a way. The method is basically the  $\kappa$  expansion, but from a finite fermion mass. This method in principle allows unitary chiral



extrapolations using only a single ensemble, however, as the parameter  $\delta m^2$  is increased, the overlap between the vacuum state, and the desired state will reduce. This lack of overlap will manifest itself through a large increase in statistical noise in any observables measured. In reality this reweighting method might be useful to explore  $\pm 25\%$  of the true ensemble mass, which could prove to be very useful when performing dynamical chiral extrapolations when using two or three different ensembles with different mass parameters.



# Appendix A

## Staggered Fermion Results

$r_{\text{md}}$	$\Delta_{\text{md}}$	$N_{\text{mv}}$	$\langle P \rangle$	$\langle A \rangle$	$E (\times 10^{-5})$
1	$3.87 \times 10^{-1}$	12712(2)	-	0	0
2	$8.71 \times 10^{-2}$	24795(3)	-	0	0
3	$1.89 \times 10^{-2}$	34196(176)	-	0	0
4	$4.08 \times 10^{-3}$	38654(14)	0.57685(22)	0.539(29)	1.48(5)
5	$8.84 \times 10^{-4}$	39420(70)	0.57688(14)	0.670(24)	1.73(7)
6	$1.91 \times 10^{-4}$	41144(34)	0.57695(16)	0.682(36)	1.73(5)
7	$4.14 \times 10^{-5}$	42023(26)	0.57693(19)	0.692(23)	1.85(6)
8	$8.97 \times 10^{-6}$	42713(36)	0.57700(24)	0.733(40)	1.75(5)
9	$1.94 \times 10^{-6}$	43098(38)	0.57701(22)	0.724(20)	1.70(5)
10	$4.20 \times 10^{-7}$	43053(24)	0.57715(10)	0.728(22)	1.69(7)
11	$9.10 \times 10^{-8}$	43285(69)	0.57697(11)	0.728(34)	1.69(9)
12	$1.97 \times 10^{-8}$	43680(5)	0.57698(7)	0.726(27)	1.69(2)
13	$4.26 \times 10^{-9}$	43785(39)	0.57705(13)	0.763(19)	1.55(4)
5	$8.84 \times 10^{-4}$	39234(10)	0.57706(4)	0.678(4)	1.75(2)

Table A.1: The measurements made when varying the MD approximation degree with staggered fermions. All of these results used an integrating stepsize  $\delta\tau = 0.02$  and consisten of 100 trajectories each, apart from the last row which is the result of a 2000 trajectory run. The errors were found using a jackknife analysis.



$\delta\tau^{\text{fer}}$	$\delta\tau^{\text{gauge}}$	$\langle P \rangle$	$\delta H$	$\langle A \rangle$	E ( $\times 10^{-5}$ )
0.01111	0.01111	0.57700(21)	0.0411(150)	0.906(11)	1.76(2)
0.02	0.02	0.57727(12)	0.181(159)	0.746(49)	2.02(12)
0.025	0.025	0.57719(13)	0.589(138)	0.631(26)	2.00(26)
0.03333	0.03333	0.57716(18)	1.85(23)	0.367(47)	1.99(18)
0.04	0.04	0.57722(10)	3.32(34)	0.231(54)	1.44(32)
0.05	0.05	-	11.0(3)	0.0140(167)	0.102(8)
0.0125	0.01	0.57723(19)	0.0327(286)	0.886(14)	1.94(3)
0.025	0.01	0.57695(26)	0.165(47)	0.786(21)	3.29(9)
0.03571	0.01	0.57684(26)	0.233(75)	0.728(27)	4.26(16)
0.05	0.01	0.57727(6)	0.762(140)	0.544(37)	4.31(29)
0.0625	0.01	0.57694(19)	2.42(24)	0.276(36)	2.63(34)
0.0833333	0.01	0.57724(14)	8.89(44)	0.0491(183)	0.592(23)
0.1	0.01	-	24.1(7)	0.000	0.000

Table A.2: The measurements made when varying the integrating stepsize  $\delta\tau$  with staggered fermions using both the standard  $\hat{U}_{\text{QFQ}}$  integrator and with multiple time steps. All of these results used  $n_{\text{md}} = 10$  and  $n_{\text{mc}} = 15$  and consist of 100 trajectories each. The errors were found using a jackknife analysis.



# Appendix B

## Nroots Results

$\delta\tau^{\text{fer}}$	$\delta H$	$\langle A \rangle$	$\langle E \rangle \times 10^{-5}$
0.20	448278(10218)	0(0)	0(0)
0.10	24.08(68)	0(0)	0(0)
0.083333	8.88(43)	0.0490(183)	0.5919(221)
0.062500	2.41(23)	0.2762(355)	2.632(338)
0.05	0.7619(1403)	0.5441(373)	4.313(294)
0.041667	0.6130(912)	0.5659(319)	3.781(213)
0.035714	0.2330(74530)	0.7284(278)	4.260(162)
0.025	0.16471(4684)	0.7865(214)	3.288(89)
0.0125	0.03272(2867)	0.8858(139)	1.937(30)

Table B.1: The measurements made when varying the integrating stepsize  $\delta\tau$  with staggered fermions when investigating nroots ( $n = 1$ ,  $V = 16^4$ , Wilson gauge action,  $\beta = 5.76$ ,  $m_f = 0.02$ ,  $n_{\text{md}} = 10$ ,  $n_{\text{mc}} = 15$ ,  $\text{res}_{\text{md}} = 10^{-6}$ ,  $\text{res}_{\text{mc}} = 10^{-9}$ ,  $\delta\tau^{\text{gauge}} = 0.01$ ). The errors were found using a jackknife analysis.

$\delta\tau^{\text{fer}}$	$\delta H$	$\langle A \rangle$	$\langle E \rangle \times 10^{-5}$
0.20	77.28(127)	0(0)	0(0)
0.10	2.322(213)	0.2942(360)	1.991(487)
0.083333	1.094(148)	0.4479(368)	2.633(432)
0.0625	0.30698(818)	0.6843(280)	3.184(261)
0.05	0.02929(533)	0.8351(225)	3.222(173)
0.041667	0.07095(3799)	0.8532(180)	2.817(119)
0.035714	0.08172(4269)	0.8351(195)	2.404(112)
0.025	0.05677(3321)	0.8716(164)	1.797(68)
0.0125	0.01888(3167)	0.8843(145)	0.9425(31)

Table B.2: The measurements made when varying the integrating stepsize  $\delta\tau$  with staggered fermions when investigating nroots ( $n = 2$ ,  $V = 16^4$ , Wilson gauge action,  $\beta = 5.76$ ,  $m_f = 0.02$ ,  $n_{\text{md}} = 10$ ,  $n_{\text{mc}} = 15$ ,  $\text{res}_{\text{md}} = 10^{-6}$ ,  $\text{res}_{\text{mc}} = 10^{-9}$ ,  $\delta\tau^{\text{gauge}} = 0.01$ ). The errors were found using a jackknife analysis.



