### ABSTRACT

Deflation Methods in Lattice QCD Travis Whyte, Ph.D. Advisor: Walter Wilcox, Ph.D.

The inversion of the Dirac operator is a necessary feature of calculating physical observables within lattice QCD. The calculation of fermionic forces within hybrid Monte Carlo and the formation of quark propagators are two examples where such an inversion is needed. The many discretizations of the Dirac operator pose an algorithmic and computational challenge due to their size and their eigenspectra. As the quark mass approaches its physical value, the low lying eigenspectra of the Dirac operator approaches zero. From this arises the phenomena of critical slowing down, where the number of iterations to obtain an approximate solution for an iterative solver increases as a power law. Deflation and multigrid are two techniques that combat the effects of critical slowing down. We present a deflated multigrid preconditioner of FGMRES for the Wilson-Dirac operator in the lattice Schwinger model. Our method of deflation within the preconditioner demonstrates a remarkable reduction in cost for the inversion of the Wilson-Dirac operator, and also displays very mild scaling with respect to lattice size.

The calculation of physical quantities arising from disconnected quark loops is one of the largest challenges in lattice QCD. A direct approach is to calculate the propagator for all lattice sites to all lattice sites. For large lattices, this approach is intractable so stochastic methods are used. The physical signal must be extracted from the noise created by these methods, and thus noise subtraction techniques are mandatory. We present deflation based noise subtraction techniques for the scalar, local vector and non-local vector operators in the quenched approximation at zero quark mass and with the inclusion of dynamical sea quarks at larger than physical pion mass. In both cases, the deflation based methods show dramatic reduction in the variance of these noisy calculations. Deflation Methods in Lattice QCD

by

Travis Whyte, B.S.

A Dissertation

Approved by the Department of Physics

Dwight P. Russell, Ph.D., Interim Chairperson

Submitted to the Graduate Faculty of Baylor University in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

Approved by the Dissertation Committee

Walter Wilcox, Ph.D., Chairperson

Ronald B. Morgan, Ph.D.

Gerald Cleaver, Ph.D.

Anzhong Wang, Ph.D.

Wickramasinghe Ariyasinghe, Ph.D.

Accepted by the Graduate School August 2020

J. Larry Lyon, Ph.D., Dean

Page bearing signatures is kept on file in the Graduate School.

Copyright © 2020 by Travis Whyte All rights reserved

# TABLE OF CONTENTS

LIST OF FIGURES LIST OF TABLES PREFACE	vii x xi
ACKNOWLEDGMENTS	xi
DEDICATION	xiii
CHAPTER ONE Lattice Quantum Chromodynamics	1 1 2
1.3 The Need for Dirac Operator Inversions1.4 The Dirac Operator in Lattice QCD	$\frac{4}{7}$
CHAPTER TWOKrylov Subspace Methods2.1Krylov Subspaces2.2GMRES and FGMRES2.3Deflated Krylov Subspace Methods	11 11 12 18
CHAPTER THREEMultigrid in Lattice QCD3.1 Introduction3.2 Classical Multigrid3.3 Adaptive Multigrid3.4 Multigrid for the Lattice Wilson-Dirac Operator3.5 Deflated GMRES in Multigrid for Lattice QCD	29 29 30 33 35 43
CHAPTER FOUR         Deflation Methods for Noise Subtraction in Disconnected Diagrams         4.1 Introduction         4.2 Disconnected Diagrams         4.3 Stochastic Trace Estimators         4.4 Noise Subtraction Techniques         4.5 Variance Reduction at Zero Quark Mass         4.6 Variance Reduction with Dynamical Configurations	60 60 63 64 66 72
CHAPTER FIVE Conclusions	77 77

<ul><li>5.2 Variance Reduction at Zero Quark Mass</li><li>5.3 Acknowledgments</li></ul>	78 79
APPENDIX A Deflated GMRES Multigrid Preconditioned FGMRES Algorithm	81
APPENDIX B	
Matlab Program for Coarsening the Wilson-Dirac Operator	83
B.1 Near Null Vector Generation	83
B.2 Chiral Doubling of the Near Null Vectors	84
B.3 Partitioning the Near Null Vectors	86
B.4 Forming the Coarsened Operator	87
BIBLIOGRAPHY	91

# LIST OF FIGURES

Figure 1.1.	The running of the strong coupling constant, $\alpha_s$	2
Figure 1.2.	A three dimensional representation of a lattice used in lattice QCD.	3
Figure 1.3.	The eigenspectra for the Wilson-Dirac operator, originating from a lattice of size $4^4$ , with $\beta = 6.0$ and $m = -1.0$	8
Figure 1.4.	The eigenspectra for the hermitian Wilson-Dirac operator $D_H$	9
Figure 2.1.	A comparison of GMRES with restarting and no restarting for the Wilson-Dirac operator for a 8 <sup>4</sup> lattice at $\kappa_{crit} \approx 0.1570$	15
Figure 2.2.	The GMRES convergence bound as a function of the number of deflated eigenvalues.	20
Figure 2.3.	A comparison of GMRES with deflated restarting and only restarting for the Wilson-Dirac operator for a 8 <sup>4</sup> lattice at $\kappa_{crit} \approx 0.1570$ . A restart length of $m = 20$ was used, and $k = 10$ eigenvectors were deflated from the system	23
Figure 2.4.	The convergence history for the Wilson-Dirac operator at $\kappa_{crit} \approx 0.1570$ for two $Z_4$ noise vectors	28
Figure 3.1.	(Left to right) The multigrid V-cycle, W-cycle, and FMG cycle for a three grid hierarchy.	32
Figure 3.2.	A comparison of multigrid with Gauss-Seidel smoothing and Gauss-Seidel on one grid	33
Figure 3.3.	An example of the blocking that occurs for a $4^2$ isotropic lattice (left) with a $2^2$ blocking to form a $2^2$ coarse lattice (right)	38
Figure 3.4.	The sparsity pattern for the coarsened Wilson-Dirac operator (left) and the fine Wilson-Dirac operator (right)	40
Figure 3.5.	The spectra for both the fine Wilson-Dirac operator and a coarsened Wilson-Dirac operator	40
Figure 3.6.	The absolute values of the components of the 1st eigenvector for the coarse (top) and fine (bottom) Wilson-Dirac operators on a lattice site by site basis.	42
Figure 3.7.	The local co-linearity for a $4^2$ coarsened lattice, $\beta = 6.0$ , and $m = -0.0675$	43

Figure 3.8.	(Top) The number of fine Dirac operator applications as a function of the mass gap for all three lattice sizes. (Bottom) The average number of coarse Dirac operator applications per outer iteration	49
Figure 3.9.	A comparison of fine equivalent Mvps as a function of the mass gap for CGNE, GMRES-DR, MG and deflated MG	50
Figure 3.10.	A comparison of fine equivalent Mvps for multiple right hand sides for GMRES-DR/GMRES-Proj, MG and deflated MG at critical mass.	51
Figure 3.11.	(Top) The number of fine Dirac operator applications as a function of mass gap for both deflated and non deflated solves with the inclusion of a partial intermediate level solve. (Bottom) The average number of intermediate operator applications as a function of mass gap.	53
Figure 3.12.	The average number of coarse Dirac operator applications as a function of mass gap for both deflated and non deflated solves with the inclusion of a partial intermediate level solve	54
Figure 3.13.	The convergence history and individual grid cost for a representative Wilson-Dirac operator from the $64^2$ lattice	55
Figure 3.14.	The convergence history and individual grid cost for a representative Wilson-Dirac operator from the $128^2$ lattice	56
Figure 3.15.	The convergence history and individual grid cost for a representative Wilson-Dirac operator from the $256^2$ lattice	57
Figure 3.16.	A comparison of the convergence history for the deflated coarse grid solver for all lattice sizes used for representative operators	57
Figure 3.17.	Fine equivalent Mvps as a function of lattice size.	59
Figure 4.1.	(Left) A connected piece of a meson correlator from lattice site $m$ to $n$ . (Right) A disconnected piece of a meson correlator from lattice sites $m$ and $n$ .	62
Figure 4.2.	The performance of the MINRES and GMRES algorithms at $\kappa_{crit}$ on the hermitian and nonhermitian systems	68
Figure 4.3.	The standard error for the local vector operator	69
Figure 4.4.	The standard error for the point-split operator	70
Figure 4.5.	The standard error for the scalar operator.	70

Figure 4.6.	The mass of the pion in dimensionless units as a function of the hopping parameter.	73
Figure 4.7.	The standard error for the local vector operator	74
Figure 4.8.	The standard error for the point-split operator	74
Figure 4.9.	The standard error for the scalar operator	75

# LIST OF TABLES

Table 3.1.	The setup and solver parameters for our deflated multigrid solve without a partial intermediate solve	46
Table 3.2.	The setup and solver parameters for our deflated multigrid solve with a partial intermediate solve	46
Table 3.3.	Approximate powers in $Mvps = (size)^{\alpha}$	58
Table 4.1.	Names of calculated operators, $\Theta$ , and their field representations	67
Table 4.2.	The legend on each subtraction plot and their representation	69
Table 4.3.	Comparison of relative efficiencies for operators with quenched configurations.	72
Table 4.4.	Comparison of relative efficiencies for operators with the dynamical configurations	75

## PREFACE

The overarching theme within this dissertation is that of deflation methods within lattice QCD. The topic of deflation arises in two different contexts: solving systems of linear equations arising from the Wilson-Dirac operator using deflated multigrid, and for variance reduction techniques necessary in the stochastic estimation of traces from disconnected diagrams arising in QCD. Both of these areas involve very specific methods arising from numerical linear algebra, and as such, it is necessary to explain the various numerical methods used within each of these areas.

The structure of this dissertation is as follows: first, I discuss lattice QCD and the importance of inverting the Dirac operator. I then discuss the numerical methods used for solving systems of linear equations and eigenvalue problems; namely, Krylov subspace methods. I then discuss multigrid methods, starting from classical multigrid and ending with deflated multigrid for lattice QCD. Finally, I will discuss our deflation techniques that we have implemented for variance reduction in the stochastic estimation of traces arising from disconnected diagrams in lattice QCD.

#### ACKNOWLEDGMENTS

Thank you to my advisors Dr. Walter Wilcox and Dr. Ron Morgan, who patiently answered all of my questions and offered endless support in my research and studies. Your guidance and encouragement was instrumental in my success within this program.

I would like to thank Dr. Cleaver, Dr. Wang and Dr. Ariyasinghe for serving on my committee. I thank all of the professors whose courses I took over the course of my time at Baylor and their investment in my education.

To my parents, Steven Whyte and Dr. Judee Tippett-Whyte, I thank you for your boundless love and support for which I am eternally grateful. You pushed me to be the very best person I could possibly be and gave me every opportunity you could provide. I could not ask for two better examples to guide my way through life.

Thank you Dr. Emily Jarvis, Dr. David Moffet, Dr. Dave Berube and Dr. Kieran Holland for offering amazing advice and support as I made the transition from biochemistry to physics. Your encouragement gave me the courage to take the leap!

And to all my other friends and family that supported and encouraged me along the way, thank you for everything!

## DEDICATION

To my wife, Giselle.

You are my inspiration and my motivation

## CHAPTER ONE

### Lattice Quantum Chromodynamics

## 1.1 Quantum Chromodynamics

Quantum Chromodynamics (QCD) is the the quantum field theory that describes the strong interaction between quarks and gluons, arising from the SU(3) gauge group. The dynamics between quarks and gluons are described by the Lagrangian density:

$$\mathscr{L}_{QCD} = \bar{\psi}_{q}^{i}(i\gamma_{\mu})(D_{\mu})_{ij}\psi_{q}^{j} - m_{q}\bar{\psi}_{q}^{i}\psi_{qi} - \frac{1}{4}F^{a}_{\mu\nu}F^{a\mu\nu}, \qquad (1.1)$$

where the fields  $\psi$  are  $4 \otimes 3$  valued spinors describing the color-spin structure of the quarks. The index i, j runs over the color index of the quarks, and the index q over the possible flavors, with  $m_q$  being the mass parameter of the theory. The gluon field strength tensor,  $F^a_{\mu\nu}$ , depends on the gluon fields,  $A^a_{\mu\nu}$ , where the index a is the color index,  $a \in [1, ..., 8]$ . The covariant derivative,  $D_{\mu}$ , is given by

$$(D_{\mu})_{ij} = \delta_{ij}\partial_{\mu} - ig_s t^a_{ij} A^a_{\mu}, \qquad (1.2)$$

where  $g_s$  is a constant related to the *strong coupling* constant,  $g_s = 4\pi\alpha_s$ , and the matrix  $t^a_{ij}$  are proportional to the Gell-Mann matrices,  $\lambda^a_{ij}$ , the eight generators of the SU(3) gauge group. QCD has experienced enormous theoretical and experimental success, particularly in the high energy regime where the strong coupling constant can be accurately approximated by perturbation theory. However, at low energy scales the value of strong coupling constant does not lend itself to perturbative expansion. As seen in Fig. 1.1 [52], the strong coupling constant increases as the value energy scale



Figure 1.1: The running of the strong coupling constant,  $\alpha_s$ 

is decreased, giving rise to the confinement of quarks. The perturbative approach, which is highly successful in Quantum Electrodynamics, is then no longer a valid approximation. As such, a different approach is necessary to explore QCD in this energy regime, particularly when examining color confinement. This was the motivation for Wilson's path integral formulation of QCD [59], and the inception of lattice QCD.

## 1.2 Lattice QCD

The idea behind lattice QCD is rather a simple one, instead of operating in the continuous theory of QCD, the theory is transcribed to a finite lattice in Euclidean space-time. In this prescription of QCD, the quark fields are placed on the lattice points, and the gluon fields are links that join neighboring lattice sites, as demonstrated in Figure 1.2 [57]. This discretization of the fields is done through the discretization of the action, which is the integral of the QCD Lagrangian density

$$S_{QCD} = \int \mathscr{L}_{QCD} \ d^4x \tag{1.3}$$



Figure 1.2: A three dimensional representation of a lattice used in lattice QCD.

