THE UNIVERSITY
of LIVERPOOL

Fermion Determinants in Lattice QCD

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor in Philosophy by Christopher Andrew Johnson

November, 2001

Declaration

I hereby declare that all the work described in this thesis is the results of my own research activities unless reference is given to others. None of this material has been previously submitted to this or any other University. All work was carried out in the Theoretical Physics Division during the period of October 1998 to November 2001.

Contributions from this work have been published, or are awaiting publication, elsewhere in the following references:

[1] E Cahill, A.C. Irving, C.A. Johnson & J. Sexton and the UKQCD collaboration hep-lat/9909131, Nucl.Phys.Proc.Suppl. 83 (2000) 825-827 'Numerical stability of Lánczos methods'

[2] E Cahill, A.C. Irving, C.A. Johnson & J. Sexton and the UKQCD collaboration, LTH/517, 'Calculating Lattice Fermion Determinants by Quadrature'

Acknowledgements

First and foremost I would like to thank my supervisor Dr. Alan Irving for his patience and guidance during this project and for his informative advice throughout the whole of my PhD. I would also like to thank Tim Jones and Ian Jack for their help early on. Thanks also go to Craig McNeile for many useful discussions, and to Hartmut Wittig for the same and for selling me a Fiat.

Outside of the department, I would particularly like to thank James Sexton for his help and insight during our collaboration together. Also thanks go to Bálint Joó for more of the same. There are many people I have met at various workshops and meetings whom I would like to thank; BUSSTEPP I found to be a very rewarding experience, in particular I would like to say thank you to Olalla for her continued friendship.

I would like to thank Simon, Kieran, David, Jude, Roz and Gavin, without whom I would never have visited the Everyman so much, and needless to say would never have had so much fun. I would particularly like to say a big thank you to Rowan for her love and support, particularly during the most crucial time of my writing-up.

Thanks go to to all those in the OAC who have constantly provided me with the opportunity to climb mountains and drink lots of beer. There are many others in Liverpool and elsewhere I would like to name, but I'm mean enough to try and squeeze this all onto one page.

Finally I would like to say the biggest 'thank you' to my parents and the rest of my family, without whom, of course, none of this would have been possible.

Abstract

The main topic of this thesis concerns efficient algorithms for the calculation of determinants of the kind of matrix typically encountered in lattice QCD. In particular an efficient method for calculating the fermion determinant is described. Such a calculation is useful to illustrate the effects of light dynamical (virtual) quarks.

The methods employed in this thesis are stochastic methods, based on the Lánczos algorithm, which is used for the solution of large, sparse matrix problems via a partial tridiagonalisation of the matrix. Here an implementation is explored which requires less exhaustive treatment of the matrix than previous Lánczos methods. This technique exploits the analogy between the Lánczos tridiagonalisation algorithm and Gaussian quadrature in order to calculate the fermion determinant. A technique for determining a number of the eigenvalues of the matrix is also presented.

A demonstration is then given of how one can improve upon this estimate considerably using variance reduction techniques, reducing the variance by a factor of order 100 with a further, equal amount of work. The variance reduction method is a two-stage process, involving a Chebyshev approximation to the quantity in question and then the subtraction of traceless operators.

The method is applied to the fermion determinant for non-perturbatively improved Wilson fermions on a $16^3 \times 32$ lattice. It is also applicable to a wider class of matrix operators.

Finally we discuss how dynamical quark effects may be simulated in a Monte Carlo process with an effective partitioning of low and high eigenmodes. This may be done via selective updating of a trial configuration which highlights the physically relevant effects of light quark modes.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Quantum Chromodynamics

1.1.1 Introduction

In nature there are four fundamental forces, the strongest of which is the short-range force which holds nuclei together in atoms and is known as the 'strong force', being some 100 times stronger than the electromagnetic force - the force which attempts to repel the positively charged protons of the nuclei apart [3, 4, 5]. The other two forces of nature being the familiar gravitational force and the weak force, the most familiar process affected by this force is that of beta decay.

Originally, the electric force and the magnetic force were thought of as separate forces until in the 1860's Maxwell found a set of equations which unified these two forces into one, electromagnetic, force. In 1967 Salam and Weinberg [6] managed to construct a single mathematical formalism which explained both the weak force and the electromagnetic force. This work was then extended in 1970 by Glashow [7] and these two forces are now sometimes referred to as one 'electroweak force'.

Of course physicists would dearly love to unite all four forces into one 'Theory of Everything' (TOE). Some believe in a 'Grand Unified Theorem' (GUT) which, on some very large energy scale (known as the *GUT-scale*), the strong force and electroweak force are united.

All four of the forces act via bosons known as *mediators*. In the case of the electromagnetic force these mediators are the massless *photons*. As a consequence of the masslessness of photons, the electromagnetic force is a long-range force. By analogy, the mediator of the gravitational force is known as the *graviton*. The graviton is also massless, hence the gravitational force is a long-range force. The weak force interacts via mediators known as the *intermediate vector bosons*. These particles are massive. Consequently the weak force is a short-range force.

In the case of the strong force, the mediators are the pions (or π-mesons). These pions are mesons, first predicted by Yukawa [8] in 1935, and eventually discovered in 1947 [9][1]. There are three varieties of pion: π^0, π^- and π^+. The protons and neutrons interact by exchanging pions. For example, if a proton emits a positively charged pion (π^+), its charge becomes neutral and it in fact becomes a neutron. However, almost immediately, a neutron will gain this pion and become a proton. A similar reverse process happens as a neutron emits a π^-. Neutrons are continuously exchanging neutral π^- particles with each other, as are protons. So whilst the individual nucleons may lose their identity, there is an overall conservation in the number of protons and neutrons.

However, as we shall see, it turns out that these exchange particles are not truly fundamental particles.

In 1961 Gell-Mann [10] (and independently Ne'eman) came up with his 'eightfold way'

[1] In 1936 Anderson discovered the muon - it was initially thought that this was Yukawa's particle, but it turned out to be a particle which is basically a heavy electron.

which described a simple arrangement consisting of all the baryons and mesons which were known to exist at that time. These particles were arranged according to their properties in a manner reminiscent of Mendeleyev's periodic table, devised a century earlier. However, there were some gaps in the assortment of particles already known to exist. In particular, the particle known as the Ω^- had not yet been discovered, although the model was able to predict its charge, mass and lifetime. In 1964 this particle was discovered as predicted and thus credence had been given to a theory which had pre-empted experiment.

The eightfold way presented a succinct way of classifying particles according to their properties, but said nothing of why these baryons and mesons existed as they did. In 1964, Gell-Mann (and independently Zweig) came up with the idea of fractionally charged elementary particles known as *quarks*, and thus the quark model was born.

The quark model asserts that every baryon is made up of three quarks (every anti-baryon is made up of three anti-quarks), and that every meson is made up of a quark, q, and an anti-quark, $\bar{q}$. Initially, three quarks were required to account for all of the hadrons (baryons and mesons) that were known to exist. These became known as up u, down d and strange s, with charges $\frac{2}{3}$, $-\frac{1}{3}$ and $-\frac{1}{3}$ respectively. The up and down quarks were attributed a 'strangeness', $S = 0$, while the strange quark has $S = 1$. All have spin $= \frac{1}{2}$ and so are fermions. The corresponding anti-quarks have opposite charge and strangeness.

It turned out however that quarks were only ever discovered experimentally in such bound states $q\bar{q}$ (mesons) and qqq (baryons). Consequently, we had a theory in which we could not observe the most fundamental constituent. This unexplained *confinement* of quarks led many physicists to be sceptical about the quark model. More importantly, the quark model also appeared to violate Pauli's exclusion principle. For example, the Δ^{++} baryon was said to consist of three identical up quarks. Pauli's exclusion principle forbids this behaviour for fermions.

It was soon noticed that the quark model had powerful implications in predicting new particles. As many new particles were predicted, and then duly discovered, the quark model was given more credence, but it was clear that it was too simple a model to describe what was happening. It also soon became clear that three quarks would not be enough to explain all of the observed hadrons, and in fact we would need six. The following quarks were added to the model: charm, c; bottom (or beauty), b; top (or truth), t.

In order to avoid a violation of Pauli's exclusion principle, we must attribute a further quantum number to our quarks, such that three otherwise identical quarks can form a bound state. This quantum number is labelled *colour*. Thus the theory of Quantum *Chromo*dynamics (QCD) was established, in analogy with the relatively trivial theory of Quantum Electrodynamics (QED) which describes the electromagnetic force.

Quantum Chromodynamics represents the underlying theory of the strong force. It is based on the $SU(3)$ gauge group which describes a theory of three colours of quarks. Conventionally, but arbitrarily, these are labelled red, green and blue. These quarks interact via gluons which themselves carry colour charge and interact not only with quarks, but also with themselves.

Each gluon carries a unit of colour, and one of anti-colour. This implies that there are nine gluons. However, a colourless gluon has never been seen, and so in fact we only have eight.

All free particles are colourless, *i. e.* they either exist as a single colour and its anti-colour, or consist of all three colours, thus producing a 'colourless' combination, by analogy with the real concept of colour.

When dealing with a gauge theory, we deal with it from the point of view of Lagrangian mechanics. In the early 1940's this formalism led Feynman to the idea of path integrals, where every possible path which can be taken by a particle is associated with a probability, interpreted as an action. In order to make physical measurements, one must sum over *all* possible paths.

If we naïvely perform calculations in QED or QCD, we will often find that our answers diverge. Later in the 1940's, a number of physicists came up with the idea of *renormalisation*. This is the idea that it is possible to absorb the infinities which appear into non-physical quantities, which can quite legitimately become infinite.

As a consequence of renormalisation, at high energies the corresponding coupling constant of the strong force (a parameter which describes the strength of a the force), becomes relatively small allowing us to expand quantities as a function of the coupling constant. Consequently *perturbative* calculations can be performed. However at low-energies the coupling constant becomes large and a perturbative expansion is no longer appropriate. In this case we must resort to a *non-perturbative* approach.

The only consistent non-perturbative approach to a gauge field theory we know of is that of 'Lattice Gauge Theory', more specifically referred to here as 'Lattice QCD'. This chapter serves as a brief overview of the ($SU(3)$) gauge theory and lattice QCD in particular, building up a formulation which attempts to deal with the large numerical simulations upon which such a theory relies.

1.1.2 Gauge theories

The colour quantum numbers which we associate with an object are internal degrees of freedom on which physical observables should have no dependence. This means that our Lagrangian must exhibit a kind of invariance to reflect this. This invariance is known as *gauge invariance*.

The QCD Lagrangian density can be written in the form [11, 4]

$$\mathscr{L}_{QCD} = -\frac{1}{2}\mathrm{Tr}F_{\mu\nu}F^{\mu\nu} + \sum_{k=1}^{n_f}\bar{\Psi}_k(i\gamma^\mu D_\mu - m_k)\Psi_k\,, \tag{1.1}$$

where $n_f = 6$ is the number of flavours of quark: up, down, strange, charm, bottom and top. Just as in general relativity, we have defined the covariant derivative

$$D^\mu\Psi = (\partial^\mu + ig\mathsf{T}_a A_a^\mu(x))\Psi\,. \tag{1.2}$$

The object $F_{\mu\nu}$ is defined by the commutator of the covariant derivative

$$F_{\mu\nu} = -ig^{-1}[D_\mu, D_\nu] = \partial_\mu A_\nu - \partial_\nu A_\mu - ig[A_\mu, A_\nu] \tag{1.3}$$

$$A_\mu = \sum_{a=1}^{8} A_\mu^a \mathsf{T}^a \tag{1.4}$$

$$\mathsf{T}^a = \frac{\lambda^a}{2} \tag{1.5}$$

where the A_a^μ are the 8 gluon fields and the λ_a are the generators of the relevant gauge group, in this case, for $SU(3)$, they are the 8 Gell-Mann matrices. The constant g is the coupling constant associated with the group.

This Lagrangian is then invariant under the following gauge transformations.

A generalised gauge transformation of the field Ψ can be written as

$$\Psi(x) \to \mathsf{U}(x)\Psi(x)\,, \tag{1.6}$$

where

$$\mathsf{U}(x) = e^{-ig\mathsf{T}_a\Lambda_a(x)}\,. \tag{1.7}$$

The covariant derivative $\mathsf{D}^\mu\Psi$ transforms in the same way as the Ψ fields.

The gauge fields transform gauge invariantly under

$$\mathsf{A}^\mu \to \mathsf{U}(x)(\mathsf{A}^\mu(x) - ig^{-1}\partial^\mu)\mathsf{U}^{-1}(x)\,. \tag{1.8}$$

The object $F_{\mu\nu}$ then transforms as

$$F^{\mu\nu}(x) \to U(x) F^{\mu\nu} U^{-1}(x)\,. \tag{1.9}$$

Then, correctly normalised, we have the gauge invariant Yang-Mills Lagrangian (the gauge field Lagrangian)

$$\mathscr{L}_{YM} = -\frac{1}{2}\mathrm{Tr}(F_{\mu\nu}F^{\mu\nu}) \tag{1.10}$$

or equivalently

$$\mathscr{L}_{YM} = -\frac{1}{4}F_a^{\mu\nu}F_{\mu\nu}^a\,. \tag{1.11}$$

These two expressions are equivalent if the generators of the gauge group are given their conventional normalisation. For the generators given in their fundamental representation, t_a, this gives

$$\mathrm{Tr}(t_a t_b) = \frac{1}{2}\delta_{ab}\,. \tag{1.12}$$

1.1.3 Renormalisation

The universe in which we live can not accommodate physical quantities which are infinite. Therefore all measurable (physical) quantities must have finite values. However, unphysical, or *bare*, quantities which are not observable, and consequently not measurable, can take any value - including that of infinity.

A Lagrangian which describes a field theory has three basic sets of physical parameters which include: the coupling constants, the fields described by the theory, and the mass. We refer to these parameters as *bare* parameters. In order to *renormalise* a theory we wish to arrange for all of the infinities to be absorbed into the bare quantities, leaving the physical quantities finite. We do this by effectively placing a momentum cut-off into the theory, this cut-off is known as a *regulator*. The physical parameters are now functions of a momentum scale, μ. The physics which is described by the parameters of our theory should not depend on the way in which we renormalise our theory. This leads us to the renormalisation group equation.

$$\mu\frac{\partial g}{\partial\mu} = -\beta(g) = g^3(\beta_0 + \beta_1 g^2 + \ldots) \tag{1.13}$$

where, for 3 colours (*i. e.* $SU(3)$), we have

$$\begin{aligned} \beta_0 &= \tfrac{1}{(4\pi)^2}\left(11 - \tfrac{2}{3}n_f\right) \\ \beta_1 &= \tfrac{1}{(4\pi)^4}\left(102 - \tfrac{38}{3}n_f\right) . \end{aligned} \tag{1.14}$$

1.1.4 Asymptotic freedom

By integrating the renormalisation group Eq. (1.13) (keeping just the β_0 term), and introducing a parameter Λ_{QCD} as an integration constant, we arrive at an expression for the running coupling α_s

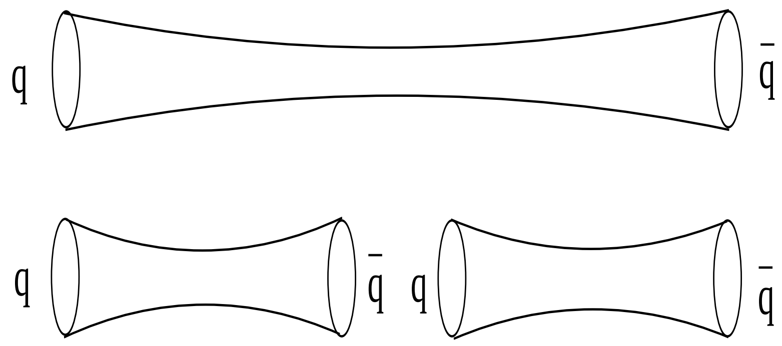
$$\alpha_s(\mu) = \frac{g^2(\mu)}{4\pi} = \frac{1}{8\pi\beta_0 \log\frac{\mu}{\Lambda_{QCD}}} . \tag{1.15}$$

It is easy to see that in Eq. (1.14), for $n_f < 17$, $\beta(g) > 0$. This implies that as μ increases then $g^2(\mu)$ decreases, and decreases to zero as $\mu \to \infty$. This is known as *asymptotic freedom*. Asymptotic freedom implies that as we reach higher energies, the theory tends towards a free-field theory.

1.1.5 Confinement

Asymptotic freedom implies that on short distance scales quarks behave as if they were free particles. We have also seen that at greater distances the effective coupling increases. This implies that it is energetically favourable, for example, for a quark and an anti-quark to remain in close proximity to each other.

However, we find a stronger condition that this. It appears that quarks are always found in bound hadronic states and never exist as free particles. Take for example the quark - anti-quark pair. We can picture these hadrons as being bound by 'strings' or tubes of colour (gluon) flux, and if one of these strings is ever broken, another quark and anti-quark pair is formed such that we now have two quark - anti-quark pairs.



One way to picture this is that if we give a quark - anti-quark pair enough energy needed to separate them, then we are providing enough energy to create a new quark - anti-quark pair, although the exact mechanism of confinement is still not fully understood.

1.1.6 Chiral symmetry

In the limit of vanishingly small degenerate quark masses,

$$m_u \sim m_d \sim m_s \to 0, \tag{1.16}$$

our QCD action satisfies another global symmetry known as *chiral symmetry.* This symmetry has an $SU(3)_L \otimes SU(3)_R$ group structure reflecting an invariance of the lagrangian under rotations in flavour space, independently for left-handed and right-handed quarks. If the masses of quark were actually zero, we would have chrial symmetry. However, this symmetry would still be broken by the vaccum. This is known as *spontaneous* symmetry braking. The spontaneous breaking of this symmetry is accompanied by the occurrence of masseless Goldstone bosons.

However, in reality, the quark masses are not zero and the associated Goldstone bosons do not have zero mass. We do however exist in a universe where quark masses are small compared with the strong mass-energy scale ($\sim 200 - 400$ MeV).

$$\begin{aligned} m_u &\sim \text{1.5 to 5 MeV} \\ m_d &\sim \text{3 to 9 MeV} \\ m_s &\sim \text{60 to 170 MeV} \end{aligned} \tag{1.17}$$

resulting in an approximate chiral symmetry. The mass of the up quark and that of the down quark are very small. This results in an approximate $SU(2)_L \otimes SU(2)_R$ symmetry. The reason for this approximate symmetry is manifest in the fact that the associated set of Goldstone bosons (the three pions) have masses which are small compared with all other hadrons.

This approximate symmetry implies a partially conserved current known as the partially conserved axial current (PCAC). This expression leads to formulae which can be used to extract quark masses from the masses of the pion, which, to leading order, is proportional to the masses of the u and d quarks. The mass of the kaon and η' mesons, to leading order, also have contributions from masses proportional to the s quarks masses.

1.2 Lattice QCD

1.2.1 Introduction

There are several good textbooks which deal with lattice gauge theory in detail [12, 13, 4, 14, 15]. Many of the recent review articles and lecture notes [16, 17, 18, 19] provide a more up-to-date discussion of the subject. I will give a brief outline here of some of the concepts which are fundamental to lattice gauge theory and will form the motivation of work presented in this thesis.

1.2.2 Discretisation

In lattice QCD we write down an action, defined on a lattice, which reduces to the continuum action in the limit in which the lattice spacing, a, tends to zero. This is done in such a way as to regularise the infinities which appear in any un-regulated field theory. The lattice effectively controls these infinities. The discrete nature of the lattice means that wavelengths less that $2a$ have no meaning. In momentum space this amounts to a momentum cut-off of $\frac{\pi}{a}$, so the lattice spacing a effectively amounts to an ultra-violet cut-off.

In order to do any kind of numerical simulations, the lattice has to be of finite size.

Unfortunately, both of these features bring with them systematic uncertainties which we must try hard to control.

Of course, we have applied a regulator to the theory, which must finally be removed. This amounts to taking our lattice spacing to zero: $a \to 0$, *i. e.* taking the continuum limit. The manner in which we do this is described later.

On the lattice four dimensional integration is replaced by a sum

$$\int d^4x \to a^4 \sum_n . \tag{1.18}$$

In order to perform an approximation of differentiation on the lattice, we replace derivatives with finite differences.

$$\begin{aligned} \partial_\mu \phi(x) &\to \tfrac{1}{a}(\phi_{n+\hat{\mu}} - \phi_n) \\ \partial_\mu^2 \phi(x) &\to \tfrac{1}{a^2}(\phi_{n+\hat{\mu}} - 2\phi_n + \phi_{n-\hat{\mu}}) \end{aligned} \tag{1.19}$$

where $\hat{\mu}$ is a four-vector of length a in the direction μ. $\mu = 0, 1, 2, 3$. and we define the

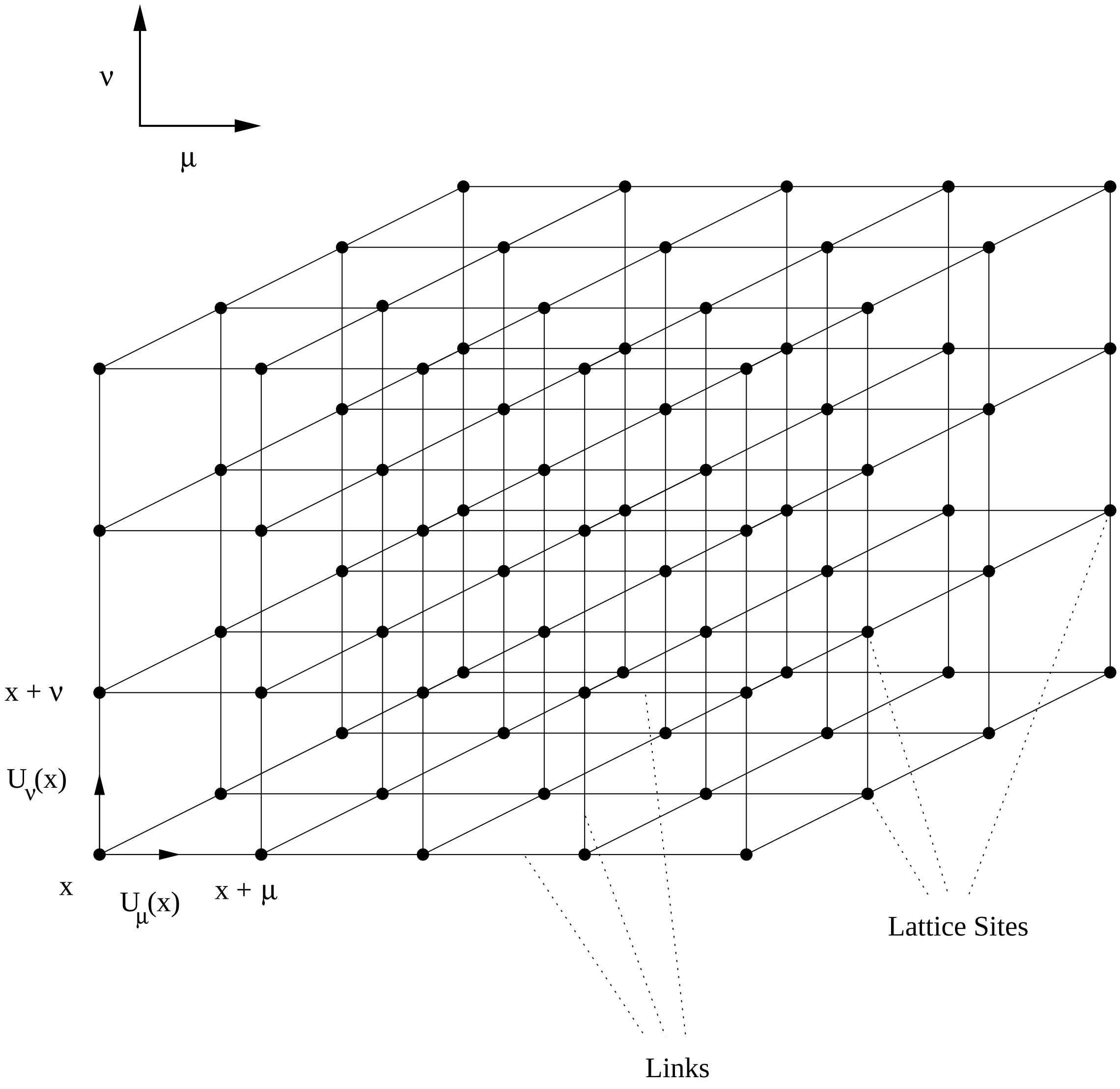


Figure 1.1: A schematic diagram of a 3-dimensional lattice. The U_μ are the $SU(3)$ link matrices.

discrete Fourier transform

$$\phi_n = \sum_{k=0}^{N} e^{2\pi ik.n/N} \phi_k \,, \tag{1.20}$$

where N is the number of samples used to calculate the Fourier transform.

1.2.3 Wick rotation

In order to perform numerical simulations, we must transform our theory from a Minkowski space to a Euclidean space. We do this via a *Wick Rotation*. Here the action is analytically continued to imaginary time $t \to it$. Our path integral is now an integral in Euclidean space, with a positive definite integrand.

$$\int e^{-iS} \to \int e^{-S} \,. \tag{1.21}$$

1.2.4 The Wilson gauge action

Let us first take another look at the gauge fields. We may study the gauge field with its Yang-Mills action separately from the interacting fermion fields.

Wilson's gauge action is given by

$$S_G = \frac{\beta}{3} \sum_x \sum \mathbb{R}\mathrm{Tr}\, \Box_{\mu\nu} \,, \tag{1.22}$$

where, for $SU(N)$, we have

$$\beta = \frac{2N}{g^2} \,. \tag{1.23}$$

So, for $SU(3)$, we have

$$\beta = \frac{6}{g^2} \,, \tag{1.24}$$

and we have defined an object, $\Box$, known as the *plaquette*.

$$\Box_{\mu\nu} = U_{\mu,x} U_{\nu,x+\mu} U^\dagger_{\mu,x+\nu} U^\dagger_{\nu,x} \,. \tag{1.25}$$

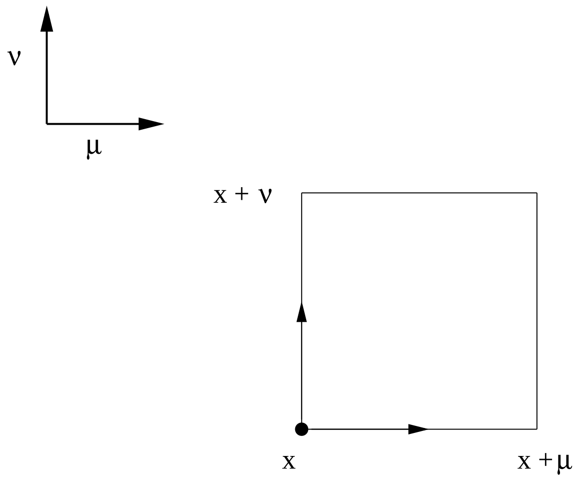


Figure 1.2: The Plaquette.

A plaquette consists of a 1×1 closed loop on the lattice which passes through the points

$$x, \quad x + a\hat{\mu}, \quad x + a\hat{\mu} + a\hat{\nu}, \quad x + a\hat{\nu}\,, \tag{1.26}$$

as shown in Fig. (1.2).

On the lattice we have the gauge fields

$$U(x,\mu) \equiv e^{-aA_\mu(x)} = 1 - aA_\mu(x) + \frac{a^2}{2}A_\mu(x)^2 + \dots\,, \tag{1.27}$$

and we define

$$A_\mu(x) = -igA^b_\mu(x)T_b\,. \tag{1.28}$$

So S_G in Eq. (1.22) reduces to

$$S_G = -\frac{\beta}{12}\sum_x\sum \mathbb{R}\mathrm{Tr}\,\Box_{\mu\nu} = -\frac{\beta}{12}\sum_x a^4\mathrm{Tr}F_{\mu\nu}(x)F^{\mu\nu}(x) + \mathcal{O}(a^6) \tag{1.29}$$

which is equal to the Yang-Mills action with relatively $\mathcal{O}(a^2)$ errors.

1.2.5 The Wilson fermion action

The Dirac equation

$$(i\gamma^\mu\partial_\mu - m)\Psi(x) = 0\,. \tag{1.30}$$

describes particles with spin $\frac{1}{2}$, and provides an interpretation of the Klein-Gordon equation which is linear in time

$$(\Box + m^2)\Psi(x) = 0 \tag{1.31}$$

where $\Box$ is the D'Alembertian, and m is the mass of the particle in question. However, we now interpret Ψ as representing a field, rather than a single particle.

$$\Box = \partial^\mu \partial_\mu = \frac{1}{c^2}\frac{\partial^2}{\partial t^2} + \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right) = \frac{1}{c^2}\frac{\partial^2}{\partial t^2} + \nabla^2 \,, \tag{1.32}$$

where we have set $c = 1$.

The free Dirac field is represented by the Lagrangian

$$\mathscr{L}(x) = \bar{\Psi}(x)(i\gamma^\mu \partial_\mu - m)\Psi(x)\,, \tag{1.33}$$

using the Euler-Lagrange equation

$$\partial_\mu \frac{\delta \mathscr{L}}{\delta(\partial_\mu \Psi)} - \frac{\delta \mathscr{L}}{\delta \Psi} = 0\,. \tag{1.34}$$

This Lagrangian can be discretised using the finite difference versions of the differentiation operator given in Eq. (1.19). This leads us to the discretised action

$$S_o(\Psi) = \sum_n \{ \frac{a^3}{2} \sum_{\mu=1}^{4} \bar{\Psi}_n \gamma_n (\Psi_{n+\hat{\mu}} - \Psi_{n-\hat{\mu}}) + ma^4 \bar{\Psi}_n \Psi_n \}\,, \tag{1.35}$$

with the Euclidean γ matrices satisfying

$$\{\gamma_\mu, \gamma_\nu\} = 2\delta_{\mu\nu}\,. \tag{1.36}$$

These matrices are shown in Appendix A.

The dispersion relation

$$S(k) = \frac{\sin^2 ak_\mu}{a^2} + m^2\,. \tag{1.37}$$

describes the behaviour of the propagator in momentum space. Unfortunately, $S(k)$ has two equal minima per dimension in each Brillouin zone, one of which occurs at $k = 0$; this

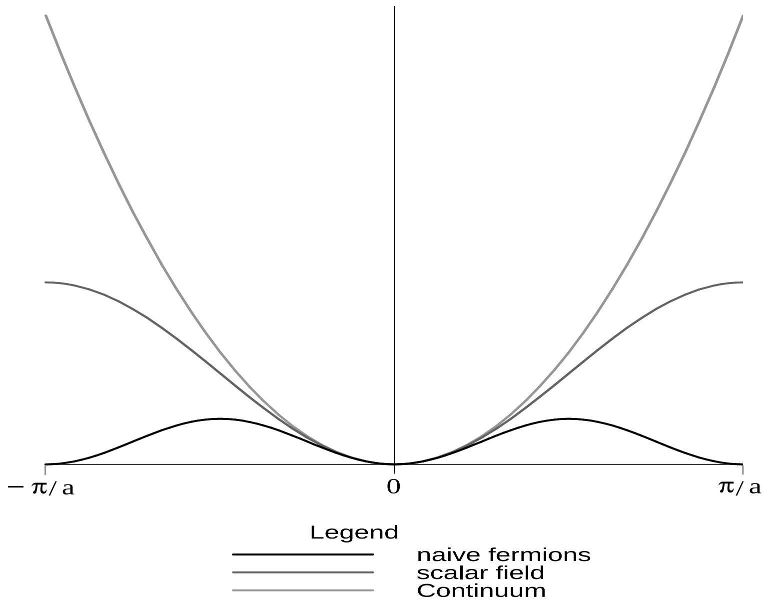


Figure 1.3: Wilson Doubling in one dimension.

has the correct continuum limit. The other occurs at $k = \pm\frac{\pi}{a}$ and this carries an infinite momentum as $a \to 0$. Fig. (1.3) shows the doubling as it arises in one dimension.

It is possible to suppress this fermionic doubling by the addition of extra terms in the Lagrangian provided the continuum limit is not affected. There are many ways to regularise this degeneracy. The one presented here is the one due to Wilson, who proposed the addition of some multiple of the following local term

$$\Delta\mathscr{L} = \frac{1}{2a}\bar{\Psi}_n(\Psi_{n+\hat{\mu}} + \Psi_{n-\hat{\mu}} - 2\Psi_n)\,. \tag{1.38}$$

This leads us to the Wilson action for free fermions:

$$S_o(\Psi) = a^4\sum_n\{\frac{1}{2a}\sum_{\mu=1}^{4}\bar{\Psi}_n[(\gamma_\mu - r)\Psi_{n+\hat{\mu}} - (\gamma_\mu + r)\Psi_{n-\hat{\mu}} - 2\Psi_n] + (m + \frac{4r}{a})\bar{\Psi}_n\Psi_n\} \tag{1.39}$$

$$S_o(\Psi) = \frac{a^3}{2}\sum_n\{\sum_{\mu=1}^{4}\bar{\Psi}_n[(\gamma_\mu - r)\Psi_{n+\hat{\mu}} - (\gamma_\mu + r)\Psi_{n-\hat{\mu}}] + \frac{1}{\kappa}\bar{\Psi}_n\Psi_n\}\,. \tag{1.40}$$

The hopping parameter κ is given by

$$\kappa = \frac{1}{2ma + 8}\,. \tag{1.41}$$

Having added this extra Wilson term, we have the following mass terms, which appear at the corners of the Brillouin zone

$$\begin{array}{lll} \text{momentum } p_\mu & \text{mass} & \text{multiplicity} \\ (0,0,0,0) & m_0 & 1 \\ (\frac{\pi}{a},0,0,0) & m_0+2\frac{r}{a} & 4 \\ (\frac{\pi}{a},\frac{\pi}{a},0,0) & m_0+4\frac{r}{a} & 6 \\ (\frac{\pi}{a},\frac{\pi}{a},\frac{\pi}{a},0) & m_0+6\frac{r}{a} & 4 \\ (\frac{\pi}{a},\frac{\pi}{a},\frac{\pi}{a},\frac{\pi}{a}) & m_0+8\frac{r}{a} & 1 \end{array} \tag{1.42}$$

So if we set $r = 1$, these extra states gain masses proportional to $1/a$ *i. e.* the cut-off scale, so are effectively removed from our theory. We now write

$$S_o(\Psi) = \frac{a^3}{2}\frac{1}{\kappa}\sum_n \bar{\Psi}_n\{1-\kappa\sum_{\mu=1}^{4}[(1-\gamma_\mu)\Psi_{n+\mu}+(1+\gamma_\mu)\Psi_{n-\mu}]\}\,. \tag{1.43}$$

So in the free-field fermion case we can write a free 'hopping term'

$$D_0(x,y) = \sum_{\mu=1}^{4}\{(1-\gamma_\mu)\delta_{x+\hat{\mu},y}+(1+\gamma_\mu)\delta_{x-\hat{\mu},y}\}\,. \tag{1.44}$$

We now include the fermion-boson interacting term

$$D_I(x,y) = \sum_{\mu=1}^{4}\{(U_\mu(x)-1)(1-\gamma_\mu)\delta_{x+\hat{\mu},y}+(U_\mu^\dagger(x-\hat{\mu})-1)(1+\gamma_\mu)\delta_{x-\hat{\mu},y}\}\,, \tag{1.45}$$

where the U_μ are the gauge fields.

So in the full fermion interacting case we have the hopping term,

$$\begin{aligned} D(x,y) &= D_0(x,y)+D_I(x,y) \\ D(x,y) &= \textstyle\sum_{\mu=1}^{4}\{U_\mu(x)(1-\gamma_\mu)\delta_{x+\hat{\mu},y}+U_\mu^\dagger(x-\hat{\mu})(1+\gamma_\mu)\delta_{x-\hat{\mu},y}\}\,, \end{aligned} \tag{1.46}$$

and a full action

$$S(\Psi) = a^3\sum_n \bar{\Psi}_n\{1-\kappa\sum_{\mu=1}^{4}[U_\mu(x)(1-\gamma_\mu)\Psi_{n+\mu}+U_\mu^\dagger(x-\hat{\mu})(1+\gamma_\mu)\Psi_{n-\mu}]\}\,, \tag{1.47}$$

where here we have absorbed a factor of $\frac{1}{\sqrt{2\kappa}}$ in to both Ψ and $\bar{\Psi}$.

We have included a term in the Wilson action which solves the problem of extra quark modes arising in the spectrum. However, this has come at a hard price. Namely that at finite a, we have explicitly broken chiral symmetry.

1.2.6 Neilson-Ninomiya "No-Go theorem"

So we have described an action which fulfills many of the properties we would like a discretised theory to have, except that we have broken chiral symmetry. Ideally, we would create an action which satisfies the following properties:

- $D(x-y)$ local.

  $f(x-y)$ must become exponentially small for large $|x-y|$.

- $D(p) = i\gamma_\mu p_\mu + \mathcal{O}(ap^2)$.

  We need our action to have the correct continuum limit behaviour.

- $D(p)$ invertible for $p \neq 0$.

  We want our propagator to have only one pole (at zero momentum) *i. e.* we don't want to introduce unwanted 'doublers'.

- $\gamma_5 D + D\gamma_5 = 0$.

  We need a lattice version of Chiral symmetry.

Neilson and Ninomiya have shown that it is impossible to create a theory on the lattice which is consistent with all four of these properties. We regard the first two properties as essential to a well-defined theory. However, the latter two properties can be adapted to provide several alternative formulations of the lattice action.

1.2.7 Alternative actions

So how do we construct such alternative formalisms? Firstly, we have Kogut-Susskind (or staggered) fermions. The idea here is that we assemble the flavour and spin components of the fermion fields at the sixteen corners of the 2^4 hypercube. This gives us four flavours of quarks, which we interpret as four physical flavours of quark. This action preserves a $U(1) \otimes U(1)$ symmetry at finite a, which is a remnant of the original chiral symmetry. However, we then have to find a way of interpreting the various flavours of fermion.

Alternatively, we can write down a property which is analogous to Chiral symmetry. This property is known as the Ginsparg-Wilson relation [20]

$$\gamma_5 D + D\gamma_5 = aD\gamma_5 D\,. \tag{1.48}$$

This idea has recently been revived as we now have access to viable actions which satisfy the Ginsparg-Wilson relation. Two formulations of such actions are the Domain Wall formalism, reviewed in [21], and the Overlap (Narayanan and Neuberger) formalism [22].