$\delta\tau^{\text{fer}}$	$\delta H$	$\langle A \rangle$	$\langle E \rangle \times 10^{-5}$
0.20	30.54(72)	0(0)	0(0)
0.10	1.283(142)	0.4224(369)	1.919(503)
0.083333	0.2032(902)	0.6092(292)	1.623(408)
0.0625	0.3291(831)	0.6926(302)	2.380(315)
0.05	0.04727(4049)	0.8516(176)	2.176(135)
0.041667	-0.002674(45069)	0.8603(175)	2.244(187)
0.035714	0.05027(3243)	0.8730(149)	1.758(95)
0.025	0.054073(3222)	0.8682(156)	1.812(63)

Table B.3: The measurements made when varying the integrating stepsize  $\delta\tau$  with staggered fermions when investigating nroots ( $n = 3$ ,  $V = 16^4$ , Wilson gauge action,  $\beta = 5.76$ ,  $m_t = 0.02$ ,  $n_{\text{md}} = 10$ ,  $n_{\text{mc}} = 15$ ,  $\text{res}_{\text{md}} = 10^{-6}$ ,  $\text{res}_{\text{mc}} = 10^{-9}$ ,  $\delta\tau^{\text{gauge}} = 0.01$ ). The errors were found using a jackknife analysis.

$\delta\tau^{\text{fer}}$	$\delta H$	$\langle A \rangle$	$\langle E \rangle \times 10^{-5}$
0.20	18.48(52)	0.000590(563)	0.3225(12)
0.10	0.8058(1246)	0.534398(354)	1.796(477)
0.083333	0.3813(833)	0.6909(314)	1.651(396)
0.0625	0.1492(469)	0.8025(213)	1.904(202)
0.05	0.08153(3644)	0.8495(177)	1.613(134)
0.041667	0.07800(3900)	0.8536(185)	1.465(148)
0.035714	0.02335(2910)	0.8969(148)	1.296(90)
0.025	0.01430(2648)	0.9035(124)	0.9220(50)

Table B.4: The measurements made when varying the integrating stepsize  $\delta\tau$  with staggered fermions when investigating nroots ( $n = 4$ ,  $V = 16^4$ , Wilson gauge action,  $\beta = 5.76$ ,  $m_t = 0.02$ ,  $n_{\text{md}} = 10$ ,  $n_{\text{mc}} = 15$ ,  $\text{res}_{\text{md}} = 10^{-6}$ ,  $\text{res}_{\text{mc}} = 10^{-9}$ ,  $\delta\tau^{\text{gauge}} = 0.01$ ). The errors were found using a jackknife analysis.

$\delta\tau^{\text{fer}}$	$\delta H$	$\langle A \rangle$	$\langle E \rangle \times 10^{-5}$
0.01	0.003961(8168)	0.9680(23)	0.6313(17hs)
0.02	0.04935(1984)	0.8979(14)	1.184(19)
0.030303	0.1494(528)	0.7643(168)	1.468(24)
0.041667	0.8621(1338)	0.5228(271)	1.309(78)
0.052632	1.592(330)	0.3730(202)	1.186(65)
0.066667	4.476(327)	0.1541(444)	0.6013(1740)
0.076923	11.81(83)	0.01058(468)	0.04704(2088)
0.090909	7842(546)	0(0)	0(0)

Table B.5: The measurements made when varying the integrating stepsize  $\delta\tau$  with staggered fermions when investigating nroots ( $n = 1$ ,  $V = 16^4$ , Wilson gauge action,  $\beta = 5.76$ ,  $m_t = 0.01$ ,  $n_{\text{md}} = 11$ ,  $n_{\text{mc}} = 16$ ,  $\text{res}_{\text{md}} = 10^{-6}$ ,  $\text{res}_{\text{mc}} = 10^{-9}$ ,  $\delta\tau^{\text{gauge}} = 0.01$ ). The errors were found using a jackknife analysis.

$\delta\tau^{\text{fer}}$	$\delta H$	$\langle A \rangle$	$\langle E \rangle \times 10^{-5}$
0.01	0.002652(5024)	0.9793(41)	0.3347(18)
0.02	0.002768(10909)	0.9634(29)	0.6186(55)
0.030303	0.05131(1467)	0.9054(134)	0.8560(127)
0.041667	0.06595(4889)	0.8580(176)	1.122(21)
0.052632	0.1294(330)	0.7909(243)	1.300(45)
0.066667	0.5001(1376)	0.6219(125)	1.267(17)
0.076923	0.6300(637)	0.5637(194)	1.326(66)
0.090909	1.589(336)	0.3676(259)	1.091(73)
0.111110	3.735(150)	0.1724(452)	0.5477(1549)
0.125	6.848(287)	0.07144(1713)	0.2121(501)
0.142857	13.27(43)	0.01146(422)	0.03778(1391)

Table B.6: The measurements made when varying the integrating stepsize  $\delta\tau$  with staggered fermions when investigating nroots ( $n = 2$ ,  $V = 16^4$ , Wilson gauge action,  $\beta = 5.76$ ,  $m_f = 0.01$ ,  $n_{\text{md}} = 11$ ,  $n_{\text{mc}} = 16$ ,  $\text{res}_{\text{md}} = 10^{-6}$ ,  $\text{res}_{\text{mc}} = 10^{-9}$ ,  $\delta\tau^{\text{gauge}} = 0.01$ ). The errors were found using a jackknife analysis.