The action is compromised of a fermion term,  $S_F$ , and a gauge term,  $S_G$ , which relies solely on the gauge fields. Upon discretization, these terms [30] take the following form:

$$S_F[\psi, \bar{\psi}, U] = a^4 \sum_{n \in \Lambda} \bar{\psi}(n) \left( \sum_{\mu=1}^4 \gamma_\mu \frac{U_\mu(n)\psi(n+\hat{\mu}) - U_\mu^{\dagger}\psi(n-\hat{\mu})}{2a} + m\psi(n) \right)$$
(1.4)

and

$$S_G[U] = \frac{\beta}{3} \sum_{n \in \Lambda} \sum_{\mu < \nu} Re \ Tr[1 - U_{\mu\nu}(n)], \tag{1.5}$$

where a is the lattice spacing and n is a point on the lattice defined by the set  $\Lambda = n = (n_1, n_1, n_3, n_4)$  where  $n_1, n_2, n_3 = 0, 1, ..., N_s - 1$  and  $n_4 = 0, 1, ..., N_t - 1$ .  $N_s$  and  $N_t$  are the spatial and time dimensions of the lattice hypercube, respectively. The prefactor of the gauge action is related to the coupling via  $\beta = \frac{6}{g^2}$ . The gauge links, U, are related to the gauge fields through the relation

$$U_{\mu}(n) = e^{iaA_{\mu}(n)}.$$
 (1.6)

The appearance of the transport vector,  $\hat{\mu}$ , is due to the discretization of the partial derivative of the quark fields, which shifts the lattice index in the direction of  $\mu$ . The discretization is given by

$$\partial_{\mu} = \frac{1}{2a} (\psi(n + \hat{\mu}) - \psi(n - \hat{\mu})).$$
(1.7)

The final term in equation 1.5 is the *plaquette*, which is the product of a four term closed loop of gauge links. The gauge and fermion actions are essentially to computing observables in lattice QCD, which can be computed from the path integral formalism.

## 1.3 The Need for Dirac Operator Inversions

Equation 1.4 can be written entirely in terms of the Dirac operator. Making the color and spin indices explicit, the naive Dirac operator is given by

$$D(n|m)_{\substack{\alpha\beta\\ab}} = \sum_{\mu=1}^{4} (\gamma_{\mu})_{\alpha\beta} \frac{(U_{\mu}(n))_{ab}\delta_{n+\hat{\mu},m} - (U_{\mu}^{\dagger}(n))_{ab}\delta_{n-\hat{\mu},m}}{2a} + m\delta_{\alpha\beta}\delta_{ab}\delta_{n,m}.$$
 (1.8)

The fermionic part of the action can then be written as

$$S_F[\psi, \bar{\psi}, U] = a^4 \sum_{n \in \Lambda} \bar{\psi}(n) D(n|m) \psi(n).$$
(1.9)

Both the gauge and fermion action appears in the path integral formalism. For a given observable  $\mathcal{O}$ , the expectation value is given by

$$\langle \mathscr{O}[\psi,\bar{\psi},U]\rangle = \frac{\int \mathscr{D}\psi \mathscr{D}\bar{\psi}\mathscr{D}U\mathscr{O}[\psi,\bar{\psi},U]e^{-S[\psi,\bar{\psi},U]}}{\int \mathscr{D}\psi \mathscr{D}\bar{\psi}\mathscr{D}Ue^{-S[\psi,\bar{\psi},U]}},\tag{1.10}$$

where the path integral measures are products of measures of all quark field components and products of measures for all link variables [30]. Given a fermionic observable, the path integral can be performed analytically. Consider the fermionic vacuum expectation value of a product of fermion fields, given by the operator

$$\langle \mathscr{O}[\psi,\bar{\psi}]\rangle_F = \langle \psi_{\alpha_1}^{(f_1)}(n_1)\psi_{\alpha_2}^{(f_2)}(n_2)...\psi_{\alpha_k}^{(f_k)}(n_k)\bar{\psi}_{\beta_1}^{(g_1)}(n_1)...\bar{\psi}_{\beta_k}^{(g_k)}(n_k)\rangle_F,$$
(1.11)

which is the n-point function of the non-interacting theory on a static gauge background [31]. In the path integral formalism, this observable takes on the form (for convenience, the lattice site dependence of the fields has been neglected)

$$\langle \mathscr{O}[\psi,\bar{\psi}] \rangle_F = \frac{1}{Z_F} \int \mathscr{D}\psi \mathscr{D}\bar{\psi}\psi^{(f_1)}_{a_1} \dots \psi^{(f_k)}_{a_k} \bar{\psi}^{(g_1)}_{\beta_1} \dots \bar{\psi}^{(g_k)}_{\beta_k} e^{-S_F[\psi,\bar{\psi}]}$$
(1.12)

where  $Z_F$  is the fermionic partition function:

$$Z_F = \int \mathscr{D}\psi \mathscr{D}\bar{\psi} e^{-S_F[\psi,\bar{\psi}]}.$$
(1.13)

This observable can be evaluated with the use of Wick's Theorem [56], which reduces it to a product of inverses of the Dirac matrix:

$$\langle \psi_{a_1}^{\alpha_1} \dots \psi_{a_k}^{\alpha_k} \bar{\psi}_{\beta_1}^{\beta_1} \dots \bar{\psi}_{\beta_k}^{\beta_k} \rangle_F = (-1)^k \sum_{P(1,2,\dots,k)} sign(P) D_{\alpha_1\beta_{P_1}}^{-1} \dots D_{\alpha_k\beta_{P_k}}^{-1}$$
(1.14)

where the sum runs over all permutations of the numbers 1, 2, ..., k and sign(P) is the

sign of the permutation P [30]. The Dirac operator inverse is then the quark propagator, which is the amplitude for a quark to propagate from one space time point to another. In the language of lattice QCD, each entry of the quark propagator connects a source at lattice point m with spin-color indices  $(\alpha_1, \alpha_1)$  to a sink point at site n with indices  $(\beta_{P_i}, b_{P_i})$ . Consider a *point source*, with fixed color, spin and spacetime indices:

$$S_0^{(m_0,\alpha_0,a_0)}(m)_a^{\alpha} = \delta(m-m_0)\delta_{\alpha\alpha_0}\delta_{aa_0}.$$
 (1.15)

Point sources are not the only type of sources used in lattice QCD, but is used in this setting for illustrative purposes. One column of the quark propagator from a site  $m_0$  to any point in the lattice, n is given by

$$D^{-1}(n|m_0)_{\substack{\beta\alpha_0\\ba_0}} = \sum_{m,\alpha,a} D^{-1}(n|m)_{\substack{\beta\alpha\\ba}} S_0^{(m_0,\alpha_0,a_0)}(m)_a^{\alpha}$$
(1.16)

This must be solved for every combination of color spin indices, totaling in 12 separate solves. In order to calculate fermionic observables in lattice QCD, we must then compute the inverse of the Dirac operator, which amounts to solving a system of linear equations. In the language of numerical linear algebra, this amounts to solving the matrix equation

$$Ax = b \tag{1.17}$$

where b is the source point (a right hand side vector), and A the Dirac operator (the matrix).

#### 1.4 The Dirac Operator in Lattice QCD

The Dirac operator is a matrix that acts in a vector space of size  $C \otimes S \otimes N_x \otimes N_y \otimes N_z \otimes N_t$ . The size of the Dirac operator is then  $12N \times 12N$ , where N is the product of the spacetime dimensions of the lattice. It is apparent that the Dirac operator is a very large matrix, even for moderately sized lattices, given its N dependence. This presents many technical challenges from the standpoint of computing and algorithmic development.

There are many discretizations of the Dirac operator in lattice QCD, all with differing properties, but all having the same continuum limit. A few of the most well known are the Wilson, staggered, overlap and domain wall formulations. The most common by far is the Wilson discretization, and the one used in the entirety of this work. The Wilson-Dirac operator is given by

$$D(n|m)_{\substack{\alpha\beta\\ab}} = (m + \frac{4}{a})\delta_{\alpha\beta}\delta_{ab}\delta_{n,m} - \frac{1}{2a}\sum_{\mu=\pm 1}^{\pm 4} (1 - \gamma_{\mu})_{\alpha\beta}U_{\mu}(n)_{ab}\delta_{n+\hat{\mu},m}$$
(1.18)

where the flavor index of the quarks have been omitted. In numerical simulations, the lattice spacing must be calculated through some observable, and so a is not explicitly used when creating the Wilson-Dirac operator for simulations. This can also be recast in terms of the *hopping matrix*, H, which collects nearest neighbor terms in the Dirac operator[30]

$$D = C(1 - \kappa H), \tag{1.19}$$

where  $\kappa$  is the hopping parameter. It is related to the quark bare mass, m, through the relation

$$\kappa = \frac{1}{2(am+4)}.\tag{1.20}$$

The constant, C, is not explicitly used as it is absorbed into the quark fields. When the quark mass approaches zero,  $\kappa$  approaches a critical value,  $\kappa_{crit}$ . The Wilson-Dirac operator is a complex, non symmetric matrix whose eigenvalues come in complex conjugate pairs, as can be seen from Figure 1.3. However, it can be made hermitian due



Figure 1.3: The eigenspectra for the Wilson-Dirac operator, originating from a lattice of size  $4^4$ , with  $\beta = 6.0$  and m = -1.0.

to its unique characteristic of  $\gamma_5$  hermiticity, where

$$D_H = D\gamma_5 \tag{1.21}$$

This hermiticity condition also holds for multiplication of  $\gamma_5$  on the left. The matrix  $D_H$  is complex and maximally indefinite, as can be seen from Figure 1.4, which uses the same Wilson-Dirac operator as 1.3 with  $\gamma_5$  multiplication. This condition also

results in the symmetry

$$D^{\dagger} = \gamma_5 D \gamma_5. \tag{1.22}$$

The matrix  $D_H$  is particularly important in noise reduction techniques for calculations of quantities arising from disconnected diagrams, which will be discussed at length in Chapter Four. The Wilson-Dirac operator, like many discretizations of the



Figure 1.4: The eigenspectra for the hermitian Wilson-Dirac operator  $D_H$ 

Dirac operator, has some numerical challenges. The size of the operator prohibits the implementation of simple iterative solvers, such as the Jacobi method or Gaussian elimination, when solving equations of the form of 1.16. As a result, more complex iterative solvers are needed in order to solve these systems of equations. Additionally, they experience a phenomena known as *critical slowing down*. As the renormalized quark mass approaches its physical value, the low lying eigenvalues approach zero. This results in a matrix that is ill-conditioned. These low lying eigenvalues inhibit the effectiveness of iterative solvers, so calculations at physical quark masses are

very expensive to perform. Simulations at unphysically large quark masses require a controlled extrapolation to the physical limit, which can be difficult and introduces additional error into the calculation. This necessitates the need for improved iterative solvers that can effectively solve equations of the form of equation 1.16 at physical values.

## CHAPTER TWO

#### Krylov Subspace Methods

## 2.1 Krylov Subspaces

Krylov subspace methods are a class of numerical iterative solvers that seek to find the solution of the linear system:

$$Ax = b \tag{2.1}$$

with  $A \in \mathbb{C}^{n \times n}$  and  $b \in \mathbb{C}^{n \times 1}$ . These methods extract an approximate solution from the subspace  $K_m$  through the projection  $b - Ax_m \perp K_m$  [49]. Here,  $x_m$  is an approximate solution and

$$K_m(A, r_0) = span\{r_0, Ar_0, A^2r_0, \dots, A^{m-1}r_0\}$$
(2.2)

is the Krylov subspace. The initial residual vector,  $r_0$ , is defined as

$$r_0 = Ax_0 - b. (2.3)$$

When a good initial guess is unavailable, this is the normalized right hand side of equation 2.1. The basis for the subspace is built through successive applications of the matrix A onto the initial residual, with orthonormalization of the basis vectors occurring with each application of the matrix. This process is known as the Arnoldi iteration [6]. The normalization coefficients obtained through the orthonormalization of the basis vectors become the entries of the matrix  $\bar{H}_m$ , where m is the number of Arnoldi iterations performed. This leads to the recurrence relation:

$$AV_m = V_{m+1}\bar{H}_m \tag{2.4}$$

where  $\bar{H}_m$  is an  $(m+1) \times m$  Hessenberg matrix, and  $V_{m+1}$  is a  $n \times (m+1)$  matrix whose columns are the basis vectors of the subspace. Algorithm 1 [49] is the Arnoldi

Algorithm 1: Arnoldi Iteration		
(1) Choose a vector $v_1$ , such that $  v_1  _2 = 1$		
(2) For $j = 1, 2,m$ Do:		
(3) Compute $h_{ij} = (Av_j, v_i)$ for $i = 1, 2,, j$		
(4) Compute $w_j := Av_j - \sum_{i=1}^j h_{ij}v_i$		
(5) $h_{j+1,j} =   w_j  _2$		
(6) If $h_{j+1,j} = 0$ then Stop		
(7) $v_{j+1} = w_j / h_{j+1,j}$		
(8) EndDo		

iteration. The output of this algorithm is the rectangular matrix, V whose columns are orthonormal and span the Krylov subspace, and the  $(m+1) \times m$  matrix, H. This is the beginning point for the GMRES [48] algorithm, which computes the approximate solution to equation 2.1.

## 2.2 GMRES and FGMRES

### 2.2.1 GMRES

The GMRES [48] algorithm begins with the recognition that the solution to 2.1 is contained in  $x_0 + K_m$ , and can be written as

$$x = x_0 + V_m y \tag{2.5}$$

where y is a  $m \times 1$  vector that is yet to be determined. The residual can be written in the following way:

$$r = b - Ax = b - A(x_0 + V_m y)$$
(2.6)

and using equation 2.4,

$$r = ||r_0||_2 v_1 - V_{m+1} \bar{H}_m y \tag{2.7}$$

$$= V_{m+1}(||r_0||_2 e_1 - \bar{H}_m y).$$
(2.8)

Because the columns of  $V_{m+1}$  are orthonormal, the residual norm can then be written as

$$||r||_2 = ||\beta e_1 - \bar{H}_m y||_2, \tag{2.9}$$

where  $\beta = ||r_o||_2$ . This equates to finding the vector y that minimizes the small least-squares problem. Once y is obtained, the approximate solution to equation 2.1 can be obtained through the use of equation 2.5. The GMRES algorithm is given by Algorithm 2. Because the GMRES algorithm forms an approximation to the solution, it may be necessary to build a large subspace to reach the desired tolerance for difficult problems. For problems where the matrix is large, as is common in lattice QCD, the storage cost of retaining the basis vectors of the subspace may be inhibitory. When this is the case, restarting the GMRES algorithm becomes a necessity.