1.2.8 Improved actions

As it stands the inclusion of an extra term in Wilson's fix for fermion doubling introduces $\mathcal{O}(a)$ discretisation errors (without Wilson's fix we have $\mathcal{O}(a^2)$ discretisation errors). Obviously, we would like to reduce the errors in the full fermion action to (at least) $\mathcal{O}(a^2)$ if possible.

There are ways of counteracting such discretisation errors. Actions which include attempts to do this are known as *improved actions.*

Sheikholeslami and Wohlert [23] proposed the addition of a second dimension 5 operator designed to counteracts the $\mathcal{O}(a)$ effects in Wilson's extra dimensional 5 term, which was introduced to fix the doubling problem. This kind of improvement is known as Symanzik improvement. This operator, A, is given by

$$\mathsf{A} = \mathsf{I} - \frac{1}{2}\kappa c_{\rm sw}\sigma_{\mu\nu}F_{\mu\nu}\,, \tag{1.49}$$

where

$$\sigma_{\mu\nu} = \frac{i}{2}[\gamma_\mu, \gamma_\nu]\,, \tag{1.50}$$

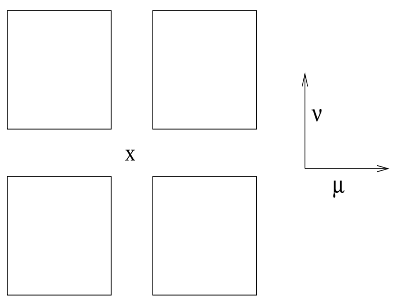
and $F_{\mu\nu}$ is the lattice field tensor.

$$F_{\mu\nu} = \frac{1}{8}(f_{\mu\nu} - f^\dagger_{\mu\nu})\,, \tag{1.51}$$

where

$$\begin{aligned} f_{\mu\nu} &= U_{\mu\ x}U^\dagger_{\nu\ x+\mu}U_{\mu\ x+\nu}U^\dagger_{\nu\ x} \\ &+ U_{\nu\ x}U^\dagger_{\mu\ x-\mu+\nu}U^\dagger_{\nu\ x-\mu}U_{\mu\ x-\nu} \\ &+ U^\dagger_{\mu\ x-\mu}U^\dagger_{\nu\ x-\mu-\nu}U_{\mu\ x-\mu-\nu}U_{\nu\ x-\nu} \\ &+ U^\dagger_{\nu\ x-\nu}U_{\mu\ x-\nu}U_{\nu\ x+\mu-\nu}U^\dagger_{\mu x}\,. \end{aligned} \tag{1.52}$$

This gives rise to 4 plaquette-like terms, and is said to resemble a four-leaf clover, giving rise to its name



The simplest case is when we set $c_{\rm sw} = 1$, which gives us a 'tree-level' approximation and we have errors of order $\mathcal{O}(a\alpha_s)$. However, the coefficient $c_{\rm sw}$ can be determined non-perturbatively [24] to give smaller errors of $\mathcal{O}(a^2)$.

1.3 Lattice QCD in Practice

In order for Lattice QCD to be useful, we need not only for it to be well-defined, we also need a way of extracting useful quantities such as hadronic masses.

1.3.1 Observables on the lattice

The expectation value of an observable F is given as

$$< F >= \frac{\int DUD\bar{\Psi}D\Psi F[U,\bar{\Psi},\Psi]e^{-S_G[U]-S_f[U,\bar{\Psi},\Psi]}}{\int DUD\bar{\Psi}D\Psi e^{-S_G[U]-S_f[U,\bar{\Psi},\Psi]}} . \tag{1.53}$$

The fermion fields Ψ and $\bar{\Psi}$ are Grassmann valued and *anti-commute i. e.*

$$\{\Psi_i, \Psi_j\} = \Psi_i\Psi_j + \Psi_j\Psi_i = 0, \qquad \forall i, j . \tag{1.54}$$

This makes fermion fields hard to manipulate numerically. Fortunately it is possible to integrate out this fermionic contribution. We define the following integration rules

$$\int d\Psi_i = 0, \qquad \int d\Psi_i\Psi_i = 1 , \tag{1.55}$$

which leads us to

$$\int D\Psi D\bar{\Psi} e^{-\bar{\Psi}\mathsf{M}[U]\Psi} = \det(\mathsf{M}[U]) . \tag{1.56}$$

This means that for two fermions we have the full QCD action

$$S = -S_G[U] - S_f[U,\bar{\Psi},\Psi] = S[U] = -\beta\sum_{\Box}(\frac{1}{3}\Re\mathrm{Tr}U_{\Box}) - \mathrm{Tr}\log(\mathsf{M}^\dagger\mathsf{M}) , \tag{1.57}$$

where we have used the identity[2]

$$\log\det(\mathsf{M}^\dagger\mathsf{M}) = \mathrm{Tr}\log(\mathsf{M}^\dagger\mathsf{M}) . \tag{1.58}$$

[2]This identity is easy to see by diagonalising the matrix.

1.3.2 Correlators - meson masses

In order to measure meson masses on the lattice, we make use of the lattice meson correlator. For example, for the π-meson correlator, $C_\pi(t)$, we have

$$C_\pi(t) = \sum_n \frac{|\langle 0|P|n\rangle|^2}{2m_n} e^{-m_n t}\,, \tag{1.59}$$

where a complete set of states is inserted and the integral given by Eq. (1.53). As $t \to \infty$, the lowest mass state dominates, and so we can perform an exponential fit to the lattice correlator to extract the mass of a particular meson.

1.3.3 The continuum limit

Now we have a basis on which to build a field theory with the lattice as a regulator. Ultimately, any regulator applied to a theory must be removed in order to give quantities a physical value. In lattice QCD this amounts to taking the lattice spacing a to zero, *i. e.* taking the continuum limit.

The way in which we regain the continuum limit is to take $a \to 0$, or equivalently $\beta \to \infty$ (see Eq. (1.24)) whilst holding the physical quantities fixed. As the lattice spacing a acts effectively as a $\frac{1}{a}$ momentum cut-off, we attribute a the units of inverse mass. Consequently, we have the dimensionless quantity M, where

$$M = ma(\beta, L) \tag{1.60}$$

where $L = Na$ is the length of one side of the lattice. Consequently we must also take the limit $L \to \infty$.

For example, we may want to consider the ratio of two masses

$$\frac{M_x}{M_y} = \lim_{\substack{a\to 0\\ L\to\infty}} \frac{am_x(a,\beta)}{am_y(a,\beta)}\,. \tag{1.61}$$

In the limit $\beta \to \infty$, these ratios would produce a constant and we would say the theory 'scales', however, we cannot work with infinite values of β and so we must find a region in which the theory scales.

In order to perform simulations, we must choose a finite set of parameters to simulate with and then take limits in a controlled manner.

1.3.4 The chiral limit

In order to simulate quarks at masses close to the mass of the u or d quark, ideally we would simulate at a value of $\kappa = \kappa_c$ known as κ-critical. This is the value at which the pion becomes massless. For values of κ not at κ_c, the bare quark mass, m_q, is given by

$$m_q = \frac{1}{2a}\left(\frac{1}{\kappa} - \frac{1}{\kappa_c}\right) . \tag{1.62}$$

Unfortunately, simulations are increasingly hard to perform as we approach $\kappa = \kappa_c$, so we must take several values of κ and extrapolate to κ_c. We wish also to be able to simulate at masses of quark close the mass of the s quark. For this we have an analogous value of κ known as κ_s. The simulations we are able to do at present have values of κ which are of the order of κ_s and so we must interpolate to κ_s.

1.4 The Fermion Determinant

1.4.1 The Matrix formulation

Now we can consider how to solve these equations using lattice techniques.

Following on from Eq. (1.46), and, for now, setting $a = 1$, we write the equation for the action in matrix form in the following way

$$S = \bar{\Psi}_x M_{x,y} \Psi_y \tag{1.63}$$

where

$$M_{x,y} = A_{x,y} - \kappa D_{x,y} \, . \tag{1.64}$$

So we write $M_{x,y}, A_{x,y}$ and $D_{x,y}$ in concise matrix notation as M, A and D, respectively.

$$\mathsf{M} = \mathsf{A} - \kappa \mathsf{D} \, . \tag{1.65}$$

1.4.2 Pre-conditioning

The amount of work required to diagonalise a matrix usually depends on the condition number $\sqrt{\lambda_{max}/\lambda_{min}}$, where λ_{max} and λ_{min} are the maximum and minimum eigenvalues of the matrix respectively. In order to reduce this condition number we can apply a form of *pre-conditioning*.

We know that if we are dealing with a matrix expression

$$\mathsf{M}\psi = \eta \tag{1.66}$$

then, for any non-singular V_1 and V_2, we can equivalently write

$$\mathsf{V}_1^{-1}\mathsf{M}\mathsf{V}_2^{-1}\mathsf{V}_2\psi = \mathsf{V}_1^{-1}\eta \, . \tag{1.67}$$

To improve the convergence properties, we want our pre-conditioner $\mathsf{V} = \mathsf{V}_1\mathsf{V}_2$ to be a good approximation to M.

Due to the way our lattice is defined, it is possible to arrange for the fermion matrix to be described in such a way that alternate sites are classified as odd or even respectively. This motivates us to consider a form of pre-conditioning known as *Red-Black Pre-conditioning*.

Red-black pre-conditioning

We can decompose the matrix M into its red (black) [even (odd)] components; each site on the lattice being given a parity, P.

$$P = (-1)^{x+y+z+t} . \tag{1.68}$$

We can work in a basis in which all even sites appear before all odd sites. Eq. (1.66) then becomes

$$\mathsf{M}\psi \equiv (\mathsf{A} - \kappa\mathsf{D})\psi = \begin{pmatrix} \mathsf{A}_{ee} & -\kappa\mathsf{D}_{eo} \\ -\kappa\mathsf{D}_{oe} & \mathsf{A}_{oo} \end{pmatrix} \begin{pmatrix} \psi_e \\ \psi_o \end{pmatrix} = \begin{pmatrix} \eta_e \\ \eta_o \end{pmatrix} . \tag{1.69}$$

We can arrange for our pre-conditioner V to imitate M by performing an *incomplete LU-factorisation* of M. For example, we can set $\mathsf{V} = \mathsf{LU}$

$$\mathsf{L} = \begin{pmatrix} \mathsf{I} & 0 \\ -\kappa\mathsf{D}_{oe}\mathsf{A}_{ee}^{-1} & \mathsf{I} \end{pmatrix} \quad \mathsf{U} = \begin{pmatrix} \mathsf{I} & -\kappa\mathsf{A}_{ee}^{-1}\mathsf{D}_{eo} \\ 0 & \mathsf{I} \end{pmatrix} \tag{1.70}$$

then we have

$$\mathsf{L}^{-1} = \begin{pmatrix} \mathsf{I} & 0 \\ \kappa\mathsf{D}_{oe}\mathsf{A}_{ee}^{-1} & \mathsf{I} \end{pmatrix} \quad \mathsf{U}^{-1} = \begin{pmatrix} \mathsf{I} & \kappa\mathsf{A}_{ee}^{-1}\mathsf{D}_{eo} \\ 0 & \mathsf{I} \end{pmatrix} . \tag{1.71}$$

We then set

$$\tilde{\mathsf{M}} = \mathsf{L}^{-1}\mathsf{M}\mathsf{U}^{-1} = \begin{pmatrix} \mathsf{A}_{ee} & 0 \\ 0 & \hat{M}_{oo} \end{pmatrix} . \tag{1.72}$$

It is then possible to define a red-black pre-conditioned matrix $\hat{\mathsf{M}}_{oo}$, which just deals with one parity of the matrix.

$$\hat{\mathsf{M}}_{oo} = \mathsf{A}_{oo} - \kappa^2\mathsf{D}_{oe}\mathsf{A}_{ee}^{-1}\mathsf{D}_{eo} . \tag{1.73}$$

The determinants of L and U are both equal to unity, so it is easy to see that $\det\tilde{\mathsf{M}} = \det\mathsf{M}$, and therefore

$$\det\mathsf{M} = \det\mathsf{A}_{ee}\det\hat{\mathsf{M}}_{oo} . \tag{1.74}$$

In our case we are interested in the $\log\det(\mathsf{H})$, where H is some Hermitian representation of the fermion determinant, *e. g.* $\mathsf{H} = \mathsf{M}^\dagger\mathsf{M}$. So we have

$$\begin{aligned}\log\det\mathsf{M}^\dagger\mathsf{M} &= \log(\det(\mathsf{A}_{ee}^\dagger\mathsf{A}_{ee})\det\mathsf{M}_{oo}^\dagger\mathsf{M}_{oo}) \\ &= 2\log\det\mathsf{A}_{ee} + \log\det\mathsf{M}_{oo}^\dagger\mathsf{M}_{oo}\,.\end{aligned} \tag{1.75}$$

The quantity $\log\det\mathsf{A}_{ee} = \mathrm{Tr}\log\mathsf{A}_{ee}$ can be calculated explicitly using, for example, $LU-$factorisation. Leaving us with the quantity $\log\det\mathsf{M}_{oo}^\dagger\mathsf{M}_{oo}$ to calculate.

So we have now defined a new matrix, where we can easily relate the determinant of the complete fermion matrix with the determinant of just one parity of the same matrix. In creating this new matrix, we have approximately doubled the value of the smallest eigenvalue and have thus reduced the condition number of the matrix by a factor of approximately $\sqrt{2}$ at the cost of a 'clover-inverse' term. We later explore how the convergence properties are improved by this form of pre-conditioning.

1.4.3 The quenched approximation

The fermion matrix M contains information about every site on the lattice, each having both 3 colour degrees of freedom and 4 spin degrees of freedom. Consequently, the matrix consists of $12V$ rows and columns. For a lattice size of $16^3 \times 32$ this results in a matrix of dimensions 1572864×1572864. Clearly such a matrix is very large and even the most up-to-date algorithms and machines cannot be expected to yield an accurate value for the determinant, or to calculate a full eigenspectrum, on any realistic time scale. In reality, we wish the lattice size to be considerably larger than this and so we must establish carefully what kind of effects the inclusion of the fermion determinant is expected to have, before we invest enormous computational resources on such calculations.

So firstly we will discuss what happens when we ignore the effect of the fermion determinant. This leads to an approximation known as the *quenched approximation*. The quenched approximation amounts to the following equivalent effects:

- ignoring virtual quark loops

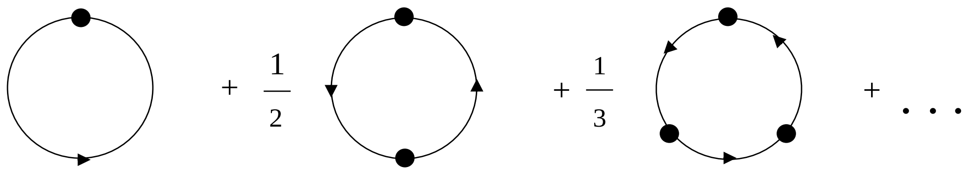
  Using Eq. (1.46) and the matrix formulation of section 1.4.1, we can see that we can write the fermion matrix $\mathsf{M}[U]$ as a fermion free-field part M_0 and an interacting part M_I

$$\begin{aligned} \mathsf{M}[U] &= \mathsf{M}_0 + \mathsf{M}_I[U] \\ &= \mathsf{M}_0(\mathsf{I} + \mathsf{M}_0^{-1}\mathsf{M}_I[U])\,. \end{aligned} \tag{1.76}$$

  Thus we can write

$$\begin{aligned} \log\det\mathsf{M}[U] &= \log\det\mathsf{M}_0 + \log\det(\mathsf{I} + \mathsf{M}_0^{-1}\mathsf{M}_I[U]) \\ &= \log\det\mathsf{M}_0 + \mathrm{Tr}\log(\mathsf{I} + \mathsf{M}_0^{-1}\mathsf{M}_I[U]) \\ &= \log\det\mathsf{M}_0 - \textstyle\sum_{i=1}^{\infty}\frac{(-1)^i}{i}\mathrm{Tr}(\mathsf{M}_0^{-1}\mathsf{M}_I[U])^{-1}\,. \end{aligned} \tag{1.77}$$

  When exponentiated and inserted into the path integral, the first term in the above expression amounts to a factor which appears in both the numerator and the denominator and thus cancels. The second term consists of summations of the trace of i products of the free fermion propagator M_0^{-1} and the gauge interacting term $\mathsf{M}_I[U]$. Omitting such a term physically amounts to ignoring the effect of quark loops.



- setting κ to zero

  - Effectively making the dynamical (virtual) quarks infinitely heavy (recall the definition of κ, $\kappa = \frac{1}{2ma+8}$)

- setting the Fermion determinant equal to unity

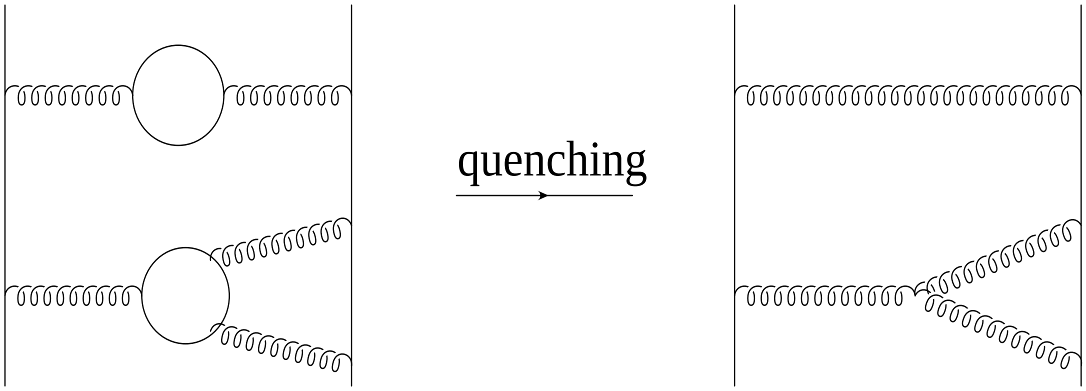


Figure 1.4: The effect of quenching is to suppress virtual quark loops.

So using the quenched approximations, we no longer have to go to the enormous computational expense of calculating the fermion determinant, but in doing so we have ignored the effects of a whole myriad of virtual particles.

1.4.4 Approximating the effects

In order to simulate the effects of full QCD [25, 26, 27, 28], we must remove the quenched approximation entirely. However, we may begin by trying to approximate the effects of dynamical quarks.

For heavy dynamical quarks *i. e.* for small values of κ, if we expand the action in powers of κ, we find that the first non-vanishing term is proportional to κ^4. Such a term corresponds to a plaquette term. Consequently, we are able to approximate the effects of dynamical quarks just by shifting the value of β in a pure gauge action. It turns out that this approximation is accurate to within 10% for some quantities, but fails completely for quantities such as the η' mass [17].

1.5 Monte Carlo Simulations

In lattice QCD, we are attempting to perform integrals of the form

$$\langle \mathcal{O} \rangle = \frac{\int [dU] e^{-S[U]} \mathcal{O}([U])}{Z}, \tag{1.78}$$

where

$$Z = \int [dU] e^{-S[U]}. \tag{1.79}$$

Even on a small lattice this amounts to a huge number of integrations. Such integrals are generally performed using *Monte Carlo integration*, where essentially we are approximating the integral with a weighted summation[15, 14, 29].

The action $S[U]$ is generally large in value, consequently only a small fraction of possible values of U will make a significant contribution to the integral in Eq. (1.78). So what we do is replace this integral with the summation

$$\langle \mathcal{O} \rangle \approx \frac{1}{N} \sum_{i=1}^{N} \mathcal{O}(\{U\}_i), \tag{1.80}$$

where $\{U\}_i$ represents a set of values of U, known as *configurations*, which fit the probability distribution, $W[U]$, where

$$W[U] = e^{-S[U]}. \tag{1.81}$$

So the task is to find a method of creating such configurations in such a way that they are biased towards large values of $W[U]$, *i. e.* such that they fit a probability distribution $\mu \propto W[U]$. This is known as *importance sampling.*

There are principally two ways of creating a set of configurations which produce the desired probability distribution. These methods are known as *static* or *dynamic*. Static algorithms create statistically independent samples from the probability distribution μ. However, as the dimension of the problem becomes large, this method becomes unfeasible. Hence, instead, we use a dynamic algorithm. Here we invent a stochastic process which is used to create a sequence of configurations, where each configuration evolves from the previous one. So the algorithm produces configurations which evolve in time, but we must remember that, in this context, the 'time' we refer to is unphysical (as is the stochastic process which we have invented), and is sometimes better referred to as 'computer time', τ. Unfortunately, such configurations are not statistically independent, and hence there exists a correlation between successive configurations. We measure this correlation in terms of something known as the *autocorrelation time*, which is essentially the average amount of time it takes for the system to 'forget' about any particular configuration.

We initiate the sequence of creating configurations, and continue the algorithm until we have reached equilibrium *i. e.* until we have a set of configurations with the desired probability distribution $\mu_{eq} = \mu$. The length of time the system takes to reach equilibrium depends heavily on the autocorrelation time. In general however we do not know a value for the autocorrelation time and have to measure this as the system evolves. In order to be confident that our system has reached equilibrium we must plot the estimates of physical observables as a function of τ. We also must be sure that equilibrium distribution is attainable by enforcing a condition known as *ergodicity.* This basically states that all possible configurations should be attainable, given any initial starting configuration.

We implement this stochastic process of creating configurations as a *Markov chain*, where we have a transition probability matrix, $\mathsf{P}(U \to U')$, which takes us from one configuration, U, to the next, U'. In order for the system to reach equilibrium, we enforce a condition known as *detailed balance*

$$\mu_{eq}(U)\mathsf{P}(U \to U') = \mu_{eq}(U')\mathsf{P}(U' \to U)\,. \tag{1.82}$$

There are a number of ways of creating a matrix P which satisfies detailed balance. The method we use is known as the Metropolis [30] method.

We define a proposal matrix P_0 which generates proposed moves $U \to U'$. This move is then accepted with probability $\alpha_{UU'}$ (and consequently rejected with probability $1-\alpha_{UU'}$). Detailed balance gives us

$$\alpha_{UU'} = \min\left(1, \frac{\mu_{eq}(U')\mathsf{P}_0(U' \to U)}{\mu_{eq}(U)\mathsf{P}_0(U \to U')}\right) . \tag{1.83}$$

In the case where P_0 satisfies *reversibility i. e.*

$$\mathsf{P}_0(U \to U') = \mathsf{P}_0(U' \to U) , \tag{1.84}$$

we have

$$\alpha_{UU'} = \min\left(1, \frac{\mu_{eq}(U')}{\mu_{eq}(U)}\right) = \min(1, e^{-\Delta S}) , \tag{1.85}$$

where

$$\Delta S = [S[U'] - S[U]] . \tag{1.86}$$

This is realised in the following algorithm

The Metropolis Algorithm

(1) Choose an initial configuration.

(2) Apply the proposal matrix to update the configuration.

(3) Calculate $\Delta S[UU'] \equiv [S[U'] - S[U]]$.

- – If $\Delta S[UU'] \leq 0$ accept: $\alpha_{UU'} = 1$. Return to (2).
  – If $\Delta S[UU'] > 0$ then accept the proposal with probability $e^{-\Delta S[UU']}$. This is implemented by generating a random number, $r \in [0,1]$, and accepting the proposal if $r \leq e^{-\Delta S}$. Return to (2).

So how do we create the matrix P_0, *i. e.* how do we create a proposed *update* to a configuration? What we do is we define a set of momenta along with the corresponding Hamilton's equations, by analogy to the theory of molecular dynamics. Thus setting up a trajectory for our system. In order to satisfy ergodicity we must regularly refresh these momenta.

Such algorithms which combine Metropolis (accept-reject) algorithms with molecular dynamics are known as *Hybrid Monte Carlo* (HMC) algorithms [31]. The HMC algorithm was employed to produce the dynamical configurations discussed in this thesis.

1.6 Parallel Computing

When it comes to running large numerical simulations on a computer, large 'matrix-vector' calculations lend themselves well to parallelisation techniques. Most of the work of the large scale calculations performed for this thesis were carried out on a Cray T3E computer. All of the gauge configurations discussed in this thesis were also created on this machine.

1.6.1 The Cray T3E

The UKQCD collaboration has access to a large subset of the 344-processor CRAY T3E housed at EPCC (Edinburgh Parallel Computing Centre), based at Edinburgh University. This machine was installed at EPCC in spring 1997.

The current T3E configuration has 450 MHz Digital Alpha(EV5.6) processors, each having a peak performance of 900 Mflops. The operating system is run over 4 processors each with 128 Mbytes RAM[3]. The system has a peak performance of 309 Gflops over the 344 application processors.

The T3E is also connected via EPCC's Cray J90 to the IBM 3494 Automatic Tape Library mass store, the total capacity of which is around 20 Tbytes.

[3]This information is based on the EPCC website "http://www.epcc.ed.ac.uk/t3e/documents/t3e-intro/node3.html".

1.6.2 QCDOC

The UKQCD collaboration, in conjunction with The University of Columbia are presently well into a project to create a new machine with a peak performance of ~ 10 Tflops and a sustained performance of ~ 5 Tflops. This project is known as QCDOC (QCD on-a-chip) which will ultimately be housed in Edinburgh.

The basic architecture of the machine will comprise of a large number of inexpensive, small, low-power processors [32]. Each processor runs at 500 MHz, with a 1 Gflop, 64-bit floating point unit.

1.6.3 PC Clusters

There is presently much interest in the possibility of creating relatively cheap supercomputers, predominantly from PC hardware. Such a system typically runs free software such as Linux. Beowulf clusters are an example of this type of architecture [33, 34].

1.7 Thesis Outline

This thesis is organised as follows. Chapter 2 gives an outline of the underlying theory on which our Lánczos methods are based and then in Chapter 3 we illustrate this theory in action together with a demonstration of the surprising numerical stability of these methods. In Chapter 4 we go on to discuss stochastic methods, and how we can apply our Lánczos methods to create stochastic estimates of quantities used in Lattice QCD. In Chapter 5 we then discuss an improvement on these methods using variance reduction techniques which we have developed. Chapter 6 discusses ways in which we may go on to make use of the methods previously described. Finally, chapter 7 gives an overview of the whole thesis summarising the conclusions, and giving ideas of how this work maybe continued.

Chapter 2

Linear systems: The Lánczos Algorithm

2.1 Introduction

A major computational challenge is the solving of large linear systems. Practitioners of Lattice QCD must take on this challenge wholeheartedly. When dealing with Fermion matrices which are even close to being physically useful, the associated matrices are huge, and even the task of finding approximate solutions is a computationally demanding one. For example in this thesis we analise lattices with lattice dimensions $16^3 \times 32$. When one considers that we have 3 colours and each site has 4 spin directions, even our red-black pre-conditioned matrix consists of 786,432 rows. For physically realistic lattices capable of containing complete hadronic wavefunctions, our lattices will have to be yet bigger still.

The type of linear solvers we deal with here are *iterative*, that is to say we start with an initial estimate of the solution to a problem, and then iteratively produce improved estimates in the hope of producing a result with is successively closer to the accurate solution. Thus, we frequently talk in general terms of the convergence properties of a particular method given the size of the problem.

The main subject of discussion in this thesis is that of the Lánczos method, therefore the majority of this chapter is devoted to Lánczos techniques along with some associated theory.

2.2 The Lánczos Algorithm

The Lánczos algorithm [35] was first introduced in the 1950's as a method intended to reduce whole matrices to tridiagonal form, and as such was not immediately successful due to the rapid breakdown of orthogonality of the resulting Lánczos vectors. This fact makes the procedure an expensive one if knowledge of the whole eigenspectrum of a matrix is required.

It was not until twenty years later that Paige [36] showed that the procedure could be used effectively to calculate a few extremal eigenvalues and their eigenvectors with high accuracy and relatively little work. During the mid 1990's Bai, Fahey and Golub [37] went on to show an analogy between the Lánczos orthogonalisation procedure and Gaussian Quadrature, which enables the Lánczos algorithm to be used effectively to approximate certain quantities, some of which are applicable to lattice QCD.

In this chapter I will describe the Lánczos algorithm itself, giving an outline of the procedure which may be used to create an approximation to the eigenspectrum of a given Hermitian matrix. I will then go on to describe some of its characteristics as applied to lattice QCD. The following chapter then illustrates the points made in this chapter, using full lattice data.

The Lánczos algorithm may be used to solve large, sparse eigenvalue problems $\mathsf{H}x =$

λx. The algorithm does not produce complete diagonalisation of the matrix but involves partial tridiagonalizations of the matrix H. It is fairly trivial to then diagonalise the tridiagonal matrix. During execution of the algorithm no full sub-matrices are produced resulting in low storage requirements. The algorithm is particularly useful if only a few, extremal eigenvalues are needed as these are accurately determined early on in the iteration procedure.

It is well known that the Lánczos procedure is beset with numerical instability when applied in finite precision arithmetic. This results in early loss of orthogonality amongst the Lánczos vectors. It turns out that despite this instability the Lánczos algorithm can be used to calculate certain expectation values, and such values are remarkably numerically stable.

The Lánczos algorithm generates a subspace by repeatedly multiplying an arbitrarily chosen seed vector q_1 by a Hermitian matrix H. The resulting subspace is known as a Krylov subspace:

$$[q_1, \mathsf{H}q_1, \mathsf{H}^2 q_1, \ldots, \mathsf{H}^{n-1} q_1] \,. \tag{2.1}$$

As H is sparse, it makes sense to compute the elements of the tridiagonal matrix, $\mathsf{T} = \mathsf{Q}^T \mathsf{H} \mathsf{Q}$, directly.

$$\mathsf{T} = \begin{pmatrix} \alpha_1 & \beta_1 & & \cdots & 0 \\ \beta_1 & \alpha_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & \beta_{n-1} \\ 0 & \cdots & & \beta_{n-1} & \alpha_n \end{pmatrix} .$$

By equating columns in $\mathsf{HQ} = \mathsf{QT}$ directly and setting $\mathsf{Q} = [q_1, q_2, ..., q_n]$, we find

$$\begin{aligned} \mathsf{H}q_k &= \beta_{k-1}q_{k-1} + \alpha_k q_k + \beta_k q_{k+1}\,, \\ \beta_0 q_0 &= 0\,. \end{aligned} \tag{2.2}$$

By orthonormality, $a_k = q_k^T \mathsf{H} q_k$.

We now compute a sequence of α's and β's to create the above tridiagonal matrix. Let $r_k = (\mathsf{H} - \alpha_k \mathsf{I})q_k - \beta_{k-1}q_{k-1}$. If r_k is non-zero, then $q_{k+1} = r_k/\beta_k$, where $\beta_k = \|r_k\|_2$. If $r_k = 0$, then the procedure breaks down, but we may already have gained information about the subspace.

From the proceeding information, we can produce an iterative algorithm.

The Lánczos Algorithm

$q_0 = 0$

q_1 (arbitrarily chosen)

Normalise q_1

$\beta_0 = 0$

Loop(n)

- $q_{n+1} := \mathsf{H} * q_n$
- $q_{n+1} := q_{n+1} - \beta_{n-1} * q_{n-1}$
- $\alpha_n := < q_n, q_{n+1} >$
- $q_{n+1} := q_{n+1} - \alpha_n * q_n$
- Set $\beta_n := ||q_{n+1}||$
- Set $q_{n+1} := \frac{q_{n+1}}{\beta_n}$
- $q_{n-1} := q_n$
- $q_n := q_{n+1}$
- $\beta_{n-1} := \beta_n$

End Loop(n)

2.3 Problems - Loss of Orthogonality

When computing the Lánczos algorithm, errors will appear due to the fact we are using finite precision arithmetic, this causes a loss of orthogonality amongst the Lánczos vectors and we soon find that the procedure is homing in on the same eigenvalues repeatedly. This leads to an excess multiplicity amongst some eigenvalues. Consequently, it takes a long time for all of the eigenvalues to be found as the procedure tries to repeatedly span the same subspace.

In the next few sections we review some of the present methods used to overcome the problems encountered when attempting to calculate eigenvalues accurately, and then we will go on to describe the method we have developed, which relies less heavily on an exhaustive calculation of the full eigenspectrum.

2.4 Analysis of the Eigenspectrum

There are a number of ways of overcoming the 'loss of orthogonality' problem, and before we describe the method developed in this project, we will describe two methods which are presently in use. These methods both show ways of determining eigenvalues correctly. The first by enforcing orthogonality amongst the Lánczos vectors, with the price of often having to perform "reorthogonalisation" steps. The second calculates the eigenvalues more naïvely, relying on the fact that the whole eigenspectrum will appear eventually. A filter is then applied to separate the correct, converged eigenvalues from the extraneous ones. The price here being that one must perform many iterations before the whole eigenspectrum will be found.

- Reorthogonalisation

- Cullum and Willoughby

Reorthogonalisation

One way of dealing with the 'loss of orthogonality' is to enforce orthogonality amongst the Lánczos vectors. The idea is to keep the Krylov space relatively small and whenever a 'breakdown' is reached, *i. e.* we lose orthogonality amongst the Lánczos vectors, we stop and explicitly re-orthogonalise the Lánczos vectors and proceed until the next breakdown is reached. This is costly both in terms of storage and arithmetic as every Lánczos vector must be stored, and we can end up having to re-orthogonalise often.

There have been studies into 'Partial Reorthogonalization (PRO)' [36], where instead of maintaining complete orthogonality amongst the Lánczos vectors, q, a compromise of 'semi-orthogonality' is made; $\cos(q_i, q_j) \leq \sqrt{\epsilon}, i \neq j$, where ϵ is the machine precision. It turns out that the best method of reorthogonalisation is to use a mixture of full-reorthogonalisation and partial-reorthogonalisation, depending on the state of orthogonality-loss and whether or not we have explicit knowledge of any of the eigenvectors.

Cullum and Willoughby

In the late 1970's Cullum and Willoughby [38] came up with a reliable implementation of the Lánczos method which could determine as much of the eigenspectrum of a large symmetric matrix as was needed. The amount of work required to calculate the entire eigenspectrum can still be substantial, and can require anything from $\sim n$ iterations, up to $10n$ iterations, for an $n \times n$ matrix, depending on the eigenvalue distribution. This method is used by [39, 40].

This method relies on the fact that the converged eigenvalues of the Lánczos tri-diagonal matrix, T_m, are eigenvalues of the matrix H. This is a consequence of the Cauchy Interlace Theorem [36], which basically states that an ordered set of eigenvalues of a minor matrix, T_m is bounded in some way by the eigenvalues of T_n (the full tridiagonal matrix of H). In fact the eigenvalues of T_m converge to the eigenvalues of T_n as $m \to n$. Then it is a matter of deciding which of the eigenvalues of T_m have actually converged. This is done in one of two ways.

Firstly, one can compare the eigenvalues of T_m with those of T_{m-1}, the minor matrix created when the last row and column of T_m are deleted, *i. e.* the tridiagonal matrix created with one fewer Lánczos sweep. Any eigenvalues which are not eigenvalues of both are those which are caught in a *hop* (see chapter 3) and are referred to as *spurious*. Then it is just a matter of setting some tolerance for determining which eigenvalues of T_m are 'close' to those of T_{m-1}.

Secondly, one can compare the eigenvalues of T_m with those of $\hat{\mathsf{T}}_2$, the minor matrix created when the *first* row and column of T_m are deleted. In [41] Cullum and Willoughby argue that any eigenvalue of T_m which is also an eigenvalue of $\hat{\mathsf{T}}_2$ is spurious.

Such methods can be applied in order to find as many of the eigenvalues as desired, but this still requires considerable effort. This method gives no information about degeneracies.

2.5 Orthogonal Polynomials

I now make an apparent digression to discuss the theory of orthogonal polynomials. As we shall see, this theory is fundamental to the work discussed in this thesis.

The theory of orthogonal polynomials is fundamental to two of the concepts discussed in this thesis. Firstly, the Bai, Fahey and Golub (BFG) method later described relies on the analogy between the Lánczos procedure and Gaussian quadrature which has orthogonal polynomials as its basis. Secondly, we make use of Chebyshev polynomials in order to make efficient low-order approximations of various functions.

2.5.1 Inner-product spaces

An inner-product space satisfies the following properties:

$$
\begin{aligned}
(f,g) &= (g,f)^* && \text{where } ^* \text{ represents the complex conjugate} \\
(\alpha f + \beta g, h) &= \alpha(f,h) + \beta(g,h) && \\
(f,f) &\geq 0 && \text{with equality if, and only if } f = 0
\end{aligned}
\tag{2.3}
$$

$$
f, g, h \in V \qquad \alpha, \beta \in F(= \mathbb{R} \text{ or } \mathbb{C})
$$

where V is a vector space over $\mathbb{R}$ or $\mathbb{C}$. We also define the *norm*

$$
||x|| \equiv \sqrt{(x,x)}\,. \tag{2.4}
$$

2.5.2 Polynomials, weights and measures

For our applications, we define an inner-product over the real functions f and g with weight ω

$$
(f,g) \equiv \int_a^b \omega(x) g(x) f(x) dx\,. \tag{2.5}
$$

Any linearly independent set of functions $\{\mathcal{F}\}$ can be orthonormalised with respect to an inner product [42]. So given a weight, $\omega(x) > 0$, over an interval $[a,b]$, we are free to take the set of functions $\{f\}$ to be a set of linearly independent polynomials $\{\mathcal{P}\}$. We can create this set of polynomials by taking the powers $1, x, x^2, \ldots$ and orthonormalising them with respect to the inner product Eq. (2.5)

$$
\int_a^b \omega(x) p_i(x) p_j(x) dx = \delta_{ij} \qquad p_i, p_j \in \{\mathcal{P}\}\,. \tag{2.6}
$$

Given a further condition (in this case we set the coefficient of the highest power in each polynomial to be positive[1]) we arrive at a unique set of polynomials. If we take such a set of polynomials $p_0(x), p_1(x), \ldots, p_n(x), p_{n+1}(x)$, orthonormal on $[a, b]$ with respect to the weight function $\omega(x)$ and let $x_1, x_2, \ldots, x_{n+1}$ be the $n+1$ zeros of $p_{n+1}(x)$, then by the fundamental theorem of Gaussian integration,

$$\int_a^b \omega(x) f(x) dx = \sum_{i=1}^{n+1} \omega_i f(x_i) \qquad \forall f \in \mathcal{P}_{2n+1}\,, \tag{2.7}$$

where $w_j = \frac{(p_n, p_n)}{p_n(x_j) p'_{n+1}(x_j)}$.