$\delta\tau^{\text{fer}}$	$\delta H$	$\langle A \rangle$	$\langle E \rangle \times 10^{-5}$
0.010	0.000346(47810)	0.9800(36)	0.2221(18)
0.020	0.003781(8671)	0.9689(46)	0.4453(34)
0.030303	0.03485(2565)	0.9270(66)	0.5761(18)
0.041667	0.07571(2453)	0.8712(20)	0.7400(36)
0.052632	0.05448(1523)	0.8582(58)	0.9358(162)
0.066667	0.2945(200)	0.7257(214)	0.9624(276)
0.076923	0.3312(1052)	0.6754(399)	1.065(67)
0.090909	1.053(190)	0.4522(585)	0.7429(947)
0.111111	1.803(144)	0.3621(363)	0.6580(678)
0.125	2.934(74)	0.1817(306)	0.3607(608)
0.142857	5.480(383)	0.07488(1684)	0.1673(376)

Table B.7: The measurements made when varying the integrating stepsize  $\delta\tau$  with staggered fermions when investigating nroots ( $n = 3$ ,  $V = 16^4$ , Wilson gauge action,  $\beta = 5.76$ ,  $m_f = 0.01$ ,  $n_{\text{md}} = 11$ ,  $n_{\text{mc}} = 16$ ,  $\text{res}_{\text{md}} = 10^{-6}$ ,  $\text{res}_{\text{mc}} = 10^{-9}$ ,  $\delta\tau^{\text{gauge}} = 0.01$ ). The errors were found using a jackknife analysis.



$\delta\tau^{\text{fer}}$	$\delta H$	$\langle A \rangle$	$\langle E \rangle \times 10^{-5}$
0.01	0.03011(229)	0.9184(58)	0.4631(69)
0.02	0.03844(4885)	0.8817(180)	1.006(21)
0.030303	0.2315(730)	0.7281(452)	0.9727(642)
0.041667	1.392(227)	0.3946(379)	0.6940(646)
0.052632	3.546(218)	0.2278(219)	0.4996(474)
0.066667	13162(961)	0(0)	0(0)

Table B.8: The measurements made when varying the integrating stepsize  $\delta\tau$  with staggered fermions when investigating nroots ( $n = 1$ ,  $V = 16^4$ , Wilson gauge action,  $\beta = 5.76$ ,  $m_f = 0.005$ ,  $n_{\text{md}} = 11$ ,  $n_{\text{mc}} = 16$ ,  $\text{res}_{\text{md}} = 10^{-6}$ ,  $\text{res}_{\text{mc}} = 10^{-9}$ ,  $\delta\tau^{\text{gauge}} = 0.01$ ). The errors were found using a jackknife analysis.

$\delta\tau^{\text{fer}}$	$\delta H$	$\langle A \rangle$	$\langle E \rangle \times 10^{-5}$
0.030303	0.05026(326)	0.8928(144)	0.6279(91)
0.041667	0.1152(400)	0.8241(212)	0.8994(211)
0.052632	0.1229(342)	0.7944(118)	1.131(136)
0.066667	0.6057(1988)	0.5679(285)	0.9711(430)
0.076823	0.7528(1502)	0.5378(298)	0.9746(568)
0.090909	1.630(207)	0.3692(422)	0.7143(569)
0.111111	3.834(109)	0.1664(332)	0.3406(679)
0.125	7.559(204)	0.04744(2069)	0.1198(51)
0.142857	14.26(73)	0.000596(222)	0.001795(676)

Table B.9: The measurements made when varying the integrating stepsize  $\delta\tau$  with staggered fermions when investigating nroots ( $n = 2$ ,  $V = 16^4$ , Wilson gauge action,  $\beta = 5.76$ ,  $m_f = 0.005$ ,  $n_{\text{md}} = 11$ ,  $n_{\text{mc}} = 16$ ,  $\text{res}_{\text{md}} = 10^{-6}$ ,  $\text{res}_{\text{mc}} = 10^{-9}$ ,  $\delta\tau^{\text{gauge}} = 0.01$ ). The errors were found using a jackknife analysis.

$\delta\tau^{\text{fer}}$	$\delta H$	$\langle A \rangle$	$\langle E \rangle \times 10^{-5}$
0.04	0.01077(1744)	0.8931(127)	0.5149(122)
0.05	0.01554(1859)	0.8126(150)	0.6263(147)
0.0625	0.07606(3229)	0.8077(151)	0.7927(349)
0.076923	0.4824(1414)	0.6311(663)	0.8523(630)
0.090909	0.8702(667)	0.5196(327)	0.8697(1243)
0.111111	1.696(337)	0.3396(358)	0.5628(529)
0.125	3.424(208)	0.1935(0.029741)	0.3397(472)
0.142857	5.477(436)	0.09041(3597)	0.1706(476)
0.166666	12.03(38)	0.03888(17671)	0.08155(2270)

Table B.10: The measurements made when varying the integrating stepsize  $\delta\tau$  with staggered fermions when investigating nroots ( $n = 3$ ,  $V = 16^4$ , Wilson gauge action,  $\beta = 5.76$ ,  $m_f = 0.005$ ,  $n_{\text{md}} = 11$ ,  $n_{\text{mc}} = 16$ ,  $\text{res}_{\text{md}} = 10^{-6}$ ,  $\text{res}_{\text{mc}} = 10^{-9}$ ,  $\delta\tau^{\text{gauge}} = 0.01$ ). The errors were found using a jackknife analysis.

# Appendix C

## Domain Wall Results

$m_{\text{val}}$	$m_{\pi}$	$\chi^2/dof$	Fit Range
0.020000	$0.390130^{+0.002832}_{-0.003183}$	1.7895	7–15
0.030000	$0.442981^{+0.002681}_{-0.002723}$	2.25062	7–15
0.040000	$0.490935^{+0.002372}_{-0.002462}$	2.47737	7–15
$m_{\text{res}}$	$-0.007143^{+0.000681}_{-0.000675}$	0.311712	

Table C.1: Fitted values of  $m_{\pi}$  from data set 1. The errors were found using a bootstrap analysis.

$m_{\text{val}}$	$m_{\pi}$	$\chi^2/dof$	Fit Range
0.020000	$0.387446^{+0.001849}_{-0.001714}$	2.70949	7–16
0.030000	$0.441740^{+0.001637}_{-0.001560}$	2.37625	7–16
0.040000	$0.490624^{+0.001481}_{-0.001495}$	2.1108	7–16
$m_{\text{res}}$	$-0.006077^{+0.000352}_{-0.000418}$	0.736242	

Table C.2: Fitted values of  $m_{\pi}$  from data set 2. The errors were found using a bootstrap analysis.

$m_{\text{val}}$	$m_{\pi}$	$\chi^2/dof$	Fit Range
0.020000	$0.383591^{+0.002264}_{-0.002116}$	0.790022	6–16
0.030000	$0.438591^{+0.002169}_{-0.001890}$	0.884075	6–16
0.040000	$0.488178^{+0.002059}_{-0.001826}$	1.10037	6–16
$m_{\text{res}}$	$-0.005328^{+0.000342}_{-0.000404}$	0.489891	

Table C.3: Fitted values of  $m_{\pi}$  from data set 3. The errors were found using a bootstrap analysis.