## Algorithm 2: GMRES

- (1) Choose an initial guess  $x_0$  and m, the dimension of the Krylov Subspace
- (2) Compute  $r_0 = b Ax_0, \beta := ||r_0||_2$  and  $v_1 := r_0/\beta$
- (3) For j = 1, 2, ...m Do:
- (4) Compute  $w_j := Av_j$
- (5) For i = 1, ..., j Do:
- $(6) h_{ij} := (w_j, v_i)$
- (7)  $w_j := w_j h_{ij} v_i$
- (8) EndDo
- (9)  $h_{j+1,j} = ||w_j||_2$ . If  $h_{j+1,j} = 0$  set m := j and go to 11
- (10)  $v_{j+1} = w_j / h_{j+1,j}$
- (11) EndDo
- (12) Define the  $(m+1) \times m$  Hessenberg matrix  $\bar{H}_m = \{h_{ij}\}_{1 \le i \le m+1, 1 \le j \le m}$
- (13) Compute  $y_m$  the minimizer of  $||\beta e_1 \bar{H}_m y||_2$  and  $x_m = x_0 + V_m y_m$

### 2.2.2 Restarting GMRES

and GoTo 1

Restarting the GMRES algorithm is a rather simple procedure. When restarting, the size of the subspace m is truncated at a predetermined value. The algorithm follows that of the basic GMRES algorithm, but when the solution is formed, the residual vector is recomputed, the first basis vector is set to  $v_1 = r/||r_0||_2$  and  $r_0 = r$ . The restarted algorithm is given by Algorithm 3. Restarting the algorithm does have

_	Algorithm 3: Restarted GMRES		
	(1)	Choose an initial guess $x_0$ and $m$ , the dimension of the Krylov Subspace.	
	(2)	Compute $r_0 = b - Ax_0, \beta :=   r_0  _2$ and $v_1 := r_0/\beta$	
	(3)	Generate the Arnoldi basis and the matrix $\bar{H}_m$ using the Arnoldi	
		algorithm	
	(4)	Compute $y_m$ the minimizer of $  \beta e_1 - \bar{H}_m y  _2$ and $x_m = x_0 + V_m y_m$	
	(5)	Compute $  r  _2 =   b - Ax_m  _2$ . If satisfied then stop. Else, set $x_0 := x_m$	

the cost of slowing convergence towards the solution, however restarts are a necessary feature of solving linear equations in lattice QCD when working with the complex Wilson-Dirac operator due to storage costs. Figure 2.1 shows the effect of restarting GMRES with a restart length of m = 20. Restarted GMRES converges much more slowly than without restarting. GMRES converged in 384 iterations before a solution was obtained. The storage cost for this calculation is 19.2 times greater than that of the restarted calculation.



Figure 2.1: A comparison of GMRES with restarting and no restarting for the Wilson-Dirac operator for a 8<sup>4</sup> lattice at  $\kappa_{crit} \approx 0.1570$ .

### 2.2.3 FGMRES

Flexible GMRES (FGMRES) [47] is a variant of GMRES that allows for flexible preconditioning, where the preconditioner is allowed to vary between iterations of Arnoldi. Preconditioning is the technique of multiplying the system of linear equations by an approximate inverse in order to reduce the difficulty of the solve. Given equation 2.1, the system can be *left* multiplied by  $M^{-1}$ , an approximate inverse of A. The left preconditioned system is then

$$M^{-1}Ax = M^{-1}b. (2.10)$$

Similarly, a *right* preconditioner can be used:

$$AM^{-1}y' = b \tag{2.11}$$

where

$$x = M^{-1}y'.$$
 (2.12)

In many applications, the preconditioner  $M^{-1}$  is precomputed and remains static throughout the course of the solve. Such is the case in polynomial preconditioning [39]. A static preconditioner can be used to accelerate convergence of linear equations with GMRES, but GMRES cannot accommodate a dynamic preconditioner. Multigrid, which will be discussed in Chapter Three, is such a method that creates a dynamic preconditioner.

The modifications to GMRES to allow for a dynamic preconditioner are relatively straightforward. Instead of building an orthonormal basis for the Krylov subspace given in equation 2.2, we use the Arnoldi iteration to build an orthonormal basis for the right preconditioned Krylov subspace [48]

$$K_m = span\{r_0, AM^{-1}r_0, ..., (AM^{-1})^{m-1}r_0\}$$
(2.13)

The main difference then lies in the basis for the subspace itself. Instead of iterating on  $v_j$  with A, flexible GMRES iterates on  $z_j = M^{-1}v_j$ , and the approximate solution to 2.1 is formed via

$$x = x_0 + Z_m y \tag{2.14}$$

where  $Z_m = [z_1, ..., z_m]$ , and y is the vector that minimizes the least squares problem. Algorithm 4 gives the full algorithm for FGMRES. The drawback of FGMRES is that the vectors that form both  $Z_{m+1}$  and  $V_{m+1}$  must both be stored. Even when restarting the algorithm, this means double the cost of storage compared to GMRES. However, this is a very powerful method for solving linear systems if your preconditioner becomes *better* throughout the course of the solve. Depending on the efficacy of the preconditioner, the restart length can be drastically reduced, which significantly reduces the storage cost of the calculation.

Algorithm 4: Fl	exible GMRI	ES
-----------------	-------------	----

- (1) Choose an initial guess  $x_0$  and a dimension m of the Krylov subspaces
- (2) Compute  $r_0 = b Ax_0$ ,  $\beta = ||r_0||_2$ , and  $v_1 = r_0/\beta$
- (3) For j = 1, ..., m, do:
- (4) Compute  $z_j = M_j^{-1} v_j$
- (5) Compute  $w := Az_i$
- (6) For i = 1, ..., j do
- (7)  $h_{i,j} := (w, v_i)$
- (8)  $w := w h_{i,j} v_i$
- (9) Compute  $h_{j+1,j} = ||w||_2$  and  $v_{j+1} = w/h_{j+1,j}$
- (10) Compute  $x_m = x_0 + Z_m y_m$  where  $y_m = argmin_y ||\beta e_1 \bar{H}_m y||_2$
- (11) Compute the residual norm. If satisfied then stop, else GoTo 2

#### 2.3 Deflated Krylov Subspace Methods

## 2.3.1 The Effect of Deflation

The convergence of the solution of linear equations is governed mainly by the distribution of eigenvalues of the matrix. If the matrix possesses eigenvalues that lie close to the origin of the complex plane, convergence of linear equations can be drastically slowed. This phenomena is known as *critical slowing down*. This is particularly true in the case of the various discretizations of the Dirac operator when quark masses are set near their physical value. One approach to overcoming this difficulty is deflation. Deflation is a catch all phrase that at its heart means to remove the components of the residual corresponding to low lying eigenvalues. This can be done through the use of a preconditioner, or as is more commonly done, by incorporating approximate eigenvectors into the Krylov subspace.

One key component of implementing deflation by incorporating approximate eigenvectors into the subspace is to keep the subspace a Krylov subspace. This can only be done by using the harmonic Ritz vectors [42]. Harmonic Ritz vectors are vectors of the form

$$\tilde{y}_i = V_m g_i \tag{2.15}$$

where  $(\theta_i, g_i)$  are the eigenpairs of the matrix

$$H_m + \beta^2 \bar{H}_m^{\dagger} e_m e_m^T \tag{2.16}$$

and  $H_m$ ,  $\bar{H}_m$  and  $V_m$  are the matrices built through the Arnoldi iteration,  $\beta$  is the m+1, m th element of  $\bar{H}_m$ , and  $e_m$  is a unit vector. The harmonic Ritz pairs  $(\theta_i, \tilde{y}_i)$  are then approximate eigenpairs of the matrix A. These harmonic Ritz vectors can

then be incorporated into the Krylov subspace:

$$K_m = span\{\tilde{y}_1, \tilde{y}_2, ..., \tilde{y}_k, A\tilde{y}_i, ..., A^{m-k-1}\tilde{y}_i\}$$
(2.17)

which then removes their contribution to the residual vector, and increases convergence. This can be seen through the convergence bound for GMRES. We can gain a rough estimate for the increase in convergence by examining the case for normal matrices [41]. For normal and nearly normal matrices, the condition number of a matrix is given by

$$\kappa = \frac{\lambda_n}{\lambda_1} \tag{2.18}$$

where  $|\lambda_1| \leq |\lambda_2| \leq ... \leq |\lambda_n|$ . A large condition number means the matrix is ill conditioned and the rate at which an approximate solution to the system of linear equations is found is small. This can happen when there are outstanding eigenvalues from the rest of the spectrum, or there are many low lying eigenvalues. The convergence bound in GMRES for matrices with real eigenvalues [41] is given by

$$\frac{||r||_2}{||b||_2} \le 2||U||||U^{-1}||(1 - \frac{2}{\sqrt{\kappa} + 1})^m$$
(2.19)

where U is the matrix that contains the eigenvectors of the matrix A, and m the dimension of the Krylov subspace. Equation 2.19 is an upper bound for the convergence in GMRES. The appearance of the condition number in the denominator of the right hand of equation 2.19 means that the residual will decrease slowly for matrices possessing a large condition number. Defining an effective condition number to be

$$\kappa_{eff} = \frac{\lambda_n}{\lambda_{k+1}} \tag{2.20}$$

which is the condition number of the matrix in the absence of the lowest lying k eigenmodes, i.e. after the eigenvalues have been deflated. From equation 2.19, A rough estimate for the speed up can then be given as

$$\sqrt{\frac{\kappa}{\kappa_{eff}}} = \sqrt{\frac{\lambda_{k+1}}{\lambda_1}}.$$
(2.21)

This means the minimization of the residual norm will proceed  $\sqrt{\frac{\lambda_{k+1}}{\lambda_1}}$  times faster than without deflation. If the k lowest lying eigenpairs are well separated from the rest of the spectrum, the rate of convergence will greatly increase. Figure 2.2 demonstrates



Figure 2.2: The GMRES convergence bound as a function of the number of deflated eigenvalues.

the effect of removing low lying eigenpairs on the convergence bound for GMRES for the matrix A = diag([0.001 : 0.01 : 1, 2 : 1 : 10]) for a fixed dimension of m = 20 for the Krylov subspace. Deflation removes the contribution of the low lying eigenpairs, which lowers the condition number of the matrix, and thus decreases the upper bound of GMRES convergence. The more eigenpairs that are removed, the more the upper bound is decreased. This guarantees that the residual norm will decrease due to the inequality of equation 2.19.

### 2.3.2 GMRES-DR

GMRES with deflated restarting (GMRES-DR) [42] is one such Krylov Subspace method that places the harmonic Ritz vectors into the Krylov subspace in order to accelerate convergence. Because the harmonic Ritz pairs are approximate eigenpairs of the matrix A, GMRES-DR calculates both the solution of linear equations and the lowest k eigenpairs of the matrix. This is particularly advantageous for systems with multiple right hand sides, such as systems of linear equations arising from stochastic trace estimators.

GMRES-DR begins in the same fashion as GMRES. After the solution to the least squares problem is formed, the harmonic Ritz pairs  $(\theta_i, V_m g_i)$  are calculated. The first k eigenvectors  $g_i$  of the matrix  $H_m + \beta^2 \bar{H}_m^{\dagger} e_m e_m^T$  are then orthonormalized against the *short residual vector*,  $c - \bar{H}_m y$ , to form the columns of the matrix  $P_k$ . The H and V matrices for the next cycle are then formed via

$$\bar{H}_k^{new} = P_{k+1}^{\dagger} \bar{H}_m P_k \tag{2.22}$$

$$V_{k+1}^{new} = V_{m+1}P_{k+1} \tag{2.23}$$

where  $H_k$  is the leading  $k + 1 \times k + 1$  portion of  $\overline{H}_m$ . The k + 1 column of  $V_{k+1}$  is then orthonormalized against the previous k columns, and the Arnoldi iteration for the next cycle begins by iterating on  $v_{k+1}$  to form the rest of  $V_{m+1}$  and  $\bar{H}_m$ . It is in this fashion that the harmonic Ritz vectors are included in the Krylov subspace in accordance with equation 2.17. The least squares problem is solved using  $c = V_{m+1}^{\dagger}r_0$ as the right hand side, and the approximate solution is formed as in GMRES. If convergence has not been obtained, set  $x_0 = x_m$  and  $r_o = r$ . The full algorithm for GMRES-DR is given by Algorithm 5.

Algorithm 5: GMRES-DR(m,k)		
(1)	Choose <i>m</i> , the size of the Krylov subspace, and <i>k</i> , the number of desired approximate eigenvectors. Choose and initial guess $x_0$ and compute $r_0 = b - Ax_0$ . Let $v_1 = r/\beta$ , where $\beta =   r_0  _2$ .	
(2)	Apply standard GMRES(m), and let $\beta = h_{m+1,m}$ . Compute the k	
	smallest eigenpairs $(\theta_i, \tilde{g}_i)$ of $H_m + \beta^2 H_m^{\dagger} e_m e_m^T$ .	
(3)	Orthonormalize the $\tilde{g}_i$ 's. Form the $m \times k$ matrix $P_k$ whose columns are	
	the orthonormalized $\tilde{g}_i$ 's.	
(4)	Extend the columns of $P_k$ to length $m + 1$ by appending a zero entry to each. Orthonormalize the vector $c - \bar{H}_m d$ against them to form the	
	$k+1$ column of $P_k$ .	
(5)	Form portions of the new H and V. Let $\bar{H}_k^{new} = P_{k+1}^{\dagger} \bar{H}_m P_k$ and $V_{k+1}^{new} = V_{m+1} P_{k+1}$ . Let $\bar{H}_k = \bar{H}_k^{new}$ and $V_{k+1} = V_{k+1}^{new}$ .	
(6)	Orthogonalize the $k + 1$ column of V against earlier columns of the new $V_{k+1}$ .	
(7)	Apply the Arnoldi iteration starting from $j = k + 1$ to form the rest of	
	$V_{m+1}$ and $\bar{H}_m$ . Let $\beta = h_{m+1,m}$ .	
(8)	Let $c = V_{m+1}^{\dagger} r_0$ . Solve $min  c - \bar{H}_m d  $ for $d$ and form the approximate solution $x_m = x_0 + V_m d$ . Form the residual vector and check for convergence. If satisfied, stop. Else, proceed.	
(9)	Compute the k smallest eigenpairs $(\tilde{\theta}_i, \tilde{g}_i)$ of $H_m + \beta^2 \bar{H}_m^{\dagger} e_m e_m^T$ .	

(10) Restart. Let  $x_m = x_0$  and  $r_0 = r$ . Go to 3.