Now for $j, k \leq n, p_j(x) p_k(x) \in \mathcal{P}_{2n+1}$. Therefore orthonormal polynomials also obey the discrete orthonormality condition

$$\sum_{k=1}^{n+1} \omega_k p_i(x_k) p_j(x_k) dx = \delta_{ij} \qquad p_i, p_j \in \{\mathcal{P}_n\}\,. \tag{2.8}$$

From [42], we can see that given an inner product which satisfies the further condition that $(xf, g) = (f, xg)$, then we have the following three-term recurrence

$$\begin{aligned} p_{-1} &\equiv 0 \\ p_1 &\equiv 1 \\ p_{n+1} &= (x - a_n) p_n(x) - b_n p_{n-1}(x), \qquad n = 0, 1, \ldots \end{aligned} \tag{2.9}$$

where

$$\begin{aligned} a_n &= \tfrac{(xp_n, p_n)}{(p_n, p_n)} \qquad && n = 0, 1 \\ b_n &= \tfrac{(p_n, p_n)}{(p_{n-1}, p_{n-1})} \qquad && n = 1, 2, \qquad b_0 = 0 \end{aligned} \tag{2.10}$$

(in the case where $p_0, p_1, \ldots, p_{n+1}$ are a set of polynomials, orthonormal with respect to the inner-product (f, g), then this reduces to $a_n = (xp_n, p_n), b_n = 1$).

Calculating polynomials in a numerically stable fashion can be quite difficult, but the above three-term recurrence leads to an efficient 'backwards' recurrence allowing for the

[1] If this leading coefficient is unity, then the polynomials are known as *monic*.

computation of expansions of the form

$$f(x) = \sum_{r=0}^{N} c_r p_r(x) \tag{2.11}$$

in a systematic, numerically stable manner.

2.5.3 Chebyshev polynomials

A classical example of a set of orthogonal polynomials is that of the Chebyshev (Tchebichef) polynomials.

First consider the form $\cos(k\theta))$. It will be noted that whilst this is not a polynomial in θ, it is a polynomial of degree k in $\cos(\theta)$. So if we replace θ with $\arccos(x)$, we have a set of functions of the form

$$T_n(x) = \cos(n \arccos x)\,, \tag{2.12}$$

which are simply polynomials in x (despite their trigonometric appearance). These polynomials are known as the Chebyshev polynomials:

$$\begin{aligned}
T_0(x) &= 1 \\
T_1(x) &= x \\
T_2(x) &= 2x^2 - 1 \\
T_3(x) &= 4x^3 - 3x \\
T_4(x) &= 8x^4 - 8x^2 + 1 \\
&\dots \\
T_{n+1}(x) &= 2xT_n(x) - T_{n-1}(x) \qquad n \geq 1\,.
\end{aligned} \tag{2.13}$$

In the language of the previous section, the Chebyshev polynomials are defined by the following properties.

The Chebyshev polynomials are orthogonal over an interval $[-1,1]$[2] with weight

$$\omega(x) = \frac{1}{\sqrt{1-x^2}}\,, \tag{2.14}$$

and the continuous orthogonality and normalisation condition

$$\int_{-1}^{1} \frac{T_i(x)T_j(x)}{\sqrt{1-x^2}}dx = \begin{cases} 0 & i \neq j \\ \frac{\pi}{2} & i = j \neq 0 \\ \pi & i = j = 0 \end{cases} \tag{2.15}$$

together with the discrete orthogonality condition

$$\sum_{k=1}^{m} T_i(x_k)T_j(x_k) = \begin{cases} 0 & i \neq j \\ \frac{m}{2} & i = j \neq 0 \\ m & i = j = 0 \end{cases}\,. \tag{2.16}$$

The recurrence relation,

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \qquad n \geq 1 \tag{2.17}$$

is easy to see by considering the identity

$$\cos((n+1)\theta) + \cos((n-1)\theta) \equiv 2\cos\theta\cos n\theta\,. \tag{2.18}$$

The zeros of the polynomial $T_n(x)$ in $[-1,1]$ are located at the points

$$x = \cos\left(\frac{\pi(k-\frac{1}{2})}{n}\right) \qquad k = 1, 2, \ldots, n\,, \tag{2.19}$$

and the $n+1$ extrema in $[-1,1]$ are located at the points

$$x = \cos\left(\frac{\pi k}{n}\right) \qquad k = 0, 1, \ldots, n\,. \tag{2.20}$$

These maxima are $T_n(x) = 1$ (and minima $T_n(x) = -1$) and over the interval $[-1,1]$, $T_n(x)$ oscillates between these extremal values exactly n times, it is this *equioscillation* which gives the Chebyshev polynomials their optimal properties when using them to approximate functions.

[2]The Chebyshev polynomials are defined outside this interval, but would then have to be expressed in terms of hyperbolic cosines.

2.5.4 Chebyshev approximations

Minmax theory [43] states that if we choose a monic polynomial, P_n, of degree n (*i. e.* P_n has leading order unity), for which $\max|P_n(x)|$ is a minimum over the interval $[-1,1]$, then there can only be one such polynomial and this will have $n+1$ alternating extrema, $+M$ and $-M$ located at the equally spaced points $-1 < x_1 < x_2 < \cdots < x_{n-1} < 1$. So, in a sense these so-called *minmax* polynomials are the polynomials which stay 'closest' to zero over the range $[-1,1]$.

In general these polynomials are hard to find. It turns out however that the Chebyshev polynomials are in fact multiples of these minmax polynomials and using the above technology the Chebyshev polynomials are easy to find.

The optimal properties of this type of polynomial are of particular importance in numerical approximations. It is straightforward to show that any polynomial, p_n can be expressed as a polynomial P_m (where $m < n$) plus a Chebyshev polynomial of order n. As this Chebyshev polynomial is optimally close to zero, we know that, of all polynomials of order m, the lower order polynomial P_m is the one which is optimally close to P_n.

This procedure can be generalised to include functions which are not polynomials. In order to find the appropriate coefficients for the Chebyshev polynomials we make use of the discrete orthogonality condition Eq. (2.16), the details of which can be found in many books on numerical analysis, *e. g.* [44, 43]. Suffice to say that we are able to make polynomial approximations to any function over a range $[-1,1]$. If the function lies outside of the interval $[-1,1]$, but inside a range, say, $[a,b]$ then we can make the transformation

$$y \equiv \frac{x - \frac{1}{2}(b+a)}{\frac{1}{2}(b-a)}\,, \tag{2.21}$$

where the approximation now becomes a polynomial in y.

In practical terms, in order to create such approximations in a numerically stable manner, we make use of a backwards recurrence Eq. (2.11) known as *Clenshaw* Recurrence. Again the details of this are given in [44].

Later we wish to generalise further the idea of approximating functions by Chebyshev polynomials to the case of matrix functions. Most of the information above is then applied to matrix case . The interval $[a, b]$ now becomes $[\lambda_{min}, \lambda_{max}]$, where λ_{max} and λ_{min} are the maximum and minimum eigenvalues of the matrix in question.

2.6 Gaussian Quadrature

As should be clear from the previous sections, we can find a set of weights, ω_j, and abscissa, x_j, such that

$$\int_a^b \omega(x) f(x) dx \approx \sum_{j=1}^{N} \omega_j f(x_j) \tag{2.22}$$

is exact if $f(x)$ is a polynomial. But what about when $f(x)$ is not a polynomial? This depends on how well the function $f(x)$ can be approximated by a polynomial. If a function is expanded as a Taylor series, then we can express this as a $(2m-1)^{th}$-degree polynomial plus higher order terms. The $(2m-1)^{th}$-degree polynomial can be thought of as a linear combination of $(2m-1)^{th}$-degree orthogonal (Chebyshev) polynomials whereas the first of these higher order terms is proportional to $f^{(2m)}$. Therefore we are able to write down an expression for the error [42]

$$E(f) = \frac{1}{(2m)! k_m^2} f^{(2m)}(\zeta), \quad a \leq \zeta \leq b\,. \tag{2.23}$$

So in the case where $f(x) = \log(x)$ we have

$$E = \frac{1}{2m k_m^2 \zeta^{2m}}\,. \tag{2.24}$$

2.7 Quadrature Approximation - Bai, Fahey and Golub

It turns out that the Lánczos procedure can be manipulated to provide information about matrix-vector inner-products $(\Psi, f(\mathsf{H})\Psi)$, where Ψ is an N component complex vector and $f(\mathsf{H})$ is a bounded matrix valued function. This was noticed by Bai, Fahey and Golub [37]. They identified a relationship between the nodes and weights needed for n-point Gaussian quadrature and the eigenvalues and eigenvectors of an $n \times n$ Lánczos (tridiagonal) matrix.

Given an arbitrary matrix function $f(\mathsf{H})$, *e. g.* $f(\mathsf{H}) = \log(\mathsf{H})$, then if H is Hermitian, and positive definite[3], we can express

$$(\Psi, f(\mathsf{H})\Psi) = \sum_{i=1}^{n} z_i f(\lambda_i) \tag{2.25}$$

$$z_i = |\Psi^\dagger \cdot y_i|^2 \tag{2.26}$$

where the λ_i are the eigenvalues of the matrix H and y_i are the eigenvectors.

This equation can be represented as an *Stieltjes* integral

$$(\Psi, f(\mathsf{H})\Psi) = \int d\mu(\Psi, \mathsf{H}) f(x)\,, \tag{2.27}$$

where the spectral measure $d\mu$ is defined as

$$d\mu(\Psi, \mathsf{H}) = \sum_{i=1}^{n} z_i \delta(x - \lambda_i)\,. \tag{2.28}$$

When applied in exact arithmetic this relationship is a precise one. Unfortunately, we have no practical way of finding all the eigenvalues of a large matrix so we have to resort to numerical methods of finding such eigenvalues which are of course only applied in finite precision arithmetic. It is possible to define an m-point Gaussian quadrature rule to estimate the above *Stieltjes* integral

$$(\Psi, f(\mathsf{H})\Psi) = \sum_{i=1}^{N} |a_i|^2 f(\lambda_i) \approx \sum_{i=1}^{m} |w_i|^2 f(\theta_i)\,, \tag{2.29}$$

[3]It turns out that in our case we only need to consider the absolute values of eigenvalues. So our matrix does not actually have to be positive definite, and indeed usually isn't.

where the sum is now over the product of some Gaussian weights w_i and a function of the abscissa θ_i. So the question is how do we find these weights and abscissa?

At this point we have to take another look at the Lánczos algorithm itself.

2.7.1 The Lánczos connection

When we create a sequence of tri-diagonal matrices T_m, we are in effect creating a sequence of characteristic polynomials, χ_m of degree m.

So for our Lánczos sequence of matrices, T_j, we have a corresponding set of characteristic polynomials, $\{\chi_j(x)\}$. This sequence satisfies a three-term recurrence relationship

$$\chi_j(x) = (\alpha_j - x)\chi_{j-1}(x) - \beta_{j-1}^2\chi_{j-2}(x) \,. \tag{2.30}$$

The existence of such a three-term recurrence relation is sufficient to imply that these polynomials form an orthogonal sequence. It can be shown [36] that these Lánczos polynomials are orthogonal with respect to the *Stieltjes* integral, with discrete measure

$$d\mu_L(x) = \sum_{k=0}^{N} \omega_k \delta(x - \theta_k) dx \,, \tag{2.31}$$

where this measure is dependent on the initial *seed* vector, q_1

$$\omega_i = (v_i^t . q_1)^2 \,. \tag{2.32}$$

We now determine what the nodes θ and weights ω actually are.

Setting $\chi_{-1} \equiv 0$ and $\chi_0 \equiv 1$, we can write the above recurrence relation as a matrix equation

$$xP(x) = \mathsf{T}_m P(x) + \beta_m \chi_m(x) e_m \tag{2.33}$$

where $P(x) = (\chi_1(x), \chi_2(x), \ldots, \chi_{m-1}(x))^T$, $e_m^T = (0, 0, \ldots, 1)$.

We can see that when, for some value of $x = \theta$, $\chi_m(x) = 0$, then $\theta P(\theta) = \mathsf{T}_m P(\theta)$ *i. e.* the eigenvalues of T_m are the nodes of the characteristic polynomial χ_m, known as the *Ritz* values and the eigenvectors are $P(\theta)$.

From [45] p48, we have

$$\begin{aligned} \omega_i^{-1} &= \textstyle\sum_{k=0}^{m}(\chi_k(\theta_i))^2 \\ &= \textstyle\sum_{k=0}^{m-1}(\chi_k(\theta_i))^2 \qquad \text{as } P_m(\theta_i) = 0 \\ &= \omega_i P^T(\theta_i)P(\theta_i) \quad . \end{aligned} \tag{2.34}$$

We know that $P(\theta_i)$ is an eigenvector of T_m, so we can see from Eq. (2.34) that $\sqrt{\omega_i}P(\theta_i)$ is a normalised eigenvector of T_m, say x_i.

We can see that if we consider the first component of the eigenvector, x_i^1, then we have

$$\begin{aligned} x_i^1 &= \sqrt{\omega_i}P^1(\theta_i) \\ &= \sqrt{\omega_i}\chi_0(\theta_i) \\ &= \sqrt{\omega_i}\cdot 1\,. \end{aligned} \tag{2.35}$$

Therefore, we can see that $\omega_i = (x_i^1)^2$, *i. e.* the weights are equal to the first component of the eigenvector squared.

2.8 Summary

We now have a method which for approximating 'trace-like' quantities $(\Psi, f(\mathsf{H})\Psi)$, which is based on the well-known Lánczos method, but does not necessarily require us to calculate the entire eigenspectrum. The aim of the next chapter is to demonstrate the numerical stability of this method with regard to quantities of interest in Lattice QCD.

Chapter 3

Lánczos in Action

3.1 Introduction

The purpose of this chapter is to demonstrate some of the features of the Bai, Fahey and Golub (BFG) method introduced in the previous chapter and then to illustrate its applicability to quantities which occur in lattice QCD. We include several illustrations which show a remarkable stability of the Lánczos method when applied in this manner. One would have thought that due to the early loss of orthogonality amongst the Lánczos vectors, this method would be hopeless for finding even approximate quantities. We will include here a demonstration of this early breakdown of orthogonality and the corresponding duplication of Ritz values. We then go on to show that this is not really a problem in our implementation, due to the self-correcting nature of the associated weight values. In particular We demonstrate the stability of 'trace-like' quantities (quantities of the form $u^T f(\mathsf{H})v$) encountered in Lattice QCD. We also include an explanation of how to identify Ritz values which correspond to approximations of genuine eigenvalues of the original matrix without resorting to such exhaustive methods as the Cullum and Willoughby methods previously described. We then give an explanation for this surprisingly simple phenomenon [46].

3.2 Toy Model

We need a simple model in which to test the theory developed in the previous chapter. Scalar field theory provides us with just such a model. Here we use a non-interacting scalar field theory *i. e.* a free boson theory. The corresponding Lagrangian is given by

$$\mathscr{L} = \frac{1}{2}(\partial_\mu \phi)(\partial^\mu \phi) - \frac{1}{2}m^2\phi^2 \,. \tag{3.1}$$

By integrating $(\partial_\mu \phi)(\partial^\mu \phi)$ over a volume, and dropping the surface term, we find

$$(\partial_\mu \phi)(\partial^\mu \phi) = -\phi\partial_\mu\partial^\mu \phi \,. \tag{3.2}$$

This leads us to an action

$$S = \frac{1}{2}\sum_{x,y} a^4 \phi(x)(\Box + m^2)_{x,y}\phi(y) \,, \tag{3.3}$$

where $\Box$ is the D'Alembertian, given on the lattice as

$$\Box = \Lambda^b_\mu \Lambda^f_\mu \,, \tag{3.4}$$

and Λ^b_μ and Λ^f_μ are the backwards and forwards derivatives

$$\begin{aligned} \Lambda^b_\mu \phi(x) &= \tfrac{1}{a}(\phi(x + a\hat{\mu}) - \phi(x)) \\ \Lambda^f_\mu \phi(x) &= \tfrac{1}{a}(\phi(x) - \phi(x - a\hat{\mu}))\,. \end{aligned} \tag{3.5}$$

So we have

$$\Box\phi = \partial_\mu \partial^\mu \phi = -\frac{2\phi_n - \phi_{n+\hat{\mu}} - \phi_{n-\hat{\mu}}}{a^2} \,, \tag{3.6}$$

which gives us a symmetric matrix representation required when using Lánczos methods.

We can calculate the eigenvalues λ_n of this matrix explicitly by performing a Fourier transform into momentum space. This model is particularly useful for our present analysis as we are able to vary most of the parameters at will. For example, we have a choice of what kind of boundary conditions to insert. Here we introduce a phase shift as follows

$$\phi(x + aL) = \phi(x)e^{i\alpha} \,, \tag{3.7}$$

which enables us to control the degeneracies in the eigenspectrum.

We can also alter the complexity of the problem by altering the number of dimensions and the size of the lattice. So for 4 dimensions we have

$$\begin{aligned} \lambda_n = m^2 + 2\sum_{\mu=1}^{4}\left[1-\cos\left(\frac{2n_\mu\pi-\alpha_\mu}{L_\mu}\right)\right] &= m^2 + 4\sum_{\mu=1}^{4}\sin^2\left[\frac{1}{2}\left(\frac{2n_\mu\pi-\alpha_\mu}{L_\mu}\right)\right] \\ &\approx m^2 + \sum_{\mu=1}^{4}(k_{n_\mu} - \frac{\alpha_\mu}{L_\mu})^2\,, \end{aligned} \tag{3.8}$$

for small momenta, k_{n_μ} where $k_{n_\mu} = \left(\frac{2n_\mu\pi}{L_\mu}\right)$.

3.2.1 Demonstration

So let us consider a simple example, which demonstrates what happens when the BFG Lánczos method is applied to the toy model described above.

Example 1: A seemingly trivial example

Here we consider a 2^3 lattice. The parameters are shown in table 3.1. As our Lánczos starting vector we use a standard vector e, where

$$\mathsf{e} = (1,0,0,\ldots,0)^T \tag{3.9}$$

then we can easily calculate what the associated weights should be. In the following example we have normalised these weights such that for each non-degenerate converged Ritz value the corresponding weight should be unity.

As shown above, we can easily find the eigenvalues analytically. These are given in Eq. (3.10). It is useful to discuss the form of this eigenspectrum, even for such a small example.

Table 3.1: Parameters for toy example 1.

Parameter	Value
No. of dimensions	3
size in x direction	2
size in y direction	2
size in z direction	2
Lattice Volume	$2^3 = 8$
x phase shift	0.1
y phase shift	0.2
z phase shift	1.2
field mass m	2.1

The true eigenvalues for example 1 (to 8 d.p.)

$$\begin{array}{rr} 1 & 4.77181992 \\ 2 & 8.07316238 \\ 3 & 8.75183658 \\ 4 & 8.76682096 \\ 5 & 12.05317904 \\ 6 & 12.06816342 \\ 7 & 12.74683762 \\ 8 & 16.04818008 \end{array} \tag{3.10}$$

We can see that by means of the phase shifts we have introduced into our boundary conditions, we have arranged for some of the eigenvalues to be almost degenerate. We now explore how the Lánczos algorithm deals with this eigenspectrum.

First of all we run the Lánczos algorithm for just a few iterations. It is instructive to look at what happens, even at this early stage.

The Ritz values after 4 iterations

	Ritz value	weight
1	4.78683406	1.01844404
2	8.53373043	2.98155596
3	12.28626957	2.98155596
4	16.03316594	1.01844404

(3.11)

Even after just 4 iterations we can see that the two extremal eigenvalues are starting to converge as their corresponding normalised weights are becoming close to unity. So now we move on and look at the situation after 6 iterations.

The Ritz values after 6 iterations

	Ritz value	weight
1	4.77181996	1.00000008
2	8.07327892	1.00055605
3	8.75945998	1.99944387
4	12.06054002	1.99944387
5	12.74672108	1.00055605
6	16.04818004	1.00000008

(3.12)

Now we see that the two smallest eigenvalues, and the two largest eigenvalues have been almost determined with the appropriate weights. The two almost degenerate pairs of eigenvalues have also been found, but in each case only one instance of each eigenvalue has appeared. Interestingly though, the doubled weights almost perfectly compensate for this.

If we look at what happens after 8 iterations, we find that all of the eigenvalues have been found to several decimal places.

The Ritz values after 8 iterations

	Ritz value	weight
1	4.77181992	1.00000000
2	8.07316238	1.00000000
3	8.75183658	1.00000000
4	8.76682096	1.00000000
5	12.05317904	1.00000000
6	12.06816342	1.00000000
7	12.74683762	1.00000000
8	16.04818008	1.00000000

(3.13)

Of course we can continue the process further. Eq. (3.14) shows the results of a run of 9 Lánczos sweeps.

The Ritz values after 9 iterations

	Ritz value	weight
1	4.77181992	1.00000000
2	8.07316238	1.00000000
3	8.75183658	1.00000000
4	8.76682096	1.00000000
5	12.05317904	1.00000000
6	12.06816342	1.00000000
7	12.74683762	1.00000000
8	15.73264277	0.00000000
9	16.04818008	1.00000000

(3.14)

What we find now is that another eigenvalue, '15.73264277', has appeared, although this has nothing to do with the true eigenspectrum, *i. e.* it is an unconverged eigenvalue. As we can see, its associated weight is very small so this incorrect eigenvalue will have little adverse effect on the eigenspectrum we have found.

Finally, we can look at what happens after 16 iterations. Eq. (3.15) shows the result of such a run, with the Ritz values shown in clusters and a third column showing the sum of weights for each cluster.

The Ritz values after 16 iterations

	Ritz value	weight	sum of weight in cluster
1	4.77181992	0.00000000	1.00000000
2	4.77181992	1.00000000	
3	8.07316238	0.01145215	1.00000000
4	8.07316238	0.98854785	
5	8.75183658	0.99997701	1.00000000
6	8.75183658	0.00002299	
7	8.76682096	0.99998885	1.00000000
8	8.76682096	0.00001115	
9	12.05317904	0.99999872	1.00000000
10	12.05317904	0.00000128	
11	12.06816342	0.99999896	1.00000000
12	12.06816342	0.00000104	
13	12.74683762	0.71401568	1.00000000
14	12.74683762	0.28598432	
15	16.04818008	0.99999964	1.00000000
16	16.04818008	0.00000036	

(3.15)

So we see that any incorrect degeneracies are compensated for according to their corresponding weights. Any non-converged Ritz values will have an associated weight which is very small.

These effects can be seen in Fig. (3.1) which shows the smoothness of functions estimated using the above data.

We have seen in this small example that the mechanism contains a self-correcting structure implemented via the weights. Either a Ritz value is unconverged, in which case it is irrelevant to the genuine eigenspectrum, and so has a weight of almost zero, or alternatively, the Ritz value has converged and so is a genuine approximation to an eigenvalue. If the Ritz value has converged then the weight is unity unless there are multiple copies of a converged Ritz value, in which case the weight is shared amongst the copies.

Furthermore, if a Ritz value appears which is attempting to mimic several eigenvalues which are in close proximity in the genuine eigenspectrum, then the weight is appropriately larger to compensate.

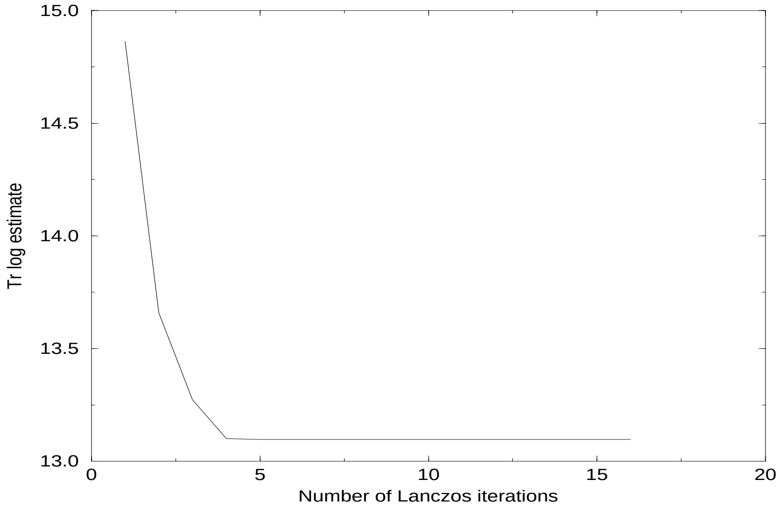


Figure 3.1: The convergence of the $\mathrm{Tr}\log(\mathsf{H})$ for the simple example of a 2^3 lattice.

Example 2: A less trivial example

This example involves a slightly bigger lattice. The parameters are given in table 3.3.

Tab. (3.2) shows a sample of the Ritz values after 100 Lánczos sweeps. We can see that this time the breakdown of orthogonality has happened earlier, and we are experiencing multiple copies of some Ritz values even before we have found the full eigenspectrum.

3.2.2 Conclusions drawn from the toy model

We have shown that in this simple case, the BFG implementation of the Lánczos method provides a viable way of calculating 'trace-like' quantities, *i. e.* quantities of the form $\phi^\dagger f(\mathsf{H})\phi$, and in these simple cases the method can also determine some detailed information about the eigenspace itself.

It turns out that most of these features generalise when we come to consider more complicated cases involving bigger lattices and different types of starting vector. In general however, the weight associated with each eigenvalue is different and is given by the Gaus-

Table 3.2: A sample of The Ritz values for example 2

Ritz value	weight
4.44033620	0.17065157
4.44033620	0.82934843
4.91557338	1.00000000
4.91817939	0.00000000
5.14837926	1.00000000
5.17094248	1.00000000
5.21561705	1.00000000
5.23771927	1.00000000
5.44906439	1.00000000
5.62361643	1.00000004
5.64617965	1.00000008
5.69085424	1.00000206
5.71295647	0.99999787
5.87898788	1.00021677
5.92370638	1.00216599
5.96851020	1.00280552
⋮	⋮
14.62702775	1.85625958
14.64759629	1.13776137
14.74691343	0.99996092
14.77170639	1.00004501
14.79649451	1.00000877
14.82128709	1.00000019
15.60326240	1.00000000
15.62805492	1.00000000
15.65284237	1.00000000
15.67763488	1.00000000
15.89932758	1.00000000
15.92412009	1.00000000
15.94890755	1.00000000
15.97350837	0.00000000
15.97370006	1.00000000

sian weight as described in the previous chapter.

3.3 Lattice QCD Results

We now go on to show that, in the more complicated cases involving full QCD lattices, we are able to calculate such 'trace-like' quantities in an efficient manner. When it comes to making a detailed examination of the eigenspace, we find that once we get past the free field calculations and on to the more physically interesting cases, we are not able

Table 3.3: Parameters for toy example 2.

Parameter	Value
No. of dimensions	3
size in x direction	7
size in y direction	7
size in z direction	7
Lattice Volume	$7^3 = 343$
x phase shift	0.1
y phase shift	0.2
z phase shift	1.2
field mass m	2.1

to determine very much about the individual eigenvalues. However, estimates of the extremal eigenvalues are still readily forthcoming. As our interest here is in 'trace-like' quantities, which, as we have shown, can be calculated without detailed knowledge of the eigenspectrum, this does not present a problem.

3.3.1 The free-field case

We now move away from scalar field theory and consider some examples from lattice QCD. As with our investigations of the scalar field theory, we first consider the free-field case.

As shown in chapter 1, we are particularly interested in the calculation of the quantity $\mathrm{Tr}\log(\mathsf{H})$, for a Hermitian representation, H, of the Fermion matrix. Fig. (3.5) shows a plot of the $\mathrm{Tr}\log(\mathsf{H})$ function against the number of Lánczos iterations for a $16^3 \times 32$ lattice in the free field case. The results are shown here for three different Hermitian representations of the fermion determinant.

As with the scalar field theory, in the free-field case we are able to calculate the eigenvalues exactly. We consider here firstly the case of the matrix $\mathsf{Q}'^2 = \mathsf{M}^\dagger\mathsf{M}$ with red-black preconditioning applied. As shown in appendix D, in the free field case the eigenvalues are given by

$$\lambda_{\mathsf{Q}'^2} = 1 + 8\kappa^2 \left(\sum_{\mu=1}^{4} \sin^2 P_\mu - \left(\sum_{\mu=1}^{4} \cos P_\mu \right)^2 \right) + 16\kappa^4 \left(\sum_{\mu=1}^{4} \sin^2 P_\mu + \left(\sum_{\mu=1}^{4} \cos P_\mu \right)^2 \right)^2 . \tag{3.16}$$

Again from appendix D we can also calculate the eigenvalues of the full matrix in the (free-field) case of $\mathsf{Q}^2 = \mathsf{M}^\dagger \mathsf{M}$

$$\lambda_{\mathsf{Q}^2} = 4\kappa^2 ((\frac{1}{2\kappa} - \sum_{\mu=1}^{4} \cos P_\mu)^2 + \sum_{\mu=1}^{4} \sin^2 P_\mu) . \tag{3.17}$$

Consequently, we can write down an expression for the eigenvalues of the matrix $\mathsf{Q} = \gamma_5 \mathsf{M}$

$$\lambda_{\mathsf{Q}} = \pm 2\kappa \sqrt{(}(\frac{1}{2\kappa} - \sum_{\mu=1}^{4} \cos P_\mu)^2 + \sum_{\mu=1}^{4} \sin^2 P_\mu) . \tag{3.18}$$

3.3.2 Determining true eigenvalues - convergence test

In order to determine all of the eigenvalues of a large Hermitian matrix using the Lánczos method, we can apply the Cullum and Willoughby method described in the previous chapter. This method usually requires that we perform a very large number of Lánczos iterations, sometimes as many as 10 times as many iterations as the size of the matrix. However, we may only wish to determine a subset of these eigenvalues to some desired accuracy. To determine which Ritz values represent genuine approximations to those found in the true eigenspectrum, rather than spurious copies, or unconverged Ritz values, we use a filter and then a convergence test.

The filter simply removes all Ritz values which have a weight of zero. In order then to determine which of the remaining Ritz values have converged to an approximation of a single genuine eigenvalue or to a genuinely degenerate set of eigenvalues, in the free-field case there are two ways we can proceed. Firstly, we can study the weight values. Here we cluster approximately equal Ritz values together, average them, and then sum the weights in each cluster. We can easily normalise the weight values such that they should be equal to unity for genuine eigenvalues, and equal to p for $p-$degenerate eigenvalues. Any Ritz values which are genuine approximations to eigenvalues will then have a corresponding weight value which is a multiple of unity.

As we will see, this method works well for extremal eigenvalues, but becomes more difficult in the case of the inner eigenvalues.

Secondly, we use a convergence test. The convergence test requires us to compute the Ritz-spectrum for m sweeps, and then compare it with that for $m-1$ sweeps. Any Ritz values which have not converged will differ significantly between the two spectra and should be removed. Obviously, we need to define what we mean by 'differ significantly' and for this we must define a tolerance. We can then identify clusters of eigenvalues.

In the case of the full interacting theory, we have no *a priori* knowledge of the weighting values, and therefore the value of non-zero weights gives no useful information to us when studying the reliability of a Ritz value as an approximation to an eigenvalue. In this case we have no alternative but to employ the convergence test.

So naturally we first study the free-field case. We use a lattice size of $16^3 \times 32$, which is the same size as the lattices we use in the full fermion interacting case. In this case the hermitian representation of our fermion matrix is $\mathsf{H} = \mathsf{M}^\dagger \mathsf{M}$ with red-black pre-conditioning applied. Eq. (3.19) shows a table of Ritz values as found using $m = 1000$ sweeps and then compares those Ritz values with those found for $m-1$ sweeps. In this table the first 41 Ritz values are shown.

The First 41 Ritz values and their unnormalised weights

	Ritz value $m-1$ iterations	Ritz value m iterations	Unnormalised weight	Approx. eigenvalue $\times$ degeneracy
1	0.0306446031359988	0.0306446031359992	0.0000000000030897	
2	0.0306446031360004	0.0306446031359997	0.0000000020730154	
3	0.0306446031360006	0.0306446031360005	0.0000000222890319	
4	0.0306446031360010	0.0306446031360006	0.0000150141519868	
5	0.0306446031360016	0.0306446031360010	0.0000002010078616	
6	0.0306446031360019	0.0306446031360016	0.0000000166035949	
7	0.0306446031360030	0.0306446031360028	0.0000000017258662	
8	0.0306446031360038	0.0306446031360036	0.0000000000019875	0.030644603136×2
9	0.0306446031360054	0.0306446031360042	0.0000000005194786	
10	0.0306446031360068	0.0306446031360073	0.0000000000850968	
11	0.0306446031360086	0.0306446031360092	0.0000000002272397	
12	0.0306446031360110	0.0306446031360117	0.0000000000368373	
13	0.0306446031360142	0.0306446031360131	0.0000000000229784	
14	0.0306446031360150	0.0306446031360160	0.0000000000048332	
15	0.0306446031360166	0.0306446031360167	0.0000000000361645	
16	0.0389326582990179	0.0389326582990161	0.0000000009453478	
17	0.0389326582990191	0.0389326582990185	0.0000000005135895	
18	0.0389326582990192	0.0389326582990195	0.0000000000000021	
19	0.0389326582990193	0.0389326582990203	0.0000305124228698	
20	0.0389326582990198	0.0389326582990204	0.0000000000248651	
21	0.0389326582990199	0.0389326582990207	0.0000000036419699	
22	0.0389326582990211	0.0389326582990210	0.0000000000035380	
23	0.0389326582990214	0.0389326582990217	0.0000000000000017	0.038932658299×4
24	0.0389326582990224	0.0389326582990232	0.0000000000020655	
25	0.0389326582990225	0.0389326582990236	0.0000000000187345	
26	0.0389326582990244	0.0389326582990249	0.0000000000049067	
27	0.0389326582990253	0.0389326582990259	0.0000000000001816	
28	0.0389326582990275	0.0389326582990271	0.0000000000000525	
29	0.0389326582990283	0.0389326582990277	0.0000000000000003	
30	0.0630469031534392	0.0630469031534399	0.0000000000000000	
31	0.0630469031534406	0.0630469031534408	0.0000000000000883	
32	0.0630469031534410	0.0630469031534416	0.0000000000000001	
33	0.0630469031534422	0.0630469031534428	0.0000000000000017	
34	0.0630469031534423	0.0630469031534435	0.0000000000000341	
35	0.0630469031534430	0.0630469031534438	0.0000000000049858	
36	0.0630469031534433	0.0630469031534448	0.0000000000000168	0.063046903153×16
37	0.0630469031534438	0.0630469031534454	0.0000062732140576	
38	0.0630469031534446	0.0630469031534459	0.0000270477311566	
39	0.0630469031534447	0.0630469031534465	0.0000000005315085	
40	0.0630469031534456	0.0630469031534469	0.0000876096321247	
41	0.0630469031534462	0.0630469031534473	0.0000011391985259	

(3.19)

Again we see the same clustering of Ritz values, and here it is easy to identify those which are genuine. As this is the free-field case we can also determine the degeneracy using the weighting values. Here for the final column we have summed the weights in each cluster to calculate the degeneracy of each approximate eigenvalue. The eigenvalues found here compare, almost to machine precision with those calculated using the analytic expressions.

Towards the centre of the Ritz-spectrum, we find (as before) the situation is a little more complicated. A sample of Ritz values from the centre of the Ritz-spectrum is shown in Eq. (3.20). We can see that it is much harder here to cluster together eigenvalues. Without the ability to sum the weights in each cluster, it is impossible to determine whether or not a Ritz value is a genuine approximation to an eigenvalue of the original matrix. Therefore in general we cannot rely on a method which relies on the weight values alone.

A sample of 20 Ritz values and their unnormalised weights taken from the centre of the Ritz spectrum

	Ritz value $m-1$ iterations	Ritz value m iterations	Unnormalised weight
490	0.8651902187177143	0.8651902179420002	0.0001831054573924
491	0.8666972451383683	0.8666972449697184	0.0007324218873553
492	0.8721403758621256	0.8721403743325195	0.0014648436261781
493	0.8734288231889906	0.8734288099536344	0.0007324221163114
494	0.8773381566520988	0.8773183249910335	0.0007588877111432
495	0.8778687811229765	0.8777894636322079	0.0007205922591292
496	0.8790711623327040	0.8790289468680872	0.0015896029052656
497	0.8797762563198102	0.8797538914214713	0.0013254224035005
498	0.8838987853479315	0.8838590899125236	0.0003894105124120
499	0.8852518156688307	0.8851570755898538	0.0009592982805032
500	0.8866683168360393	0.8864695226469931	0.0008703642357137
501	0.8878130216301972	0.8876732341992877	0.0006978801991992
502	0.8898742946449756	0.8898593896096665	0.0007440253678314
503	0.8936690112843909	0.8936119985882448	0.0006791319083856
504	0.8950369623334760	0.8949861545156468	0.0029025752531177
505	0.8975283490074054	0.8966030417984104	0.0004016751578368
506	0.8990416971461762	0.8977574702330426	0.0020612557220313
507	0.9007416089822343	0.8999469628923339	0.0010111109722436
508	0.9025231133198425	0.9014896061821442	0.0012048319841091
509	0.9037793379370265	0.9029033001311656	0.0019055017019806

(3.20)

The convergence test is the only reliable method for determining true eigenvalues. We now study the full fermion interacting theory, where we use the convergence test to find out which Ritz values are (to a desired accuracy) true approximations of eigenvalues of the original matrix.

3.3.3 $16^3 \times 32$ full lattice QCD results

Having performed an eigenvalue analysis in the free-field cases in the same manner as for the scalar field case, we have found that the Ritz values follow the same pattern regarding degeneracies, weight clusters, and in particular smooth convergence of the $\mathrm{Tr}\log(\mathsf{H})$ quantity (as seen in Fig. (3.5)). Taking note of this we immediately move on to the full fermion interacting case using large lattices of the size presently in production, *i. e.* $16^3 \times 32$.

Here we have taken one $(\beta, \kappa, c_{\mathrm{sw}}) = (5.2, 0.1355, 2.0171)$ [26] configuration and run our Lánczos BFG method on the matrix $\mathsf{H} = \mathsf{M}^\dagger \mathsf{M}$ with red-black pre-conditioning applied. We now look at samples of the Ritz values which were found.