$m_{\text{val}}$	$m_{\pi}$	$\chi^2/dof$	Fit Range
0.020000	$0.373471^{+0.001635}_{-0.001605}$	1.06044	8–16
0.030000	$0.428783^{+0.001560}_{-0.001553}$	1.47946	8–16
0.040000	$0.479044^{+0.001529}_{-0.001458}$	1.82003	8–16
$m_{\text{res}}$	$-0.004284^{+0.000222}_{-0.000231}$	1.06968	

Table C.4: Fitted values of  $m_{\pi}$  from data set 4. The errors were found using a bootstrap analysis.

$m_{\text{val}}$	$m_{\text{res}}$	$\chi^2/dof$	Fit Range
0.020000	$0.011883^{+0.000101}_{-0.000119}$	1.29208	7–14
0.030000	$0.011674^{+0.000097}_{-0.000109}$	1.28611	7–14
0.040000	$0.011500^{+0.000095}_{-0.000104}$	1.28701	7–14
$-m_{\text{res}}$	$0.012258^{+0.000118}_{-0.000137}$	0.0193822	

Table C.5: Fitted values of  $m_{\text{res}}$  from data set 1. The errors were found using a bootstrap analysis.



$m_{\text{val}}$	$m_{\text{res}}$	$\chi^2/dof$	Fit Range
0.020000	$0.011742^{+0.000080}_{-0.000076}$	1.46174	7–15
0.030000	$0.011549^{+0.000083}_{-0.000071}$	1.62917	7–15
0.040000	$0.011389^{+0.000079}_{-0.000070}$	1.75333	7–15
$-m_{\text{res}}$	$0.012090^{+0.000098}_{-0.000088}$	0.0302606	

Table C.6: Fitted values of  $m_{\text{res}}$  from data set 2. The errors were found using a bootstrap analysis.

$m_{\text{val}}$	$m_{\text{res}}$	$\chi^2/dof$	Fit Range
0.020000	$0.010957^{+0.000067}_{-0.000078}$	0.949423	8–14
0.030000	$0.010777^{+0.000066}_{-0.000073}$	1.02914	8–14
0.040000	$0.010626^{+0.000063}_{-0.000072}$	1.0777	8–14
$-m_{\text{res}}$	$0.011283^{+0.000083}_{-0.000090}$	0.0298768	

Table C.7: Fitted values of  $m_{\text{res}}$  from data set 3. The errors were found using a bootstrap analysis.

$m_{\text{val}}$	$m_{\text{res}}$	$\chi^2/dof$	Fit Range
0.020000	$0.010880^{+0.000059}_{-0.000059}$	1.00306	9–15
0.030000	$0.010701^{+0.000055}_{-0.000054}$	1.08747	9–15
0.040000	$0.010553^{+0.000052}_{-0.000052}$	1.20074	9–15
$-m_{\text{res}}$	$0.011199^{+0.000070}_{-0.000071}$	0.0498049	

Table C.8: Fitted values of  $m_{\text{res}}$  from data set 4. The errors were found using a bootstrap analysis.

$m_{\text{val}}$	$m_\rho$	Fit Range	$\chi^2/dof$
0.020000	$0.679102^{+0.025864}_{-0.024308}$	9–14	0.948594
0.030000	$0.708836^{+0.018933}_{-0.018083}$	9–14	1.00814
0.040000	$0.733272^{+0.014230}_{-0.013562}$	9–14	1.0781
$-m_\rho$	$0.627506^{+0.037500}_{-0.036327}$		0.0128028
$a$	$1.291533^{+0.044514}_{-0.043109}$		

Table C.9: Fitted values of  $m_\rho$  from data set 1. The errors were found using a bootstrap analysis.

$m_{\text{val}}$	$m_\rho$	Fit Range	$\chi^2/dof$
0.020000	$0.651279^{+0.016998}_{-0.015754}$	8–16	1.03006
0.030000	$0.683165^{+0.012081}_{-0.011034}$	8–16	0.977714
0.040000	$0.714115^{+0.009428}_{-0.008771}$	8–16	0.784661
$-m_\rho$	$0.588923^{+0.024832}_{-0.022735}$		0.00099057
$a$	$1.393576^{+0.029527}_{-0.027342}$		

Table C.10: Fitted values of  $m_\rho$  from data set 2. The errors were found using a bootstrap analysis.

$m_{\text{val}}$	$m_\rho$	Fit Range	$\chi^2/dof$
0.020000	$0.660201^{+0.014139}_{-0.013475}$	8–16	1.02566
0.030000	$0.684790^{+0.010323}_{-0.010037}$	8–16	1.22492
0.040000	$0.711636^{+0.008139}_{-0.008040}$	8–16	1.30742
$-m_\rho$	$0.607705^{+0.019695}_{-0.019522}$		0.00759155
$a$	$1.330925^{+0.023070}_{-0.023143}$		

Table C.11: Fitted values of  $m_\rho$  from data set 3. The errors were found using a bootstrap analysis.

$m_{\text{val}}$	$m_\rho$	Fit Range	$\chi^2/dof$
0.020000	$0.654280^{+0.011103}_{-0.009716}$	8–15	1.25851
0.030000	$0.680018^{+0.008020}_{-0.007365}$	8–15	1.81491
0.040000	$0.707011^{+0.006604}_{-0.005764}$	8–15	2.34282
$-m_\rho$	$0.600964^{+0.015382}_{-0.013924}$		0.00410566
$a$	$1.347540^{+0.018276}_{-0.016700}$		

Table C.12: Fitted values of  $m_\rho$  from data set 4. The errors were found using a bootstrap analysis.

$m_{\text{res}}$
$-0.004611^{+0.000862}_{-0.001187}$

Table C.13: Fitted value of  $m_{\text{res}}$  from dynamical  $m_\pi$  extrapolation using R data sets. The errors were found using a bootstrap analysis.

$m_{\text{res}}$
$-0.002054^{+0.000359}_{-0.000399}$

Table C.14: Fitted value of  $m_{\text{res}}$  from dynamical  $m_\pi$  extrapolation using RHMC data sets. The errors were found using a bootstrap analysis.

$m_{\text{val}}$	$m_{\text{res}}$
$-m_{\text{res}}$	$0.010414^{+0.000173}_{-0.000172}$

Table C.15: Fitted value of  $m_{\text{res}}$  from the R data sets. The errors were found using a bootstrap analysis.

$m_{\text{val}}$	$m_{\text{res}}$
$-m_{\text{res}}$	$0.010371^{+0.000146}_{-0.000147}$

Table C.16: Fitted value of  $m_{\text{res}}$  from the RHMC data sets. The errors were found using a bootstrap analysis.