The effects of deflation can be observed in the convergence history of the solution. Figure 2.3 displays the residual norm as a function of the number of matrix vector products for the Wilson-Dirac operator at  $\kappa = \kappa_{crit}$ , for both GMRES-DR and GMRES. At  $\kappa_{crit}$ , the quark mass is zero, and the low part of the eigenspectra for



Figure 2.3: A comparison of GMRES with deflated restarting and only restarting for the Wilson-Dirac operator for a 8<sup>4</sup> lattice at  $\kappa_{crit} \approx 0.1570$ . A restart length of m = 20 was used, and k = 10 eigenvectors were deflated from the system.

the Wilson-Dirac operator has many eigenvalues near zero. The convergence history for GMRES-DR shows a sharp decline in the residual norm, whereas the residual norm for GMRES is monotonically decreasing. The sharp decline for GMRES-DR indicates that the approximate k eigenvectors included in the subspace have converged accurately enough for their contribution to be removed from the residual norm.

### 2.3.3 MINRES-DR

Minimum residual with deflated restarting (MINRES-DR) [3] is an iterative solver that computes solutions to linear equations for symmetric indefinite matrices while simultaneously computing the k lowest lying eigenvectors of the matrix A. It follows the same basic structure as GMRES-DR, with a few notable differences. The
Arnoldi iteration that is used for forming the orthonormal basis for the Krylov subspace is GMRES-DR is replaced with the Lanczos iteration [36], given by Algorithm 6. The Lanczos iteration is specific for symmetric matrices, and employs what is called the "Three Term Recurrence". In the Arnoldi iteration, all of the basis vectors must be retained for full orthonormalization. With symmetric matrices, this is not necessary, and only three of the basis vectors need to be orthonormalized at a time. The Lanzcos iteration results in a recurrence relation in the same way that the Arnoldi iteration does. Given a symmetric matrix, A, the Lanczos iteration yields the recurrence relation:

$$AV_m = V_{m+1}\bar{T}_m \tag{2.24}$$

in direct analogy with the Arnoldi recurrence relation. In this instance,  $\overline{T}_m$  is symmetric and tridiagonal, except for a leading  $(k+1) \times (k+1)$  portion. Because MINRES-DR

Algorithm 6: Symmetric Lanczos Iteration	
(1) Choose an initial vector $v_1$ of 2-norm unity. Set $\beta_1 = 0, v_0 = 0$	
(2) For $j = 1, 2,, m$ Do:	
$(3) \qquad w_j := Av_j - \beta_j v_{j-1}$	
(4) $\alpha_j := (w_j, v_j)$	
(5) $w_j := w_j - \alpha_j v_j$	
(6) $\beta_{j+1} :=   w_j  _2$ . If $\beta_{j+1} = 0$ then Stop	
(7) $v_{j+1} := w_j / \beta_{j+1}$	
(8) EndDo	

computes and deflates harmonic Ritz vectors as well, the MINRES-DR algorithm proceeds in the same fashion as GMRES-DR. The full algorithm for MINRES-DR is given by Algorithm 7.

# Algorithm 7: MINRES-DR(m,k)

- (1) Choose m, the maximum size of the subspace, and k, the desired number of approximate eigenvectors. If there is an initial guess,  $x_0$ , then the problem becomes  $A(x - x_0) = r_0$ .
- (2) Apply m iterations of the standard symmetric Lanczos algorithm. Fully reorthogonalize all the Lanczos vectors.
- (3) Compute the k smallest harmonic Ritz values,  $\theta_i$ , and let  $g_i$  be the corresponding vectors of unit length.
- (4) Let  $c_{m+1} = V_{k+1}^{\dagger} r_0$ . Solve the least squares problem  $min||c_{m+1} \bar{T}_m d||$ for d and set  $\tilde{x} = x_0 + V_m d$ . Then  $r = r_0 - A\tilde{x} = r_0 - V_{m+1}\bar{T}_m d$ . If satisfied with convergence of the linear equations and the eigenvalues, stop. If not, let  $x_0 = \tilde{x}$  and  $r_0 = r$ , and continue.
- (5) Let P be the  $m + 1 \times k + 1$  matrix whose first k columns come from orthonormalizing the  $g_i$  vectors (and adding zeros for the m + 1 row). Let  $e_m$  be the mth coordinate vector of length m. Then the k + 1column of P is the vector  $[-t_{m+1,m}\bar{T}_m^{\dagger}e_m, 1]^T$  orthonormalized against the previous columns. The new V and T matrices are formed from the old ones:  $V_{k+1}^{new} = V_{m+1}P$  and  $\bar{T}_k^{new} = P^{\dagger}\bar{T}_m P_{m,k}$ , where  $P_{m,k}$  has the first m columns and k rows of P. Set  $V_{k+1} = V_{k+1}^{new}$  and  $\bar{T}_k = \bar{T}_k^{new}$ .
- (6) Compute  $w = Av_{k+1} \sum_{i=1}^{k} t_{k+1,i}v_i$ , then  $t_{k+1,k+1} = v_{k+1}^{\dagger}w$  and  $w = w t_{k+1,k+1}v_{k+1}$ . Let  $t_{k+2,k+1} = ||w||_2$ . Set  $v_{k+2} = w/||w||_2$ . Next, compute the other  $v_i$  vectors for i = k+3, ..., m+1, using the standard Lanczos iteration.  $\overline{T}_m$  is formed at the same time. Also, reorthogonalize the  $v_i$  vectors as desired. Go to step 3.

### 2.3.4 GMRES-Project

In many areas of science and engineering, linear systems of equations with multiple right had sides present themselves. One such example is for calculating matrix elements of disconnected loops diagrams in lattice QCD, which will be discussed in length in Chapter Four. The goal then is to calculate the solutions,  $x^{(i)}$ , to the system

$$Ax^{(i)} = b^{(i)}. (2.25)$$

In cases where there are many right hand sides, it becomes advantageous to use acceleration techniques, rather than treating each right hand side as its own distinct problem. GMRES-Project [44], is one such acceleration technique. GMRES-Project uses previously generated eigenvector information of the matrix A to increase the rate of convergence for subsequent right hand sides.

Given the system of 2.25, the i = 1 system of linear equations is solved with GMRES-DR. In addition to the recurrence relation of equation 2.4, a smaller recurrence relation is also generated during GMRES-DR:

$$AV_k = V_{k+1}\bar{H}_k,\tag{2.26}$$

where  $V_k$  is the  $n \times k$  matrix whose columns span the subspace of approximate eigenvectors, and  $\bar{H}_k$  is the leading  $(k + 1) \times k$  portion of  $\bar{H}_m$ . Because  $V_k$  contains the eigenvector information of the matrix, it can be used via projection methods to accelerate the subsequent right hand sides. This eigenvector information is carried over to the i = 2, ..., n right hand side systems. The eigenvector information carried by  $V_k$  and  $\bar{H}_k$  is incorporated into subsequent right hand sides via either Minres or Galerkin Projections. GMRES-Project uses Minres Projection alternated between cycles of GMRES. The Minres Projection, given by Algorithm 8, works by forming an initial

gorithm 8: Minres Projection
(1) Let the current approximation solution be $x_0$ and the current system of
equations be $A(x - x_0) = r_0$ . Let $V_{k+1}$ and $\overline{H}_k$ be the matrices from 2.26
(2) Solve $min  c - \bar{H}_k d  $ , where $c = V_{k+1}^{\dagger} r_0$ .
(3) The new approximate solution is $x_k = x_0 + V_k d$ .
(4) THe new residual vector is $r_k = r_0 - AV_k d = r_0 - V_{k+1}\bar{H}_k d$ .

 $\mathbf{A}$ 

approximate solution by solving the small least squares problem

$$min||c - \bar{H}_k d||, \qquad (2.27)$$

where  $c = V_k^{\dagger} r_0$ . The initial approximate solution is then formed via  $x_k = x_0 + V_k d$ , and the residual is  $r_k = r_0 - V_{k+1} \overline{H}_k d$ . This initial solution and residual is then used in a cycle of GMRES. This process of alternating projection after restarts repeats until the desired convergence has been reached, as illustrated in Algorithm 9.

Algorithm 9: GMRES(m)-Project(k)
(1) For right hand side number <i>i</i> , after applying the initial guess $x_0^{(i)}$ , let the system of equations be $A(x^{(i)} - x_0^{(i)} = r_0^{(i)})$ .
(2) If it is known that the right-hand sides are related, project of the
previous computed solution vectors.
(3) Apply the Minres Projection for $V_k$ . This uses the $V_{k+1}$ and $\bar{H}_k$ matrices developed while solving the first right hand side with GMRES-DR.

- (4) Apply one cycle of GMRES(m).
- (5) Test the residual norm for convergence (can also test during the GMRES cycles). If not satisfied, go back to step 3.

Figure 2.4 shows the effect of deflation immediately upon the start of the iterative solve. Because the first right hand sides is solved with GMRES-DR, no eigenvector information is yet available, so convergence is slower. Because the eigenvector information is available upon the outset of the second right hand sides, convergence occurs at a faster rate for GMRES-Proj.



Figure 2.4: The convergence history for the Wilson-Dirac operator at  $\kappa_{crit} \approx 0.1570$  for two  $Z_4$  noise vectors.

### CHAPTER THREE

### Multigrid in Lattice QCD

### 3.1 Introduction

Multigrid is a technique that was developed originally for the use of boundary value problems arising from discretized partial differential equations [22]. The goal of multigrid is to shift critical slowing down on a fine grid arising from the discretization of the differential equation to coarser grid levels, where it can be dealt with more efficiently. As discussed in Chapter Two, exceptional eigenvalues can drastically decrease the performance of iterative solvers. While deflation removes the contribution of these eigenvalues to the residual, multigrid seeks to move their contribution to the error to smaller, related operators. While this can be easily performed for the discretization of differential equations by creating operators with coarser meshes, it is much more complicated for lattice QCD [17]. In addition, critical slowing down may still play a role in increasing the expense of the the coarse grid solve, and acceleration techniques such as deflation may still be necessary for quick convergence.

Not only does multigrid recursively shift critical slowing down to smaller operators, it also implicitly helps with the issue of *strong scaling*. Strong scaling is the phenomena that occurs for linear solvers where the the performance, or time, of the calculation deviates from linear with an increasing number of CPU/GPU cores [8][19]. As the lattice hypervolume increases, more computational power becomes necessary. Because most of the work in multigrid is done on the coarse grid, multigrid reduces the amount of computation power necessary compared to conventional means. This chapter will discuss the basis of multigrid, and the changes that are necessary for lattice QCD, as well as the results for our study using deflated GMRES as coarse grid solver for solving linear systems arising from lattice QCD [55]. While multigrid has been developed for the Wilson [17], staggered [23] with improvements [53], domain wall [27] and the overlap [18] discretizations, deflation in conjunction with multigrid has only been explored in a limited context [46][53][43]. In [46], multigrid was used in a deflation like context for variance reduction of stochastic trace estimators. In [53], deflation was used for the normal equations, and in [43], the eigenvectors of the Wilson-Dirac operator were calculated on the coarse grid and prolonged to the fine grid for deflation. We demonstrate that our method of deflated GMRES is incredibly efficient, and dramatically reduces the overall cost of the inversion of the Wilson Dirac operator in the lattice Schwinger model [50].

# 3.2 Classical Multigrid

The discretization of partial differential equations can be accomplished using approximations to derivatives in the form of either finite difference methods or finite elements. Both methods permit the use of arbitrary mesh sizes, so the accuracy of the solution to the system of linear equations can be calculated to a desired precision by either increasing or decreasing the mesh size. Classic iterative schemes, such as the Jacobi method or the Gauss Seidel method, are effective at reducing high frequency components of the error, but struggle with reducing low frequency components. On meshes of a coarser spacing, the low frequency components of the fine mesh behave like high frequency components [1]. This gives rise to the simple idea to use a coarser mesh to accelerate the solution on the finest mesh level. This section is primarily intended as an introduction to the basic concepts and the jargon of multigrid, with an illustrative example.

In the following discussion, we will refer to the finest mesh, or grid, with h, and the coarse grid with h'. Given a fine grid operator,  $A_h$ , we require operators that move between the fine grid and coarse grids. The *restriction operator*,  $R_h^{h'}$ , moves a fine grid operator to the coarse grid. If v is a coarse grid vector, and u is a fine grid vector, then the action of  $R_h^{h'}$  is

$$v = R_h^{h'} u. aga{3.1}$$

Similarly, in order to transfer a vector from the coarse grid to the fine grid, we use an *interpolation*, or *prolongation*, operator,  $I_{h'}^h$ . The action of the interpolation operator is then

$$u = I_{h'}^h v. aga{3.2}$$

This also gives a prescription for forming a coarse grid operator,  $A_{h'}$ , through the relation

$$A_{h'} = R_h^{h'} A_h I_{h'}^h. ag{3.3}$$

These are then the three ingredients needed for a multigrid prescription. The typical usage of multigrid is to use a small number of iterations with either Gauss-Seidel or the Jacobi method to reduce high frequency components of the error on the fine grid. This process is called *smoothing* because it produces error whose high frequency components have been reduced, or smoothed. The residual is then restricted to the coarse grid where it can be solved exactly, and the result is interpolated back up to the fine grid. This defines the *V-cycle* of a multigrid solver. Given the system of linear equations Au = f, a two grid V-cycle proceeds according to Algorithm 10 [1]. It is worth noting that multigrid is not constrained to only V-cycles. Other common cycles

### Algorithm 10: Classic multigrid

- (1) Iterate  $\nu$  times on Au = f with Gauss-Seidel or Jacobi to obtain  $\tilde{u}$ .
- (2) Form the residual vector  $r = A\tilde{u} f$ .
- (3) Restrict the residual to the coarse grid with  $r_h = R_h^{h'} r$ .
- (4) Solve  $A_{h'}e = r_h$ .
- (5) Interpolate and add e to the fine grid solution:  $\tilde{u} = \tilde{u} + I_{h'}^h e$
- (6) Iterate  $\nu$  times on Au = f with initial guess  $\tilde{u}$
- (7) If the residual norm is less than a specified tolerance, quit. Else, go to (1).

are the W-cycle and Full multigrid (FMG), illustrated in Figure 3.1 where the level of coarsening is denoted by the integer l. The fine grid level is denoted by l = 1.



Figure 3.1: (Left to right) The multigrid V-cycle, W-cycle, and FMG cycle for a three grid hierarchy.