Fig. (3.2) shows the number of eigenvalues which can be determined using the convergence test with a given tolerance, *i. e.* it shows how many eigenvalues can be found within a given accuracy. Here a convergence test was applied after having filtered out Ritz values with a weight of zero. The convergence test compares the value of an eigenvalue λ_m of T_m with the corresponding eigenvalue λ_{m-1} of T_{m-1}. The normalised difference, $\frac{\lambda_m - \lambda_{m-1}}{\lambda_m}$, is then taken and, if this difference is less than the tolerance, the corresponding Ritz value is considered to have converged to a true eigenvalue of the original matrix. We are then left with a set of 'true' eigenvalues which can then be clustered together to determine how many distinct eigenvalues of H we have found within the given accuracy. Obviously we need to set a tolerance for determining when a weight is considered to be 'zero', and also when comparing 'true' eigenvalues for distinguishing clusters. Both of these tolerances were set to 0.0000001.

We also now look at the numerical stability of the Lánczos BFG method when applied to the calculation of $\mathrm{Tr}\log(\mathsf{H})$, on a lattice of the type encountered in full lattice QCD. Fig. (3.3) shows this convergence in the case where $\mathsf{H} = \mathsf{M}^\dagger \mathsf{M}$ with red-black

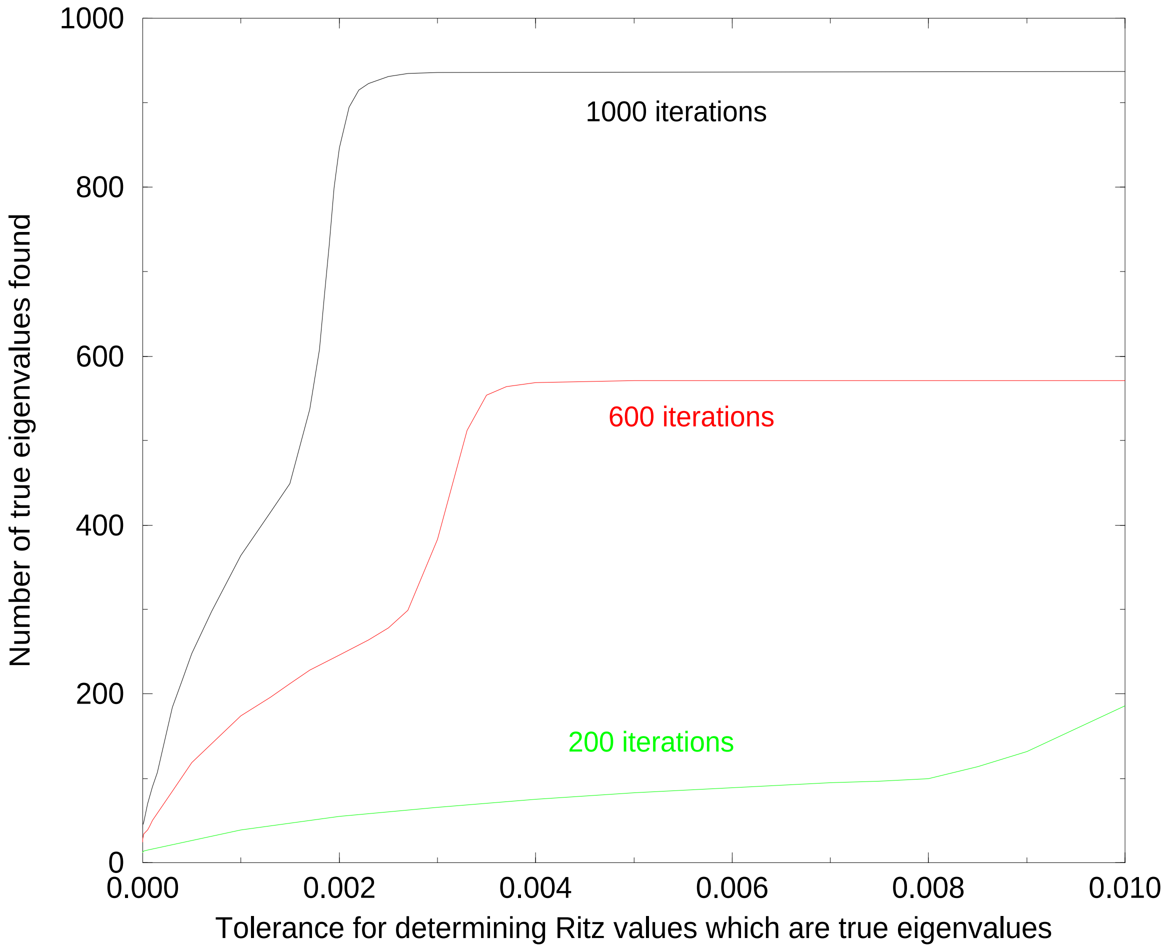


Figure 3.2: The number of Ritz values which have converged to genuine approximations of true eigenvalues determined by a particular tolerance as described in the text. $\mathsf{H} = \mathsf{M}^\dagger\mathsf{M}$ with red-black pre-conditioning applied.

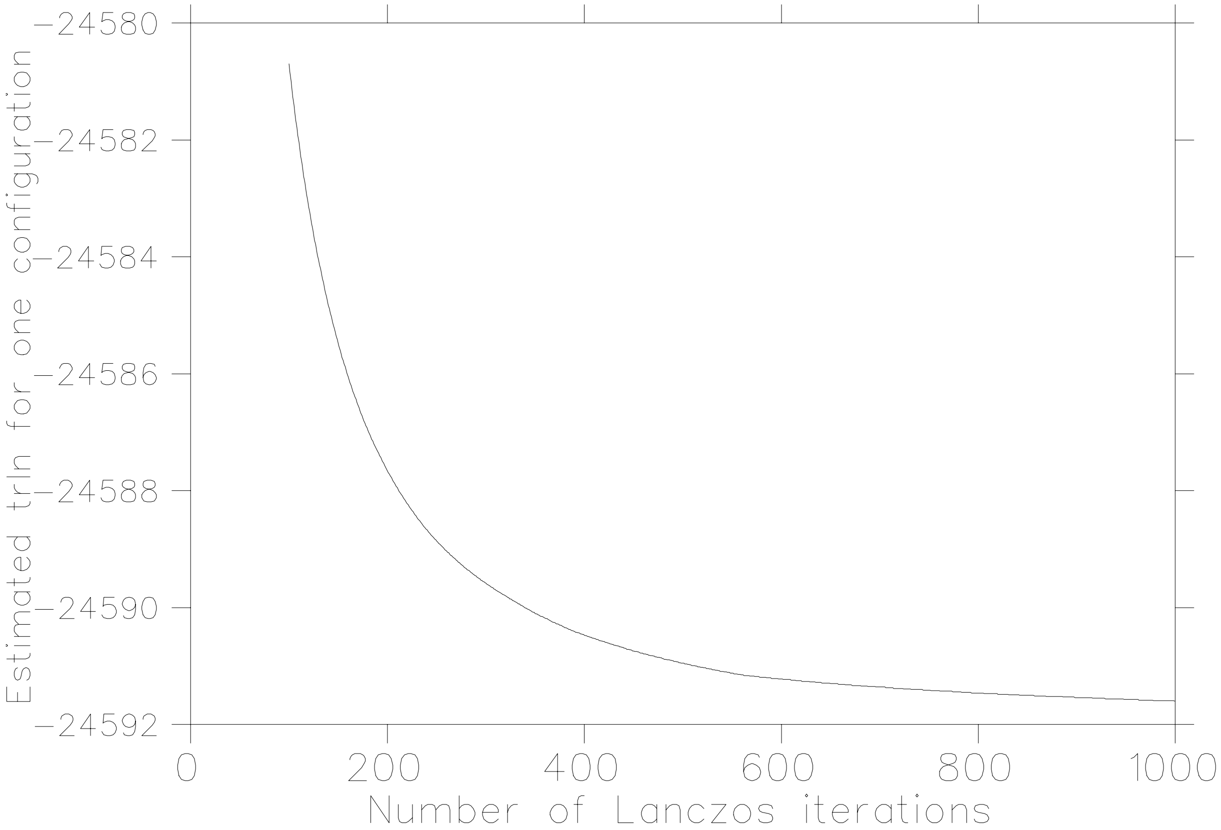


Figure 3.3: Convergence history of the $16^3 \times 32$ lattice.

pre-conditioning applied. We later look at what happens with different Hermitian representations of H. First of all however, we require an explanation of this surprising stability.

3.4 Explanation of Numerical Stability

Fig. (3.4) shows the behaviour of a sample of Ritz values which have been multiplied by their corresponding weights. So here is a description of what actually happens. To start with, the algorithm proceeds with no apparent problem, then after, say, 150 iterations a Lánczos vector fails to be orthogonal to one of its predecessors due to the effects of finite precision arithmetic. After a few more steps, the Lánczos vectors become linearly dependent. As this happens, a Ritz value converges to an eigenvalue of H (the original matrix to be diagonalised). The algorithm then 'forgets' that it has found this eigenvalue and starts to compute it again.

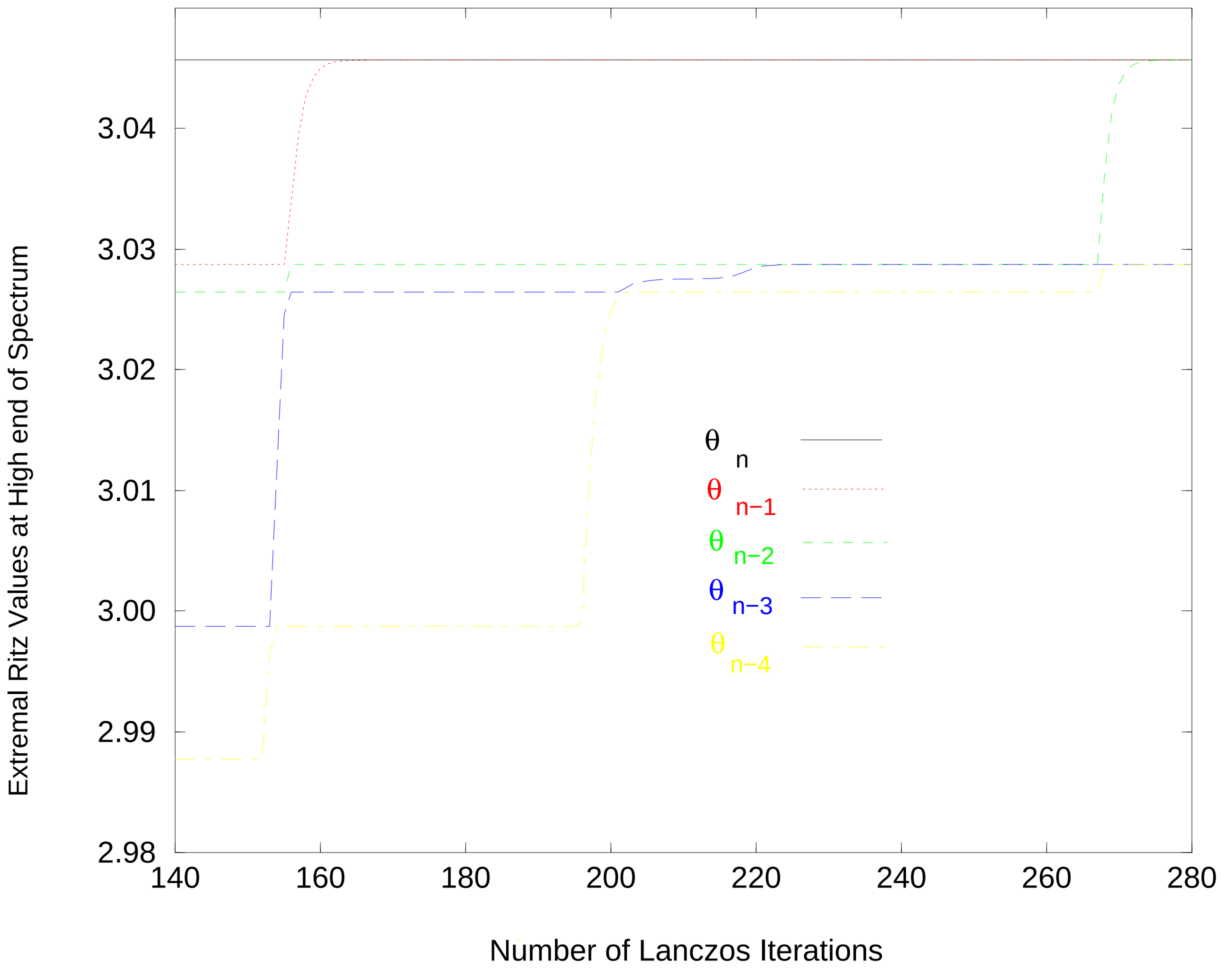


Figure 3.4: The behaviour of the first few Ritz values for the $16^3 \times 32$ lattice.

It turns out that it is still possible to gain accurate information about some of the extremal eigenvalues. One would have thought that the false degeneracy amongst the Ritz values would be catastrophic for most calculations involving the eigenvalues, for example when summing the eigenvalues as in a trace calculation. This is actually not necessarily such a problem if one has some way of compensating for both the false degeneracy and the corresponding 'hopping' of the Ritz values, which gives a Ritz value a totally incorrect value if caught 'mid-hop'.

It turns out that the weights we described earlier have an in-built mechanism which compensates for the false degeneracy of the Ritz values. The mechanism for this is surprisingly simple. Basically, whenever we have a 'cluster' of Lánczos vectors which are correspondingly not linearly independent, we consequently produce a set of approximately equal Ritz values. However, when the corresponding tri-diagonal matrix is diagonalised, a set of orthonormal eigenvalues are always produced and the sum of weights corresponding to a cluster of 'identical' Ritz values turns out to be an invariant quantity. Thus for 'trace-like' quantities such as the ones considered here, we have a reliable method of evaluation despite a breakdown of orthogonality which would naïvely seem disastrous.

3.5 Different Hermitian Representations

As explained in the previous chapter, the Lánczos method requires that the matrix to be diagonalised, H, is *Hermitian.* Unfortunately M itself is not Hermitian (although it is $\gamma_5 - Hermitian$). We can however use M to construct a matrix which is Hermitian. Here we consider the Hermitian matrices, $\gamma_5\mathsf{M}$, $\mathsf{M}^\dagger\mathsf{M}$ and $\mathsf{M}^\dagger_{ee}\mathsf{M}_{ee}$ ($\mathsf{M}^\dagger\mathsf{M}$ with red-black (odd-even) pre-conditioning applied.) There exists a simple relationship between the $\mathrm{Tr}\log(\mathsf{H})$ quantities for all the matrices described above, so we are free to use whichever of these gives the best convergence properties.

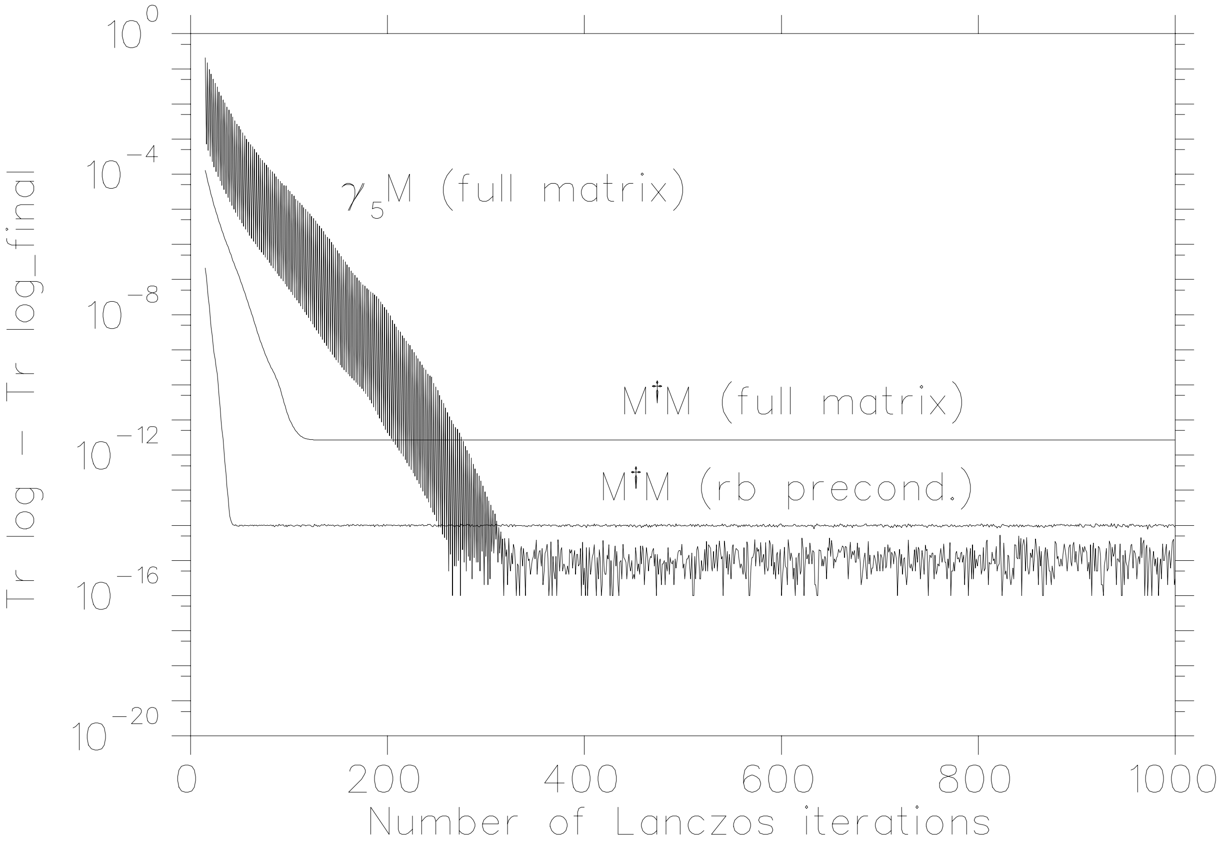


Figure 3.5: The convergence of the normalised $\mathrm{Tr}\log(\mathsf{H})$ for $16^3 \times 32$ lattice in the free-field case. This plot shows the convergence of several different representations of the matrix H.

Fig. (3.5) shows a comparison between the convergence rates of these matrices for the $\mathrm{Tr}\log(\mathsf{H})$ quantity for the free-field case. Fig. (3.6) shows the same result for the full fermion interacting case.

3.6 Conclusion

So we have seen that we have a reliable method of calculating 'trace-like' quantities using the BFG Lánczos method and this method appears to be very efficient for the $\mathrm{Tr}\log\mathsf{H}$ quantity which we wish to calculate. In particular we have found that the most efficient Hermitian representation of the Fermion matrix H is $\mathsf{H} = \mathsf{M}^\dagger\mathsf{M}$ with red-black preconditioning applied.

We also have an efficient way of determining a few eigenvalues efficiently, with a specified

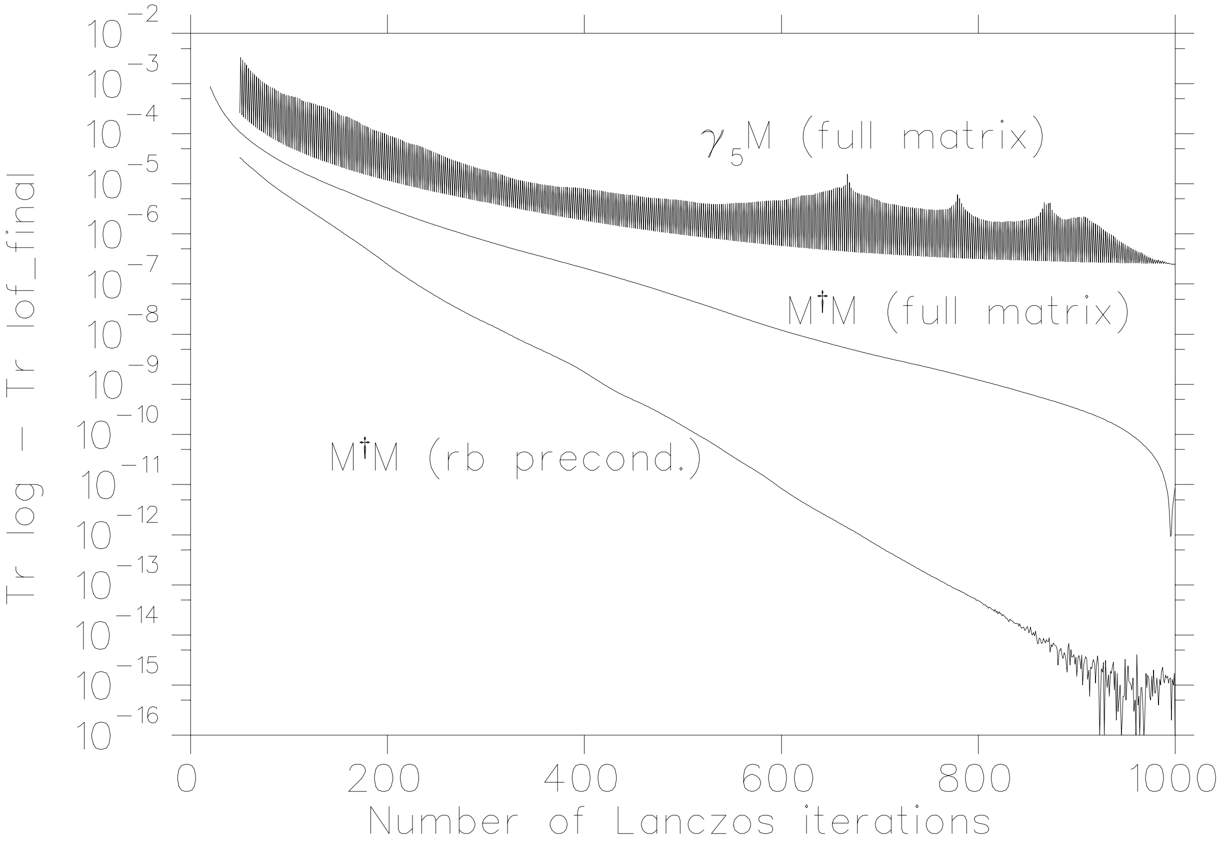


Figure 3.6: The convergence of the normalised $\mathrm{Tr}\log(\mathsf{H})$ for $16^3 \times 32$ lattice in the full fermion interacting case. This plot shows the convergence of several different representations of the matrix H.

accuracy using the same Lánczos BFG methods with an additional filter and convergence test.

Chapter 4

Stochastic Evaluations

4.1 Introduction

This chapter collects together some of the ideas of necessary to produce stochastic estimates of 'trace-like' quantities described previously.

4.2 Variances

The variance of a quantity is one of the most important tools in any statistical analysis. It provides a measure of the average fluctuation of a set of data about its average value. Of course, it says nothing of where this fluctuation comes from, *i. e.* whether the fluctuation is due to some natural randomness of the data in question, or whether it is due to some random statistical error in the measurement.

The variance is formally defined as

$$\sigma^2 = \langle (X - \langle X \rangle)^2 \rangle \,, \tag{4.1}$$

where $\langle X \rangle$ represents the expectation value of the variable x.

In conventional statistics, the variance of a sample of n data points, $x_i \in \mathbb{R}$, is defined by

$$\sigma^2 = \frac{1}{n-1} (\sum_{i=1}^{n} x_i^2 - n\overline{x}^2) \,, \tag{4.2}$$

where $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the average value of x, which is either known explicitly (and then often denoted μ), or calculated statistically.

Similarly if $x_i \in \mathbb{C}$, we often use the following representation of

$$\sigma^2 = \frac{1}{n-1}(\sum_{i=1}^{n} |x_i|^2 - n|\overline{x}|^2)\,, \tag{4.3}$$

which then ensures $\sigma^2 \in \mathbb{R}$.

4.3 Correlations

Another important concept in statistics is that the *correlation* of two quantities. This defines to what extent two variable tend to synchronise their fluctuations. Formally, we have

$$C(X,Y) = \langle (X - \langle X\rangle)(Y - \langle Y\rangle)\rangle\,. \tag{4.4}$$

We define the following, normalised correlation

$$C(x,y) = \frac{\langle (x - \langle x\rangle)(y - \langle y\rangle)\rangle}{\sigma_x \sigma_y}\,. \tag{4.5}$$

4.4 Noisy Estimators

In order to create a stochastic estimate, we choose our noise to have the following properties

$$\begin{aligned} \langle \phi_i \rangle &= 0 \\ \langle \phi_i^* \phi_j \rangle &= \delta_{ij}\,. \end{aligned} \tag{4.6}$$

Then we can calculate trace quantities in the following manner

$$\mathrm{Tr} f(\mathsf{H}) = \langle \phi^\dagger f(\mathsf{H}) \phi \rangle = \sum_{ij} \langle \phi_i^* (f(\mathsf{H}))_{ij} \phi_j \rangle\,. \tag{4.7}$$

We would also like

$$\langle \phi_i \phi_j \rangle = 0\,. \tag{4.8}$$

When determining which type of noise to use, we need to test to what extent the constraints given in Eq. (4.6) are obeyed. From [47] we can write such a test in the form

$$\begin{aligned} C_1 &= \tfrac{1}{N(N-1)} \textstyle\sum_{i\neq j} |\langle \phi_i^* \phi_j \rangle|, \\ C_2 &= \sqrt{\tfrac{1}{N}(\langle \phi_i^* \phi_j \rangle - 1)^2}\,. \end{aligned} \tag{4.9}$$

We consider two types of noise here, namely, Gaussian noise and Z_2 noise. Gaussian noise simply consists of a set of random numbers which satisfy a Gaussian distribution with a given variance. Such a distribution can consist of either real or complex numbers. Z_2 noise consist of a random distribution of 1's and -1's *e. g.* $1, 1, -1, 1, -1, -1, 1, -1, \ldots$. Technically, Z_2 noise fulfills the criteria of Eq. (4.9) better than Gaussian noise, but we find that the difference in performance is less than 10%. We also find that, for our analysis, we are able to handle the expectations of variances better using Gaussian noise, therefore the remaining part of this thesis makes use of Gaussian noise, unless otherwise stated.

In practice we generate a whole set, N_ϕ, of noise vectors, $\phi = (\phi_1, \phi_2, \phi_3, \ldots, \phi_n)^T$ and $\phi^\dagger = (\phi_1^*, \phi_2^*, \phi_3^*, \ldots, \phi_n^*)$, of length n, where each of the components, ϕ_i, satisfy Eq. (4.6). We then create an estimate, E_f, for the trace of a matrix function, f, as follows

$$E_f = \frac{1}{N_\phi} \sum_{i=1}^{N_\phi} \phi^{i\dagger} f(\mathsf{H}) \phi^i\,, \tag{4.10}$$

where, from now on, 'up' indices on our noise vectors, ϕ, indicate different randomly chosen (Gaussian) noise vectors, and 'down' indices indicate the individual components of these noise vectors.

The variance is given by

$$\begin{aligned} \sigma^2(E_f) &= \langle (E_f - \langle E_f \rangle)^2 \rangle \\ &= \langle E_f^2 \rangle - \langle E_f \rangle^2\,. \end{aligned} \tag{4.11}$$

We have defined

$$\langle E_f \rangle = \mathrm{Tr} f(\mathsf{H}) = \lim_{N_\phi \to \infty} \frac{1}{N_\phi} \sum_{i=1}^{N_\phi} \phi^{i\dagger} f(\mathsf{H}) \phi^i\,. \tag{4.12}$$

So we now turn our attention to the quantity $\langle E_f^2 \rangle$. This involves the expansion of quantities of the form

$$\langle \phi_i^\dagger \phi_j \phi_k^\dagger \phi_l \rangle \, . \tag{4.13}$$

So we start with this expansion, assuming that we are using Gaussian noise

$$\begin{aligned} \langle \phi_i^{a*} \phi_j^a \phi_k^{b*} \phi_l^b \rangle &= \langle \phi_i^{a*} \phi_j^a \rangle \langle \phi_k^{b*} \phi_l^b \rangle + \langle \phi_i^{a*} \phi_l^b \rangle \langle \phi_k^{b*} \phi_j^a \rangle \\ &= \delta_{ij} \delta_{kl} + \delta_{ab} \delta_{il} \delta_{kj} \, . \end{aligned} \tag{4.14}$$

So we can now calculate $\langle E_f^2 \rangle$ as required

$$\begin{aligned} \langle E_f^2 \rangle &= \tfrac{1}{N_\phi^2} \textstyle\sum_{a,b=1}^{N_\phi} \langle \phi_i^{a*} (f(\mathsf{H}))_{ij} \phi_j^a \phi_k^{b*} (f(\mathsf{H}))_{kl} \phi_l^b \rangle \\ &= \delta_{ij} \delta_{kl} (f(\mathsf{H}))_{ij} (f(\mathsf{H}))_{kl} + \tfrac{1}{N_\phi} \delta_{il} \delta_{kj} (f(\mathsf{H}))_{ij} (f(\mathsf{H}))_{kl} \\ &= (\mathrm{Tr} f(\mathsf{H}))^2 + \tfrac{1}{N_\phi} \mathrm{Tr}((f(\mathsf{H}))^2) \, . \end{aligned} \tag{4.15}$$

Consequently, we have

$$\sigma^2(E_f) = \frac{1}{N_\phi} \mathrm{Tr}((f(\mathsf{H}))^2) \, . \tag{4.16}$$

4.5 Monte Carlo

In order to perform a Metropolis accept-reject step (as shown in Chapter 1), we have to compare the actions of two consecutive configurations. Recall our acceptance is given by

$$\alpha_{UU'} = \min(1, e^{-\Delta S}) \, , \tag{4.17}$$

where

$$\Delta S = [S[U'] - S[U]] \, . \tag{4.18}$$

This means that we need to make unbiased estimates of the quantity $e^{-\Delta S}$, given the unbiased estimates of both $S[U']$ and $S[U]$, which are given from our Lánczos estimates.

4.5.1 Unbiased estimators

If we were simply to take estimates of $S[U']$ and $S[U]$, and then exponentiate the difference, the result would be biased. In order to create an unbiased estimate of $e^{-\Delta S}$, we can employ a method due to Bhanot and Kennedy [48] and further analysed in [49]. Basically the idea is to write the exponential as a Taylor expansion, where the Taylor coefficients are interpreted as probabilities.

So we start with an estimate of $S_f = \mathrm{Tr}\log(\mathsf{H})$ for both U and U', both evaluated with the same noise vector, ϕ^i. We then calculate their difference

$$\Delta S_i[UU'] = \phi^{i\dagger}(\log \mathsf{H}[U'] - \log \mathsf{H}[U])\phi^i \,. \tag{4.19}$$

We then do this for a set of noise vectors, such that we have a set of estimates $\{x_j\} = \{\Delta S_i[UU']\}$. So we now wish to calculate an unbiased estimate of $\langle e^x \rangle$. We write the Taylor expansion in the following manner

$$\langle e^x \rangle = 1 + x_1 \left(1 + \frac{1}{2} x_2 \left(1 + \frac{1}{3}(1 + \ldots)\right)\right) \,. \tag{4.20}$$

We then evaluate $\langle e^x \rangle$ by first setting $\langle e^x \rangle = 1 + x_1$, where x_1 is randomly chosen from $\{x_j\}$. We then add on $x_1 x_2$ with probability $\frac{1}{2}$, again x_2 is taken randomly from $\{x_j\}$. If $x_1 x_2$ is not chosen, we terminate the procedure. We then continue to add on $x_1 x_2 x_3$ with probability $\frac{1}{3}$, or alternatively the procedure is terminated. Again, x_3 is randomly chosen from $\{x_j\}$.

This whole procedure is continued until termination. We then repeat the procedure many times and take the average of our estimates for e^x. We repeat the procedure until the average value of e^x converges to some desired accuracy.

4.5.2 Random numbers

In numerical analysis, and particularly when using stochastic methods, we often deal with random numbers. Of course it is impossible to generate truly random numbers on a computer, and so we must appeal to algorithms which produce *pseudo-random numbers*. The basic idea is that the algorithm takes a *seed* number and produces a sequence of numbers which appear to be unconnected, but in actual fact are totally reproducible. We use pseudo-random number generators which take a user-defined seed number and then modify it depending on the gauge configuration in question. From this modified seed number, a whole set of pseudo-random random vectors are produced. At present these pseudo-random numbers are completely reproducible, provided the number of processors is kept constant. Versions of the algorithm, which are independent of the number of processors do exist, but are much slower in performance.

4.6 Making Tr $\log(\mathsf{H})$ Estimates

In chapter 3, we saw that the convergence of an estimate of the $\mathrm{Tr}\log(\mathsf{H})$ quantity was smooth for an estimate made using a single noise vector. In order to produce a stochastic estimate of $\mathrm{Tr}\log(\mathsf{H})$, we must make estimates using many noise vectors. The question then must be, how do such estimates vary as we go from one noise vector to the next. Fig. (4.1) illustrates what happens. We can measure the fluctuation over a set of, say, 100 noise vectors simply by measuring the standard deviation. For the full $\mathrm{Tr}\log(\mathsf{H})$. We measure this standard deviation to be ~ 30. In order to compare this fluctuation with the convergence of the normalised $\mathrm{Tr}\log(\mathsf{H})$ illustrated in Fig. (3.6), we measure the fluctuation of the normalised $\mathrm{Tr}\log(\mathsf{H})$ quantity and find this standard deviation to be $\sim 10^{-4}$.

Choosing the number of Lánczos sweeps to perform amounts effectively to choosing the level of convergence we require. We must ensure that for each noise vector, the $\mathrm{Tr}\log(\mathsf{H})$ quantity has converged relative to the fluctuation from noise to noise. If we choose a convergence of $\sim 10^{-6}$ relative to a fluctuation of $\sim 10^{-4}$, then we can see from Fig. (3.6) that we require roughly 200 Lánczos sweeps to achieve this when using $\mathsf{H} = \mathsf{M}^{\dagger}\mathsf{M}$ with red-black pre-conditioning applied. In the case $\mathsf{H} = \gamma_5 \mathsf{M}$, we see that 500 sweeps would

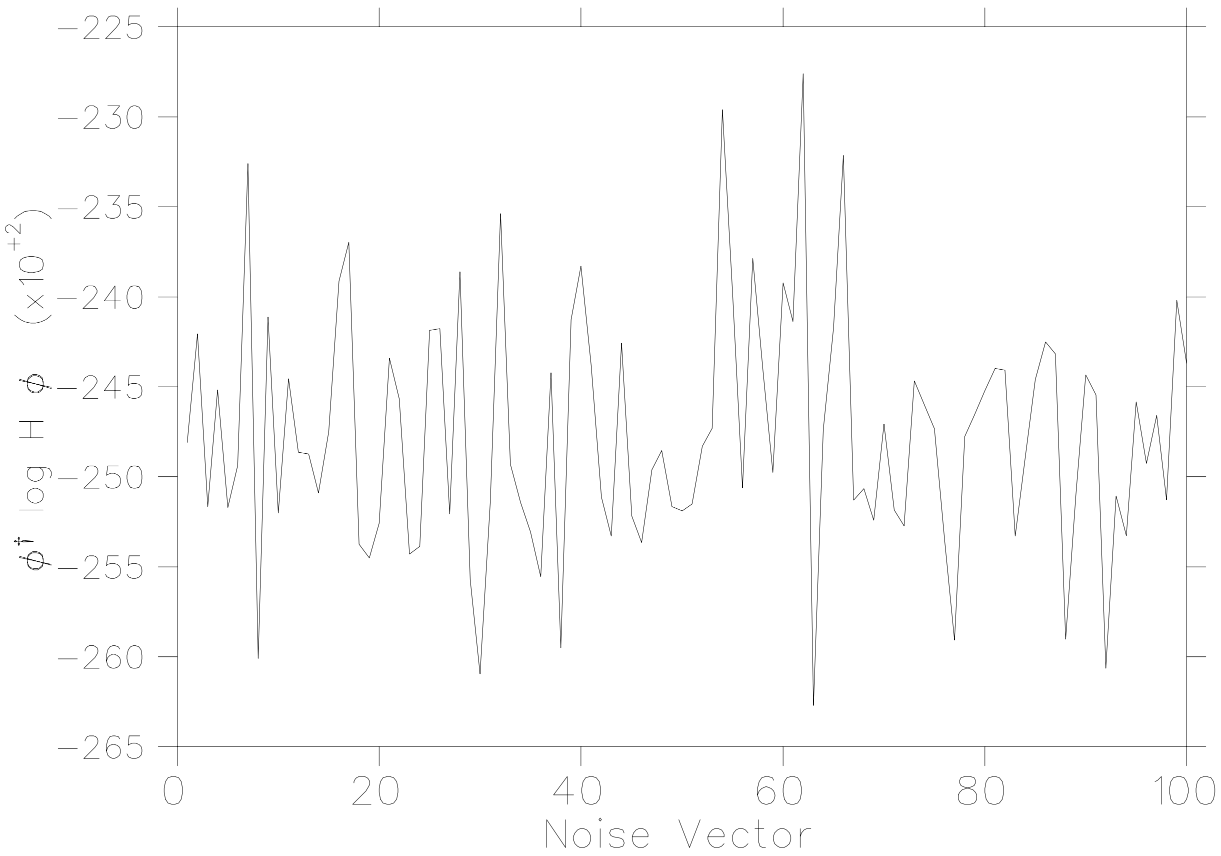


Figure 4.1: The fluctuation of $\phi^{i\dagger} \log \mathsf{H} \phi^i$ for 100 different noise vectors. $\mathsf{H} = \mathsf{M}^\dagger \mathsf{M}$ with red-black pre-conditioning applied.

be required.

4.7 Conclusion

So we now have a method of creating stochastic estimates of the $\mathrm{Tr}\log(\mathsf{H})$ quantity using Gaussian noise. It is necessary to use just 200 Lánczos sweeps when using $\mathsf{H} = \mathsf{M}^\dagger\mathsf{M}$ with red-black pre-conditioning applied. We also have a reliable way of estimating the errors in terms of the variances which occur when using Gaussian noise.

Chapter 5

Variance Reduction Methods

5.1 Introduction

In this chapter I describe two variance reduction techniques which, it turns out, can be used together to produce a significant improvement to the calculation of traces using the stochastic methods described in the previous chapters. This method requires relatively little extra work on top of that already done using the Lánczos BFG method.

The basic Lánczos BFG idea is to make a stochastic estimate, E_f, of the $\mathrm{Tr} f(\mathsf{H})$ quantity in the following way.

$$E_f = \frac{1}{N_\phi} \sum_{i=1}^{N_\phi} \sum_{k=1}^{N_{lanc}} \omega_k^i f(\theta_k^i) \,, \tag{5.1}$$

where N_ϕ is the number of noise vectors used in the stochastic estimate and N_{lanc} is the number of Lánczos sweeps used (ω_k and θ_k are the weights and abscissa as described earlier).