$m_{\text{val}}$	$m_{\rho}$
$-m_{\rho}$	$0.587131^{+0.031762}_{-0.032007}$
$a$	$1.399778^{+0.114901}_{-0.099073}$

Table C.17: Fitted value of  $m_{\rho}$  from the R data sets. The errors were found using a bootstrap analysis.

$m_{\text{val}}$	$m_{\rho}$
$-m_{\rho}$	$0.594444^{+0.023193}_{-0.022226}$
$a$	$1.364793^{+0.072267}_{-0.068793}$

Table C.18: Fitted value of  $m_{\rho}$  from the RHMC data sets. The errors were found using a bootstrap analysis.



# Appendix D

## QCDOC Implementation

QCDOC has been designed specifically to perform lattice QCD simulations very efficiently, therefore the implementation of the above algorithms was done with the QCDOC hardware in mind. Here the QCDOC implementation of some of the performance critical code is explained.

### D.1 Optimising Matrix Inversion for QCDOC

The conjugate gradient algorithm, and in general all Krylov inversion methods, operate at an efficiency lower than that of the matrix-vector product. This is because of the decreased computation to bandwidth ratio of the linear algebra compared to the matrix-vector product. This problem can be alleviated by calculating the scalar products concurrently with VAXPY operations, but this overhead can not be eliminated completely (e.g., for ASQTAD fermions, matrix-vector product efficiency  $\approx 40\%$ , conjugate gradient  $\approx 38\%$ ). This may seem like a small reduction in operating efficiency, however, for the case of the multi-shift solver, the problem is much more severe due to the increased linear algebra per

iteration. The double pass formulation [83] of the multi-shift solver removes all of the overhead of the linear algebra, but has some severe caveats which render it unsuitable for RHMC. The most severe of these is that it will only evaluate the summed partial fraction, as opposed to the individual poles. This renders it unuseable for the RHMC force calculation. The conventional multi-shift solver must therefore be optimised for QCDOC.

The multi-shift solver has  $2n+1$  VAXPY operations as well as two scalar products per iteration. The VAXPY operations are

$$p_{k+1}^\sigma = \zeta_k^\sigma r_k - \alpha_k^\sigma p_k^\sigma \quad (\text{D.1})$$

$$r_{k+1} = r_k - \beta_k A p_{k+1} \quad (\text{D.2})$$

$$x_{k+1}^\sigma = x_k^\sigma + \beta_k^\sigma p_{k+1}^\sigma, \quad (\text{D.3})$$

and the required scalar products are

$$c_k = \langle r_k, r_k \rangle \quad (\text{D.4})$$

$$d_k = \langle \psi_k, A \psi_k \rangle. \quad (\text{D.5})$$

The scalar products can be calculated concurrently with the update to the residual vector (equation (D.2)), and when the  $A p_{k+1}$  matrix-vector product is taken respectively. This leaves just equations D.1 and D.3 upon which improvements must be made.

The QCDOC floating point unit has in addition to standard operations, an instruction which allows the evaluation of  $v = ax + y$ , this is the multiply-add instruction [84]. Notice that individual elements of equations D.2 and D.3 can be evaluated using only a single multiply-add floating point instruction, however, the same cannot be said of the D.1 since there is an extra scalar multiplication involved. This equation could be recast into this form if the equation was divided through by  $\alpha_k^\sigma$ , this factor would have to be restored in the calculation

of  $x_{k+1}^\sigma$ .

$$\hat{p}_{k+1}^\sigma = \frac{\zeta_k^\sigma}{\alpha_k^\sigma} r_k - p_k^\sigma \quad (\text{D.6})$$

$$x_{k+1}^\sigma = x_k^\sigma + \alpha_k^\sigma \beta_k \hat{p}_{k+1}^\sigma \quad (\text{D.7})$$

But since only the value of  $\hat{p}_k^\sigma$  from previous iterations would be known, this factor has to be included from all previous iterations. The equations required are thus

$$\hat{p}_{k+1}^\sigma = \zeta_k^\sigma \left( \prod_{l=1}^k \alpha_l^\sigma \right)^{-1} r_k - \hat{p}_k^\sigma, \quad (\text{D.8})$$

$$r_{k+1} = r_k - \left( \prod_{l=1}^k \alpha_l \right) \beta_k A \hat{p}_{k+1}, \quad (\text{D.9})$$

$$x_{k+1}^\sigma = x_k^\sigma + \left( \prod_{l=1}^k \alpha_l^\sigma \right) \beta_k^\sigma \hat{p}_{k+1}^\sigma, \quad (\text{D.10})$$

which are all in the form of multiply-add, and hence optimal for evaluation on QCDOC.

If the  $\hat{p}$  update is cycled to the end of the loop, as opposed to the beginning, then

$$x_{k+1}^\sigma = x_k^\sigma + \left( \prod_{l=1}^k \alpha_l^\sigma \right) \beta_k^\sigma \hat{p}_{k+1}^\sigma \quad (\text{D.11})$$

$$\hat{p}_{k+2}^\sigma = \zeta_{k+1}^\sigma \left( \prod_{l=1}^{k+1} \alpha_{k+1}^\sigma \right)^{-1} r_{k+1} - \hat{p}_{k+1}^\sigma. \quad (\text{D.12})$$

It can now be seen that there is a common input vector to the  $x$  and  $\hat{p}$  vector updates, namely  $\hat{p}_k$ . This suggests that if these operations were combined, then the computation to bandwidth ratio would be increased by a factor of  $\frac{4}{3}$ . Of course the  $x$  update must be calculated before the  $\hat{p}$  update to reproduce the algorithm correctly.

This procedure is of course applicable to any architecture, but the QCDOC in particular is problematic for such a recasting. When fields are stored in the EDRAM, the prefetching EDRAM controller will only simultaneously operate

with two input streams, and two output streams. For standard VAXPY operations this requirement is of course met, but for this proposed scheme this is not true. Proceed naïvely then would result with an overall reduction in performance because of the need to switch input streams continuously. Fortunately, it is possible to circumvent this problem entirely.

The solution is to interleave two of the input streams into a single double-length stream, that way it is possible to load three streams of data simultaneously without a bottleneck. Obviously, then all operations which concern the interleaved data would have to be re-written to reflect this. For the unshifted shift updates, any two of  $r$ ,  $\hat{p}$  or  $x$  can be interleaved, however, since the Dirac operator must act on the  $\hat{p}$  vector it is easiest to interleave the  $r$  and  $x$  vectors. The shifted  $p^\sigma$  and  $x^\sigma$  are interleaved, but since the Dirac operator is not applied here this can be done without complication.