To demonstrate the speed up gained by using multigrid, the one dimensional Poisson equation is solved using Gauss-Seidel on multiple grids and on one grid as a comparison [25]. The Poisson equation in one dimension is given by

$$\frac{\partial^2}{\partial x^2}\phi = f(x). \tag{3.4}$$

This partial differential equation commonly arises in electrostatics, gravitation and fluid dynamics. For this example, we solve the Poisson equation with boundary conditions u(0) = 0 and u(1) = 0 on the interval 0 < x < 1, with f(x) = -1 and a fine grid spacing of  $h = \frac{1}{32}$ . Figure 3.2 gives the error of the solution as a function of the Gauss-Seidel iteration count. The use of multiple grids greatly increases the convergence rate of the solve in comparison to solely using one grid.



Figure 3.2: A comparison of multigrid with Gauss-Seidel smoothing and Gauss-Seidel on one grid.

# 3.3 Adaptive Multigrid

If no information about the underlying mesh of the problem is available, then the hierarchy of grids must be generated in some fashion directly from the matrix. In *adaptive multigrid* this is done through the use of near null vectors [20]. Because the goal is to transfer critical slowing down to coarser grid levels, near null vectors approximate the low lying eigenvectors of the matrix A. Given the eigenvalue equation

$$Av_i = \lambda_i v_i, \tag{3.5}$$

if  $\lambda_i \ll 1$ , then the eigenvalue equation reduces to

$$Av_i \approx 0.$$
 (3.6)

This implies that the low lying eigenspace that contributes to critical slowing down can be approximated by approximately solving the homogeneous equation

$$Ax \approx 0 \tag{3.7}$$

to a specified tolerance, where x is a near null vector. The success in the using near null vectors, at least in the context of lattice QCD operators, can be attributed to *local coherence* [40]. The eigenvectors of the Wilson-Dirac operator are locally coherent in that they are similar to one another on a local scale. This means that the eigenvectors are similar within certain domains of the lattice [29]. This in part explains the success of multigrid in lattice QCD. Due to the effects of local coherence, the near null vectors must be partitioned into aggregates [21]. These near null vectors then become the columns of the prototypical prolongator:

$$\tilde{P} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_{n_{vec}} \end{bmatrix}.$$
(3.8)

Equation 3.8 is the array whose columns are the near null vectors, where  $n_{vec}$  is the number of near null vectors that are necessary to remove critical slowing down. Since each near null vector is a *global* quantity, the aggregation procedure groups elements of the matrix together based on some commonality, usually by phase or magnitude of the matrix elements [7]. The elements of the near null vectors are then zeroed out based on the aggregation criteria, which forms the block diagonal structure of the

# prolongator

$$P = \begin{pmatrix} \begin{bmatrix} \mathbf{X}_1 \end{bmatrix} & & & \\ & \begin{bmatrix} \mathbf{X}_2 \end{bmatrix} & & \\ & & \mathbf{X}_2 \end{bmatrix}$$
(3.9)  
$$& & \ddots & \\ & & & \begin{bmatrix} \mathbf{X}_n \end{bmatrix} \end{pmatrix}.$$

In equation 3.9, every block corresponds to an aggregate of the near null vectors, where n is the number of aggregates within the matrix. There are modifications that can be performed to the prolongator operator, such as using a smoothing operator based on an estimator of the operator's spectral density [21], however, this allows for a basic transfer between gird hierarchies. A minimum requirement of the prolongator is that the interpolation of an eigenvector must reconstruct it to an accuracy that is proportional to its corresponding eigenvalue [16]. In lattice QCD, there are further modifications to the near null vectors based on symmetry considerations, which will be discussed in the next section.

#### 3.4 Multigrid for the Lattice Wilson-Dirac Operator

Lattice QCD cannot benefit from the methods of classical multigrid, or even adaptive multigrid in its naïve form. In classical multigrid, the differential equation can be discretized with an arbitrary mesh size. This is not possible in lattice QCD, as the gauge configurations that are used in the formation of the Dirac operator are dependent on the lattice size used to create them. Generally speaking, gauge configurations arising from different lattice scales are not correlated. It is only after sampling from many gauge configurations that the correct physics is obtained. Multigrid must therefore proceed from an initial approximation of the low lying eigenspace directly from the Dirac operator in an adaptive fashion [17].

One important consideration in the formation of multigrid for lattice QCD is the preservation of  $\gamma_5$  hermiticity on coarse grids created from the original Wilson-Dirac operator, given by equation 1.18 [23]. The  $\gamma_5$  hermiticity condition is given by

$$D = \gamma_5 D^{\dagger} \gamma_5 \tag{3.10}$$

where  $\gamma_5$  is the product of the four Dirac matrices. This is done to ensure that the coarse grid operators also have eigenvalues that come in complex conjugate pairs, and is accomplished through a process called *chiral doubling*. Chiral doubling projects out the left and right handed components of the near null vectors:

$$n_{\pm} = \frac{1}{2} (1 \pm \gamma_5) n \tag{3.11}$$

where n is a near null vector and  $n_{\pm}$  are two chirally doubled near null vectors. The adaptive process and chiral doubling are essential features of a successful coarsening for the Wilson-Dirac operator.

#### 3.4.0.1 The Full Coarsening Procedure.

The full coarsening procedure for the lattice Wilson-Dirac operator proceeds as follows:

- (1) Calculate  $n_{vec}$  near null vectors, and globally orthonormalize them
- (2) Chirally double the near null vectors
- (3) Partition the chirally doubled near null vectors based on partitions within the fine lattice

- (4) Locally orthonormalize the resultant vectors
- (5) Form the prolongator, whose columns are the partitioned, orthonormalized, chirally doubled near null vectors
- (6) Form the coarse grid operator,  $\hat{D}$ , where  $\hat{D} = P^{\dagger}DP$

Each of these processes deserves some elaboration. We begin by calculating  $n_{vec}$  near null vectors, where  $n_{vec}$  is the number of near null vectors coming from approximately solving the homogeneous equation. In practice, this is done by converting the homogeneous equation into the residual equation

$$DD^{\dagger}e = -DD^{\dagger}\psi, \qquad (3.12)$$

where  $\psi$  is a random gaussian distributed vector [23]. This residual equation is solved to a tolerance of  $10^{-4}$ . The solution to the homogeneous equation is then constructed as  $n = e + \psi$ , where *n* is the near null vector. This conversion to the residual equation is done for stability, and using the normal equations can also be employed. This process is performed  $n_{vec}$  times to yield near null vectors that approximate the low lying eigenspace of the fine operator.

The near null vectors must be chirally doubled in order to conserve  $\gamma_5$  hermiticity of the coarse grid operators. Each vector produced from the lattice are global quantities, thus they have a dependence on each lattice site. In lattice QCD, each lattice site has 12 numbers, coming from the SU(3) gauge group and Dirac algebra basis. The chiral doubling then occurs within the Dirac space on each lattice site as

$$n_{\pm}(x) = \frac{1}{2}(1 \pm \gamma_5)n(x). \tag{3.13}$$

In order to define the coarse grid degrees of freedom and coarse grid lattice, the near

null vectors need to be partitioned based on partitions of the original lattice [9]. This is done by defining a block of neighboring lattice points,  $x_{sub}$ , and letting

$$\hat{n}_{\pm}(x) = \begin{cases} n_{\pm}(x) & \text{if } x \in x_{sub}, \\ 0 & \text{if } x \notin x_{sub}. \end{cases}$$
(3.14)

This is performed for each left and right handed chiral vector for each block of neighboring lattice points. Because we have left and right handed chiral null vectors, we have  $2n_{vec}$  degrees of freedom per coarse site [23], with the number coarse sites being determined by the number of lattice points contained within  $x_{sub}$ .



Figure 3.3: An example of the blocking that occurs for a  $4^2$  isotropic lattice (left) with a  $2^2$  blocking to form a  $2^2$  coarse lattice (right).

The vectors  $\hat{n}_{\pm}$  then become the columns of the prolongator, P. This process is repeated for each of the chirally doubled near null vectors for each block of neighboring lattice points, filling the columns of the prolongator. Due to the orthonormalization done on the columns of the prolongator, we have the additional property that

$$R = P^{\dagger}.\tag{3.15}$$

The restriction and prolongation operators allow us to move vectors between fine and coarse grid levels. A coarse grid vector v can be prolonged to the fine grid in the same way as equation 3.2 via

$$u = Pv \tag{3.16}$$

where u is a fine grid vector. The restriction operator also allows us to do the reciprocal calculation from the fine grid to the coarse grid as in equation 3.1:

$$v = P^{\dagger}u. \tag{3.17}$$

These operators also allow us to create the coarse grid operator,  $\hat{D}$ , via  $\hat{D} = P^{\dagger}DP$ . This is then the operator for a  $N_c \times N_c$  lattice with  $2n_{vec}$  degrees of freedom per coarse lattice site, where  $N_c = \frac{N_{latt}}{size(x_{sub})}$  (in the case of an isotropic two dimensional lattice), and the size of the operator is  $2n_{vec} \times N_c \times N_c$ . Figure 3.4 shows the sparsity pattern of the coarsened Wilson-Dirac operator and the fine Wilson-Dirac operator in the lattice Schwinger model. The y axis denotes the row index of the operator and the x axis the column index, with blue markers denoting the non zero elements of the operator. The value nz denotes the total number of non zero elements of the operator. The fine grid operator corresponds to a lattice of size  $16^2$ , and the coarsened operator was created with 4 near null vectors and a partioning of  $4^2$  in the original fine lattice. As can be observed through comparison of the sparsity patterns, the sparsity pattern is maintained, but with the additional  $2n_{vec}$  degrees of freedom for each lattice site. The coarsened operator then corresponds to a  $4^2$  lattice with 8 numbers per lattice site. in contrast to the  $16^2$  fine lattice with 2 numbers per lattice site. This procedure can be used recursively to form even coarser operators. The procedure is merely repeated using  $\hat{D}$  as the operator to be coarsened, to form a third coarse operator  $\hat{\hat{D}}$ .

Figure 3.5 shows the spectra for the fine and coarse Wilson-Dirac operators in the lattice Schwinger model for a lattice of size  $16^2$ , with  $\beta = 6.0$  and m = -0.0675. The coarse operator was formed with the coarsening procedure previously outline with  $n_{vec} = 8$ , and a partitioning of size  $4^2$ . The use of near null vectors approximates the



Figure 3.4: The sparsity pattern for the coarsened Wilson-Dirac operator (left) and the fine Wilson-Dirac operator (right).



Figure 3.5: The spectra for both the fine Wilson-Dirac operator and a coarsened Wilson-Dirac operator

low lying eigenspace of D, so the low lying eigenspectra of D approximates that of D. We can also observe that the ratio for the maximum eigenvalue and minimum value of the coarsened operator is smaller than that for the fine grid operator, resulting in a smaller condition number on the coarse grid. This means that the coarse grid operator is less ill conditioned than that of the fine operator, so a solve on the coarse grid is more easily performed. This reduction in the condition number of the matrix through multiple grids is akin to the reduction of the condition number with deflation.

In addition, the eigenvectors of the coarsened operator must have components that resemble those of the fine operator. Because eigenvectors are global quantities, they have components that extend to every point of the lattice. This poses a challenge to classical multigrid that cannot be satisfied, but the adaptive process creates a coarse operator that satisfies this criteria. Figure 3.6 shows the sum of the absolute value of the elements for the 1st eigenvector of both the coarse and fine grid operators used in Figure 3.5. It is apparent that the shape of the coarse operator's lowest lying eigenvector resembles that of the fine operator.

Though Figure 3.6 is a reassuring qualitative assessment of the shift of critical slowing down to coarser grids, we can observe this phenomena through the use of the local co-linearity [23] of the eigenvectors, given by

$$||(1 - PP^{\dagger})v_{\lambda}||_2 \tag{3.18}$$

where  $v_{\lambda}$  is an eigenvector of the fine grid lattice operator. The local co-linearity gives a quantitative measurement of how well the eigenspace of the fine operator is replicated on the coarse grid. The process of coarsening seeks to minimize this function of the prolongator for the lowest lying eigenvalues, while not increasing it for higher eigenmodes. If the value of this function exceeds one, then the eigenspace is not being replicated properly.



Figure 3.6: The absolute values of the components of the 1st eigenvector for the coarse (top) and fine (bottom) Wilson-Dirac operators on a lattice site by site basis.



Figure 3.7: The local co-linearity for a  $4^2$  coarsened lattice,  $\beta = 6.0$ , and m = -0.0675.

Figure 3.7 displays the local co-linearity for the coarsened lattice of figures 3.5 and 3.6. For the lowest lying eigenmode, the local co-linearity is near zero, reflecting the qualitative assessment from figure 3.6. This is one assurance that critical slowing down has been shifted to the coarse level. The local co-linearity of the higher eigenmodes is not increased beyond one, indicating that the eigenspace of the fine operator is not being inaccurately represented on the coarse lattice. The solid line represents the limit of the allowed local colinearity before the contribution of the higher eigenmodes to the error begins to become detrimental.

# 3.5 Deflated GMRES in Multigrid for Lattice QCD

Because the multigrid procedure recursively shifts critical slowing down to coarser operators, the coarse grid problem can still be a difficult problem due to the low lying eigenvalues of D now being approximated in coarsened operators. For lattice QCD operators that are generated from very fine lattices, the coarse grid operator may still be large, and as such deflation can be beneficial.

While deflation can greatly improve the cost of the solve in terms of matrix vector products, it can be prohibitively expensive in terms of storage when the Dirac operator in use is very large. Even though the rate of convergence may be slower, Conjugate Gradient (CG) [35] is often used in lieu of deflation methods for very large operators, since no storage of eigenvectors is required and the required. The hierarchy of operators constructed in multigrid allows for deflation to be used on the coarse level, where the storage costs of retaining the eigenvectors are greatly reduced due to the smaller size of the coarse grid operator.