If using Gaussian noise vectors, we can write down an explicit formula for the variance, as described in the previous chapter

$$\sigma^2(E_f) = \frac{1}{N_\phi^2} \sum_{i=1}^{N_\phi} \sum_{k=1}^{m} \omega_k^i (f(\theta_k^i))^2 \,. \tag{5.2}$$

An obvious way of reducing this variance is to increase the number of noise vectors used, but this is expensive. A less expensive way is to employ some kind of variance reduction technique. The rest of the chapter describes such techniques. Unless otherwise stated, in this chapter $\mathsf{H} = \mathsf{M}^\dagger \mathsf{M}$ with red-black pre-conditioning applied.

5.2 Method 1 - Using Chebyshev Polynomials

First of all we consider ways of reducing the variance of the quantity estimated by the Lánczos BFG method. In calculating a quantity $\mathrm{Tr} f(\mathsf{H})$ using such a method, we arrive at a variance given by Eq. (5.2). Of course this variance is dependent on the function f in question. So in general when estimating a quantity $\mathrm{Tr} f'(\mathsf{H})$, where

$$f'(\theta) = f(\theta) - g(\theta)\,, \tag{5.3}$$

we will arrive at a new quantity with a different associated variance. Furthermore, if the quantity g is a good approximation to f, then it is also reasonable to suppose that the variance of this new quantity will be lower than the variance of the original quantity. So how can we choose such a quantity in a way which will be useful to us? The question really is how we would then go about reconstructing an estimate of the original value $\mathrm{Tr} f(\mathsf{H})$ from this new quantity $\mathrm{Tr} f'(\mathsf{H})$. One way we can do this is to add back on the quantity in terms of the matrix itself.

So, labelling our new estimate of $\mathrm{Tr} f(\mathsf{H})$ as E'_f, we have

$$E'_f = \frac{1}{N_\phi} \sum_{i=1}^{N_\phi} \sum_{k=1}^{N_{lanc}} \omega_k^i (f(\theta_k^i) - g(\theta_k^i)) + \frac{1}{N_\eta} \sum_{i=1}^{N_\eta} \eta^{i\dagger} g(\mathsf{H}) \eta^i\,, \tag{5.4}$$

where $\eta^\dagger g(\mathsf{H})\eta$ is a direct stochastic estimate of $\mathrm{Tr} g(\mathsf{H})$, included to reconstruct the estimate of $\mathrm{Tr} f(\mathsf{H})$ as required.

If we choose our function g carefully, then we will reduce the variance of the first 'Lánczos' quantity in Eq. (5.4), but what about the second term involving $g(\mathsf{H})$? This is also a stochastic term, and so will have an associated variance. We need then to find a form for the function g, which will enable us to reduce the overall variance of E'_f compared with the original estimate E_f. A candidate for such a function g needs to satisfy two criteria. Firstly, it must be a good approximation to f in the sense that the variance of the stochastic estimate of $\mathrm{Tr} f'$ is small. Secondly, we must be able to produce an estimate of the second stochastic term in Eq. (5.4) in such a way that this too has a small variance. When discussing variances as being 'small', we mean that we want the overall variance of E'_f to be significantly smaller than that of E_f.

So how do we go about finding a useful form for g? This is where Chebyshev polynomials come in. Chapter (2) gives a detailed description of these polynomials. The beauty of these well-studied polynomials is basically that they may be used to produce a good approximation of the quantity in question at a relatively low order. Consequently, these polynomials fulfill both of the criteria for our function g. Firstly they can easily be manipulated to give a good approximation to f. Secondly, the fact that they provide a good approximation at low-order means that we can make a good stochastic estimate of the second term in Eq. (5.4) *i. e.* we can afford to make N_η quite large.

So now we have our new estimate E'_f given by

$$E'_f = L + C \,, \tag{5.5}$$

where

$$L = \frac{1}{N_\phi} \sum_{i=1}^{N_\phi} \sum_{k=1}^{N_{lanc}} \omega_k^i (f(\theta_k^i) - C^n(\theta_k^i)) \,, \qquad C = \frac{1}{N_\eta} \sum_{i=1}^{N_\eta} \eta^{i\dagger} C^n(\mathsf{H}) \eta^i \,. \tag{5.6}$$

$C^n(\theta)$ is a Chebyshev approximation of $f(\theta)$ created with linear combinations of Chebyshev polynomials; n is the (as yet undetermined) order of this Chebyshev approximation. $C^n(\mathsf{H})$ is this same Chebyshev approximation written as a polynomial in the matrix H.

So now we need to quantify this theory to see how it would work in practice. First of all we define a ratio r, such that

$$r = \frac{N_\eta}{N_\phi}\,. \tag{5.7}$$

The next section gives the outline of a model which can be used to calculate an optimal choice for the various parameters discussed above.

5.2.1 Variance reduction model

We want to make estimates of the quantity $\mathrm{Tr}\log(\mathsf{H})$, where H is some Hermitian representation of the Fermion matrix. So we now set $f(\mathsf{H}) = \log(\mathsf{H})$.

So we now have two quantities to deal with; the quantities we have just defined as 'L' and 'C'. Both of these stochastic quantities, separately have a variance associated with them and both require different amounts of work[1] to attain a value with a given variance. So now we set up a model which can be used to find the optimal parameters to use in practice. Our goal is to achieve the lowest overall variance with the minimum of overall work.

We start with the quantity L. For a given configuration, we can plot the variance of this quantity as a function of n, the order of the Chebyshev polynomial. The easiest way of doing this is just to measure the variance of L for different values of n and then plot these values. Fig. (5.1), more specifically the curve annotated 'no traceless operators', shows such a plot.

An analysis of this log-plot enables us to write down an expression for the variance σ_L^2 in terms of n

$$\sigma_L^2 = \frac{935530n^{-2.68}}{N_\phi}\,. \tag{5.8}$$

It is fairly straightforward to write down the amount of work W_L required in such an estimate. This is independent of n as the Chebyshev polynomial is calculated on the Ritz

[1] By 'work' we mean the amount of computer time in terms of the number of 'matrix-vector' calculations.

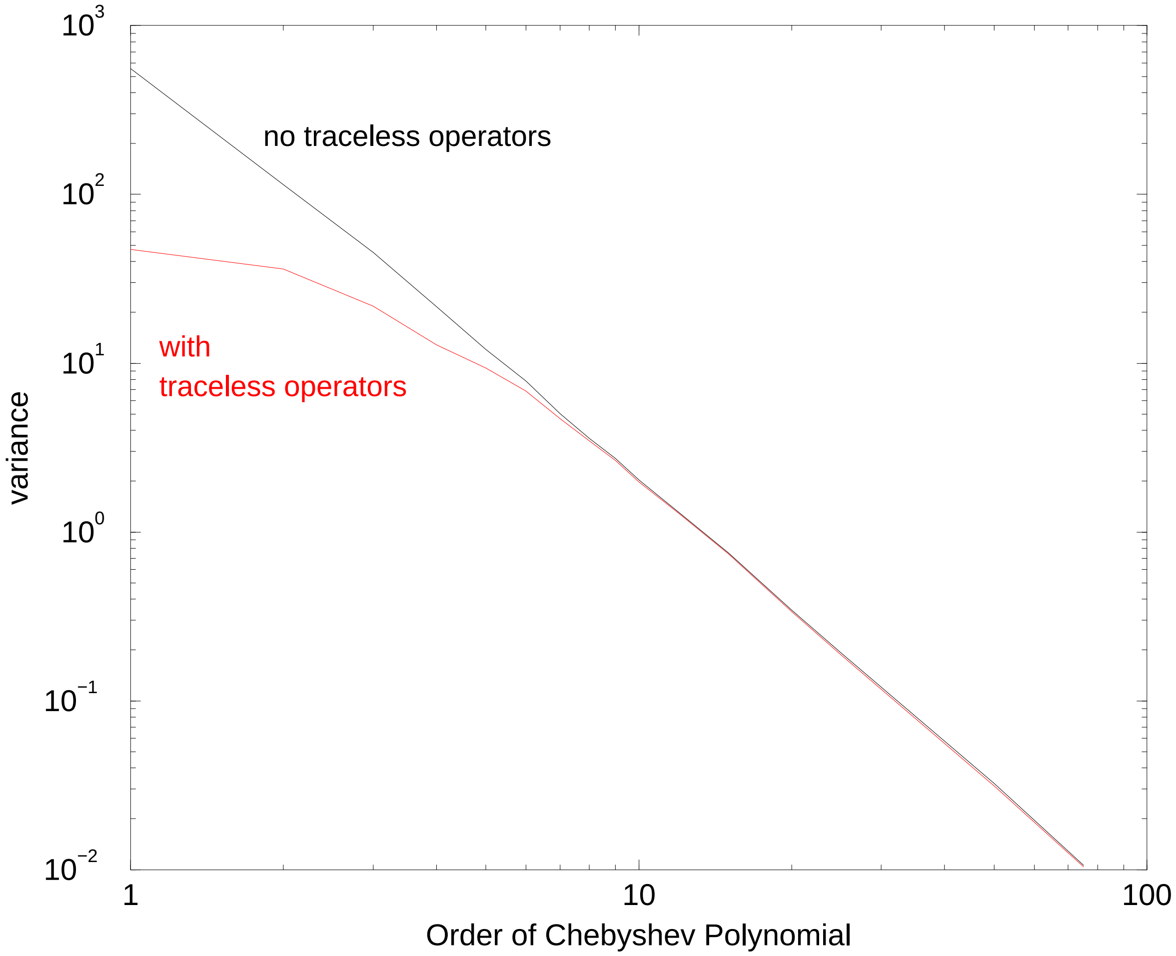


Figure 5.1: The variance of the Lánczos quantity (L) as a function of the order of Chebyshev. This plot was created using $N_\phi = 100$.

values, after any matrix-vector multiplications have been done *i. e.* it is 'off-line' work done by a single processor and does not take part in the calculations done in parallel. Such 'off-line' calculations are considered of insignificant cost, which we usually think of in terms of processor-hours.

Here we define one unit of work to be the amount of work required to do one matrix-vector operation $\mathsf{H}\Psi$, where, in this case, $\mathsf{H} = \mathsf{M}^\dagger\mathsf{M}$ (red-black pre-conditioned). This corresponds to 4 single-parity $\mathsf{D}_{oe}\Psi_e$ operations. Sometimes this one unit of work will be referred to as '1 Krylov Sweep'.

So we write

$$W_L = N_\phi N_{lanc}\,, \tag{5.9}$$

where N_ϕ is the number of noise vectors used in the Lánczos procedure and N_{lanc} is the number of Lánczos sweeps.

So we now turn our attention to the quantity C. Again, we can plot variance in terms of the order of Chebyshev polynomial. As before we do this by creating several estimates of the variable C for several values of n. Fig. (5.2) shows such a plot, again the curve annotated 'no traceless operators' is the relevant one. We can see that after $\sim n = 8$ this variance is roughly constant. So we can write

$$\sigma_C^2 = \frac{400000}{N_\eta}\,, \qquad n > 8\,. \tag{5.10}$$

As with the quantity C, we can write down the amount of work in terms of n. This is simply a product of the number of noisy estimates, N_η, used to estimate C and the order of Chebyshev polynomial, n, used for each estimate

$$W_C = N_\eta n\,. \tag{5.11}$$

We can now write expressions for the total work W and total variance σ^2 of our new

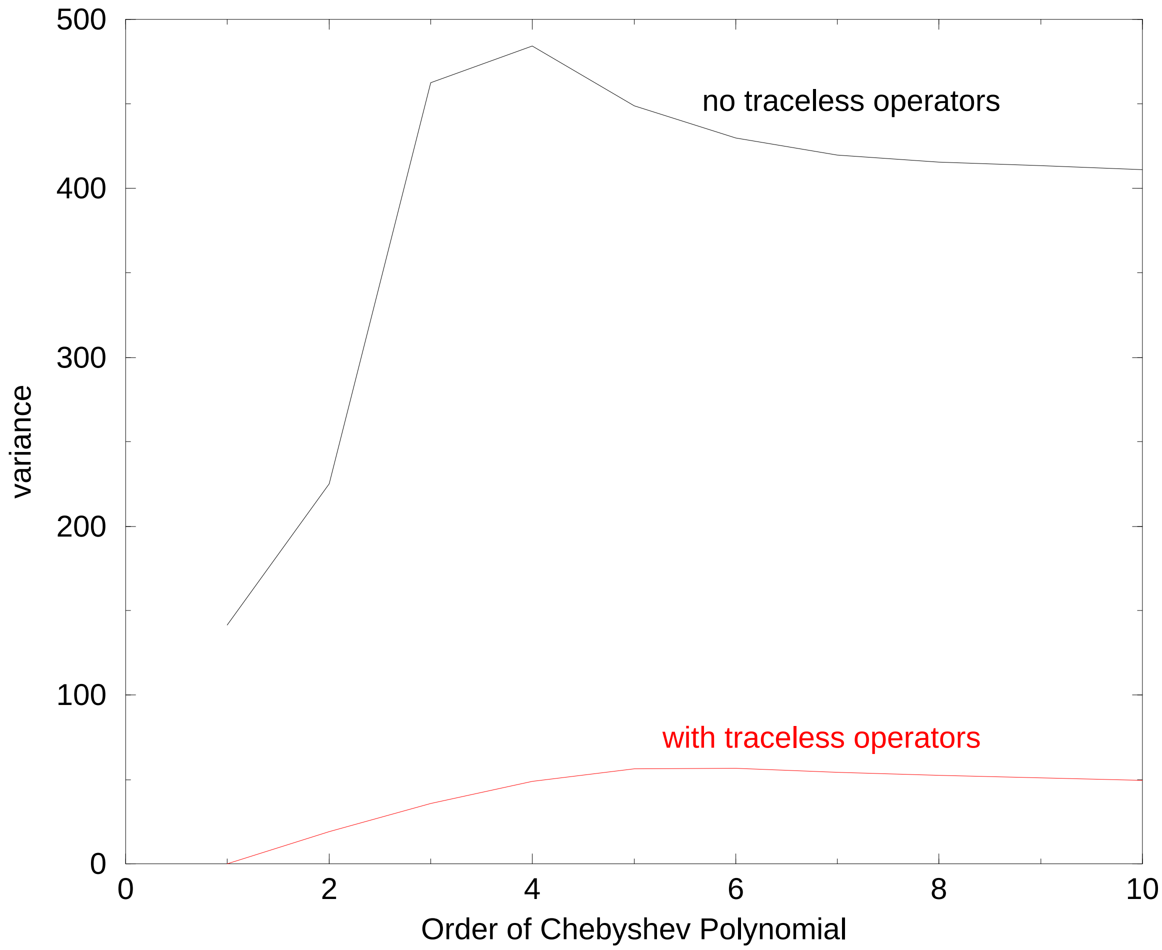


Figure 5.2: The variance of the quantity (C) as a function of the order of Chebyshev. This plot was created using $N_\eta = 1000$.

quantity E'_f

$$\begin{aligned} W &= W_L + W_C \\ \sigma^2 &= \sigma_L^2 + \sigma_C^2 \,. \end{aligned} \tag{5.12}$$

In order to find optimal values for the various parameters which appear in the work and variance expressions above, we will start by fixing certain quantities. Later we can see the effect of varying these quantities also.

If we fix N_{lanc} and N_ϕ, we can write the expressions in Eq. (5.12) in such a way that the only unknown parameters we are left with are n and r and we have two constraints to choose from; either we fix the amount of work and then find optimum values of n and r in terms of a minimum variance, or alternatively we fix the variance and minimise the amount of work.

For example, for convergence reasons we have already used $N_{lanc} = 200$. In order to give us a reasonable variance for our Lánczos estimates (compared with the fluctuation from gauge to gauge), we have already set $N_\phi = 100$. This means that we have already performed 20,000 matrix-vector multiplications, *i. e.* $W_L = 20,000$. Of course in principal both of the parameters N_ϕ and N_{lanc} can be varied, and we look at these situations later.

If, for example, we decided that we could afford ourselves enough computer time to do another equal amount of work on top of that already done by the Lánczos part of the calculation, then we are effectively fixing the total amount of work, W_0 to $W_0 = 40,000$. We can therefore write the corresponding value $r = r_0$ in terms of n

$$r_0 = \frac{120}{n} \,. \tag{5.13}$$

We can then substitute this value of $r = r_0$ into the variance expression in Eq. (5.12) and then find the optimal value of n_0 *i. e.* by finding a value for $n = n_0$ which gives this variance a minimum. We can then substitute this value, n_0 back into Eq. (5.13) to find

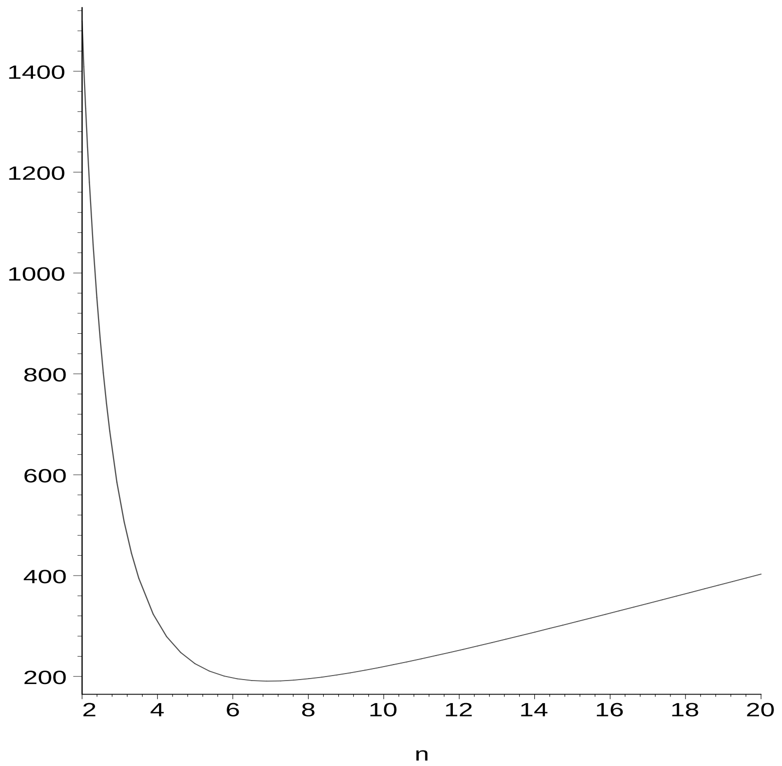


Figure 5.3: The variance for a fixed amount of work as a function of n.

r_0.

Fig. (5.3) shows the variance as a function of n for a fixed amount of work. This minimisation procedure gives $(n_0, r_0) \approx (7, 28.9)$ and achieves a variance of $\sigma^2 = 190.8$, compared with an original variance of 4071.6. So our new variance is a factor of ~ 21.3 smaller than the original. *i. e.* the error is a factor of ~ 4.6 smaller.

Similarly, we could decide upon a fixed variance. Say for example we wanted to achieve a factor of 10 reduction in the variance, what would be the optimal values of (n, r) to use? Again, we refer to these optimal values as (n_0, r_0).

Fig. (5.4) shows the amount of work as a function of n. Here we find $(n_0, r_0) \approx (5, 13.5)$. The total work required amounts to ~ 27000, a factor of ~ 1.35 times the work we have already done on just the Lánczos part of the calculation alone.

We can of course go on and use this procedure to work out optimal parameters by also

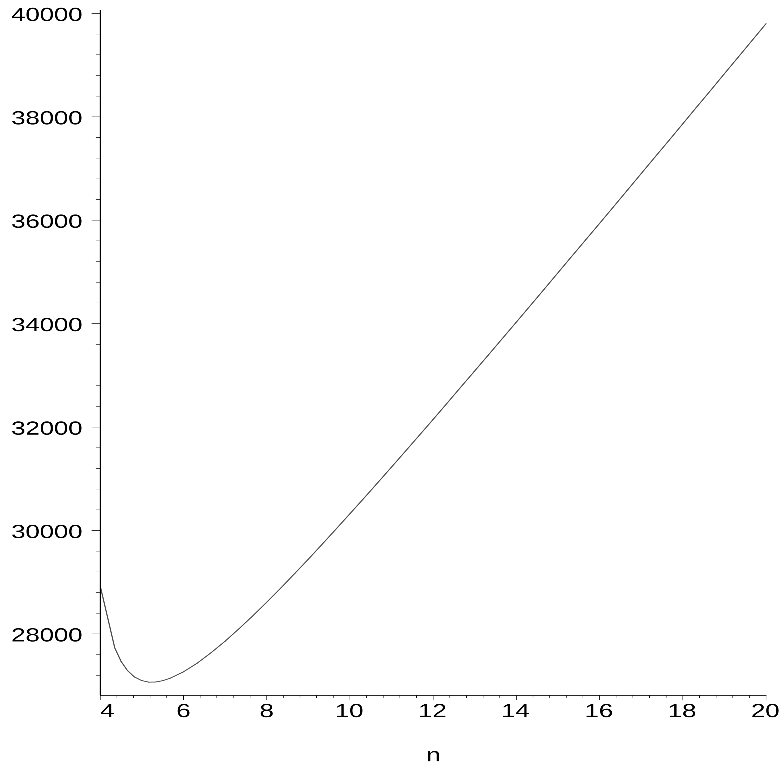


Figure 5.4: The work for a fixed amount of variance as a function of n.

varying N_ϕ *i. e.* by varying the amount of work done on the Lánczos part of the calculation. Fig. (5.5) shows the work as a function of N_ϕ for various different fixed variances.

So we now have a model which can be used to predict the overall variance of an estimate, given a set of input parameters. It appears that within the scale of the amount of work we have afforded ourselves *i. e.*$\sim 40,000$ matrix-vector calculations, we can achieve an overall reduction of ~ 20.

We now illustrate some results to check that this model works in practice.

5.2.2 Results

Table 5.1 shows the resulting reduction in the error using $(n, r) = (14, 12.5)$ (values which we will later find to be useful). These results were created using the $(\beta, \kappa, c_{\text{sw}}) = (5.2, 0.1355, 2.0171)$ [26] configurations.

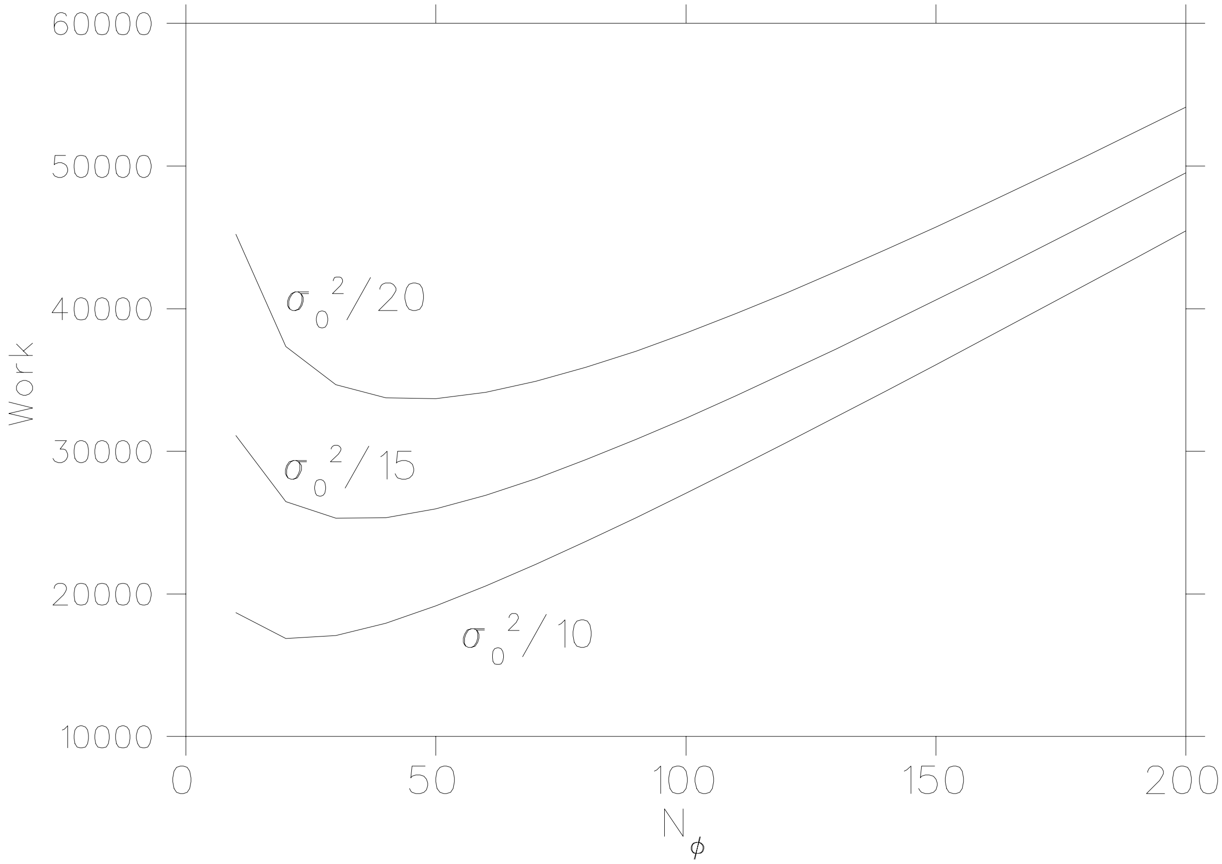


Figure 5.5: The work as a function of N_ϕ for various fixed variances. The variances shown are multiples of $\sigma_0^2 = 4071.6$, the variance resulting from a calculation involving just Lánczos with $N_\phi = 100$ and no Chebyshev subtraction.

The above model predicts that for the test configuration the variance would be reduced as follows

$$\begin{aligned} \sigma^2 : &\quad 4071.6 \quad \rightarrow \quad 327.9 \\ \sigma : &\quad 63.8 \quad \rightarrow \quad 18.1\,. \end{aligned} \tag{5.14}$$

We can see that the results fit this model.

Table 5.1: The $\mathrm{Tr}\log(\mathsf{M}_{ee}^{\dagger}\mathsf{M}_{ee})$ quantity for the first 10 configurations. Here we see the effect of variance reduction using method 1 alone. $n = 14$, $N_{\phi} = 100$ and $r = 12.5$.

Configuration	$\mathrm{Tr}\log(\mathsf{M}_{ee}^{\dagger}\mathsf{M}_{ee})$ Lánczos 100 noise	$\mathrm{Tr}\log(\mathsf{M}_{ee}^{\dagger}\mathsf{M}_{ee})$ with variance reduction
16400	-24645.19 $\pm$ 64.2	-24546.80 $\pm$ 18.62
16440	-24467.70 $\pm$ 64.2	-24396.95 $\pm$ 18.05
16480	-24624.87 $\pm$ 64.3	-24671.15 $\pm$ 18.34
16520	-24640.77 $\pm$ 64.2	-24513.01 $\pm$ 18.97
16560	-24504.47 $\pm$ 64.2	-24540.12 $\pm$ 18.22
16600	-24801.28 $\pm$ 64.3	-24766.38 $\pm$ 18.83
16640	-24611.95 $\pm$ 64.2	-24475.89 $\pm$ 18.88
16680	-24648.44 $\pm$ 64.3	-24622.16 $\pm$ 19.05
16720	-24556.10 $\pm$ 64.2	-24559.54 $\pm$ 18.49
16760	-24947.00 $\pm$ 64.4	-24849.35 $\pm$ 18.46

5.2.3 Summary of method 1

We can see that within the limits of work which we have allowed ourselves, we are able to gain roughly a factor of 20 in variance reduction using method 1. We will now go on to describe a second variance reduction method.

5.3 Method 2 - Traceless Operators

5.3.1 Introduction

Here we introduce a second variance reduction method, which attempts to subtract off unwanted noise from a stochastic estimate directly. Say we were to make a stochastic estimate of a lattice trace quantity $\mathrm{Tr} f(\mathsf{H})$. For small values of κ, we can perform a hopping parameter expansion[2] of the quantity, $\mathrm{Tr} f(\mathsf{H})$, and we can see that a number terms appear which are either traceless, or have traces which we can calculate explicitly in terms of Wilson loops. It is obvious then that their inclusion into the estimate of $\mathrm{Tr} f(\mathsf{H})$ is introducing unnecessary stochastic noise into the calculation which we can remove. This idea is described in [50] and our application of this method is described here.

Essentially, instead of making an estimate of a quantity $\mathrm{Tr} f(\mathsf{H})$ using purely Lánczos, as in Eq. (5.1), we make a new estimate

$$E'_f = \frac{1}{N_\phi} \sum_{i=1}^{N_\phi} \sum_{k=1}^{N_{lanc}} \omega_k^i f(\theta_k^i) - \frac{1}{N_\phi} \sum_{i=1}^{N_\phi} \sum_{q=1}^{N_q} \phi^{i\dagger} \lambda_q \mathsf{Q}_q \phi^i \tag{5.15}$$

where the Q_q are effectively traceless operators, whose inclusion should not contribute to the overall estimate except in such a way as to reduce the variance. The λ_q are a set of, as yet undetermined, coefficients.

So how do we go about deciding which coefficients to use? The next section illustrates the kind of terms which appear in the hopping parameter expansion. We can then determine which may be useful as 'traceless operators'.

5.3.2 Analysing the hopping parameter expansion

First we consider the Hermitian representation of the fermion determinant $\mathsf{M}^\dagger \mathsf{M}$,

$$\mathsf{M}^\dagger \mathsf{M} = (\mathsf{A} - \kappa^2 \mathsf{D}^\dagger \mathsf{A}^{-1} \mathsf{D}^\dagger)(\mathsf{A} - \kappa^2 \mathsf{D} \mathsf{A}^{-1} \mathsf{D}) \,. \tag{5.16}$$

[2] An expansion of the quantity in κ.

If we were to perform a hopping parameter expansion of a quantity involving a second order polynomial in $\mathsf{M}^\dagger\mathsf{M}$, then, up to order $\mathcal{O}(\kappa^3)$ we have something of the form

$$\begin{aligned}
X(\mathsf{M}^\dagger\mathsf{M}) \approx\ & a_0\mathsf{I} + a_1(\mathsf{M}^\dagger\mathsf{M}) + a_2(\mathsf{M}^\dagger\mathsf{M})^2 \\
\approx\ & a_0\mathsf{I} + a_1\mathsf{A}^2 + a_2\mathsf{A}^4 \\
& -\kappa^2(a_1\mathsf{D}^\dagger\mathsf{A}^{-1}\mathsf{D}^\dagger\mathsf{A} + a_1\mathsf{A}\mathsf{D}\mathsf{A}^{-1}\mathsf{D} + a_2\mathsf{A}^2\mathsf{D}^\dagger\mathsf{A}^{-1}\mathsf{D}^\dagger\mathsf{A} \\
& \quad +a_2\mathsf{A}^3\mathsf{D}\mathsf{A}^{-1}\mathsf{D} + a_2\mathsf{D}^\dagger\mathsf{A}^{-1}\mathsf{D}^\dagger\mathsf{A}^3 + a_2\mathsf{A}\mathsf{D}\mathsf{A}^{-1}\mathsf{D}\mathsf{A}^2) \\
& +\kappa^4(a_1\mathsf{D}^\dagger\mathsf{A}^{-1}\mathsf{D}^\dagger\mathsf{D}\mathsf{A}^{-1}\mathsf{D} + a_2\mathsf{A}^2\mathsf{D}^\dagger\mathsf{A}^{-1}\mathsf{D}^\dagger\mathsf{D}\mathsf{A}^{-1}\mathsf{D} \\
& \quad +a_2\mathsf{D}^\dagger\mathsf{A}^{-1}\mathsf{D}^\dagger\mathsf{A}\mathsf{D}^\dagger\mathsf{A}^{-1}\mathsf{D}^\dagger\mathsf{A} + a_2\mathsf{D}^\dagger\mathsf{A}^{-1}\mathsf{D}^\dagger\mathsf{A}^2\mathsf{D}\mathsf{A}^{-1}\mathsf{D} \\
& \quad +a_2\mathsf{A}\mathsf{D}\mathsf{A}^{-1}\mathsf{D}\mathsf{D}^\dagger\mathsf{A}^{-1}\mathsf{D}^\dagger\mathsf{A} + a_2\mathsf{A}\mathsf{D}\mathsf{A}^{-1}\mathsf{D}\mathsf{A}\mathsf{D}\mathsf{A}^{-1}\mathsf{D} \\
& \quad +a_2\mathsf{A}^2\mathsf{D}^\dagger\mathsf{A}^{-1}\mathsf{D}^\dagger\mathsf{D}\mathsf{A}^{-1}\mathsf{D}\mathsf{A}^2) \\
& +\mathcal{O}(\kappa^6)\,.
\end{aligned} \tag{5.17}$$

Remembering that the clover term $\mathsf{A} = \mathsf{I} + \kappa\mathsf{S}$, where $\mathsf{S} = \frac{1}{2}c_{\rm sw}\sigma_{\mu\nu}F_{\mu\nu}\mathsf{I}$, we can write

$$\begin{aligned}
\kappa^2\mathsf{A}(\kappa)\mathsf{D}\mathsf{A}^{-1}(\kappa)\mathsf{D} \approx\ & \kappa^2\mathsf{D}\mathsf{A}^{-1}\mathsf{D} + \kappa^3\mathsf{S}\mathsf{D}\mathsf{A}^{-1}\mathsf{D} \\
\approx\ & \kappa^2\mathsf{D}\mathsf{A}^{-1}\mathsf{D} + \kappa^3\mathsf{S}\mathsf{D}\mathsf{D} + \mathcal{O}(\kappa^4) \\
\approx\ & \kappa^2(\mathsf{D}\mathsf{A}^{-1}\mathsf{D} + \mathsf{A}\mathsf{D}\mathsf{D} - \mathsf{D}\mathsf{D}) + \mathcal{O}(\kappa^4)\,,
\end{aligned} \tag{5.18}$$

where we have used a Taylor expansion for one of the A^{-1} terms.

When it comes to considering operators to include in our subtraction scheme, this further analysis of the hopping parameter expansion prompts us to consider not just the operators which appear in a basic hopping parameter expansion, as given in Eq. (5.17), but also many other simple operators which involve $\mathsf{A}, \mathsf{A}^{-1}, \mathsf{D}$ and $\mathsf{D}^\dagger$.

So we now have a number of operators which may be useful to us. We now need to classify them in terms of their traces. These operators fall into three categories:

- Those which are explicitly traceless, *i. e.* D^n where n is odd; also operators such as D^2, $\mathsf{A}\mathsf{D}$, etc.

- Those which are not explicitly traceless, but where the trace may be calculable, *e. g.* A, A^2, A^3, A^4, A^5, $\mathsf{D}^\dagger\mathsf{D}$, D^4, etc. See Appendix B and C.

- Those which are complicated and have a trace which is not easy to write down[3] *e. g.* $\mathsf{A}\mathsf{D}^\dagger\mathsf{A}^{-1}\mathsf{D}^\dagger\mathsf{D}\mathsf{A}^{-1}\mathsf{D}$.

The inclusion of operators belonging to the second category depends upon how much work is involved in calculating their corresponding traces. In some cases these traces are calculated easily, see appendices B and C. A summary is given here

$$\begin{aligned}
\mathsf{A} &= V\,, \\
\mathsf{D}^\dagger\mathsf{D} &= 192V\,, \\
\mathsf{D}\mathsf{D}\mathsf{D}^\dagger\mathsf{D}^\dagger &= 12V(256 + 96\langle P\rangle)\,, \\
\mathsf{D}\mathsf{D}^\dagger\mathsf{D}\mathsf{D}^\dagger &= 12V(448 - 96\langle P\rangle)\,, \\
\mathsf{D}^4 &= 12V(-96\langle P\rangle)\,.
\end{aligned} \tag{5.19}$$

where $\langle P\rangle = \frac{1}{3}\sum_\Box \Re\mathrm{Tr}U_\Box$ (the *plaquette*) and V is the volume of the lattice. These terms can be found by evaluating traces of products of $1 \pm \gamma_\mu$ projection operators and gauge links, U, arising in the Wilson action. Unfortunately, for higher order terms, or more complex terms involving the clover term, A, the calculation of these traces is not so easy and consequently the cost of including such terms may outweigh the benefit in terms of noise reduction.

So we now need to look at these terms to determine which may be of help to us in our variance reduction scheme.

5.3.3 Determining the optimum value of the λ_q

Having at least an idea of which traceless operators Q_q to include in the subtraction scheme, we now need to determine a set of coefficients λ_q to go with them. This section

[3] Of course it may be possible to calculate such terms stochastically, if their convergence properties are good enough.

discusses some of the ways we may go about this.

We wish to minimise the variance (with respect to noise) of the following basic quantity

$$\frac{1}{N_\phi}\sum_{i=1}^{N_\phi}\phi^{i\dagger}(f(\mathsf{H})-\sum_p\lambda_p Q_p)\phi^i\,, \tag{5.20}$$

where H is a Hermitian matrix, Q_p is a set of p traceless operators , the ϕ^i's are randomly generated Gaussian noise vectors defined earlier and λ_p is a set of, as yet undefined, coefficients.

Firstly, however, we must decide exactly what it is we are trying to minimise. We wish to minimise the variance with respect to noise, but we have a choice as to the definition of the variance, *i. e.* we have some freedom when chosing between Eq. (4.2) or Eq. (4.3), depending upon whether we need to calculate the whole quantity, E'_T, or whether we wish to calculate the real and imaginary parts separately.

We know that

$$\Im \mathrm{Tr} f(\mathsf{H})=0\,, \tag{5.21}$$

as H is Hermitian, therefore we are interested in an accurate calculation of just the real quantity $\mathrm{Tr}f(\mathsf{H})$ with the subtraction of real traceless operators. Therefore the quantity in question is

$$\frac{1}{N_\phi}\sum_{i=1}^{N_\phi}\left(\Re(\phi^{i\dagger}f(\mathsf{H})\phi^i)-\Re(\phi^{i\dagger}\sum_p\lambda_p Q_p\phi^i)\right)\,, \tag{5.22}$$

and we will define its variance using Eq. (4.2), giving us

$$\sigma^2\propto\sum_{i=1}^{N_\phi}\left(\Re(\phi^{i\dagger}f(\mathsf{H})\phi^i)-\sum_p\Re(\phi^{i\dagger}\lambda_p Q_p\phi^i)\right)^2-\frac{1}{N_\phi}\{\sum_{i=1}^{N_\phi}\Re(\phi^{i\dagger}(f(\mathsf{H})-\sum_p\lambda_p Q_p)\phi^i)\}^2\,. \tag{5.23}$$

In order to determine a set of coefficients, λ_p, which minimise the variance due to noise, we differentiate Eq. (5.23) with respect to λ_p and then set the result equal to zero.