The only slight consideration that has to be made when using such an interleaving strategy is what coarseness of interleaving to use. The coarseness of the interleaving must be such that a cache miss does not occur when performing an operation using only one vector of the interleaved data stream, e.g., when the  $r$  vector is updated, the other data stream required is the  $Ap$  vector, and not the  $x$  vector with which the  $r$  is interleaved. On QCDOC a cache line has length 768 bytes [85], this implies that an interleave of 12 doubles should be optimal.

For the case of the ASQTAD (or staggered) fermion matrix  $\mathcal{M}$  that is used in dynamical simulations, the mass parameter is a multiple of the identity matrix. Hence it is just a constant shift, like the shifts applied when using rational approximations. Therefore, the mass parameter can be renormalised to include the smallest shift that appears in the rational approximation, and the smallest shift subtracted from all shifts, i.e.,

$$m \rightarrow \sqrt{4m^2 + \beta_0}, \quad (\text{D.13})$$



$$\beta_k \rightarrow \beta_k - \beta_0. \quad (\text{D.14})$$

By renormalising the mass like this, some linear algebra is eliminated in the multi-shift solver, i.e., when calculating  $p_i A p_i$  as described in §3.8.2. Unfortunately this optimisation cannot be applied to Wilson or domain wall fermion because the mass parameter of  $M^\dagger M$  is not a simple shift.

Solver	$N_{\text{mass}}$	Percentage of peak
naïve	1	30%
optimised	1	39%
naïve	10	20%
optimised	10	35%

Table D.1: The percentage of peak performance obtained from using the multi-mass solver on a QCDOC with ASQTAD fermions. The naïve solver uses optimised assembler for the linear algebra, but no other optimisations.

## D.2 Asqtad RHMC Force Implementation

The RHMC ASQTAD force term has been optimised for use on the QCDOC. The calculation involves many SU(3) operations of the form

$$X = X + cUV, \quad (\text{D.15})$$

$$X = X + cUVW, \quad (\text{D.16})$$

where  $X$ ,  $U$ ,  $V$  and  $W$  are SU(3) matrices, and  $c$  is a scalar. These operations are the SU(3) equivalents of the VAXPY operations define in appendix D.1. Each matrix-matrix multiply involves 198 flops, and a scalar-matrix multiply 18 flops. When calculating simple matrix-matrix multiply operations, it is possible to obtain around 80% of the peak performance of the floating point unit, however,

the inclusion of the summation spoils this performance considerably because of the reduced flops to load ratio. As a result the ASQTAD force calculation has an efficiency of around 36% of peak performance.

As an interesting side note, there is actually a method for cutting down the number of floating point operations required to perform  $SU(3)$  multiplication. The naïve evaluation of  $SU(3) \times SU(3)$  involves 198 flops, these originate from four  $3 \times 3$  matrix multiplies, and two matrix additions, i.e.,

$$X = AB \quad (\text{D.17})$$

$$= (A_r + iA_i)(B_r + iB_i) \quad (\text{D.18})$$

$$= A_r B_r - A_i B_i + i(A_r B_i + A_i B_r), \quad (\text{D.19})$$

where  $X, A, B \in SU(3)$ . It is possible to rewrite multiplication of complex scalars in terms of 3 scalar multiplications and 5 additions, instead of the usual 4 multiplications and 2 additions [23], and there is no reason why this procedure cannot be applied to matrices. Observe equation (D.19) can be rewritten

$$X = A_r B_r - A_i B_i + i((A_r + A_i)(B_r + B_i) - A_r B_r - A_i B_i), \quad (\text{D.20})$$

which only involves three matrix multiplications and five matrix additions. Hence, with this reformulation, the  $SU(3)$  multiplication only costs 180 flops since matrix addition is cheaper than multiplication. If a calculation is being performed where the same matrices are being repeatedly reused, e.g., the ASQTAD RHMC force calculation, then it could be beneficial to precalculate the sum of real and imaginary components for all matrices. In the limit of infinite data reuse, this would lead to the multiplication operation only costing 162 flops, representing an 18% reduction over the naïve method. This method could be useful in reducing the cost of the ASQTAD force term calculation, but it is hard to implement efficiently on a QCDOC because of the reduced register reuse compared to the naïve calculation and so has not been exploited in this work.

# Bibliography

- [1] H. J. Rothe. Lattice gauge theories: An Introduction. *World Sci. Lect. Eprints Phys.*, 59:1–512, 1997.
- [2] Kenneth G. Wilson. Quarks and strings on a lattice. *New Phenomena In Subnuclear Physics. Part A. Proceedings of the First Half of the 1975 International School of Subnuclear Physics, Erice, Sicily, July 11 - August 1, 1975*, ed. A. Zichichi, Plenum Press, New York, 1977, p. 69, CLNS-321.
- [3] Holger Bech Nielsen and M. Ninomiya. No go theorem for regularizing chiral fermions. *Phys. Lett.*, B105:219, 1981.
- [4] John B. Kogut and Leonard Susskind. Hamiltonian formulation of Wilson’s lattice gauge theories. *Phys. Rev.*, D11:395, 1975.
- [5] B. Bunk, M. Della Morte, K. Jansen, and F. Knechtli. Locality with staggered fermions. *Nucl. Phys.*, B697:343–362, 2004 (hep-lat/0403022).
- [6] A. Hart and E. Muller. The locality of the square-root method for improved staggered quarks. *Phys. Rev.*, D70:057502, 2004 (hep-lat/0406030).
- [7] Kenneth G. Wilson. Confinement of quarks. *Phys. Rev.*, D10:2445–2459, 1974.
- [8] Tetsuya Takaiishi. Heavy quark potential and effective actions on blocked configurations. *Phys. Rev.*, D54:1050–1053, 1996.