To examine the effects of deflation on the coarse grid, we work in the two dimensional lattice Schwinger model [50], or compact QED, which shares many physical characteristics with four dimensional lattice QCD. Compact QED uses the U(1) group, and exhibits chiral symmetry breaking and confinement [45]. As such, it is an excellent algorithmic testing ground for the full 4D theory of QCD. 10 Gauge configurations for lattices of size  $64^2$ ,  $128^2$ , and  $256^2$  at a coupling of  $\beta = 6.0$  were created using QCDLAB 1.0 [15]. The coarsening procedure for the two dimensional lattice Schwinger model is directly analogous with that of four dimensional lattice QCD, with the replacement of

$$\gamma_5 \to \sigma_3. \tag{3.19}$$

This substitution results in the following stencil for the Wilson-Dirac operator in the 2D Schwinger model[15]:

$$D(n|m) = (m + \frac{2}{a})\delta_{nm} - \frac{1}{2a}\sum_{\mu=1}^{2} [(1 - \sigma_{\mu})U_{\mu,n}\delta_{n,m+\hat{\mu}} + (1 + \sigma_{\mu})U_{\mu,n-\hat{\mu}}\delta_{n,m-\hat{\mu}}], \quad (3.20)$$

with an analogous hermiticity condition of the Wilson-Dirac operator being

$$D = \sigma_3 D^{\dagger} \sigma_3. \tag{3.21}$$

With these replacements, the coarsening procedure proceeds as discussed in the previous section. A hierarchy of three grids was created for this study. While multigrid can be used as a linear solver in its own right, the typical use of multigrid in lattice QCD is to use it as a preconditioner of an outer Krylov solver. To this end, we use FGM-RES as an outer solver with the resultant vector of the V-cycle from multigrid being used as a right preconditioner. We consider the use of coarse grid deflation in three contexts: (1) coarse grid deflation with only smoothing performed on the fine and intermediate levels, (2) coarse grid deflation with a partial solve on the intermediate grid, and (3) the effect of coarse grid deflation for multiple right hand sides.

Our particular method of deflation is based on the recognition that because we employ MG as a preconditioner for FGMRES, every outer iteration of FGMRES represents *a new right hand side*. To this end, we used GMRES-DR as a coarse grid solver for the first outer iteration, followed by GMRES-Project for every other outer iteration. This allows us to reap the benefits of deflation without the increased storage cost of deflating on the fine grid alone, and at every iteration of the outer solver, we receive a better approximation to the Wilson-Dirac operator inverse.

We consider the performance of coarse grid deflation in three contexts: (1) coarse grid deflation without a partial intermediate solve, (2) coarse grid deflation with an intermediate solve, and (3) the effect of coarse grid deflation for multiple right hand sides. The parameters for the setup and solvers for our deflated coarse grid *without* an intermediate solve are given in Table 3.1. The parameters for our deflated coarse grid solve *with* an intermediate solve vary only slightly, so only the level 2 and 3 parameters are given in Table 3.2, the rest remains the same.

Solve Segment	Segment Parameters	Parameter Used
setup	operator	$DD^{\dagger}$
	solver	CG
	max iterations	250
	tolerance per near null vector	$10^{-4}$
	number of near null vectors per level	12
	size of partioning	$4^2$
	number of levels	3
level 1	operator	D
	solver	FGMRES(8)
	pre-, post-smoother	GMRES(2)
	residual tolerance	$10^{-8}$
level 2	operator	$\hat{D}$
	pre-, post-smoother	GMRES(2)
level 3	operator	$\hat{\hat{D}}$
	solver, 1st outer iteration	GMRES-DR(200,40)
	residual tolerance	$10^{-15}$
	solver, subsequent outer iterations	GMRES-Project(200,40)
	residual tolerance	10 <sup>-8</sup>

Table 3.1: The setup and solver parameters for our deflated multigrid solve without a partial intermediate solve.

Table 3.2: The setup and solver parameters for our deflated multigrid solve with a partial intermediate solve.

Solve Segment	Segment Parameters	Parameter Used
level 2	operator	$\hat{D}$
	solver	GMRES(8)
	pre-, post-smoother	GMRES(2)
	residual tolerance, max cycles	.2  b  , 10
level 3	operator	$\hat{\hat{D}}$
	solver, 1st outer iteration	GMRES-DR(200,40)
	residual tolerance	$10^{-8}$
	solver, subsequent outer iterations	GMRES-Project(200,40)
	residual tolerance	$10^{-2}$

At this point, it becomes prudent to mention that the remarkably high restart length of GMRES-DR/GMRES-Proj are used for comparison sake. In order to de-

termine the efficacy of our deflated multigrid preconditioner, we compare to the conventional solvers of CG on the normal equations (CGNE),  $D^{\dagger}D$ , GMRES-DR, and multigrid preconditioned FGMRES using CGNE as a coarse grid solver. A restart length of m = 200 and k = 40 eigenvectors were needed in GMRES-DR for all masses and lattice sizes in order for the calculation to converge in a reasonable amount of time. To compare the efficacy of coarse grid vs. fine grid deflation, we chose the same parameters for our coarse grid deflation for a strict comparison, as we are then dealing with the same deflation space. For our comparisons with an intermediate partial solve, it was found that using the same residual tolerances for CGNE as were used for our deflated coarse grid solve resulted in a large increase in outer iterations of FGMRES. To keep the number fine matrix vector products consistent with one another, CGNE was run to a tolerance of  $10^{-8}$  for all iterations. This potentially could have been avoided through the inclusion of a W-cycle, where the coarse grid is visited twice during every outer iteration, as is performed in [7]. Since we aim to reduce the cost of the coarse grid, this method was avoided. Throughout this study, Z(4)noise vectors were used as a right hand side, and all reported values are averaged over ten solves. Due to the use of random gaussian noise vectors in the adaptive coarse grid operator formation, no coarse grid operator is identical, even if they come from the same original Dirac operator. To that end, the performance of each solve is also slightly different. In order to obtain statistically relevant results, we perform our simulations with ten gauge configurations and average our results.

An indication that critical slowing down has been relayed to the coarsest grid level is a constant number of fine grid operator applications as the lattice size increases and the mass gap decreases towards zeros. The mass gap is defined as  $m_{gap} = m - m_{crit}$  where  $m_{crit}$  is the value at which the quark mass is zero. As the mass gap decreases to zero, exceptional eigenvalues arise in the eigenspectra of the Wilson-Dirac operator that contribute to slowing down. We first consider results for our deflated solve without a partial intermediate solve. Since we only smooth on the intermediate level, the number of intermediate grid operator applications is constant irrespective of fermion mass and lattice size.

Figure 3.8 (top) displays the number of fine grid operator applications, or matrix vector products (Mvps) as a function of the mass gap for our deflated multigrid method. The number of fine mvps is nearly constant for all three lattice sizes and mass values, indicating that critical slowing down has been shifted to the coarse grid level. Figure 3.8 (bottom) shows the number of coarse grid matrix vector products averaged over the number of outer iterations. The number of coarse operator applications increasing drastically as the mass gap approaches zero, indicative of critical slowing down. Deflation significantly reduces the number of coarse operator applications. In the most dramatic case, deflation has reduced the average number of coarse operator applications for the  $256^2$  lattice to that of the  $64^2$  lattice without deflation.

To examine the total cost of the calculation, we recast the intermediate and coarse operator applications in terms of fine equivalent mvps:

$$Fine \ Equivalent \ Mvps = Fine \ Mvps + \frac{n_{int}}{n_{fine}} \times Int \ Mvps + \frac{n_{coarse}}{n_{fine}} \times Coarse \ Mvps$$
(3.22)

where  $n_{fine}$ ,  $n_{int}$  and  $n_{coarse}$  are the size of the of the Dirac operator for the fine, intermediate and coarse grids, respectively. Figure 3.9 shows the performance of all iterative solvers under consideration for the 256<sup>2</sup> lattice, cast in terms of fine Dirac operator application equivalence. When evaluated in terms of fine equivalent mvps, non deflated MG is nearly as expensive as CGNE. Performing a deflated solve on the coarse grid significantly reduces the number of fine equivalent mvps. It outperforms MG without deflation and is more effective than pure deflation on the finest grid.



Figure 3.8: (Top) The number of fine Dirac operator applications as a function of the mass gap for all three lattice sizes. (Bottom) The average number of coarse Dirac operator applications per outer iteration.



Figure 3.9: A comparison of fine equivalent Mvps as a function of the mass gap for CGNE, GMRES-DR, MG and deflated MG.

Coarse grid deflation has a surprising effect when it is observed over multiple right hand sides. Figure 3.10 displays a comparison of coarse grid deflated multigrid to non deflated multigrid and GMRES-DR/GMRES-Proj for ten right hand sides for a lattice of size 256<sup>2</sup>. In order to obtain eigenvectors accurate enough for deflation, GMRES-DR was over converged to a tolerance of  $10^{-15}$  for the first right hand side, and subsequent right hand sides were solved to a tolerance of  $10^{-8}$ . This is mimicked in our deflation parameters for deflated coarse grid solves, however only the *the first outer iteration of the first right hand side* is solved to a tolerance of  $10^{-15}$ . Subsequent outer iterations and right hand sides are solved with GMRES-Proj to a tolerance of  $10^{-8}$ .



Figure 3.10: A comparison of fine equivalent Mvps for multiple right hand sides for GMRES-DR/GMRES-Proj, MG and deflated MG at critical mass.

fine grid for all right hand sides. Naïvely, we would expect traditional deflation and coarse grid deflated multigrid to be equivalent over subsequent right hand sides, since the solvers are now operating on the same deflated subspace. Since coarse grid deflated multigrid is more effective than GMRES-DR/GMRES-Project over subsequent right hand sides, this displays the surprising effect that coarse grid deflation and multigrid work in a synergistic fashion to obtain a performance that is better than either method can achieve on their own.

The previous discussion took place for results that do not include a partial solve on the intermediate grid, where the number of intermediate operator applications is a fixed number. We now consider results where the intermediate grid is solved to a fixed residual. Thus, it is necessary to examine the average number of intermediate operator applications to ensure that critical slowing down is not shifted back up to the intermediate grid. Figure 3.11 displays the fine (top) and average intermediate (bottom) operator applications for a lattice of size  $256^2$ . Performing an intermediate solve has reduced the number of fine grid operator applications for both deflated and non deflated multigrid. The number of average intermediate operator applications is constant for all reported values of the mass gap, indicating that critical slowing down has remained on the lowest grid level, despite the tolerance used in Table 3.2. Deflation can then be used to reduce the coarse grid solve even further through the use of a highly truncated solve while not sacrificing performance on the fine or intermediate grids. Figure 3.12 displays the average number of the coarse Dirac operator applications arising from the same system as Figure 3.11 for both the deflated and non deflated methods. As the mass gap approaches zero, the number of coarse operator applications increases as a very low power law, indicating that critical slowing down has been removed almost entirely from the coarse grid solve as well, not only from the fine and intermediate grids.

The reduction of the coarse grid solve can be seen through the convergence history of a representative calculation. Figures 3.13, 3.14, and 3.15 shows the convergence history and the contribution of all grids to the total cost of the calculation for



Figure 3.11: (Top) The number of fine Dirac operator applications as a function of mass gap for both deflated and non deflated solves with the inclusion of a partial intermediate level solve. (Bottom) The average number of intermediate operator applications as a function of mass gap.



Figure 3.12: The average number of coarse Dirac operator applications as a function of mass gap for both deflated and non deflated solves with the inclusion of a partial intermediate level solve.

an optimized calculation for lattices of size  $64^2$ ,  $128^2$  and  $256^2$ , respectively. These calculations were done with a Dirac operator having a mass gap of  $10^{-4}$ , a restart length of m = 20 and k = 10 eigenvectors deflated from the coarse grid. While the large restart length and large number of deflated eigenvectors was used for direct comparison to traditional deflation, the calculation is more efficient with smaller Krylov parameters. Using a smaller restart length and deflation length has the additional benefit of reducing the storage costs of the coarse grid. For inversions of the Dirac operator arising from very large lattices, the coarse grid operator may still be very large and have a significant storage requirement. Smaller Krylov parameters are thus advantageous in this regard. The tolerances for the run are those from Table 3.2 for both the deflated and non deflated solves. For the  $256^2$  lattice, it can be observed that the non deflated method did not converge in the maximum allowed number of cycles.



Figure 3.13: The convergence history and individual grid cost for a representative Wilson-Dirac operator from the  $64^2$  lattice.



Figure 3.14: The convergence history and individual grid cost for a representative Wilson-Dirac operator from the  $128^2$  lattice.

This optimized solve reveals an additional trait of deflated multigrid preconditioners. Because the tolerances of the coarse grid operator solve are the same for both the deflated and non deflated preconditioners, we would expect that the solutions arising from the coarse grid inversion are the same, irrespective of the iterative solver. Figure 3.15 demonstrates that this is not the case. In the non deflated case, the number of V-cycles, and thus the number of outer iterations, is dramatically larger than that of the the deflated case. This demonstrates that deflated multigrid preconditioners possess superior inherent performance in comparison to its non deflated counterpart.

Figure 3.16 displays a comparison of the convergence history for the coarse grid deflated solver for all lattice sizes. Each operator arising from these lattices is 4 times larger than the last. Despite the large increase in operator size, the deflated coarse grid solver has similar performance for each of the lattice sizes.



Figure 3.15: The convergence history and individual grid cost for a representative Wilson-Dirac operator from the  $256^2$  lattice.



Figure 3.16: A comparison of the convergence history for the deflated coarse grid solver for all lattice sizes used for representative operators.

Method	α
DEFLATED MG WITH PARTIAL SOLVES	$0.13 \pm 0.04$
DEFLATED MG	$0.25\pm0.05$
GMRES-DR	$0.48\pm0.06$
NONDEFLATED MG WITH PARTIAL SOLVE	$0.72\pm0.10$
CGNE	$0.76\pm0.10$
NONDEFLATED MG	$0.81\pm0.11$

Table 3.3: Approximate powers in  $Mvps = (size)^{\alpha}$ .

To examine the scaling of this method, we examine the performance of each iterative solver as a function of lattice size at constant mass gap of  $10^{-4}$ , shown in Figure 3.17. Our method of coarse grid deflation with partial intermediate solves exhibits the mildest dependence on lattice size. The estimate for *Fine equivalent mvps* =  $(L^2)^{\alpha}$  is given in Table 3.3. Deflated multigrid is thus a particular robust method to combat the effects of strong scaling. As the lattice size increases, the number of processors used for the inversion must necessarily increase. However, the speedup gained from the addition of more processors begins to deviate from linear [19]. Our method of deflated multigrid demonstrates that the lattice size dependence of the method is very mild, and thus a smaller number of processors could be used for the inversion.



Figure 3.17: Fine equivalent Mvps as a function of lattice size.
# CHAPTER FOUR

Deflation Methods for Noise Subtraction in Disconnected Diagrams

# 4.1 Introduction

Quark effects arising in disconnected diagrams are important contributors to many physical quantities in lattice QCD, such as form factors [5][4], scattering states [24] and the strange quark content of the nucleon [37][51][32]. The evaluation of disconnected diagrams in lattice QCD is one of the world's largest computing challenges. It is physically intractable to calculate all the necessary propagators arising from disconnected quark loops when very large lattices are used, so stochastic methods are used to estimate the propagators needed for the computation of correlator functions. The use of stochastic methods introduces inherent noise into the system, which increases the noise to signal ratio of these calculations, and at large time slices, the signal cannot be extracted in any statistically meaningful way. Despite this drawback, the stochastic estimation of propagators from disconnected diagrams is the only method that can be used to compute these important quantities. This necessitates the need for noise subtraction methods. In this chapter, I discuss how disconnected loops arise in lattice QCD and the basics of noise subtraction applied to these calculations. I also discuss our results for deflated noise subtraction techniques at zero quark mass in the quenched approximation, as well as with gauge configurations where the effect of dynamical sea quarks are taken into account.