Assuming $(\frac{1}{N_\phi}\sum_{i=1}^{N_\phi}\Re\phi^{i\dagger}Q_p\phi^i)^2 \ll \sum_{i=1}^{N_\phi}(\Re(\phi^{i\dagger}Q_p\phi^i))^2$, we obtain

$$\sum_p\sum_{i=1}^{N_\phi}\Re(\phi^{i\dagger}f(\mathsf{H})\phi^i)\Re(\phi^{i\dagger}Q_p\phi^i)-\frac{1}{N_\phi}\sum_p\sum_{i=1}^{N_\phi}\Re(\phi^{i\dagger}f(\mathsf{H})\phi^i)\sum_{i=1}^{N_\phi}\Re(\phi^{i\dagger}Q_p\phi^i)=\sum_p\lambda_p\sum_{i=1}^{N_\phi}\Re(\phi_i^\dagger Q_p\phi^i)^2\,. \tag{5.24}$$

This then leads us to an optimisation procedure for the λ's.

Optimisation 1a

$$\begin{aligned}\Lambda &= (\lambda_1,\lambda_2,...,\lambda_p)^T\,,\\ A_{pq} &= \textstyle\sum_{i=1}^{N_\phi}\Re(\phi^{i\dagger}Q_p\phi^i)\Re(\phi^{i\dagger}Q_q\phi^i)\,,\\ V_q &= \textstyle\sum_{i=1}^{N_\phi}\Re(\phi^{i\dagger}f(\mathsf{H})\phi^i)\Re(\phi^{i\dagger}Q_p\phi^i)-\frac{1}{N_\phi}\sum_p\sum_{i=1}^{N_\phi}\Re(\phi^{i\dagger}f(\mathsf{H})\phi^i)\sum_{i=1}^{N_\phi}\Re(\phi^{i\dagger}Q_p\phi^i)\,,\\ \Lambda &= A^{-1}V\,.\end{aligned} \tag{5.25}$$

Note that ignoring the second term in Eq. (5.24) leads to the same optimisation procedure as in [50], *i. e.*

Optimisation 1b

$$\begin{aligned}\Lambda &= (\lambda_1,\lambda_2,...,\lambda_p)^T\,,\\ A_{pq} &= \textstyle\sum_{i=1}^{N_\phi}\Re(\phi^{i\dagger}Q_p\phi^i)\Re(\phi^{i\dagger}Q_q\phi^i)\,,\\ V_q &= \textstyle\sum_{i=1}^{N_\phi}\Re(\phi^{i\dagger}f(\mathsf{H})\phi^i)\Re(\phi^{i\dagger}Q_p\phi^i)\,,\\ \Lambda &= A^{-1}V\,.\end{aligned} \tag{5.26}$$

This is valid if the term $\sum_{i=1}^{N_\phi}\Re(\phi^{i\dagger}Q_p\phi^i)$ is very small[4].

It is possible to simplify the procedure further by condensing the two terms of V_q using the properties of Gaussian noise defined earlier.

[4]In the Gaussian case, using $\sim$ 1000 noise vectors, we find $\sum_{i=1}^{N_\phi}\Re(\phi^{i\dagger}Q_p\phi^i)\approx 10$ which makes this term too large to ignore.

First consider the following:

$$\Re((\phi^{i\dagger} f(\mathsf{H})\phi^i)^* \phi^{i\dagger} Q_p \phi^i) = \Re(\phi^{i\dagger} f(\mathsf{H})\phi^i)\Re(\phi^{i\dagger} Q_p \phi^i) + \Im(\phi^{i\dagger} f(\mathsf{H})\phi^i)\Im(\phi^{i\dagger} Q_p \phi^i)\,. \quad (5.27)$$

Due to the realness of the quantity $\mathrm{Tr} f(\mathsf{H})$, we note that $\Im(\phi^{i\dagger} f(\mathsf{H})\phi^i) = 0$, which gives

$$\Re((\phi^{i\dagger} f(\mathsf{H})\phi^i)^* \phi^{i\dagger} Q_p \phi^i) = \Re(\phi^{i\dagger} f(\mathsf{H})\phi^i)\Re(\phi^{i\dagger} Q_p \phi^i)\,. \quad (5.28)$$

Using what we know about expectations over Gaussian noise (see section (4.4)), we can say the following

$$\langle \phi^\dagger f(\mathsf{H})\phi\rangle_\phi = \lim_{N_\phi\to\infty} \frac{1}{N_\phi}\sum_{i=1}^{N_\phi} \Re(\phi^{i\dagger} f(\mathsf{H})\phi^i) = \mathrm{Tr}(f(\mathsf{H}))\,, \quad (5.29)$$

$$\Re\langle \phi^\dagger Q\phi\rangle_\phi = \lim_{N_\phi\to\infty} \frac{1}{N_\phi}\sum_{i=1}^{N_\phi} \Re(\phi^{i\dagger} Q\phi^i) = \Re\mathrm{Tr}(Q)\,. \quad (5.30)$$

We can also say

$$\Re\langle \phi^\dagger f(\mathsf{H})^\dagger Q\phi\rangle_\phi = \lim_{N_\phi\to\infty} \frac{1}{N_\phi}\sum_{i=1}^{N_\phi} \Re(\phi^{i\dagger} f(\mathsf{H})^\dagger Q\phi^i) = \Re\mathrm{Tr}(f(\mathsf{H})^\dagger Q)\,. \quad (5.31)$$

This means that in the limit $n\to\infty$, Eq. (5.24) becomes

$$\sum_p \sum_{i=1}^{N_\phi} \Re(\phi^{i\dagger} f(\mathsf{H})\phi^i)\Re(\phi^{i\dagger} Q_p\phi^i) - N_\phi \sum_p \mathrm{Tr}(f(\mathsf{H}))\Re\mathrm{Tr}(Q_p) = \sum_p \lambda_p \sum_{i=1}^{N_\phi} \Re(\phi^{i\dagger} Q_p \phi^i)^2\,. \quad (5.32)$$

We can use our rules concerning expectations over Gaussian noises, along with Eq. (5.28), to expand the first term in Eq. (5.32)

$$\begin{aligned}
\sum_p \sum_{i=1}^{N_\phi} \Re(\phi^{i\dagger} f(\mathsf{H})\phi^i)\Re(\phi^{i\dagger} Q_p\phi^i) &= \sum_p \Re \sum_{i=1}^{N_\phi} (\phi^{i\dagger} f(\mathsf{H})\phi^i)^* \phi^{i\dagger} Q_p \phi^i \\
&\approx N_\phi \sum_p \Re\mathrm{Tr}(f(\mathsf{H})^\dagger Q_p) + N_\phi \sum_p \mathrm{Tr}(f(\mathsf{H})^\dagger)\Re\mathrm{Tr}(Q_p) \\
&\approx N_\phi \sum_p \Re\mathrm{Tr}(f(\mathsf{H})^\dagger Q_p) + N_\phi \sum_p \mathrm{Tr}(f(\mathsf{H}))\Re\mathrm{Tr}(Q_p)\,.
\end{aligned} \quad (5.33)$$

So in the limit $n\to\infty$ Eq. (5.32) LHS now becomes

$$\sum_p \sum_{i=1}^{N_\phi} \Re(\phi^{i\dagger} f(\mathsf{H})\phi^i)\Re(\phi^{i\dagger} Q_p\phi^i) - N_\phi \sum_p \mathrm{Tr}(f(\mathsf{H}))\Re\mathrm{Tr}(Q_p) = N_\phi \sum_p \Re\mathrm{Tr}(f(\mathsf{H})^\dagger Q_p)\,. \quad (5.34)$$

Which leads us to an alternative optimisation procedure for the λ's.

Optimisation 2

$$\begin{aligned}
\Lambda &= (\lambda_1, \lambda_2, ..., \lambda_p)^T \,, \\
A_{pq} &= \textstyle\sum_{i=1}^{N_\phi} \Re(\phi^{i\dagger} Q_p \phi^i) \Re(\phi^{i\dagger} Q_q \phi^i) \,, \\
V_q &= \textstyle\sum_{i=1}^{N_\phi} (\Re\phi^{i\dagger} f(\mathsf{H})^\dagger Q_q^\dagger \phi^i) \,, \\
\Lambda &= A^{-1} V \,.
\end{aligned} \tag{5.35}$$

Alternatively, one can chose to minimise the entire variance over both real and imaginary parts, *i. e.* the variance of the form Eq. (4.3).

$$\sigma^2 \propto \sum_p \sum_{i=1}^{N_\phi} \phi^{i\dagger} (f(\mathsf{H}) - \lambda_p Q_p)^\dagger (f(\mathsf{H}) - \lambda_p Q_p) \phi^i \,. \tag{5.36}$$

This soon leads us to a third optimisation procedure.

Optimisation 3

$$\begin{aligned}
\Lambda &= (\lambda_1, \lambda_2, ..., \lambda_p)^T \,, \\
A_{pq} &= \textstyle\sum_{i=1}^{N_\phi} (\phi^{i\dagger} Q_p^\dagger Q_q \phi^i) \,, \\
V_q &= \textstyle\sum_{i=1}^{N_\phi} (\phi^{i\dagger} f(\mathsf{H})^\dagger Q_q \phi^i) \,, \\
\Lambda &= A^{-1} V \,.
\end{aligned} \tag{5.37}$$

So we now have three possible optimisation procedures we may use to find an optimal choice of coefficients for our subtraction method. Unfortunately, Optimisations 2 and 3 require that we have access to the quantity $\phi^{i\dagger} f(\mathsf{H})^\dagger Q_q \phi^i$, whereas Optimisation 1 requires only the separate quantities $\Re(\phi^{i\dagger} f(\mathsf{H}) \phi^i)$ and $\Re(\phi^{i\dagger} Q_p \phi^i)$. The latter quantities are readily available to us as we must calculate them anyway, however, the former quantity is not readily available to us, and would require considerable additional work to calculate. For this reason we proceed to use Optimisation 1, more specifically, Optimisation 1a.

5.3.4 Test Chebyshev model

We can test the optimisation procedure using a very low-order polynomial approximations of a simple function.

As our function, we take a simple 3^{rd} order polynomial in H, where, in this case, $\mathsf{H} = \gamma_5 \mathsf{M}$. We then perform a hopping parameter expansion of this polynomial to see which terms appear. We then use our optimisation procedure (Optimisation 1, above) to find a coefficient, λ, for each term. We then subtract off each term, in turn, with the associated coefficient. Fig. (5.6) shows the result of subtracting-off each of these terms. We can see how the variance is reduced as we subtract off more terms.

5.3.5 Applying the traceless operator methods

Fig. (5.7) shows the effect of subtracting traceless operators from the Lánczos quantity 'L'. We can see that in this case we gain roughly a factor of ~ 10 in variance reduction.

5.3.6 Summary of method 2

We can see that approximately a factor of 10 reduction in variance is achievable by carrying out very little extra work on top of that already performed by the Lánczos procedure. By comparison with method 1, where we allowed ourselves a further, equal amount of work on top of the work performed by the Lánczos procedure, we gained a factor of 20 in variance reduction. Here if we were to allow ourselves a similar amount of work, we would gain approximately the same factor of 20 in variance reduction.

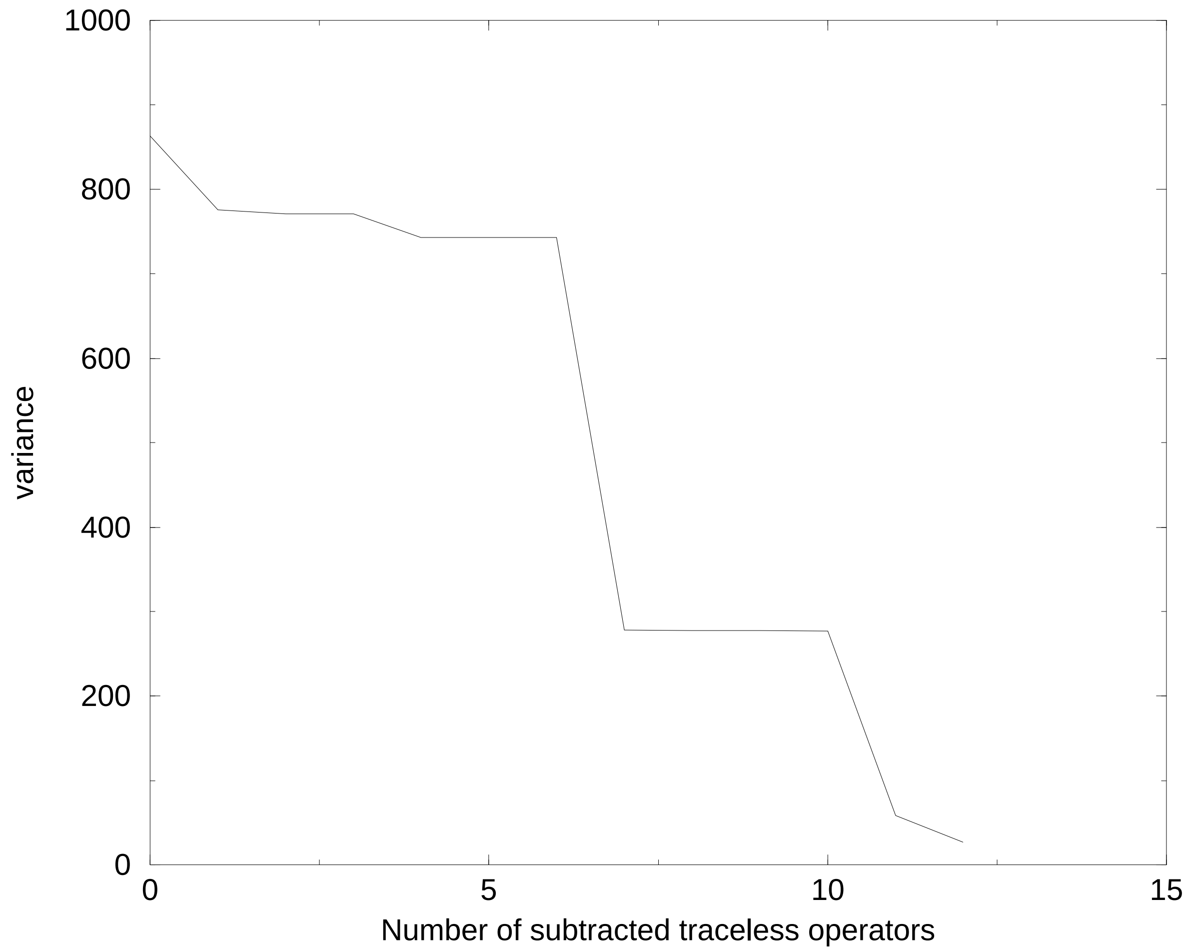


Figure 5.6: The variance of a stochastic estimate of a simple 3^{rd} order polynomial. The variance is shown as the each term of the hopping parameter expansion is subtracted off.

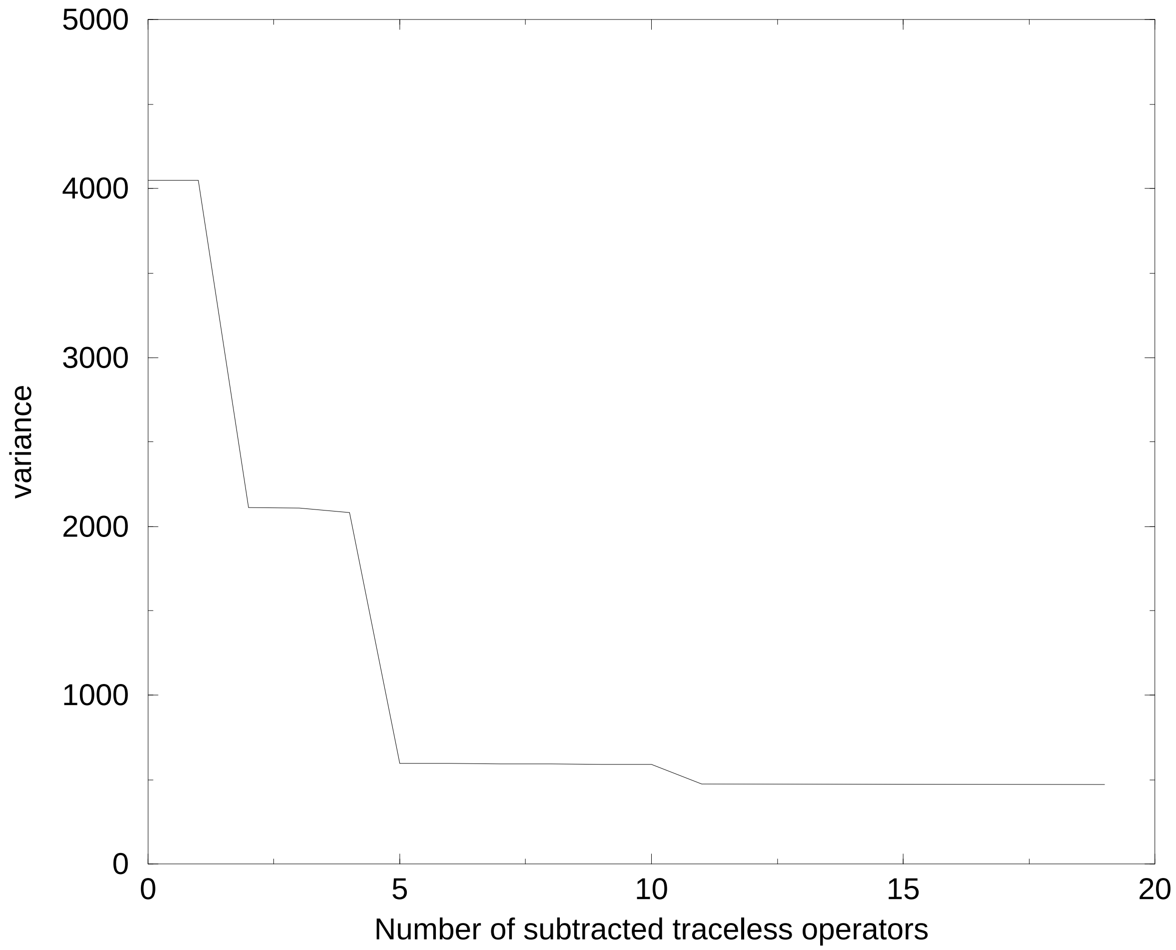


Figure 5.7: The variance of the Lánczos quantity (L) shown as various traceless operators are subtracted off. The effect shown is the cumulative effect of subtracting each operator in turn.

5.3.7 Using the ideas from both method 1 and method 2

So now we have two reliable ways of reducing the variance of an estimate of 'trace-like' quantities with not much to choose between them in terms of their efficiency. Up until now we have used both ideas separately, but it turns out that we are able to combine the ideas of both methods into a new, combined method which can easily be used to achieve an overall factor of 100 in variance reduction.

This combined method initially proceeds as with method 1. We make an estimate using the Lánczos quantity L and then subtract off a Chebyshev polynomial approximation of some low order. We then make an estimate of the quantity C as before. However, this time we also improve upon the estimate of C using traceless operators in the same manner as we did in method 2 for the Lánczos quantity.

Fig. (5.8) shows the variance reduction we can achieve using the subtraction of traceless operators from a 10^{th} order Chebyshev approximation of $\mathrm{Tr} f(\mathsf{H}) = \mathrm{Tr}\log(\mathsf{H})$, where $\mathsf{H} = \mathsf{M}^{\dagger}\mathsf{M}$.

Again we see approximately a factor of 10 reduction in the variance for this quantity. This prompts us to construct another (n, r) model as we did for method 1.

5.3.8 A new variance reduction model

Just as with the first variance reduction model described in section 5.2, we are able to model the variance reduction here using similar parameters. Following the same procedure as before, we create a variance reduction model. We use the appropriate plots contained in Fig. (5.1) and Fig. (5.2) to create equations analogous to Eqs. (5.8,5.9,5.10,5.11).

We see from the plot in Fig. (5.1) that when traceless operators are applied to the Lánczos quantity 'L', then for $n > 8$ the corresponding variance is approximately the same as it

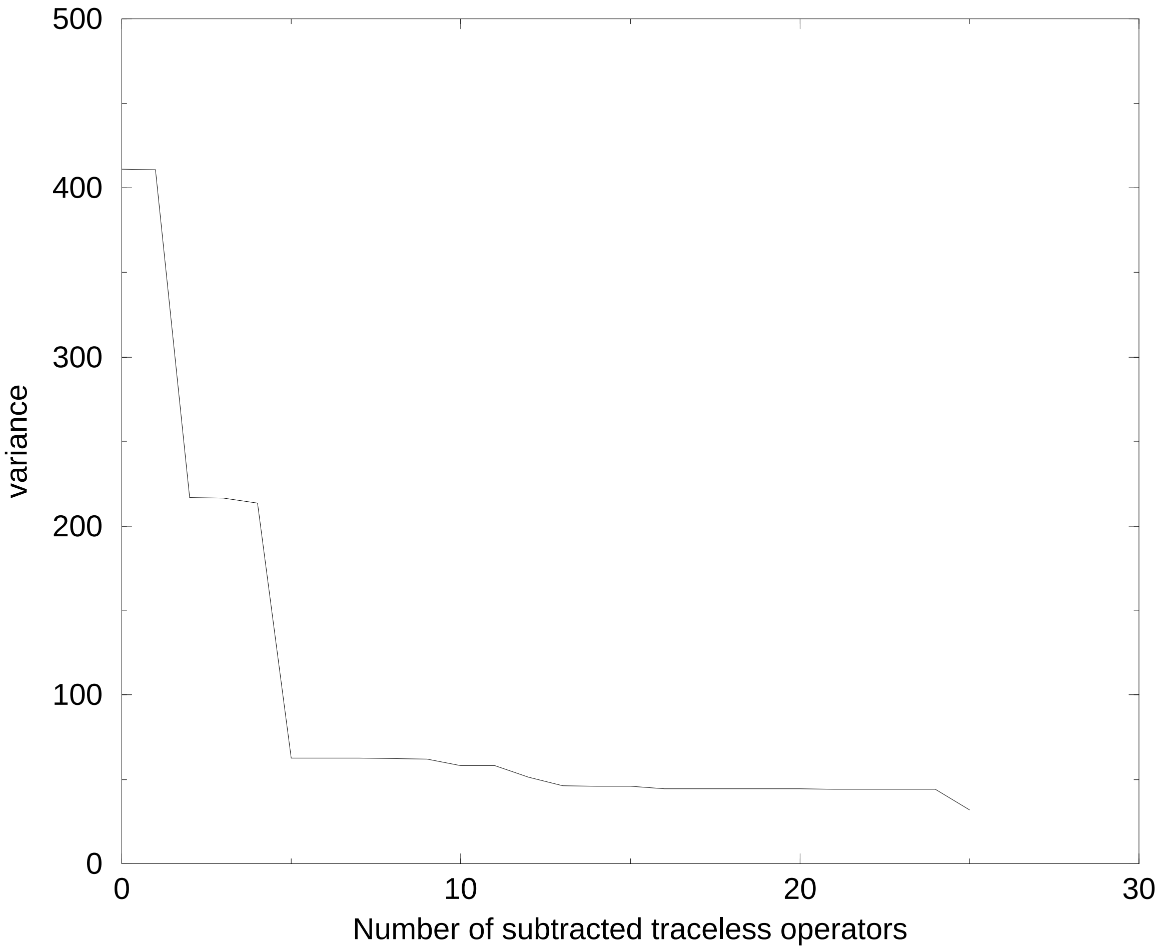


Figure 5.8: The variance of the Chebyshev (matrix) quantity (C) shown as various traceless operators are subtracted off. The effect shown is the cumulative effect of subtracting each operator in turn. The operators subtracted off were: I, A, D, A^2, D^2, A^3, $\mathsf{D}^3, \ldots$, A^7, D^7, $\mathsf{D}^\dagger\mathsf{D}\mathsf{D}^\dagger\mathsf{D}$, ADD, $\mathsf{A}^2\mathsf{D}^2$, AD^3, $\mathsf{A}^2\mathsf{D}^3$, $\mathsf{A}^3\mathsf{D}^3$, AD^5, $\mathsf{A}^2\mathsf{D}^5$, $\mathsf{A}^3\mathsf{D}^5$, $\mathsf{D}^\dagger\mathsf{A}^{-1}\mathsf{D}$.

is when no traceless operators are applied. So the equation for σ_L^2 remains essentially the same as before

$$\sigma_L^2 = \frac{935530n^{-2.68}}{N_\phi}\,, \qquad n > 8\,. \tag{5.38}$$

If we assume then that it is not worth applying the subtraction of traceless operators to the quantity L, then our equation for the work remains the same

$$W_L = N_\phi N_{lanc}\,. \tag{5.39}$$

Fig. (5.2) shows the effect of applying the subtraction of traceless operators to our quantity 'C'. As before, for $n > 8$, we have a value for the variance which is approximately independent of n. So we can write a new expression for σ_C^2

$$\sigma_C^2 = \frac{40400}{N_\eta}\,, \qquad n > 8\,. \tag{5.40}$$

We can also write a new expression for the work. The creation of traceless operators requires work on top of that already done to create the Chebyshev estimate.

For example, in order to create the operator D^4 on a single parity, we must do 4 single-parity matrix-vector multiplications. This is equivalent to 1 Krylov sweep. Of course in creating the operator D^4 we also get D, D^2 and D^3 for free. Typically a 'clover-vector' multiplication $\mathsf{A}v$ requires approximately 15% of the work of the equivalent $\mathsf{D}v$ operation. So we estimate the inclusion of a set of all of the useful traceless operators mentioned above is equivalent to approximately an extra 3 matrix-vector multiplications.

So we now write

$$W_C = N_\eta(n+3)\,. \tag{5.41}$$

We now have a new model to work with. We can go through the same process as we did for method 1. Fig. (5.9) shows the amount of work required for various amounts of

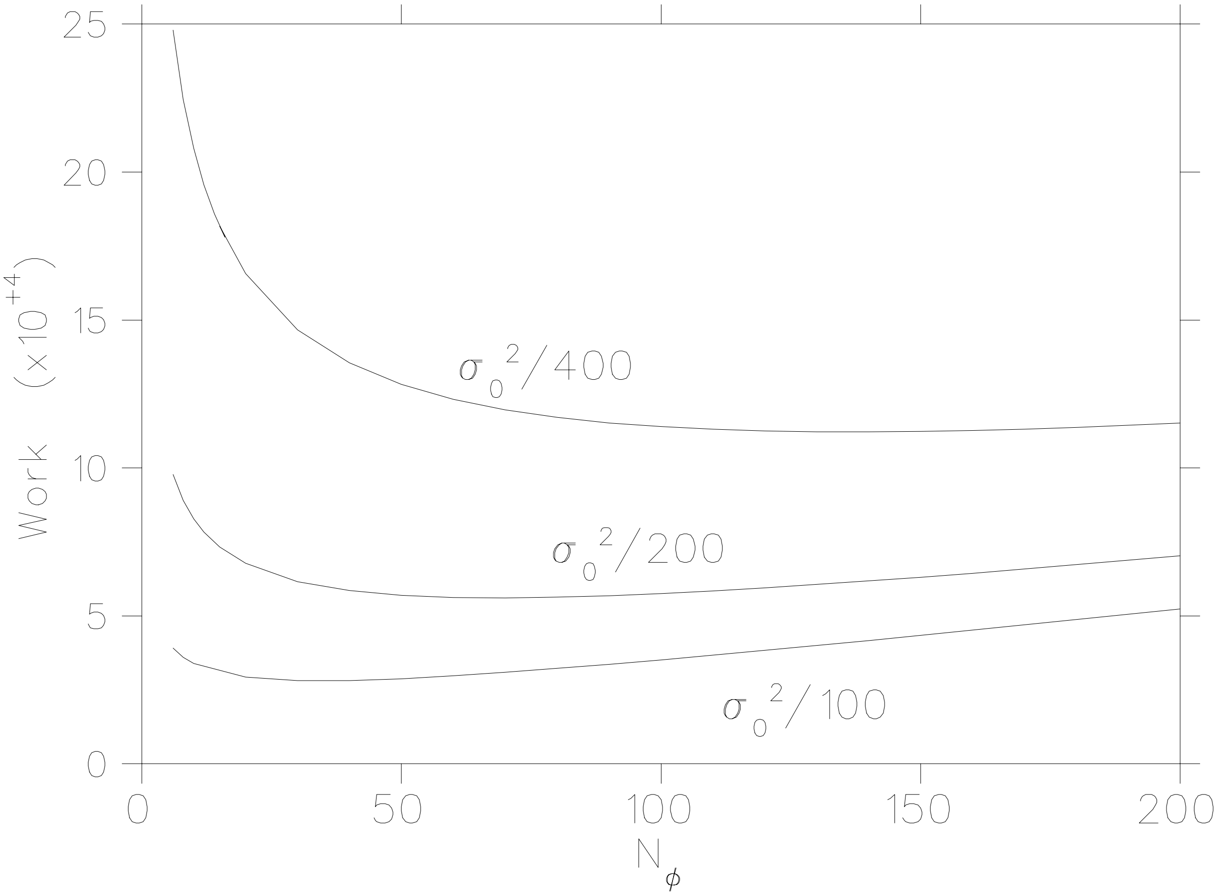


Figure 5.9: The work as a function of N_ϕ for various fixed variances for our new method combining both methods of variance reduction. The variances shown are multiples of $\sigma_0^2 = 4071.6$, the variance resulting from a calculation involving just Lánczos with $N_\phi = 100$ and no Chebyshev subtraction or traceless operator subtraction.

variance reduction for differing values of N_ϕ. We can see that factors of 100 in variance reduction are readily achievable.

5.4 Summary

Tab. (5.4) shows the results of the variance reduction method applied to the 200 $(\beta, \kappa, c_{\rm sw}) = (5.2, 0.1355, 2.0171)$ [26] configurations. The parameters we have used are $(n, r) = (14, 12.5)$. These were found to be optimal for our fixed amount of work *i. e.* $W_0 = 40,000$. Here we see a clear factor of ~ 100 in variance reduction, *i. e.* a factor of ~ 10 in the error.

So we have managed to combine two separate variance reduction ideas into one two-stage variance reduction method. We have then shown a way of optimising the parameters in terms of the overall work necessary to create an estimate of a 'trace-like' quantity such as $\mathrm{Tr}\log(\mathsf{H})$ and in terms of the overall variance of this stochastic estimate.

We now summarise these methods.

1) Firstly, instead of using Lánczos to calculate the $\mathrm{Tr}f(\mathsf{H})$ quantity, we use it to estimate a different, smaller quantity. We do this by subtracting off a low order Chebyshev polynomial approximation to the $\mathrm{Tr}f(\mathsf{H})$ as a function of the Ritz values. In order then to retrieve the $\mathrm{Tr}f(\mathsf{H})$ quantity of interest, we add back on the polynomial, but this time as a function of the matrix itself. The idea here is that the first (Lánczos) quantity is a small quantity with a small associated variance. The second (matrix) quantity is a relatively inexpensive quantity to calculate (by virtue of its low-order), and can therefore be calculated with higher accuracy using more noise vectors.
$$E'_f = \underbrace{E_f(\theta) - C^n(\theta)}_{\text{Lánczos}} + \underbrace{C^n(\mathsf{H})}_{\text{high statistics}} \tag{5.42}$$
where E'_f is our new estimate for $\mathrm{Tr}f(\mathsf{H})$ and C^n is our low-order Chebyshev polynomial.

2) The second step is a noise reduction step. We improve upon the second (matrix) quantity by subtracting off traceless operators in such a way that the variance due to stochastic noise is optimally reduced [50].

Table 5.2: The $\mathrm{Tr}\log(\mathsf{H})$ values for all 201 configurations analysed. $\mathsf{H} = \mathsf{M}^\dagger \mathsf{M}$ (red-black pre-conditioning applied).

Config. no.	$\mathrm{Tr}\log(\mathsf{H})$	Config. no.	$\mathrm{Tr}\log(\mathsf{H})$	Config. no.	$\mathrm{Tr}\log(\mathsf{H})$
16400	-24546.99 ± 6.22	19080	-24919.38 ± 6.20	21760	-24717.14 ± 6.15
16440	-24397.14 ± 6.35	19120	-24808.35 ± 6.21	21800	-24646.78 ± 6.08
16480	-24671.32 ± 6.08	19160	-24317.11 ± 6.41	21840	-24683.05 ± 6.15
16520	-24513.19 ± 6.28	19200	-24726.73 ± 6.18	21880	-24723.98 ± 6.15
16560	-24540.31 ± 6.15	19240	-24781.89 ± 6.35	21920	-24784.95 ± 6.46
16600	-24766.55 ± 6.39	19280	-24706.50 ± 6.18	21960	-24212.53 ± 6.26
16640	-24476.06 ± 6.24	19320	-24608.21 ± 6.24	22000	-24617.67 ± 6.15
16680	-24622.39 ± 6.44	19360	-24626.98 ± 6.27	22040	-24978.74 ± 6.19
16720	-24559.75 ± 6.25	19400	-24528.18 ± 6.43	22080	-24909.90 ± 6.23
16760	-24849.59 ± 6.20	19440	-24905.18 ± 6.11	22120	-24547.13 ± 5.95
16800	-24711.21 ± 6.24	19480	-24855.57 ± 6.35	22160	-24525.33 ± 6.26
16840	-24848.56 ± 6.11	19520	-24856.49 ± 6.34	22200	-24561.23 ± 5.94
16880	-24714.28 ± 6.57	19560	-24752.60 ± 6.34	22240	-24763.63 ± 6.32
16920	-24407.74 ± 6.19	19600	-24672.32 ± 6.35	22280	-24691.30 ± 6.53
16960	-24699.22 ± 6.16	19640	-24419.72 ± 6.38	22320	-24951.30 ± 6.17
17000	-24808.81 ± 6.18	19680	-24500.93 ± 5.97	22360	-24718.82 ± 6.10
17040	-24792.07 ± 6.36	19720	-24488.60 ± 6.26	22400	-24965.14 ± 6.19
17080	-24781.13 ± 6.32	19760	-24804.69 ± 6.17	22440	-24818.02 ± 6.44
17120	-24560.29 ± 5.98	19800	-24546.72 ± 6.39	22480	-25118.27 ± 6.11
17160	-25010.61 ± 6.11	19840	-24853.16 ± 6.06	22520	-24858.05 ± 6.34
17200	-24604.34 ± 6.38	19880	-24660.68 ± 6.29	22560	-24585.15 ± 6.45
17240	-24739.26 ± 5.95	19920	-25030.91 ± 6.19	22600	-24781.53 ± 6.17
17280	-24663.84 ± 6.31	19960	-24805.48 ± 6.15	22640	-24752.83 ± 6.37
17320	-24756.89 ± 6.24	20000	-25131.57 ± 6.22	22680	-24634.61 ± 6.34
17360	-24826.72 ± 6.11	20040	-24719.98 ± 6.26	22720	-24843.87 ± 6.17
17400	-24602.90 ± 6.27	20080	-24575.96 ± 6.24	22760	-24677.52 ± 6.22
17440	-24907.49 ± 6.27	20120	-24669.23 ± 6.36	22800	-24651.25 ± 6.15
17480	-24702.55 ± 6.10	20160	-24749.82 ± 6.12	22840	-24652.45 ± 6.30
17520	-24576.92 ± 5.97	20200	-24967.52 ± 6.17	22880	-24515.63 ± 6.18
17560	-24360.43 ± 6.34	20240	-24854.81 ± 6.40	22920	-24733.73 ± 6.31
17600	-24962.30 ± 6.12	20280	-24561.73 ± 6.13	22960	-24740.00 ± 6.58
17640	-24709.21 ± 5.86	20320	-24780.24 ± 6.22	23000	-24740.46 ± 6.12
17680	-24830.70 ± 6.24	20360	-24609.15 ± 6.36	23040	-24484.48 ± 6.34
17720	-24663.69 ± 6.27	20400	-24597.56 ± 6.28	23080	-24609.63 ± 6.21
17760	-24759.13 ± 6.18	20440	-24820.44 ± 6.39	23120	-24764.73 ± 6.15
17800	-24693.43 ± 6.20	20480	-25040.43 ± 6.24	23160	-24825.84 ± 6.31
17840	-24547.56 ± 6.03	20520	-24623.80 ± 6.31	23200	-25101.38 ± 6.32
17880	-24661.92 ± 6.32	20560	-24579.93 ± 6.13	23240	-24894.45 ± 6.46
17920	-24747.24 ± 6.21	20600	-24876.19 ± 6.31	23280	-24902.44 ± 6.25
17960	-24738.99 ± 6.20	20640	-24624.55 ± 6.26	23320	-24918.31 ± 6.01
18000	-24886.45 ± 6.30	20680	-24596.11 ± 6.10	23360	-24731.67 ± 6.28
18040	-24821.32 ± 6.03	20720	-24507.79 ± 6.27	23400	-24840.56 ± 6.38
18080	-24762.41 ± 6.21	20760	-24482.35 ± 6.12	23440	-24731.31 ± 6.12
18120	-24820.55 ± 6.39	20800	-24721.35 ± 6.26	23480	-24855.05 ± 6.25
18160	-24522.67 ± 6.29	20840	-24646.83 ± 6.18	23520	-24574.95 ± 6.26
18200	-24267.32 ± 6.18	20880	-24869.78 ± 6.37	23560	-24871.38 ± 6.17
18240	-24698.86 ± 6.27	20920	-24423.66 ± 6.24	23600	-24551.08 ± 6.30
18280	-24837.16 ± 6.33	20960	-24770.07 ± 6.13	23640	-24776.15 ± 6.20
18320	-24719.74 ± 6.07	21000	-24734.61 ± 6.19	23680	-24507.33 ± 6.24
18360	-24800.87 ± 6.21	21040	-24692.32 ± 6.21	23720	-24958.22 ± 6.31
18400	-24892.87 ± 6.06	21080	-24749.61 ± 6.10	23760	-24719.70 ± 6.28
18440	-24676.06 ± 6.41	21120	-24645.92 ± 6.21	23800	-24732.50 ± 6.32
18480	-24754.49 ± 6.05	21160	-24794.08 ± 6.16	23840	-24643.84 ± 6.29
18520	-24676.38 ± 6.12	21200	-24651.49 ± 6.28	23880	-24819.50 ± 6.22
18560	-24587.29 ± 6.32	21240	-24782.15 ± 5.94	23920	-24793.09 ± 6.34
18600	-24407.95 ± 5.97	21280	-24875.31 ± 6.17	23960	-24586.31 ± 6.19
18640	-24778.18 ± 6.10	21320	-24828.66 ± 6.37	24000	-24724.33 ± 6.26
18680	-24732.28 ± 6.45	21360	-24677.95 ± 6.06	24040	-24840.08 ± 6.13
18720	-24645.46 ± 6.20	21400	-24657.51 ± 6.35	24080	-24790.41 ± 6.04
18760	-24699.57 ± 6.11	21440	-24860.75 ± 6.09	24120	-24687.40 ± 6.46
18800	-24520.41 ± 5.91	21480	-24978.23 ± 6.18	24160	-24991.94 ± 6.29
18840	-24379.38 ± 6.22	21520	-24788.82 ± 6.11	24200	-24775.26 ± 6.36
18880	-24676.38 ± 6.22	21560	-24742.28 ± 6.34	24240	-24678.84 ± 6.08
18920	-24645.28 ± 6.34	21600	-24520.66 ± 6.23	24280	-24495.39 ± 6.09
18960	-24718.57 ± 6.31	21640	-24649.88 ± 6.32	24320	-24727.18 ± 6.01
19000	-24901.44 ± 6.28	21680	-24879.84 ± 6.47	24360	-24891.41 ± 6.26
19040	-24690.76 ± 6.23	21720	-24737.07 ± 6.51	24400	-24835.39 ± 6.26

Chapter 6

Applications to Dynamical Fermion Simulation

6.1 Introduction

This chapter describes some of the possible ways we may apply the technology developed during the course of this thesis. In particular we discuss the case of dynamical fermions. Hybrid Monte Carlo (HMC) methods are presently used to simulate dynamical fermions. However as one proceeds to ever-lighter quark masses, HMC algorithms become increasingly costly and the creation of light dynamical configurations becomes ever-more difficult. The effects of light fermionic modes are subtle in comparison with the effects which are due to the gauge action alone. Consequently, with access to only a small number of light dynamical configurations, it is difficult to see the effects which are due specifically to light quark modes. This chapter is primarily concerned with techniques which we may use to highlight the effects of light quark modes in a Monte Carlo simulation. Essentially, there are two possible paths we may follow. Firstly we can simply simulate with a truncated action described in this chapter. This will result in a truncated probability measure, biased towards the creation of lighter quark modes. Alternatively, we may replace the missing modes in the truncated action using an approximate loop action.