- [9] P. de Forcrand et al. Renormalization group flow of SU(3) lattice gauge theory: Numerical studies in a two coupling space. *Nucl. Phys.*, B577:263–278, 2000 (hep-lat/9911033).
- [10] Mark G. Alford, W. Dimm, G. P. Lepage, G. Hockney, and P. B. Mackenzie. Lattice QCD on small computers. *Phys. Lett.*, B361:87–94, 1995 (hep-lat/9507010).
- [11] Kostas Orginos, Doug Toussaint, and R. L. Sugar. Variants of fattening and flavor symmetry restoration. *Phys. Rev.*, D60:054503, 1999 (hep-lat/9903032).
- [12] Satchidananda Naik. On-shell improved lattice action for qcd with suskind fermions and asymptotic freedom scale. *Nucl. Phys.*, B316:238, 1989.
- [13] Claude W. Bernard et al. Quenched hadron spectroscopy with improved staggered quark action. *Phys. Rev.*, D58:014503, 1998 (hep-lat/9712010).
- [14] G. Peter Lepage and Paul B. Mackenzie. On the viability of lattice perturbation theory. *Phys. Rev.*, D48:2250–2264, 1993 (hep-lat/9209022).
- [15] Paul H. Ginsparg and Kenneth G. Wilson. A remnant of chiral symmetry on the lattice. *Phys. Rev.*, D25:2649, 1982.
- [16] Rajamani Narayanan and Herbert Neuberger. Chiral fermions on the lattice. *Phys. Rev. Lett.*, 71:3251–3254, 1993 (hep-lat/9308011).
- [17] Vadim Furman and Yigal Shamir. Axial symmetries in lattice QCD with Kaplan fermions. *Nucl. Phys.*, B439:54–78, 1995 (hep-lat/9405004).
- [18] Pavlos M. Vranas. Chiral symmetry restoration in the schwinger model with domain wall fermions. *Phys. Rev.*, D57:1415–1432, 1998 (hep-lat/9705023).
- [19] Y. Aoki et al. Domain wall fermions with improved gauge actions. *Phys. Rev.*, D69:074504, 2004 (hep-lat/0211023).



- [20] T. Blum et al. Quenched lattice qcd with domain wall fermions and the chiral limit. *Phys. Rev.*, D69:074502, 2004 (hep-lat/0007038).
- [21] Herbert Neuberger. Vector like gauge theories with almost massless fermions on the lattice. *Phys. Rev.*, D57:5417–5433, 1998 (hep-lat/9710089).
- [22] Rajan Gupta. Introduction to lattice QCD. 1997 (hep-lat/9807028).
- [23] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1992.
- [24] Ferenc Niedermayer. Exact chiral symmetry, topological charge and related topics. *Nucl. Phys. Proc. Suppl.*, 73:105–119, 1999 (hep-lat/9810026).
- [25] Anna Hasenfratz and Francesco Knechtli. Flavor symmetry and the static potential with hypercubic blocking. *Phys. Rev.*, D64:034504, 2001 (hep-lat/0103029).
- [26] Claude Bernard et al. Topological susceptibility with the improved asqtad action. *Phys. Rev.*, D68:114501, 2003 (hep-lat/0308019).
- [27] S. Okubo. Phi meson and unitary symmetry model. *Phys. Lett.*, 5:165–168, 1963.
- [28] G. Zweig. Fractional charged particles and SU(6). *Symmetries in Elementary Particle Physics*, 1965.
- [29] Jugoro Iizuka. Systematics and phenomenology of meson family. *Prog. Theor. Phys. Suppl.*, 37:21–34, 1966.
- [30] N. Cabibbo and E. Marinari. A new method for updating su(n) matrices in computer simulations of gauge theories. *Phys. Lett.*, B119:387–390, 1982.

- [31] I. Montvay and G. Munster. Quantum fields on a lattice. *Cambridge, UK: Univ. Pr. (1994) 491 p. (Cambridge monographs on mathematical physics)*.
- [32] D. H. Weingarten and D. N. Petcher. Monte carlo integration for lattice gauge theories with fermions. *Phys. Lett.*, B99:333, 1981.
- [33] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid monte carlo. *Phys. Lett.*, B195:216–222, 1987.
- [34] R. T. Scalettar, D. J. Scalapino, and R. L. Sugar. New algorithm for the numerical simulation of fermions. *Phys. Rev.*, B34:7911–7917, 1986.
- [35] A. D. Kennedy and Pietro Rossi. Classical mechanics on group manifolds and applications to hybrid monte carlo. *Nucl. Phys.*, B327:782, 1989.
- [36] 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.
- [37] P. J. Rossky, J. D. Doll, and H. L. Friedman. Smart Monte-Carlo Simulation. *J. Chem. Phys.*, 69:5628, 1978.
- [38] S. Duane and J. B. Kogut. The theory of hybrid stochastic algorithms. *Nucl. Phys.*, B275:398, 1986.
- [39] S. Aoki et al. Polynomial hybrid Monte Carlo algorithm for lattice QCD with odd number of flavors. *Phys. Rev.*, D65:094507, 2002 (hep-lat/0112051).
- [40] Tetsuya Takaishi and Philippe de Forcrand. Testing and tuning new symplectic integrators for hybrid Monte Carlo algorithm in lattice QCD. 2005 (hep-lat/0505020).
- [41] Philippe de Forcrand and Tetsuya Takaishi. Fast fermion Monte Carlo. *Nucl. Phys. Proc. Suppl.*, 53:968–970, 1997 (hep-lat/9608093).

- [42] Roberto Frezzotti and Karl Jansen. A polynomial hybrid Monte Carlo algorithm. *Phys. Lett.*, B402:328–334, 1997 (hep-lat/9702016).
- [43] A. D. Kennedy, Ivan Horvath, and Stefan Sint. A new exact method for dynamical fermion computations with non-local actions. *Nucl. Phys. Proc. Suppl.*, 73:834–836, 1999 (hep-lat/9809092).
- [44] Claude W. Bernard et al. The QCD spectrum with three quark flavors. *Phys. Rev.*, D64:054506, 2001 (hep-lat/0104002).
- [45] A. D. Kennedy. Approximation theory for matrices. *Nucl. Phys. Proc. Suppl.*, 128C:107–116, 2004 (hep-lat/0402037).
- [46] A. D. Kennedy and J. Kuti. Noise without noise: A new monte carlo method. *Phys. Rev. Lett.*, 54:2473–2476, 1985.
- [47] S. Aoki et al. An exact algorithm for any-flavor lattice QCD with Kogut-Susskind fermion. *Comput. Phys. Commun.*, 155:183–208, 2003 (hep-lat/0208058).
- [48] Steven A. Gottlieb. Cost of dynamical quark simulations with improved staggered quarks. *Nucl. Phys. Proc. Suppl.*, 106:189–190, 2002 (hep-lat/0112039).
- [49] A. D. Kennedy, Ivan Horvath, and Stefan Sint. Monte Carlo actions for Non-Local Actions I. *In preparation*.
- [50] Massimo Campostrini and Paolo Rossi. A comparison of numerical algorithms for dynamical fermions. *Nucl. Phys.*, B329:753, 1990.
- [51] J. C. Sexton and D. H. Weingarten. Hamiltonian evolution for the hybrid Monte Carlo algorithm. *Nucl. Phys.*, B380:665–678, 1992.