#### 4.2 Disconnected Diagrams

In order to begin the discussion of disconnected diagrams, it is necessary to begin by discussing Euclidean correlators. The correlators are given by the relation

$$C(n_t) \equiv \langle O(n_t)\bar{O}(0)\rangle = \sum_k \langle 0|\hat{O}|k\rangle \langle k|\hat{O}^{\dagger}|0\rangle e^{-n_t a E_k}, \qquad (4.1)$$

where the *interpolators*, O and  $\overline{O}$ , are defined by the Hilbert space operators,  $\hat{O}$  and  $\hat{O}^{\dagger}$ , that annihilate and create the states from the vaccuum under consideration, and  $n_t$  is a time slice of the lattice. The correlator functions describe how particle states propagate through the lattice [30]. As an example, consider the meson interpolator for the neutral pion. Ignoring the contribution of the strange and charm sea quarks for simplicity, it is given by

$$O_{\pi^0}(n) = \frac{1}{\sqrt{2}} (\bar{u}(n)\Gamma u(n) - \bar{d}(n)\Gamma d(n))$$
(4.2)

where  $\Gamma = \gamma_5$ , u and d and the fields of the up and down quarks, respectively. Making the spin and color indices explicit, the correlator is then given by

$$\langle O(n)\bar{O}(m)\rangle = \frac{1}{2} \langle \bar{u}(n)_{c_1}^{\alpha_1} \Gamma_{\alpha_1\beta_1} u(n)_{c_1}^{\beta_1} \bar{u}(m)_{c_2}^{\alpha_2} \Gamma_{\alpha_2\beta_2} u(m)_{\beta_2}^{\beta_2} \\ -\bar{u}(n)_{c_3}^{\alpha_3} \Gamma_{\alpha_3\beta_3} u(n)_{\beta_3}^{\beta_3} \bar{d}(m)_{c_4}^{\alpha_4} \Gamma_{\alpha_4\beta_4} d(m)_{\beta_4}^{\beta_4} \\ -\bar{d}(n)_{c_5}^{\alpha_5} \Gamma_{\alpha_5\beta_5} d(n)_{\beta_5}^{\beta_5} \bar{u}(m)_{c_6}^{\alpha_6} \Gamma_{\alpha_6\beta_6} u(m)_{\beta_6}^{\beta_6} \\ -\bar{d}(n)_{c_7}^{\alpha_7} \Gamma_{\alpha_7\beta_7} d(n)_{\beta_7}^{\gamma_7} \bar{d}(m)_{c_8}^{\alpha_8} \Gamma_{\alpha_8\beta_8} d(m)_{\beta_8}^{\beta_8} \rangle$$

$$(4.3)$$

We can reorder the fields and use the fact that the fermionic expectation values factorize with respect to quark flavors [30], and use Wick's Theorem to obtain the correlator in terms of the two point functions for the up and down quarks to obtain

$$\langle O(n)\bar{O}(m)\rangle = -\frac{1}{2}\Gamma_{\alpha_{2}\beta_{2}}D_{u}^{-1}(m|n)_{\beta_{2}\alpha_{1}}\Gamma_{\alpha_{1}\beta_{1}}D_{u}^{-1}(n|m)_{\beta_{1}\alpha_{2}} + \frac{1}{2}\Gamma_{\alpha_{1}\beta_{1}}D_{u}^{-1}(n|n)_{\beta_{1}\alpha_{1}}\Gamma_{\alpha_{2}\beta_{2}}D_{u}^{-1}(m|m)_{\beta_{2}\alpha_{2}} + \frac{1}{2}\Gamma_{\alpha_{3}\beta_{3}}D_{u}^{-1}(n|n)_{\beta_{3}\alpha_{3}}\Gamma_{\alpha_{4}\beta_{4}}D_{d}^{-1}(m|m)_{\beta_{4}\alpha_{4}} + u \leftrightarrow d$$

$$(4.4)$$

where  $D_u^{-1}$  and  $D_d^{-1}$  are the inverses of the Dirac operator, i.e. the propagators of the up and down quarks, respectively. Summing over indices, yields

$$\langle O(n)\bar{O}(m)\rangle = -\frac{1}{2}Tr[\Gamma D_{u}^{-1}(m|n)]Tr[\Gamma D_{u}^{-1}(n|m)]$$

$$+\frac{1}{2}Tr[\Gamma D_{u}^{-1}(n|n)]Tr[\Gamma D_{u}^{-1}(m|m)]$$

$$+\frac{1}{2}Tr[\Gamma D_{u}^{-1}(n|n)]Tr[\Gamma D_{d}^{-1}(m|m)] + u \leftrightarrow d$$

$$(4.5)$$

where the trace is taken in color-spin space. The first term in equation 4.5 is a *connected* piece. An up quark is propagated from the lattice point m to the lattice point n, and an anti-up quark is propagated from n to m. The other two terms are referred to as *disconnected* pieces. A quark is propagated from a lattice point back to the same lattice point. The final term is the exchange terms when the up and down quarks are interchanged. Figure 4.1 gives an illustrative example of the difference between connected and disconnected contributions. In the quenched approximation,



Figure 4.1: (Left) A connected piece of a meson correlator from lattice site m to n. (Right) A disconnected piece of a meson correlator from lattice sites m and n.

the contributions of disconnected pieces are disregarded. This is done by setting the fermionic determinant to one. The evaluation of the disconnected pieces arising in the calculation of correlation functions requires the calculation of the quark propagators from all spatial coordinates to all spatial coordinates within the lattice [2]. This would require  $N_s^3$  inversions of the Dirac operator, where  $N_s$  is the spatial extent of the lattice. The cost of such a calculation is prohibitively expensive. When the contribution of disconnected pieces are desired, stochastic methods to estimate the trace are used.

### 4.3 Stochastic Trace Estimators

The trace of the inverse Dirac operator arising in correlators can be estimated stochastically using noise vectors. Given the linear system:

$$Dx = \eta \tag{4.6}$$

with  $\eta$  a noise vector, the expectation value of any element of  $D^{-1}$  can be extracted as

$$D_{ik}^{-1} = \lim_{N \to \infty} \frac{1}{N} \sum_{n}^{N} x_{i}^{(n)} \eta_{k}^{*(n)}$$

$$= \langle x_{i} \eta_{k} \rangle.$$
(4.7)

Since the exact matrix element can only be obtained in the limit of an infinite number of noise vectors, the number of noises must be truncated at a finite value. This truncation is the source of the noise arising in signals from disconnected loop calculations, necessitating the need for variance reduction methods.

The variance in the trace estimation can be shown to only depend on the off diagonal elements of some matrix Q if Z(4) noise vectors are used. The entries of Z(4) noise vectors are that of the points of the unit circle in the complex plane. Using specific properties of Z(4) noise vectors, it can be shown [11] that the variance of the trace estimation is given by

$$V\left[Tr\left(QX_{Z(N\geq3)}\right)\right] = \frac{1}{N}\sum_{i\neq j}|q_{ji}|^{2}.$$
(4.8)

where X is the projection matrix formed by taking the outer product of the noise vectors. The goal then is to form a traceless matrix,  $\tilde{Q}$ , that approximates the off diagonal elements of Q, such that

$$\langle Tr(QX) \rangle = \left\langle Tr\left\{ \left( Q - \tilde{Q} \right) X \right\} \right\rangle$$

$$(4.9)$$

However, it is very difficult to form traceless matrices in lattice QCD due to their size. The practical method is then to form an approximation to inverse Wilson-Dirac operator, and add back the trace of the approximation to ensure the trace remains unchanged under the subtraction correction.

# 4.4 Noise Subtraction Techniques

The expectation value of any operator in lattice QCD is given by the equation

$$\langle \bar{\psi} \Theta \psi \rangle = -Tr(\Theta D^{-1}),$$
(4.10)

where  $\Theta$  is the operator under consideration. Using noise vector subtraction techniques, the trace takes the form

$$Tr\left(\Theta D^{-1}\right) = \frac{1}{N} \sum_{n}^{N} \left(\eta^{(n)\dagger}\Theta\left(x^{(n)} - \tilde{x}^{(n)}\right)\right) + Tr\left(\Theta \tilde{D}^{-1}\right).$$
(4.11)

where N is the number of noise vectors being used,  $x^{(n)}$  is the solution to equation 4.3, and  $\tilde{x^n}$  is the approximate solution formed using the approximation to the operator inverse,  $\tilde{D^{-1}}$ . This calculation then involves the inversion of the Dirac operator N times. There are a variety of ways to form  $\tilde{D^{-1}}$ , and the way in which it is formed defines the type of subtraction. The subtraction methods demonstrated are Perturbative Subtraction (PS), Polynomial Subtraction (POLY), Eigenvalue Subtraction (ES), Hermitian Forced Eigenvalue Subtraction (HFES), and the two combination methods, Hermitian Forced Pertubative Subtraction (HFPS) and Hermitian Forced Polynomial Subtraction (HFPOLY). For self-consistency, we list the various forms that the variance reduced trace estimation takes. For a detailed derivation of these terms, see [10][11][12][33][34][38][54].

$$PS: \frac{1}{N} \sum_{n}^{N} \left( \eta^{(n)\dagger} \cdot \Theta \left( x^{(n)} - \tilde{x}_{pert}^{(n)} \right) \right) + Tr \left( \Theta \tilde{D}_{pert}^{-1} \right)$$
(4.12)

$$POLY: \frac{1}{N} \sum_{n}^{N} \left( \eta^{(n)\dagger} \Theta \left( x^{(n)} - \tilde{x}_{poly}^{(n)} \right) \right) + Tr \left( \Theta \tilde{D}_{poly}^{-1} \right)$$
(4.13)

$$ES: \frac{1}{N} \sum_{n}^{N} \left( \eta^{(n)\dagger} \Theta \left( x^{(n)} - \tilde{x}_{eig}^{(n)} \right) \right) + Tr \left( \Theta \tilde{D}_{eig}^{-1} \right)$$
(4.14)

$$HFES: \frac{1}{N} \sum_{n}^{N} \left( \eta^{(n)\dagger} \Theta \left( x^{(n)} - \tilde{x}_{eig}^{\prime(n)} \right) \right) + Tr \left( \Theta \gamma_5 \tilde{D}_{eig}^{\prime-1} \right)$$
(4.15)

$$HFPS: \frac{1}{N} \sum_{n}^{N} \left( \eta^{(n)\dagger} \left[ \Theta x^{(n)} - \Theta \tilde{x}'_{eig}^{(n)} - \left( \Theta \tilde{x}_{pert}^{(n)} - \Theta \tilde{x}'_{eigpert}^{(n)} \right) \right] \right) + Tr \left( \Theta \gamma_5 \tilde{D'}_{eig}^{-1} \right) + Tr \left( \Theta \tilde{D}_{pert}^{-1} - \Theta \gamma_5 \tilde{D'}_{eigpert}^{-1} \right)$$

$$(4.16)$$

$$HFPOLY : \frac{1}{N} \sum_{n}^{N} \left( \eta^{(n)\dagger} \left[ \Theta x^{(n)} - \Theta \tilde{x}'_{eig}^{(n)} - \left( \Theta \tilde{x}_{poly}^{(n)} - \Theta \tilde{x}'_{eigpoly}^{(n)} \right) \right] \right) + Tr \left( \Theta \gamma_5 \tilde{D'}_{eig}^{-1} \right) + Tr \left( \Theta \tilde{D}_{poly}^{-1} - \Theta \gamma_5 \tilde{D'}_{eigpoly}^{-1} \right)$$
(4.17)

The last four subtraction methods are the deflation type subtraction methods. The ES method uses the low lying eigenpairs of the non hermitian Wilson-Dirac operator, while the others use the low lying eigenvectors of the hermitian Wilson operator. The Wilson operator can be put into a hermitian form through the use of the  $\gamma_5$  operator in the following fashion:

$$D' = D\gamma_5 \tag{4.18}$$

It is important to do the multiplication by  $\gamma_5$  on the right in order to preserve cyclic properties of the trace that fail in finite noise space [12]. It should be noted here that the non normality of the Wilson-Dirac operator results in an increase of the variance when using the non hermitian eigenpairs [33]. The form that the corrected trace estimators take all involve a trace correction term, due to the difficulty of forming a traceless matrix. In the case of the combination type methods, an additional correction term is necessary so that the eigenspace of the polynomial approximation does not overlap with the eigenspace of the approximation using the lowest lying hermitian Wilson operator eigenpairs. If this is not properly accounted for, then the variance of the calculation would be increased.

#### 4.5 Variance Reduction at Zero Quark Mass

As mentioned previously, the Wilson-Dirac operator becomes ill conditioned as the value of the bare quark mass approaches zero. The value of the hopping parameter at which the quark mass is zero is referred to as  $\kappa_{critical}$ . In this section, we report results for subtraction methods, averaged over ten gauge configurations, which take place using the value of the hopping parameter corresponding to  $\kappa_{critical} \approx 0.1570$  in the quenched approximation for a lattice of size  $24^3 \times 32$ . This lattice corresponds to a Wilson-Dirac operator of size n = 5,308,416. In the quenched approximation, the value of the fermion determinant is set to one during the generation of the gauge configurations. This results in the background fermion disconnected loop contributions not being accounted for, however it is a decent approximation. 200 Z(4) noise vectors were used for the calculation of the error. The eigenpairs of the non-hermitian Wilson Dirac operator, and the solution to the first right hand side were calculated using GMRES-DR(200,160). These eigenpairs were used for both the ES method and to accelerate the convergence of subsequent right hand sides using GMRES-Proj(200,160). The eigenpairs of the hermitian Wilson-Dirac operator were calculated with MINRES-DR(200,160), and were used for the HFES, HFPS and HFPOLY methods. It should be noted that the calculation of the low lying eigenpairs of the hermitian Wilson-Dirac operator is a particularly difficult numerical problem. Figure 4.2 shows the convergence of the linear equations for both the hermitian and non hermitian operators. The spectra of the hermitian Wilson-Dirac operator is maximally indefinite, with many eigenvalues close to zero, making the calculation of the low lying eigenspace difficult. The convergence of the eigenvalues can be observed in the "deflation knee" present in the convergence history of the linear equations.