6.2 Analysis of the gauge configurations

An obvious quantity to study is σ_T, the fluctuation of the $\mathrm{Tr}\log(\mathsf{H})$ quantity as we go from gauge to gauge.

In our case we have noisy estimates of quantities of $T = \mathrm{Tr}\log(\mathsf{H})$ for the 200 configurations $((\beta, \kappa, c_{\mathrm{sw}}) = (5.2, 0.1355, 2.0171)$ [26]). Due to the effects of noise, if we measure the fluctuation from gauge to gauge, we find that we have an apparent increase in the value of σ_T, say σ'_T.

$$\sigma'^2_T = \sigma^2_T + \sigma^2_\phi \tag{6.1}$$

where σ_ϕ is the variance due to noise.

$$\sigma^2_\phi \propto \frac{1}{N_\phi} \tag{6.2}$$

so we can write

$$\sigma'^2_T(N_\phi) = \sigma^2_T + \frac{a}{N_\phi} \tag{6.3}$$

where a is a constant. We can therefore extract a value for σ^2_T by plotting σ'^2_T against $\frac{1}{N_\phi}$. Fig. (6.1) shows such a plot and we find that $\sigma'^2_T \approx 43400$ (and $\sigma_T \approx 208.3$).

We can measure the normalised correlation between P (the average value of the plaquette) and T. In the absence of any noise in the measurement of T, we can write this correlator as $C_0(P,T)$. This is given by

$$C_0(P,T) = \frac{\langle (P - \langle P \rangle)(T - \langle T \rangle) \rangle}{\sigma_P \sigma_T}\,. \tag{6.4}$$

When our estimates of T are noisy ones, we have

$$C(P,T,N_\phi) = \frac{\langle (P - \langle P \rangle)(T - \langle T \rangle) \rangle}{\sigma_P \sigma'_T} \tag{6.5}$$

and we can see that

$$C(P,T,N_\phi) = b(N_\phi) C_0(P,T) \tag{6.6}$$

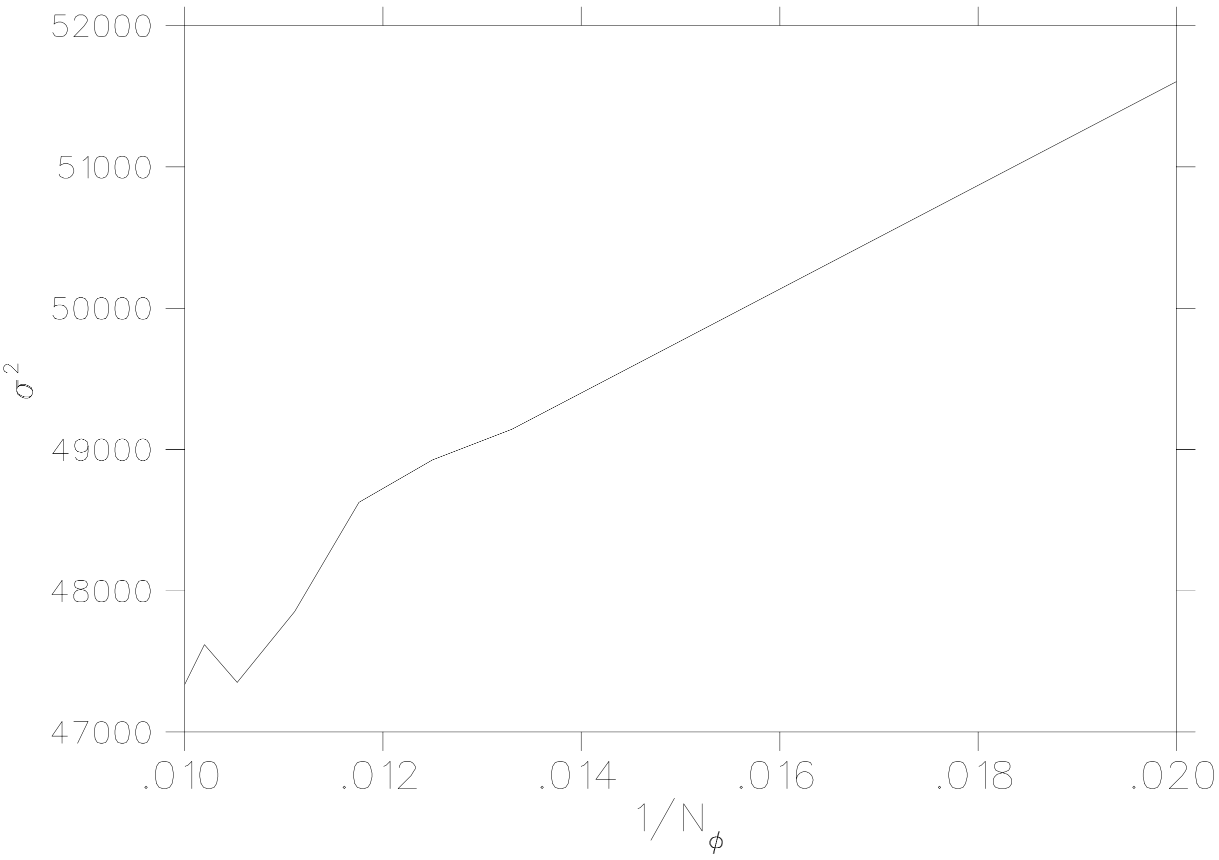


Figure 6.1: A plot of $\sigma_T'^2$ against $\frac{1}{N_\phi}$ in order to extrapolate to $\sigma_T'^2$, the fluctuation from gauge to gauge in the absence of noise.

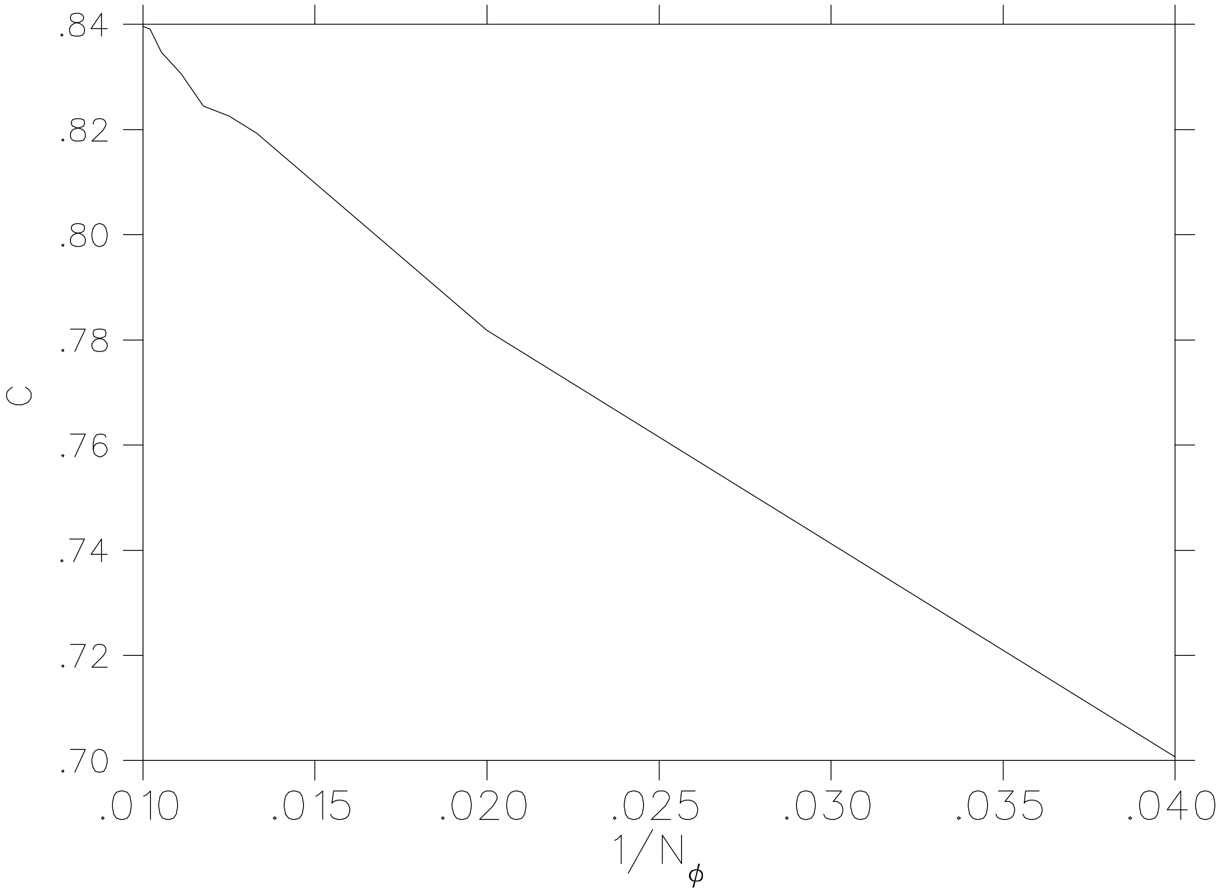


Figure 6.2: A plot of $C(P,T,N_\phi)$ against $b(N_\phi)$ to extrapolate to $C_0(P,T)$, the correlation of the plaquette and $\mathrm{Tr}\log(\mathsf{H})$ in the absence of noise.

where

$$b(N_\phi) = \left[1 + \frac{a}{N_\phi \sigma_T^2}\right]^{-\frac{1}{2}}. \tag{6.7}$$

So having measured σ_T and a (from Eq. (6.7)), we can now measure C_0, by plotting C against $1/N_\phi$ and extrapolating to $N_\phi \to \infty$. Fig. (6.2) shows such a plot and we find $C_0(P,T) \approx 88\%$.

6.3 Truncated Fermion Actions

A hopping parameter expansion shows that, to low order in κ, the value of $\mathrm{Tr}\log(\mathsf{H})$ is simply proportional to $\langle P \rangle$, the plaquette value. One would therefore expect a certain amount of correlation between these two quantities as we move from gauge to gauge. We would expect this value to diminish as we reduce the mass of the quarks (*i. e.* as we increase κ). Figs. (6.4,6.5,6.6) illustrates the correlation between the $\mathrm{Tr}\log(\mathsf{H})$ quantity and the plaquette value for each configuration. Each quantity is normalised to its average.

We would like to study directly the effects arising from the long range (infrared) behaviour of light dynamical quarks. In order to do this we employ a truncation method. Some work has been presented in [51, 52, 53], where a hard truncation of the eigenvalues is employed after having calculated the full eigenspectrum using the Lánczos method with the Cullum and Willoughby filtering techniques applied [38]. Here we adopt a smooth cut-off using the available Ritz values which is designed to give a smooth convergence in terms of the (small) number of Lánczos iterations.

The truncation is achieved by weighting the Ritz values with a truncation profile function

$$\frac{1}{2}(1 - \tanh b\frac{\theta - \theta_c}{\theta_{min} - \theta_c})\,, \tag{6.8}$$

where b is chosen to control the smoothness of the cut-off and θ_{min} is the minimum Ritz value. The value of θ_c is chosen to represent an energy-momentum cut-off. This is done by relating the eigenvalues of the fermion matrix in the free-field case to a corresponding energy. This idea is quantified in appendix D. A particular energy is chosen as a cut-off. We then relate this energy to the appropriate eigenvalue, θ_c. This value of θ_c can then be used as the cut-off value in the full interacting case. The cut-off now refers to what we class as a pseudo energy-scale cut-off.

We are effectively introducing an energy cut-off in order to separate out the high and low eigenmodes. For heavy quarks, the majority of the contribution to the fermion determinant comes from short-range (ultra-violet) effects. With lighter quarks, there are contributions from both ultra-violet and infra-red (long-range) parts of the spectrum. Using a truncation method, one should be able to illustrate the long-range effects of the light quarks.

Fig. (6.3) shows the fluctuation of the full $\mathrm{Tr}\log(\mathsf{H})$ quantity compared with that of the truncated $\mathrm{Tr}\log(\mathsf{H})$ quantity. If we measure the correlation between the plaquette and the new, truncated, $\mathrm{Tr}\log(\mathsf{H})$ quantity, we see that the correlation drops. We have measured this correlation to be $\sim 56\%$. This drop in correlation is anticipated as we expect the long-range modes of the light quarks to have a lower correlation with the plaquette.

6.4 Monte Carlo Simulations

One possible way we may go about incorporating a truncated algorithm into a Monte Carlo scheme, is via an accept-reject step. During the execution of a Metropolis accept-reject step, we take the current configuration and propose a new, candidate configuration as an update to the current one. In order to satisfy the desired probability distributions, we then compare the full action of each configuration and accept or reject the proposed update depending on the difference between these two actions. One alternative to this may be to compare the difference between two actions which have been truncated using the above energy-truncation method.

Obviously if we have truncated the action in such a way that our accept-reject step takes into account only the low-lying fermion modes, we still need to deal with the rest of the spectrum. The possibilities for constructing approximate actions have been discussed in [54, 51, 55]. These are based on the idea of approximating the ultra-violet part of the spectrum with an effective loop action.

In order to produce a workable algorithm based on the above technology, we need to have a reasonable acceptance rate for our candidate configurations. According to [56], we have an acceptance probability α given by

$$\alpha(\Delta(S)) = erfc(\frac{1}{2}\sqrt{\sigma^2(\Delta(S))})\,. \tag{6.9}$$

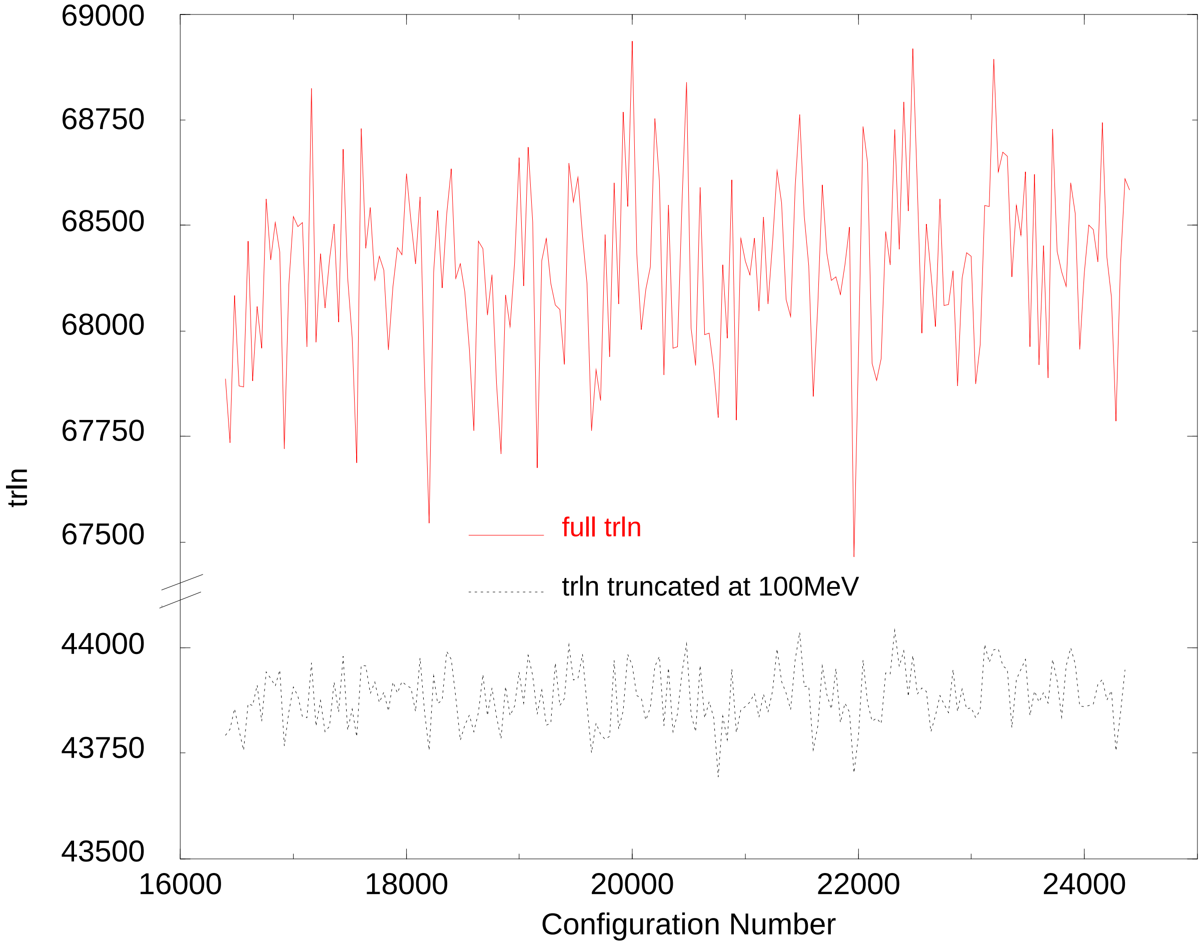


Figure 6.3: The full $\mathrm{Tr}\log(\mathsf{H})$ compared with that of the truncated value. The truncation energy is 100 MeV.

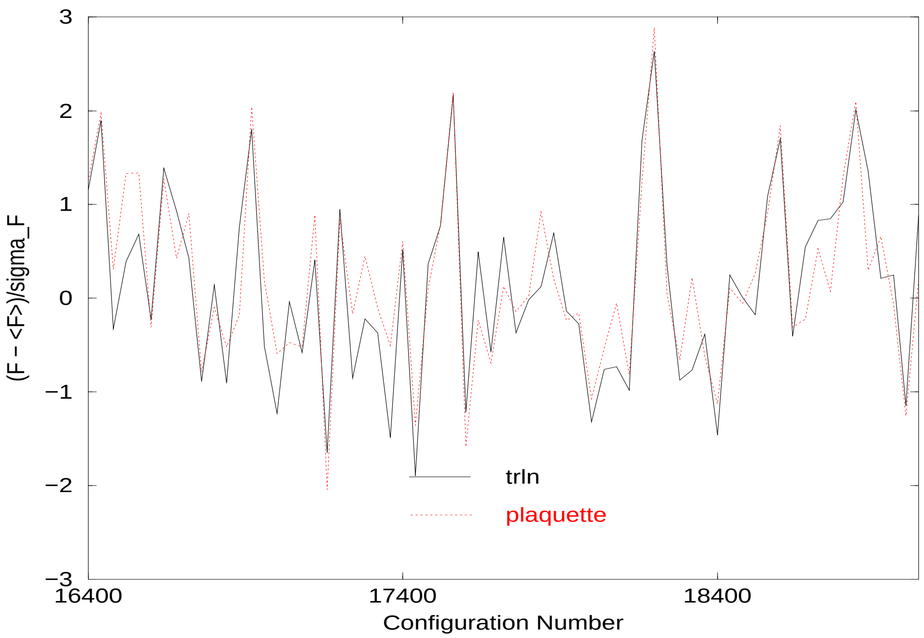


Figure 6.4: Correlation of full $\mathrm{Tr}\log(\mathsf{H})$ compared with that of the plaquette value. Both values are normalised to their average.

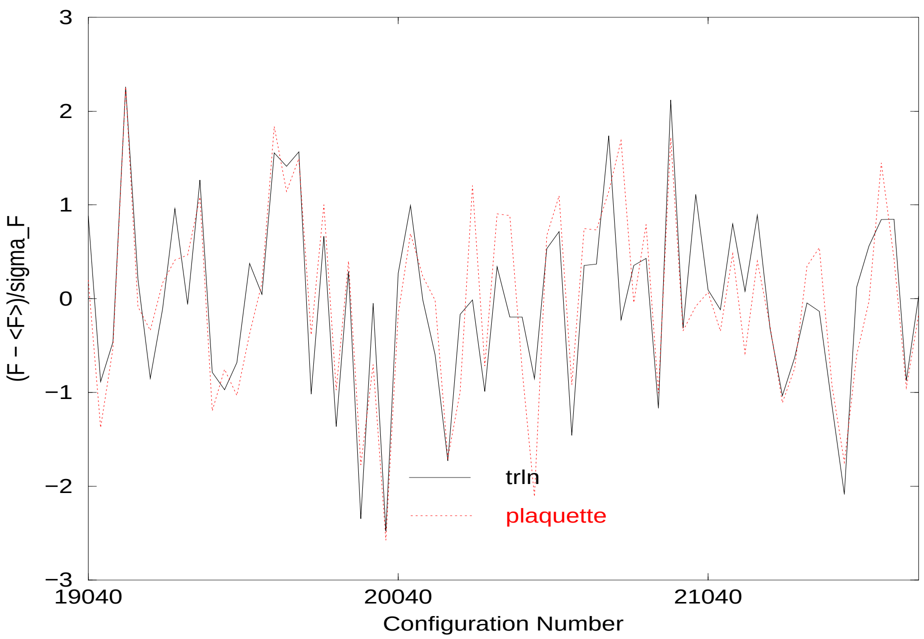


Figure 6.5: A continuation of Fig. (6.4).

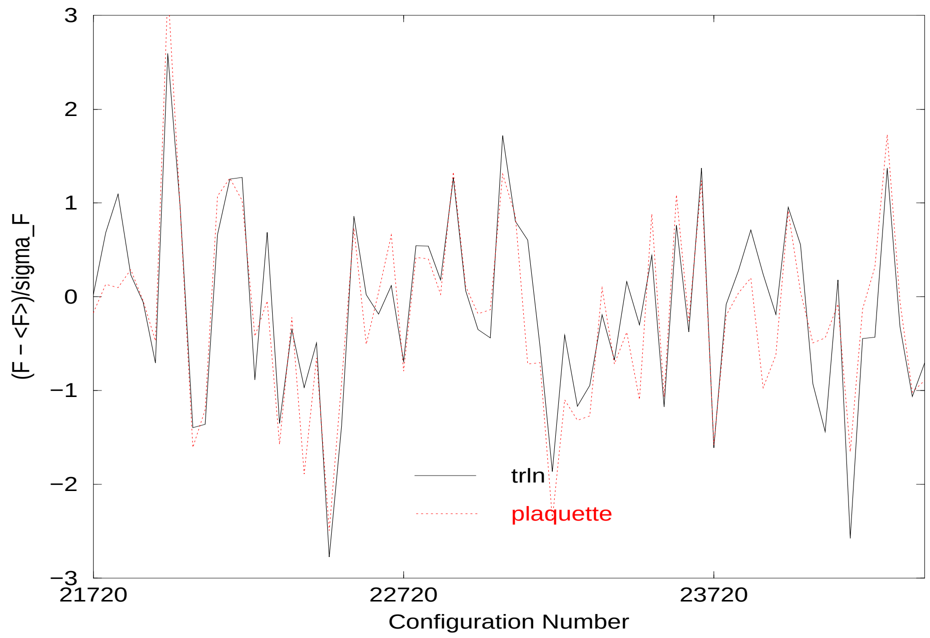


Figure 6.6: A continuation of Fig. (6.4).

where $\sqrt{\sigma^2(\Delta(S))}$ refers to the fluctuation of $\Delta(S)$ from one configuration to the next. In order to achieve an acceptance of 50%, we need to have a fluctuation of $\sigma^2(\Delta(S)) \approx 1$, while only a 10% acceptance is achieved if $\sigma^2(\Delta(S)) \approx 5$.

In our case we have analysed configurations which are separated by 40 trajectories. Fig. (6.7) shows the variance, $\sigma_T'^2(\theta_c, N_\phi)$, *i. e.* the fluctuation of the $\mathrm{Tr}\log(\mathsf{H})$ quantity as we move from gauge to gauge. This fluctuation is shown as a function of the pseudo-energy cut-off. Obviously, the average fluctuation of $\Delta(S)$ between consecutive configurations for the full action, will be much less than the fluctuation of S itself, for configurations separated by 40 trajectories updating, say, a loop action. However, we can see, this fluctuation is very high, implying that our value of $\sigma^2(\Delta(S))$ will be very much greater than unity. Unfortunately, this implies that the acceptance rate for such an algorithm would be very low. However, it is clear that more investigation is needed into the precise nature of such an algorithm.

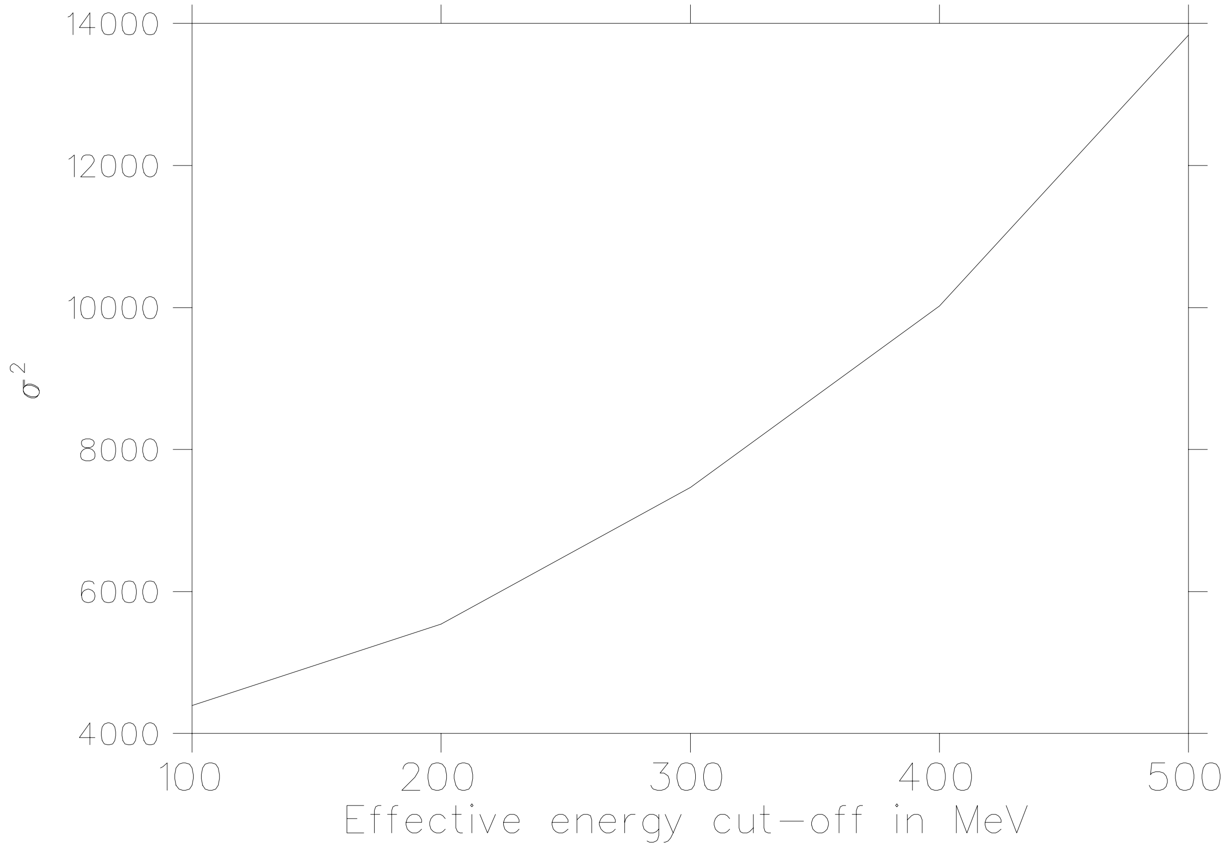


Figure 6.7: The fluctuation of the $\mathrm{Tr}\log(\mathsf{H})$ quantity as we go from gauge to gauge, shown as a function of the pseudo-energy cutoff

6.5 Conclusions

We have measured the quantity $T = \mathrm{Tr}\log(\mathsf{H})$ for the 200 configurations referred to throughout this thesis. We have then performed statistical analysis on these configurations, allowing us to measure the fluctuation of the quantity T as we move from gauge to gauge. We have also measured the correlation between the quantity T and the quantity $\langle P \rangle$, the average plaquette value for each configuration. This quantity was measured over the same 200 configurations. We have then discussed some preliminary ideas with regard to using our technology in a dynamical simulation, designed to highlight the effect of light quark modes. The ideas here are based on a truncated determinant combined with an effective loop action.

Chapter 7

Conclusions

7.1 Introduction

This chapter is intended as a short summary of the work completed for this thesis along with some discussion of the direction any future work may take.

We have found that the Bai, Fahey and Golub (BFG) approach to the Lánczos method is applicable to the stochastic evaluation of 'trace-like' quantities which appear in lattice QCD. In particular we have found that relatively few Lánczos iterations are required to produce a reasonable estimate of such quantities. We have also found that the BFG method can be combined with a filter and convergence test to evaluate a subset of the eigenvalues of a large, sparse matrix, such as the fermion matrix, on which our investigations were concentrated.

We have implemented the fermion matrix in three different Hermitian forms, namely, $\mathsf{M}^\dagger\mathsf{M}$ (with red-black pre-conditioning applied), $\gamma_5\mathsf{M}$ (full matrix) and $\mathsf{M}^\dagger\mathsf{M}$ (full lattice). We have found that due to the large decrease of the matrix condition number when red-black pre-conditioning is applied, the convergence of trace quantities of the matrix $\mathsf{M}^\dagger\mathsf{M}$ (with red-black pre-conditioning applied) is clearly better than for the other two representations.

We have found that in addition to the estimates produced in the BFG method, we can improve upon our estimates considerably by applying variance reduction. We have studied two forms of variance reduction, one which is based on Chebyshev approximations and one which involves the subtraction of traceless operators. Both of these methods work well individually and each achieves a factor of 10 reduction in the variance. However, it also turns out that both methods can be combined together producing a multiplicative reduction in the variance, giving an overall factor of 100.

We have then explored the idea of using our technology to illustrate the effects of light dynamical quarks. In the final chapter we have done some analysis of the $\mathrm{Tr}\log(\mathsf{H})$ quantity, measured on each configuration using our technology. This has led us to some discussion of a possible Monte Carlo algorithm which could be employed to highlight the effects the light dynamical quark modes.

As we have seen, the effects of light dynamical quarks are quite subtle and can consequently become swamped by the effects of other modes. Present algorithms, such as HMC, base their updating on all quark modes. Consequently, using such algorithms, it is difficult to realise the effects which are due solely to light dynamical quarks. Clearly the next step in these investigations would involve using the technology to investigate the feasibility of creating a viable simulation algorithm, capable of concentrating on the physically relevant long-range modes. Thus efficiently illustrating the effects of light dynamical quarks.

Appendix A

Matrix Notation and Convention

A.1 γ-matrices

$$\gamma_1 = \begin{pmatrix} 0 & 0 & 0 & i \\ 0 & 0 & i & 0 \\ 0 & -i & 0 & 0 \\ -i & 0 & 0 & 0 \end{pmatrix}, \quad \gamma_2 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \quad \gamma_3 = \begin{pmatrix} 0 & 0 & i & 0 \\ 0 & 0 & 0 & -i \\ -i & 0 & 0 & 0 \\ 0 & i & 0 & 0 \end{pmatrix},$$

$$\gamma_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}, \quad \gamma_5 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

A.2 σ-matrices

$$\sigma_{12} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \sigma_{13} = \begin{pmatrix} 0 & -i & 0 & 0 \\ i & 0 & 0 & 0 \\ 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \end{pmatrix}, \quad \sigma_{14} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix},$$

$$\sigma_{23} = \begin{pmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{pmatrix}, \quad \sigma_{24} = \begin{pmatrix} 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \\ 0 & -i & 0 & 0 \\ i & 0 & 0 & 0 \end{pmatrix}, \quad \sigma_{34} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}.$$

Appendix B

Clover Terms

B.1 Introduction

In this appendix we describe how the clover term is constructed and how the trace of this term (and its corresponding higher powers) may be calculated.

B.2 The Clover Term

As we saw in chapter 1, the 'Clover' term is given by

$$\mathsf{A} = \mathsf{I} - \frac{1}{2}\kappa c_{\rm sw}\sigma_{\mu\nu}F_{\mu\nu}\,, \tag{B.1}$$

where

$$\sigma_{\mu\nu} = \frac{i}{2}[\gamma_\mu, \gamma_\nu]\,, \tag{B.2}$$

and $F_{\mu\nu}$ is the lattice field tensor.

$$F_{\mu\nu} = \frac{1}{8}(f_{\mu\nu} - f^\dagger_{\mu\nu})\,, \tag{B.3}$$

where

$$\begin{aligned} f_{\mu\nu} &= U_{\mu\ x}U^\dagger_{\nu\ x+\mu}U_{\mu\ x+\nu}U^\dagger_{\nu\ x} \\ &+ U_{\nu\ x}U^\dagger_{\mu\ x-\mu+\nu}U^\dagger_{\nu\ x-\mu}U_{\mu\ x-\nu} \\ &+ U^\dagger_{\mu\ x-\mu}U^\dagger_{\nu\ x-\mu-\nu}U_{\mu\ x-\mu-\nu}U_{\nu\ x-\nu} \\ &+ U^\dagger_{\nu\ x-\nu}U_{\mu\ x-\nu}U_{\nu\ x+\mu-\nu}U^\dagger_{\mu x}\,. \end{aligned} \tag{B.4}$$

So we can write the clover term in the form $\mathsf{A} = \mathsf{I} + \kappa c_{\mathrm{sw}} \mathsf{S}$ and remembering that the $\sigma_{\mu\nu}$ are symmetric and the $F_{\mu\nu}$ are anti-symmetric, we write S as

$$\mathsf{S} = -\frac{1}{2}\sigma_{\mu\nu}F_{\mu\nu} = -\sum_{\mu<\nu}\sigma_{\mu\nu}F_{\mu\nu} = \begin{pmatrix} W_0 & W_1 & W_2 & W_3 \\ W_1^\dagger & -W_0 & W_3^\dagger & -W_2 \\ W_2 & W_3 & W_0 & W_1 \\ W_3^\dagger & -W_2 & W_1^\dagger & -W_0 \end{pmatrix}, \tag{B.5}$$

where the blocks W are given in terms of the 3×3 matrices

$$\begin{aligned} W_1 &= F_{12} \\ W_2 &= iF_{13} + F_{23} \\ W_3 &= -F_{34} \\ W_4 &= iF_{24} - F_{14}\,. \end{aligned} \tag{B.6}$$

We are now in a position to calculate the trace of various powers of the clover term A. As can be seen above, the matrix S is clearly traceless, this means that

$$\mathrm{Tr}(\mathsf{A}) = \mathrm{Tr}(\mathsf{I})\,. \tag{B.7}$$

We know consider

$$\begin{aligned} \mathrm{Tr}(\mathsf{A}^2) &= \mathrm{Tr}(\mathsf{I} + 2\mathsf{S} + \mathsf{S}^2) \\ &= \mathrm{Tr}(\mathsf{I} + \mathsf{S}^2)\,. \end{aligned} \tag{B.8}$$

So now we need to calculate $\mathrm{Tr}(\mathsf{S}^2)$

$$= \mathrm{Tr}(\frac{1}{4}(\sigma_{\mu\nu}F_{\mu\nu})^2) = 4\mathrm{Tr}(F_{12}^2 + F_{13}^2 + F_{14}^2 + F_{23}^2 + F_{24}^2 + F_{34}^2)\,. \tag{B.9}$$

So then we have

$$\mathrm{Tr}(\mathsf{A}^2) = \mathrm{Tr}(\mathsf{I} + 4\kappa^2 c_{\mathrm{sw}}^2(F_{12}^2 + F_{13}^2 + F_{14}^2 + F_{23}^2 + F_{24}^2 + F_{34}^2))\,. \tag{B.10}$$

It turns out that all odd-powers of the matrix S are traceless, so $\mathrm{Tr}(\mathsf{A}^3)$ is now easy to calculate

$$\begin{aligned}
\mathrm{Tr}(\mathsf{A}^3) &= \mathrm{Tr}(\mathsf{I} + 3\kappa c_{\mathrm{sw}}\mathsf{S} + 3\kappa^2 c_{\mathrm{sw}}^2\mathsf{S}^2 + \kappa^3 c_{\mathrm{sw}}^3\mathsf{S}^3) \\
&= \mathrm{Tr}(\mathsf{I} + 3\kappa^2 c_{\mathrm{sw}}^2\mathsf{S}^2) \\
&= \mathrm{Tr}(\mathsf{I} + 12\kappa^2 c_{\mathrm{sw}}^2(F_{12}^2 + F_{13}^2 + F_{14}^2 + F_{23}^2 + F_{24}^2 + F_{34}^2))\,.
\end{aligned} \tag{B.11}$$

Similarly we can go on and calculate higher powers,

$$\begin{aligned}
\mathrm{Tr}(\mathsf{S}^4) = {} & \\
& 4F_{12}^4 + 4F_{13}^4 + 4F_{23}^4 + 4F_{24}^4 + 4F_{14}^4 \\
& +24F_{12}^2F_{34}^2 + 4F_{34}^4 + 8F_{12}^2F_{13}^2 \\
& +8F_{12}^2F_{23}^2 + 8F_{12}^2F_{24}^2 \\
& +8F_{12}^2F_{14}^2 + 8F_{13}^2F_{23}^2 \\
& -32F_{12}F_{34}F_{13}F_{24} \\
& +32F_{12}F_{34}F_{23}F_{14} - 32F_{13}F_{23}F_{24}F_{14} \\
& +8F_{34}^2F_{13}^2 + 8F_{34}^2F_{23}^2 \\
& +8F_{13}^2F_{14}^2 + 24F_{23}^2F_{14}^2 \\
& +24F_{13}^2F_{24}^2 + 8F_{23}^2F_{24}^2 \\
& +8F_{34}^2F_{24}^2 + 8F_{34}^2F_{14}^2 \\
& +8F_{24}^2F_{14}^2\,.
\end{aligned} \tag{B.12}$$

This immediately leads us to the value of $\mathrm{Tr}(\mathsf{A}^4)$ and $\mathrm{Tr}(\mathsf{A}^5)$. It is possible to continue in this fashion to calculate higher-order terms, however these calculations become increasingly more complicated.