- [52] 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 (hep-lat/0209037).
- [53] M. R. Hestenes and E. Stiefel. *J. Res. Nat. Bur. Standards*, 49:409, 1952.
- [54] A. Frommer, V. Hannemann, B. Nockel, T. Lippert, and K. Schilling. Accelerating Wilson fermion matrix inversions by means of the stabilized biconjugate gradient algorithm. *Int. J. Mod. Phys.*, C5:1073–1088, 1994 (hep-lat/9404013).
- [55] Beat Jegerlehner. Krylov space solvers for shifted linear systems. 1996 (hep-lat/9612014).
- [56] 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 (hep-lat/9509012).
- [57] Y. Aoki et al. Lattice QCD with two dynamical flavors of domain wall fermions. 2004 (hep-lat/0411006).
- [58] E. Y. Remez. General computational methods of chebyshev approximation. *US Atomic Energy Commission*, 1962.
- [59] E. I. Zolotarev. *Zap. Imp. Akad. Nauk. St. Petersburg*, 30:5, 1877.
- [60] *GNU Multiple Precision Arithmetic Library*. <http://www.swox.com/gmp>.
- [61] Martin Hasenbusch. Speeding up the Hybrid-Monte-Carlo algorithm for dynamical fermions. *Phys. Lett.*, B519:177–182, 2001 (hep-lat/0107019).
- [62] M. Hasenbusch and K. Jansen. Speeding up the Hybrid-Monte-Carlo algorithm for dynamical fermions. *Nucl. Phys. Proc. Suppl.*, 106:1076–1078, 2002 (hep-lat/0110180).



- [63] M. Hasenbusch and K. Jansen. Speeding up the HMC: QCD with clover-improved Wilson fermions. *Nucl. Phys. Proc. Suppl.*, 119:982–984, 2003 (hep-lat/0210036).
- [64] M. Hasenbusch and K. Jansen. Speeding up lattice QCD simulations with clover-improved wilson fermions. *Nucl. Phys.*, B659:299–320, 2003 (hep-lat/0211042).
- [65] A. D. Kennedy and Brian Pendleton. Cost of the generalised hybrid Monte Carlo algorithm for free field theory. *Nucl. Phys.*, B607:456–510, 2001 (hep-lat/0008020).
- [66] A. D. Kennedy and Brian Pendleton. Cost of generalised HMC algorithms for free field theory. *Nucl. Phys. Proc. Suppl.*, 83:816–818, 2000 (hep-lat/0001031).
- [67] R. G. Edwards, Ivan Horvath, and A. D. Kennedy. Non-reversibility of molecular dynamics trajectories. *Nucl. Phys. Proc. Suppl.*, 53:971–973, 1997 (hep-lat/9608020).
- [68] R. G. Edwards, Ivan Horvath, and A. D. Kennedy. Instabilities and non-reversibility of molecular dynamics trajectories. *Nucl. Phys.*, B484:375–402, 1997 (hep-lat/9606004).
- [69] Karl Jansen and Chuan Liu. Study of Liapunov exponents and the reversibility of molecular dynamics algorithms. *Nucl. Phys. Proc. Suppl.*, 53:974–976, 1997 (hep-lat/9607057).
- [70] Balint Joo et al. Instability in the molecular dynamics step of hybrid Monte Carlo in dynamical fermion lattice QCD simulations. *Phys. Rev.*, D62:114501, 2000 (hep-lat/0005023).
- [71] G. H. Golub and C. F. V. Loan. *Matrix Computations*. The Johns Hopkins University Press, 1996.

- [72] Martin Hasenbusch. Full QCD algorithms towards the chiral limit. *Nucl. Phys. Proc. Suppl.*, 129:27–33, 2004 (hep-lat/0310029).
- [73] J. B. Kogut and D. K. Sinclair. Finite dt dependence of the Binder cumulants for 3-flavor QCD at finite temperature and isospin density. 2005 (hep-lat/0504003).
- [74] M. A. Clark, B. Joo, and A. D. Kennedy. Comparing the R algorithm and RHMC for staggered fermions. *Nucl. Phys. Proc. Suppl.*, 119:1015–1017, 2003 (hep-lat/0209035).
- [75] Thomas A. DeGrand and Pietro Rossi. Conditioning techniques for dynamical fermions. *Comput. Phys. Commun.*, 60:211–214, 1990.
- [76] Ronald B. Morgan and Walter Wilcox. Deflation of eigenvalues for GMRES in lattice QCD. *Nucl. Phys. Proc. Suppl.*, 106:1067–1069, 2002 (hep-lat/0109009).
- [77] Dean Darnell, Ronald B. Morgan, and Walter Wilcox. Deflation of eigenvalues for iterative methods in lattice QCD. *Nucl. Phys. Proc. Suppl.*, 129:856–858, 2004 (hep-lat/0309068).
- [78] Stephen M. Pickles. Block algorithms for quark propagator calculation. *Nucl. Phys. Proc. Suppl.*, 63:961–963, 1998 (hep-lat/9709082).
- [79] P. de Forcrand. UV-filtered fermionic Monte Carlo. *Nucl. Phys. Proc. Suppl.*, 73:822–824, 1999 (hep-lat/9809145).
- [80] Anna Hasenfratz and Thomas A. DeGrand. Heavy dynamical fermions in lattice QCD. *Phys. Rev.*, D49:466–473, 1994 (hep-lat/9304001).
- [81] Constantia Alexandrou, Philippe de Forcrand, Massimo D’Elia, and Haralambos Panagopoulos. Efficiency of the UV-filtered multiboson algorithm. *Phys. Rev.*, D61:074503, 2000 (hep-lat/9906029).

- [82] A. M. Ferrenberg and R. H. Swendsen. New monte carlo technique for studying phase transitions. *Phys. Rev. Lett.*, 61:2635–2638, 1988.
- [83] Herbert Neuberger. Minimizing storage in implementations of the overlap lattice-Dirac operator. *Int. J. Mod. Phys.*, C10:1051–1058, 1999 (hep-lat/9811019).
- [84] K. Dockser. ‘Honey, I Shrunk the Supercomputer’, - The PowerPC 440 FPU Brings Supercomputing to IBM’s Blue Logic Library. *IBM MicroNews*, 7(4):27–29, 2001.
- [85] IBM Corporation. *Book E: Enhanced PowerPC Architecture*. March 2000.