Appendix C

Traces of Operators

C.1 Introduction

A large part of the variance reduction method described in this thesis requires the inclusion of traceless operators. These are operators which appear in a hopping parameter expansion and are either explicitly traceless, or can be made traceless if we can find a way of working out the what the trace actually is.

C.2 Explicitly Traceless Operators

Many terms which appear in the hopping parameter expansion are explicitly traceless. For example,

- $\mathsf{D}^{2n+1}, \mathsf{A}^m\mathsf{D}^{2n+1}$
- $\mathsf{D}^2, \mathsf{A}^n\mathsf{D}^2$

where A is a clover term, and $m, n = 1, 2, 3, \dots$.

C.3 Calculable Traces

To begin with we consider the 'D' term which appears in the Dirac-Wilson action

$$\begin{aligned} D(x,y) &= \textstyle\sum_\mu \{U_\mu(x)(1-\gamma_\mu)\delta_{x+\hat{\mu},y} + U^\dagger_\mu(x-\hat{\mu})(1+\gamma_\mu)\delta_{x-\hat{\mu},y}\} \\ D^\dagger(y,x) &= \textstyle\sum_\mu \{U_\mu(y)(1+\gamma_\mu)\delta_{y+\hat{\mu},x} + U^\dagger_\mu(y-\hat{\mu})(1-\gamma_\mu)\delta_{y-\hat{\mu},x}\}\,. \end{aligned} \tag{C.1}$$

For simplicity we write

$$\begin{aligned} \mathsf{D} &= U_1(1-\gamma_\mu) + U^\dagger_1(1+\gamma_\mu) \\ \mathsf{D}^\dagger &= U_1(1+\gamma_\mu) + U^\dagger_1(1-\gamma_\mu)\,. \end{aligned} \tag{C.2}$$

C.3.1 Tr($\mathsf{D}^\dagger\mathsf{D}$)

Using this simplified notation

$$\mathrm{Tr}(\mathsf{D}^\dagger\mathsf{D}) = \mathrm{Tr}((U_1(1-\gamma_\mu) + U^\dagger_1(1+\gamma_\mu))(U_2(1+\gamma_\nu) + U^\dagger_2(1-\gamma_\nu)))\,. \tag{C.3}$$

We receive contributions from diagram picture in Fig. (C.1).

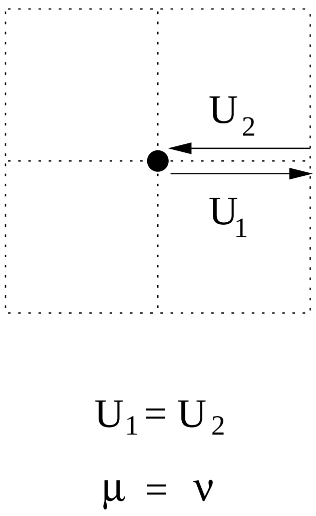


Figure C.1: The only contribution to Tr($\mathsf{D}^\dagger\mathsf{D}$).

So we have

$$\begin{aligned}
\mathrm{Tr}(\mathsf{D}^\dagger\mathsf{D}) &= \mathrm{Tr}(U_1(1+\gamma_\mu)U_1^\dagger(1+\gamma_\mu)+U_1^\dagger(1-\gamma_\mu)U_1(1-\gamma_\mu)) \\
&= \mathrm{Tr}(U_1U_1^\dagger(1+2\gamma_\mu+\gamma_\mu^2)+U_1^\dagger U_1(1-2\gamma_\mu+\gamma_\mu^2)) \\
&= \mathrm{Tr}_{colour}(U_1^\dagger U_1)\mathrm{Tr}_{spin}(2(1+\gamma_\mu)+2(1-\gamma_\mu)) \\
&= \mathrm{Tr}_{colour}(1)\mathrm{Tr}_{spin}(4) \\
&= \underbrace{3}_{colour} \times \underbrace{4\times 4}_{spin} \times \underbrace{4}_{\mu-directions} \\
&= 192\,.
\end{aligned} \tag{C.4}$$

So, over the whole lattice of volume V, we have

$$\mathrm{Tr}(\mathsf{D}\mathsf{D}^\dagger) = 192V\,. \tag{C.5}$$

C.3.2 $\mathbf{Tr}(\mathsf{D}\mathsf{D}^\dagger\mathsf{D}\mathsf{D}^\dagger)$

$$\begin{aligned}
&\mathrm{Tr}(\mathsf{D}\mathsf{D}^\dagger\mathsf{D}\mathsf{D}^\dagger) = \\
&\quad (U_1(1-\gamma_\mu)+U_1^\dagger(1+\gamma_\mu))(U_2(1+\gamma_\nu)+U_2^\dagger(1-\gamma_\nu)) \\
&\quad \times(U_3(1-\gamma_\rho)+U_3^\dagger(1+\gamma_\rho))(U_4(1+\gamma_\sigma)+U_4^\dagger(1-\gamma_\sigma)))\,.
\end{aligned} \tag{C.6}$$

Then we are left with the following non-zero trace terms

$$\begin{aligned}
\mathrm{Tr}(\mathsf{D}\mathsf{D}^\dagger\mathsf{D}\mathsf{D}^\dagger) = \mathrm{Tr}(& \\
&+U_1U_2U_3^\dagger U_4^\dagger(1-\gamma_\mu)(1+\gamma_\nu)(1+\gamma_\rho)(1-\gamma_\sigma) \\
&+U_1U_2^\dagger U_3U_4^\dagger(1-\gamma_\mu)(1-\gamma_\nu)(1-\gamma_\rho)(1-\gamma_\sigma) \\
&+U_1U_2^\dagger U_3^\dagger U_4(1-\gamma_\mu)(1-\gamma_\nu)(1+\gamma_\rho)(1+\gamma_\sigma) \\
&+U_1^\dagger U_2U_3^\dagger U_4(1+\gamma_\mu)(1+\gamma_\nu)(1+\gamma_\rho)(1+\gamma_\sigma) \\
&+U_1^\dagger U_2^\dagger U_3U_4(1+\gamma_\mu)(1-\gamma_\nu)(1-\gamma_\rho)(1+\gamma_\sigma) \\
&+U_1^\dagger U_2U_3U_4^\dagger(1+\gamma_\mu)(1+\gamma_\nu)(1-\gamma_\rho)(1-\gamma_\sigma))\,.
\end{aligned} \tag{C.7}$$

We consider each of these terms individually,

1. $\mathrm{Tr}(U_1U_2U_3^\dagger U_4^\dagger(1-\gamma_\mu)(1+\gamma_\nu)(1+\gamma_\rho)(1-\gamma_\sigma))\,.$

   This corresponds to the following three planar diagrams shown in Fig. (C.2).

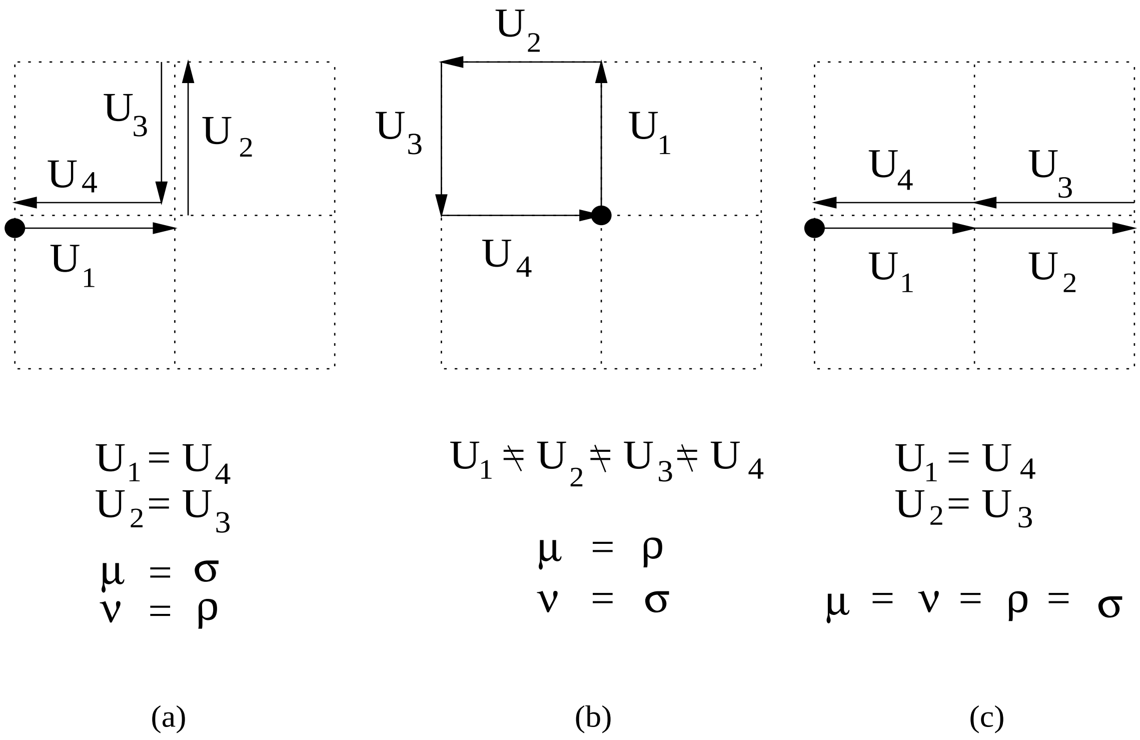


Figure C.2: Three contributions to $\mathrm{Tr}(\mathrm{D}\mathrm{D}^\dagger\mathrm{D}\mathrm{D}^\dagger)$.

Diagram (a)

$$\begin{aligned} &= \mathrm{Tr}(U_1U_2U_2^\dagger U_1^\dagger(1-\gamma_\mu)(1+\gamma_\nu)(1+\gamma_\nu)(1-\gamma_\mu)) \\ &= \mathrm{Tr}_{colour}(1)\mathrm{Tr}_{spin}(0) \\ &= 0\,. \end{aligned} \tag{C.8}$$

Diagram (b)

$$\begin{aligned} &= \mathrm{Tr}(U_1U_2U_3^\dagger U_4^\dagger(1-\gamma_\mu)(1+\gamma_\nu)(1+\gamma_\mu)(1-\gamma_\nu)) \\ &= \mathrm{Tr}_{colour}(U_1U_2U_3^\dagger U_4^\dagger)\mathrm{Tr}_{spin}(-2) \\ &= \underbrace{3\langle P\,\rangle}_{colour} \quad\times\quad \underbrace{-2\times 4}_{spin} \quad\times\quad \underbrace{8}_{\mu-dircetions} \quad\times\quad \underbrace{6}_{\nu-dircetions} \\ &= -96\times 12\langle p\rangle\,. \end{aligned} \tag{C.9}$$

Diagram (c)

$$\begin{aligned} &= \mathrm{Tr}(U_1U_2U_2^\dagger U_1^\dagger(1-\gamma_\mu)(1+\gamma_\mu)(1+\gamma_\mu)(1-\gamma_\mu)) \\ &= \mathrm{Tr}_{colour}(1)\mathrm{Tr}_{spin}(0) \\ &= 0\,. \end{aligned} \tag{C.10}$$

2. $\mathrm{Tr}(U_1 U_2^\dagger U_3 U_4^\dagger (1-\gamma_\mu)(1-\gamma_\nu)(1-\gamma_\rho)(1-\gamma_\sigma))$.

This gives rise to two planar diagrams shown in Fig. (C.3).

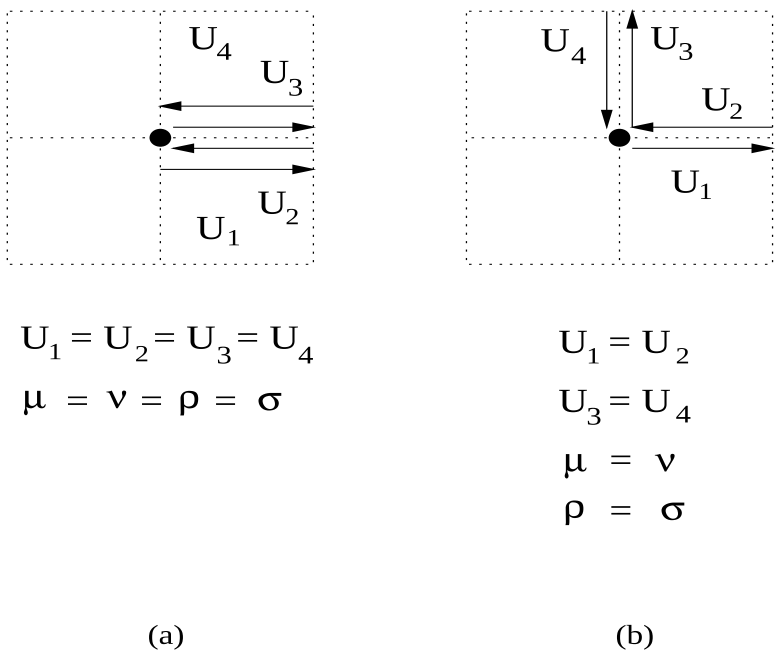


Figure C.3: Two contributions to $\mathrm{Tr}(\mathsf{D}\mathsf{D}^\dagger\mathsf{D}\mathsf{D}^\dagger)$.

Diagram (a)

$$\begin{aligned}
&= \mathrm{Tr}(U_1 U_1^\dagger U_1 U_1^\dagger (1-\gamma_\mu)(1-\gamma_\mu)(1-\gamma_\mu)(1-\gamma_\mu)) \\
&= \mathrm{Tr}_{colour}(1)\mathrm{Tr}_{spin}(8) \\
&= \underbrace{3\times 1}_{colour} \quad\times\quad \underbrace{4\times 8}_{spin} \quad\times\quad \underbrace{8}_{\mu-dircetions} \\
&= 12\times 64\,.
\end{aligned} \tag{C.11}$$

Diagram (b)

$$\begin{aligned}
&= \mathrm{Tr}(U_1 U_1^\dagger U_3 U_3^\dagger (1-\gamma_\mu)(1-\gamma_\mu)(1-\gamma_\rho)(1-\gamma_\rho)) \\
&= \mathrm{Tr}_{colour}(1)\mathrm{Tr}_{spin}(8) \\
&= \underbrace{3\times 1}_{colour} \quad\times\quad \underbrace{4\times 8}_{spin} \quad\times\quad \underbrace{8}_{\mu-dircetions} \quad\times\quad \underbrace{6}_{\rho-dircetions} \\
&= 12\times 384\,.
\end{aligned} \tag{C.12}$$

3.

$$\mathrm{Tr}(U_1 U_2^\dagger U_3^\dagger U_4 (1-\gamma_\mu)(1-\gamma_\nu)(1+\gamma_\rho)(1+\gamma_\sigma))\,. \tag{C.13}$$

This gives rise to two planar diagrams shown in Fig. (C.4).

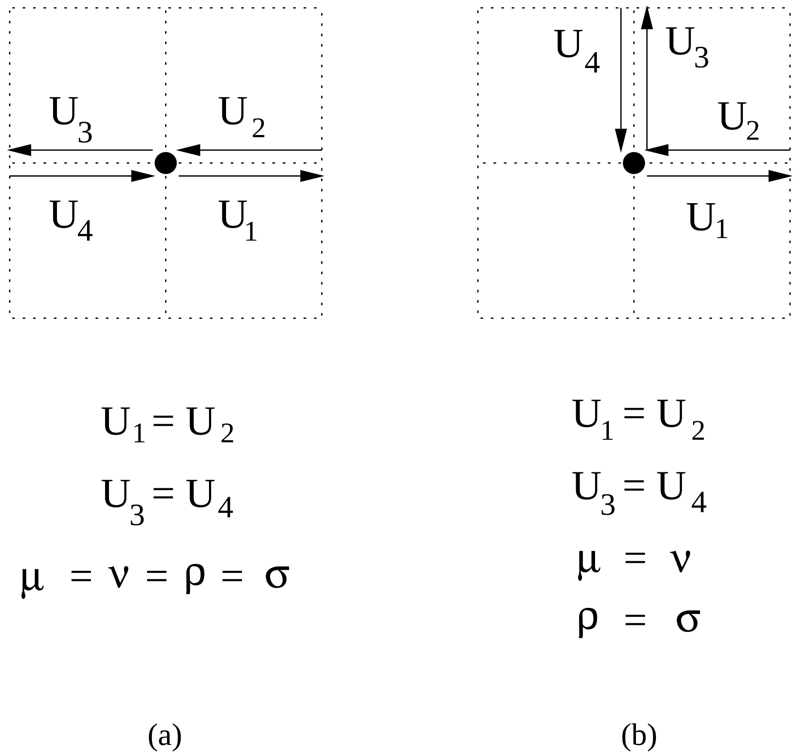


Figure C.4: Two contributions to $\mathrm{Tr}(\mathsf{D}\mathsf{D}^\dagger\mathsf{D}\mathsf{D}^\dagger)$.

Diagram (a)

$$\begin{aligned} &= \mathrm{Tr}(U_1 U_2 U_2^\dagger U_1^\dagger (1-\gamma_\mu)(1+\gamma_\mu)(1+\gamma_\mu)(1-\gamma_\mu) \\ &= 0\,. \end{aligned} \tag{C.14}$$

Diagram (b)

$$\begin{aligned} &= \mathrm{Tr}(U_1 U_1 U_3^\dagger U_3^\dagger (1-\gamma_\mu)(1+\gamma_\mu)(1+\gamma_\rho)(1-\gamma_\rho) \\ &= 0\,. \end{aligned} \tag{C.15}$$

4. $\mathrm{Tr}(U_1^\dagger U_2 U_3^\dagger U_4 (1+\gamma_\mu)(1+\gamma_\nu)(1+\gamma_\rho)(1+\gamma_\sigma))\,.$

   By symmetry this gives the same as term 2.

5. $\mathrm{Tr}(U_1^\dagger U_2^\dagger U_3 U_4 (1+\gamma_\mu)(1-\gamma_\nu)(1-\gamma_\rho)(1+\gamma_\sigma))\,.$

   By symmetry this gives the same as term 1.

6. $\mathrm{Tr}(U_1^\dagger U_2 U_3 U_4^\dagger (1+\gamma_\mu)(1+\gamma_\nu)(1-\gamma_\rho)(1-\gamma_\sigma))\,.$

By symmetry this gives the same as term 3.

So finally, by summing all of these terms, over a lattice with volume V, we have

$$\mathrm{Tr}(\mathsf{D}\mathsf{D}^\dagger\mathsf{D}\mathsf{D}^\dagger) = 12V(448 - 96\langle P\rangle)\,. \tag{C.16}$$

C.3.3 Tr($\mathsf{D}\mathsf{D}\mathsf{D}^\dagger\mathsf{D}^\dagger$)

Using similar techniques, we proceed as above

$$\begin{aligned}
&\mathrm{Tr}(\mathsf{D}\mathsf{D}\mathsf{D}^\dagger\mathsf{D}^\dagger) = \\
&\quad (U_1(1-\gamma_\mu) + U_1^\dagger(1+\gamma_\mu))(U_2(1-\gamma_\nu) + U_2^\dagger(1+\gamma_\nu)) \\
&\qquad \times(U_3(1+\gamma_\rho) + U_3^\dagger(1-\gamma_\rho))(U_4(1+\gamma_\sigma) + U_4^\dagger(1-\gamma_\sigma)))\,.
\end{aligned} \tag{C.17}$$

We have the following non-zero terms

$$\begin{aligned}
\mathrm{Tr}(\mathsf{D}\mathsf{D}\mathsf{D}^\dagger\mathsf{D}^\dagger) \;=\; \mathrm{Tr}(& \\
& U_1U_2U_3^\dagger U_4^\dagger(1-\gamma_\mu)(1-\gamma_\nu)(1-\gamma_\rho)(1-\gamma_\sigma) \\
& +U_1U_2^\dagger U_3U_4^\dagger(1-\gamma_\mu)(1+\gamma_\nu)(1+\gamma_\rho)(1-\gamma_\sigma) \\
& +U_1U_2^\dagger U_3^\dagger U_4(1-\gamma_\mu)(1+\gamma_\nu)(1-\gamma_\rho)(1+\gamma_\sigma) \\
& +U_1^\dagger U_2U_3U_4^\dagger(1+\gamma_\mu)(1-\gamma_\nu)(1+\gamma_\rho)(1-\gamma_\sigma) \\
& +U_1^\dagger U_2U_3^\dagger U_4(1+\gamma_\mu)(1-\gamma_\nu)(1-\gamma_\rho)(1+\gamma_\sigma) \\
& +U_1^\dagger U_2^\dagger U_3U_4(1+\gamma_\mu)(1+\gamma_\nu)(1+\gamma_\rho)(1+\gamma_\sigma)\,.
\end{aligned} \tag{C.18}$$

Again, we consider each term individually and draw the corresponding diagrams. This does not present any new diagrams which have not been dealt with in the case of $\mathsf{D}\mathsf{D}^\dagger\mathsf{D}\mathsf{D}^\dagger$.

We find the result

$$\mathrm{Tr}(\mathsf{D}\mathsf{D}\mathsf{D}^\dagger\mathsf{D}^\dagger) = 12V(256 + 96\langle P\rangle)\,. \tag{C.19}$$

C.3.4 Tr($\mathsf{D}\mathsf{D}\mathsf{D}\mathsf{D}$)

A similar procedure yields the result

$$\mathrm{Tr}(\mathsf{D}\mathsf{D}\mathsf{D}\mathsf{D}) = 12V(-96\langle P\rangle)\,. \tag{C.20}$$

C.3.5 Higher order traces

We can follow through to generalise these result to operators involving higher orders of D *e. g.* D^6. Such operators require non-planar diagrams and are more difficult to calculate. We have found that the inclusion of such operators in a subtraction scheme, such as the one described in this thesis yield little benefit in terms of variance reduction. Consequently, we do not discuss such operators here.

C.3.6 Related traces

It is possible to use the above trace calculations, together with well-known trace rules to calculate the trace values for other related quantities which are relevant to our studies. For example,

$$\mathrm{Tr}(\mathsf{D}\mathsf{A}^{-1}\mathsf{D}) = \mathrm{Tr}(\mathsf{A}^{-1}\mathsf{D}\mathsf{D}) = 0\,. \tag{C.21}$$

C.4 Summary

Table C.1: Summary of operator traces.

Operator	Value
$\mathrm{Tr}(\mathsf{D}\mathsf{D}\mathsf{D}^\dagger\mathsf{D}^\dagger)$	$12V(256 + 96\langle P\rangle)$
$\mathrm{Tr}(\mathsf{D}\mathsf{D}^\dagger)$	$192V$
$\mathrm{Tr}(\mathsf{D}^4)$	$12V(-96\langle P\rangle)$
$\mathrm{Tr}(\mathsf{D}^\dagger\mathsf{D}\mathsf{D}^\dagger\mathsf{D})$	$12V(448 - 96\langle P\rangle)$

Table C.1 gives a summary of the traces calculated in this appendix.

Appendix D

Hermitian Representations of Fermion Matrices

D.1 The Eigenvalues of Q^2 for $\mathsf{Q} = \gamma_5 \mathsf{M}$ in the Free Fermion Case

We find the eigenvalues of Q, via $\mathsf{Q}^2 = \mathsf{M}^\dagger \mathsf{M}$, noting that $\lambda_{\mathsf{Q}} = \pm\sqrt{\lambda_{\mathsf{Q}^2}}$. So we are able to determine the eigenvalues of Q, but not the sign of these eigenvalues. However, this is not a problem here as we only need to find the absolute value of each eigenvalue of Q, in order to then relate this eigenvalue back to that of Q^2. For convenience, we set $a = 1$.

$$\mathsf{M} = \mathsf{I} - \kappa \mathsf{D}\,. \tag{D.1}$$

So

$$\begin{aligned} \mathsf{Q}^2 &= (\gamma_5 \mathsf{M})^2 = \mathsf{I} - \gamma_5 \kappa \mathsf{D} \gamma_5 - \kappa \mathsf{D} + \kappa^2 \gamma_5 \mathsf{D} \gamma_5 \mathsf{D} \\ &= \mathsf{I} - \kappa(\mathsf{D}^\dagger + \mathsf{D}) + \kappa^2 \mathsf{D}^\dagger \mathsf{D}\,. \end{aligned} \tag{D.2}$$

By taking a Fourier transform to momentum space,

$$\begin{aligned} \mathsf{D}_{xy} &= -\tfrac{2i}{L^2} \textstyle\sum_P e^{-iP(y-x)} \sum_\mu \gamma_\mu \sin P_\mu + i \cos P_\mu \\ \mathsf{D}^\dagger_{xy} &= \tfrac{2i}{L^2} \textstyle\sum_P e^{-iP(y-x)} \sum_\mu \gamma_\mu \sin P_\mu - i \cos P_\mu\,, \end{aligned} \tag{D.3}$$

therefore

$$\mathsf{D}_{xy} + \mathsf{D}^\dagger_{xy} = \frac{4}{L^2} \sum_P e^{-iP(y-x)} \sum_\mu \cos P_\mu \tag{D.4}$$

and

$$\mathsf{D}_{xy}\mathsf{D}^{\dagger}_{xy} = \frac{4}{L^2}\sum_{P} e^{-iP(y-x)}(\sum_{\mu}\sin^2 P_\mu + (\sum_{\mu}\cos P_\mu)^2)\,. \tag{D.5}$$

So we can write

$$\lambda_{\mathsf{Q}^2} = 4\kappa^2((\frac{1}{2\kappa} - \sum_{\mu}\cos P_\mu)^2 + \sum_{\mu}\sin^2 P_\mu)\,, \tag{D.6}$$

and therefore

$$\lambda_{\mathsf{Q}} = \pm 2\kappa\sqrt{((\frac{1}{2\kappa} - \sum_{\mu}\cos P_\mu)^2 + \sum_{\mu}\sin^2 P_\mu)}\,. \tag{D.7}$$

D.2 Eigenvalues of $\mathsf{Q}^2 = \mathsf{M}^{\dagger}\mathsf{M}$ in the Continuum Limit

We need to calculate $(\frac{1}{2\kappa}\gamma_5\mathsf{M})^2$ is as $a \to 0$. So we replace a in the equation for λ_{Q^2}.

$$\begin{aligned}
\lambda_{\mathsf{Q}^2} &= a^6\{(\tfrac{1}{2\kappa} - \textstyle\sum_\mu \cos(aP_\mu))^2 + \sum_\mu \sin^2(aP_\mu)\} \\
\lambda_{\mathsf{Q}^2} &= a^6\{(ma+4)^2 - (2ma+8)\textstyle\sum_\mu \cos(aP_\mu) + (\sum_\mu \cos(aP_\mu))^2 + \sum \sin^2(aP_\mu)\}\,.
\end{aligned} \tag{D.8}$$

In the continuum limit (for $\mathsf{M}^{\dagger}\mathsf{M}$)

$$\lim_{a\to 0}(\cos(aP_\mu)) = 1 \qquad \lim_{a\to 0}(\frac{\sin(aP_\mu)}{a}) = P_\mu\,. \tag{D.9}$$

Therefore we can write

$$\lambda_{\mathsf{Q}^2} \approx a^8\{m^2 + P_\mu^2\}\,. \tag{D.10}$$

D.3 The Eigenvalues of $\mathsf{M}^{\dagger}\mathsf{M}$ in the Freefield Case where Red-Black Pre-conditioning Has Been Applied

So we have the red-black pre-conditioned matrix $\mathsf{M}^{(e)}$, defined as

$$\mathsf{M}^{(e)} = \mathsf{I} - \kappa^2\mathsf{D}^{eo}\mathsf{D}^{oe}, \qquad \mathsf{M}^{(e)\dagger} = \mathsf{I} - \kappa^2\mathsf{D}^{eo\dagger}\mathsf{D}^{oe\dagger} \tag{D.11}$$

and therefore

$$\mathsf{M}^{\dagger}\mathsf{M} = \mathsf{M}^{(e)\dagger}\mathsf{M}^{(e)} = \mathsf{I} - \kappa^2(\mathsf{D}^{\dagger}\mathsf{D}^{\dagger} + \mathsf{D}\mathsf{D}) + \kappa^4\mathsf{D}^{\dagger}\mathsf{D}^{\dagger}\mathsf{D}\mathsf{D} \tag{D.12}$$

$$\mathsf{D}^\dagger\mathsf{D}^\dagger\mathsf{D}\mathsf{D} \Rightarrow 16(\sum_\mu \sin^2 P_\mu + (\sum_\mu \cos P_\mu)^2)^2 \,. \tag{D.13}$$

From before

$$\mathsf{D}^\dagger\mathsf{D}^\dagger + \mathsf{D}\mathsf{D} \Rightarrow 8\cos P_\mu \,. \tag{D.14}$$

So we have

$$\lambda = 1 + 8\kappa^2(\sum \sin^2 P_\mu - (\sum \cos P_\mu)^2) + 16\kappa^4(\sum \sin^2 P_\mu + (\sum \cos P_\mu)^2)^2 \,. \tag{D.15}$$

D.3.1 Eigenvalues of $\mathsf{M}^\dagger\mathsf{M}$ (red-black pre-conditioned) in the continuum limit

We need to calculate the eigenvalues of $(\frac{1}{2\kappa})^2\mathsf{M}^\dagger\mathsf{M}$ in the continuum limit. As before we replace a

$$\lambda = a^6\frac{1}{4\kappa^2} + 2(a^2P_\mu^2 - 16) + 4\kappa^2(a^2P_\mu + 16)^2 \,. \tag{D.16}$$

Using a Taylor expansion of κ in a

$$\kappa^2 = \frac{1}{(2ma+8)^2} \approx \frac{1}{64} - \frac{1}{128}ma + \frac{3}{1024}m^2a^2 \,, \tag{D.17}$$

we have

$$\lambda \approx 4a^8\{m^2 + P_\mu^2\} \,. \tag{D.18}$$

D.4 Lattice Units

In order to convert from lattice units to physical units, we need to set the lattice scale. We start by setting $\hbar c = 1$

$$\hbar c = 1 = 197.3 \text{ MeV fm} \,. \tag{D.19}$$

For example, if we have a lattice with spacing $a = 0.1$ fm, then we can write

$$a = 0.1 \text{ fm} = \frac{0.1}{197.3} \text{ MeV}^{-1} = 0.5068 \text{ GeV}^{-1} \,. \tag{D.20}$$

Thus we are able to relate our lattice units to an pseudo-energy scale.

Bibliography

[1] UKQCD, E. Cahill, A. Irving, C. Johnson, and J. Sexton, Nucl. Phys. Proc. Suppl. **83**, 825 (2000), hep-lat/9909131.

[2] E. Cahill, J. C. Sexton, A. C. Irving, and C. A. Johnson, Calculating lattice fermion determinants by quadrature, LTH/517.

[3] D. Griffiths, *Introduction to Elementary Particles* (John Wiley and Sons, Inc., 1987).

[4] Cheng and Li, *Gauge theory of elementary particle physics* (Clarendon Press, 1984).

[5] F. Halzen and A. D. Martin, *Quarks and Leptons* (John Wiley and Sons, Inc., 1984).

[6] S. Weinberg, Phys. Rev. Lett. **19**, 1264 (1967).

[7] S. Glashow, J. Iliopoulos, and L. Maiani, Phys.Rev. **D2**, 1285 (1970).

[8] H. Yukawa, Proc. Phys. Math Society Japan **17,38** (1935).

[9] M. Conversi, E. Pancini, and O. Piccioni, Phys. Rev. **71**, 209 (1947).

[10] M. Gell-Mann and Y. Ne'eman, *The Eightfold Way* (New York: Benjamin, 1964).

[11] D.Balin and A.Love, *Introduction to Gauge Theory* (Institute of Physics, 1986).

[12] I. Montvay and G. Munster, *Quantum Fields on a Lattice* (Cambridge University Press, 1994).

[13] H. J. Roethe, *Lattice Gauge Theories* (World Scientific, 1992).

[14] M. Creutz, *Quantum Fields on the Computer* (World Scientific, 1992).

[15] M. Creutz, *Quarks, Gluons and Lattices* (Cambridge University Press, 1983).

[16] R. Gupta, (1997), hep-lat/9807028.

[17] H. Wittig, (1999), hep-ph/9911400.

[18] D. G. Richards, (1999), nucl-th/0006020.

[19] M. Di Pierro, (2000), hep-lat/0009001.

[20] P. H. Ginsparg and K. G. Wilson, Phys. Rev. **D25**, 2649 (1982).

[21] T. Blum and A. Soni, (1997), arXiv:hep-lat/9712004.

[22] R. Narayanan and H. Neuberger, Phys. Rev. Lett. **71**, 3251 (1993), arXiv:hep-lat/9308011.

[23] B. Sheikholeslami and R. Wohlert, Nucl. Phys. **B259**, 572 (1985).

[24] ALPHA, K. Jansen and R. Sommer, Nucl. Phys. **B530**, 185 (1998), hep-lat/9803017.

[25] UKQCD, C. R. Allton *et al.*, Phys. Rev. **D60**, 034507 (1999), hep-lat/9808016.

[26] UKQCD, C. R. Allton *et al.*, Phys. Rev. **D65**, 054502 (2002), hep-lat/0107021.

[27] UKQCD, C. Allton, Nucl. Phys. Proc. Suppl. **106**, 212 (2002), hep-lat/0111011.

[28] A. Duncan, E. Eichten, and H. Thacker, Phys. Rev. **D63**, 111501 (2001), hep-lat/0011076.

[29] D. P. Landau and K. Binder, *A Guide to Simulations in Statistical Physics* (Cambridge University Press, 2000).

[30] N. Metropolis, A. Rosenbluth, A. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953).

[31] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, Phys. Lett. **B195**, 216 (1987).

[32] D. Chen *et al.*, (2000), hep-lat/0011004.

[33] S. Gottlieb, Nucl. Phys. Proc. Suppl. **94**, 833 (2001), hep-lat/0011071.

[34] X.-Q. Luo *et al.*, Nucl. Phys. Proc. Suppl. **106**, 1046 (2002), hep-lat/0202007.

[35] C. Lanczos, J. Res. Nat. Bur. Standards **B45**, 225 (1950).

[36] B. N. Parlett, *The Symmetric Eigenvalue Problem* (Prentice-Hall/SIAM, Englewood Cliffs, NJ, 1980).

[37] Z. Bai, M. Fahey, and G. Golub, Some large scale matrix computation problems, 1995, SCCM-95-09 (School of Engineering, Stanford University).

[38] J. Cullum and R. Willoughby, J. Comp. Phys. **44**, 329 (1981).

[39] T. Kalkreuter, Comput. Phys. Commun. **95**, 1 (1996), hep-lat/9509071.

[40] X.-Q. Luo, Mod. Phys. Lett. **A16**, 1615 (2001), hep-lat/0107013.

[41] J. Cullum and R. Willoughby, Sparse Matrix Proceedings , 220 (1978).

[42] P. Davis and P. Rabinowitz, *Methods of Numerical Integration* (Academic Press, 1975).

[43] L. Fox and I. B. Parker, *Chebyshev Polynomials in Numerical Analysis* (Oxford University Press, 1968).

[44] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes* (Cambridge University Press, 1992).

[45] G. Szego, *Orthogonal Polynomials* (American Mathematical Society, 1939).

[46] E. Cahill, *Krylov Subspace Methods and the Calculation of Matrix Determinants*, PhD thesis, School of Mathematics Trinity College University of Dublin, 1999.

[47] S.-J. Dong and K.-F. Liu, Phys. Lett. **B328**, 130 (1994), hep-lat/9308015.

[48] G. Bhanot and A. D. Kennedy, Phys. Lett. **B157**, 70 (1985).

[49] L. Lin, K. F. Liu, and J. H. Sloan, Phys. Rev. **D61**, 074505 (2000), hep-lat/9905033.

[50] C. Thron, S. J. Dong, K. F. Liu, and H. P. Ying, Phys. Rev. **D57**, 1642 (1998), hep-lat/9707001.

[51] A. Duncan, E. Eichten, and H. Thacker, Phys. Rev. **D59**, 014505 (1999), hep-lat/9806020.

[52] A. Duncan, E. Eichten, and H. Thacker, Nucl. Phys. Proc. Suppl. **83**, 449 (2000), hep-lat/9909123.

[53] A. Duncan, E. Eichten, and H. Thacker, Nucl. Phys. Proc. Suppl. **83**, 795 (2000), hep-lat/9908057.

[54] UKQCD, A. C. Irving *et al.*, Phys. Rev. **D58**, 114504 (1998), hep-lat/9807015.

[55] A. Duncan, E. Eichten, R. Roskies, and H. Thacker, Phys. Rev. **D60**, 054505 (1999), hep-lat/9902015.

[56] A. C. Irving, J. C. Sexton, and E. Cahill, Nucl. Phys. Proc. Suppl. **63**, 967 (1998), hep-lat/9708004.