Name	Representation	Total operators
Scalar	$Re[ar{\psi}(x)\psi(x)]$	1
Local Vector	$Im[ar{\psi}(x)\gamma_{\mu}\psi(x)]$	4
Point-Split Vector	$\kappa Im \left[ \bar{\psi}(x+a_{\mu})(1+\gamma_{\mu})U^{\dagger}_{\mu}(x)\psi(x) \right]$	4
	$-\kappa Im\left[\bar{\psi}(x)(1-\gamma_{\mu})U_{\mu}(x)\psi(x+a_{\mu})\right]$	

Table 4.1: Names of calculated operators,  $\Theta$ , and their field representations.



Figure 4.2: The performance of the MINRES and GMRES algorithms at  $\kappa_{crit}$  on the hermitian and nonhermitian systems.

The operators under consideration for this study are given in Table 4.1, which respond best to unparticle, or undiluted, noise vectors. In [58], it is shown that the ratio of the variance for partitioned noise vectors to the variance of unpartitioned noise vectors is large for the scalar, local vector and point-split vector operators, meaning that these three operators experience better variance reduction if they are not subjected to partitioning based on some degree of freedom within the lattice. In contrast, it is found that the pseudoscalar, local axial, point-split axial and tensor operators exhibit variance reduction when they are subjected to some partitioning pattern. Thus, we choose to examine the scalar, local vector and point-split vector operators so that we can see the variance reduction due to only our subtraction methods.

While the local and point-split (or non-local) vectors have an operator for each spacetime direction, the results are independent of the dimension, so only the oper-

ators in the time direction are reported. Each of the subtraction methods and their representation are listed in table 4.2.

Symbol	Method	Representation
NS	No Subtraction	Blue Bursts
$\mathbf{ES}$	Eigenspectrum Subtraction	Green Circles
HFES	Hermitian Forced - Eigenspectrum Subtraction	Magenta Crosses
$\mathbf{PS}$	7th Order Perturbative Subtraction	Red Bursts
POLY	7th Order Polynomial Subtraction	Green Bursts
HFPS	HFES and PS combination	Black Diamonds
HFPOLY	HFES and POLY combination	Pink Diamonds

Table 4.2: The legend on each subtraction plot and their representation.



Figure 4.3: The standard error for the local vector operator.

The standard error as a function of the number of deflated eigenvectors for the local vector, point-split vector and scalar operator are reported in figures 4.3, 4.4 and 4.5, respectively. Because the quark mass is inversely related to the value of the



Figure 4.4: The standard error for the point-split operator.



Figure 4.5: The standard error for the scalar operator.

hopping parameter, the perturbative expansion of the approximation to the inverse of the Wilson-Dirac operator becomes less accurate. This results in the standard error increasing as the quark mass approaches its physical values, and at  $\kappa_{critical}$ , the standard error has approached that of no subtraction. In contrast, the hermitian deflation methods are not subject to the same approximation, and the standard error reduction becomes better at  $\kappa_{critical}$ .

A number of intriguing features can be observed from the results at zero quark mass. The first and foremost is that of the discontinuity between NS, PS, and POLY with their deflated counterparts. The large discontinuity indicates that the variance is mostly dominated by the low lying eigenmodes. Secondly, the standard error plateaus at approximately 30 eigenvectors. This phenomena is referred to as "deflation saturation": the variance is saturated by the low lying eigenmodes, and the higher modes do not contribute as much. Lastly, there appears to be a synergy in the variance reduction when a subtraction method is combined with deflation, most significantly for the scalar operator. The separation between the error of HFPOLY and HFPS is larger than that between PS and POLY, their non deflated counterparts.

We define the relative efficiency, RE, of two methods as

$$RE \equiv \left(\frac{1}{\delta y^2} - 1\right) \times 100, \tag{4.19}$$

where  $\delta y$  is the relative error bar ratio and  $\delta y^2$  is the relative variance. Because the standard error goes as the square root of the variance, the relative efficiency gives a direct comparison between two methods. Table 4.3 gives the relative efficiencies for the POLY, HFES, HFPS and HFPOLY subtraction methods. HFPOLY sees the most dramatic reduction in the standard error, both in comparison to no subtraction and the standard method of perturbative subtraction. The relative efficiency is greater than 1000% for all operators, meaning the variance has been reduced by over an

Subtraction	Scalar		Local $J_4$		Point-Split $J_4$	
Type	vs. NS	vs. PS	vs. NS	vs. PS	vs. NS	vs. PS
POLY	8.9%	2.8%	16.4%	0.1%	49.5%	1.1%
HFES	634%	593%	496%	413%	180%	89.2%
HFPS	972%	911%	1970%	1680%	1800%	1180%
HFPOLY	1350%	1270%	2070%	1770%	2220%	1470%

 Table 4.3: Comparison of relative efficiencies for operators with quenched configurations.

order of magnitude in comparison to no subtraction and perturbative subtraction. Because the variance is proportional to computer time, the same calculation can be performed an order of magnitude faster than with conventional methods.

# 4.6 Variance Reduction with Dynamical Configurations

Our subtraction methods were also tested on configurations generated by the MILC collaboration [13]. These configurations were generated using  $N_f = 2 + 1 + 1$  flavors of dynamical sea quarks using the Highly Improved Staggered Quark action [28], with a pion mass of 306.9(5) MeV, on a lattice of size  $16^3 \times 48$ . In the dynamical case, the value of the fermion determinant is allowed to take on its true value, and is a better representation of the full QCD theory. An analysis of the pion correlators in the quenched approximation at three different values of the hopping parameter for all ten configurations was performed to determine the value of the hopping parameter corresponding the mass of the pion for these configurations. The mass of the pion can be extracted from the correlator through the use of the effective mass plateau [30]. Because equation 4.1 can be written as an expansion of energy states, the effective mass of the ground state is given by the ground state energy. The effective mass can therefore be extracted via

$$m_{eff}(n_t + \frac{1}{2}) = ln \frac{C(n_t)}{C(n_t + 1)}$$
(4.20)

The values of the masses are then fit to the equation [14] that relates the mass of the pion in dimensionless units to the hopping parameter

$$(m_{\pi}a)^2 = B(\frac{1}{\kappa} - \frac{1}{\kappa_{critical}}), \qquad (4.21)$$

where B is a constant.



Figure 4.6: The mass of the pion in dimensionless units as a function of the hopping parameter.

Figure 4.6 displays the mass of the pion in dimensionless units for three different values of the hopping parameter, with the error bars determined from jackknife analysis. The value of the hopping parameter corresponding to the mass of the pion for these configurations determined from the fit was found to be  $\kappa = 0.1453$ , with  $\kappa_{critical} = 0.1481$ . This value of kappa roughly corresponds to  $\kappa \approx 0.1567$  in the quenched case [26]. Other than the value of the hopping parameter, the subtraction methods were performed in the same manner as in the quenched case.



Figure 4.7: The standard error for the local vector operator.



Figure 4.8: The standard error for the point-split operator.



Figure 4.9: The standard error for the scalar operator.

 Table 4.4: Comparison of relative efficiencies for operators with the dynamical configurations

Subtraction	Scalar		Loca	al $J_4$	Point-Split $J_4$	
Type	vs. NS	vs. PS	vs. NS	vs. PS	vs. NS	vs. PS
POLY	22.8%	6.6%	35.0%	-0.1%	93.4%	5.2%
HFES	134%	104%	120%	62.4%	60.0%	-13.2%
HFPS	192%	153%	332%	220%	417%	181%
HFPOLY	260%	217%	436%	230%	505%	229%

Figures 4.7, 4.8 and 4.9 show the standard error for the lattices generated with dynamical sea quarks. A similar but less pronounced discontinuity is observed between the deflated subtraction methods and their non deflated counterparts. While the analogous value of the hopping parameter in the quenched case is near  $\kappa_{critical}$ , this value of  $\kappa$  corresponds to an unphysically large pion mass. It is common to perform simulations at larger than physical pion mass in lattice QCD due to the effects of

critical slowing down. Results can then be extrapolated to the physical limit. Because similar characteristics to quenched zero mass subtraction methods at large pion mass with dynamical sea quarks, it may indicate that deflation is even more effective when contributions from dynamical sea quarks are taken into effect. We would then expect deflated subtraction methods to be even more effective when using gauge ensembles with dynamical sea quarks at physical pion mass.

Table 4.4 displays the relative efficiencies for the POLY, HFES, HFPS and HF-POLY subtraction methods. HFPOLY remains the most effective method for reducing the standard error across all operators. Similar qualitative behavior is observed between the quenched and dynamical cases. It should be noted that the non normality of the non hermitian case (ES) is greater than that in the quenched case.

# CHAPTER FIVE

### Conclusions

# 5.1 Deflated GMRES in Multigrid for Lattice QCD

We have demonstrated that the use of coarse grid deflation has many benefits to computing the quark propagator. While not only reducing the cost of the coarse grid solve, a deflated multigrid preconditioner displays some surprising characteristics. 1). The performance of a deflated preconditioner displays a better performance than a predeflated fine grid operator, as seen through an analysis of multiple right hand sides. 2) Deflated multigrid preconditioners display an increased performance in comparison to non deflated multigrid preconditioners. This effect is not limited to the reduction of coarse grid matrix vector products. While keeping all parameters constant between the deflated and non deflated preconditioners, deflated multigrid preconditioners display a large improvement over its non deflated counterpart. Even though the solution from the coarse grids in both cases should be theoretically the same, the deflated preconditioner sees no increase in the number of outer iterations, while the same is not true for its non deflated counterpart. 3) The deflated preconditioner also demonstrates very mild scaling in terms of lattice size dependence in comparison to conventional iterative solvers. This is an improvement in overcoming the strong scaling problem.

Multigrid is not only constrained to calculating approximate solutions of linear systems. It can also be used to accelerate eigenpair computation within lattice QCD. In the work of [29], a multigrid technique based on domain decomposition was used to calculate the eigenpairs of the hermitian Wilson-Dirac operator by solving the correction equation of shifted linear systems on coarser grids. A similar technique may be possible by using the Full Approximation Scheme, which approximates the fine grid solution on coarse grid levels, rather than solving the correction equation.

#### 5.2 Variance Reduction at Zero Quark Mass

We have demonstrated the benefits of deflation in regards to the variance of the stochastic trace estimators arising from lattice QCD, in particular, that of the Wilson-Dirac operator. While the lowest eigenmodes of the non hermitian Wilson-Dirac operator do not exhibit variance reduction due to their non normality, the lowest lying eigenmodes of the hermitian Wilson-Dirac operator are very effective at reducing the variance of the trace estimation in the quenched approximation. In particular, deflation methods in combination with other variance reduction techniques display a large reduction of the variance. At zero quark mass, the deflation type methods exhibit large discontinuities from their non deflated counterparts. This demonstrates that the variance of the trace estimation is dominated by the low lying eigenspace. The higher eigenmodes only slightly contribute to the variance, giving rise to the effect of *deflation saturation*. This is beneficial, as only a small number of low lying eigenmodes need be calculated in order to obtain significant variance reduction. In testing the effectiveness of our deflation methods on gauge configurations that include the effects of the dynamical sea quarks, it was found that the low lying eigenspace dominance is present to a greater degree. At the unphysical pion mass used for the generation of these configurations, the deflation type methods display a small discontinuity, indicating the presence of low eigenmode dominance. As the mass of the pion is decreased to the physical point, we expect that deflation type variance techniques will display even greater low eigenmode dominance, in analogy with the quenched case, and the variance reduction will be even greater.

While deflation methods show considerable benefit for variance reduction associated with stochastic trace estimators of lattice QCD, they are limited in their ability as only a small number of low lying eigenvalues can be feasibly calculated. New advancements in polynomial methods [39] show great promise. The polynomial methods in presented in this work are limited to small degrees due to numerical instability in the calculation of coefficients. Work in progress within our research group using the methods of [39] shows dramatic reduction in the variance, even for a very small number of noise vectors used.

Multigrid has also been shown to be effective in variance reduction for stochastic trace estimators within lattice QCD. In the work of [46], a multigrid correction scheme was used to reduce the variance of quantities arising from disconnected diagrams. This work employed similar correction methods to the fine grid solution as outlined in Chapter Three. A potential avenue is to use FAS in order to approximate the inverse to the Wilson-Dirac operator on the fine grid, rather than generate a correction to the fine grid solution.

#### 5.3 Acknowledgments

We thank the University Research Committee of Baylor University for their support of this project, whose funding secured release time for me to complete these projects. We thank Randy Lewis for the use of his QQCD program, and for Dr. Abdou Abdel-Rehim and Victor Guerrero for their contributions to the noise subtraction program. We thank Carlton DeTar and Doug Toussaint for making the MILC configurations available to us, and we thank Jim Hetrick for his helpful analysis of them. I would also like to thank Robert Edwards and Jianwei Qiu of The Center for Theoretical and Computational Physics at Thomas Jefferson National Laboratory for hosting me in the summer of 2019, and Raza Sufian, Luka Leskovic and Balint Joo for their helpful discussions during my time there. APPENDICES

# APPENDIX A

### Deflated GMRES Multigrid Preconditioned FGMRES Algorithm

This chapter will detail the deflated GMRES multigrid preconditioned FGM-RES algorithm (with an intermediate solve). In the following algorithms, "hat" notation is used to denote quantities on the intermediate grid, and "double hat" notation is used to denote quantities on the coarse grid. Steps of the algorithm invoking "Solve" are to be solved to a predetermined residual norm. Due to the length of the algorithm, it will be broken up into two parts: the three grid deflated GMRES V-cycle, given by Algorithm 12 and the overall preconditioned FGMRES algorithm, given by Algorithm 11. The output of  $Cycle(v_i)$  is the right preconditioner for FGMRES.

Algorithm 11: Deflated	l GMRES	Multigrid	Preconditioned	FGMRES
------------------------	---------	-----------	----------------	--------

(1)	Start: Choose an initial guess $x_0$ and dimensions of the fine, intermediate and coarse grid Krylov subspaces: $m_{fine}$ , $m_{int}$ and $m_{coarse}$ and the
	number of approximate coarse grid eigenvectors to deflate, $k_{coarse}$
(2)	Compute $r_0 = b - Ax_0$ , $\beta =   r_0  _2$ , and $v_1 = r_0/\beta$
(3)	For $j = 1,, m_{fine}$ , do:
(4)	Compute $z_j = Cycle(v_j)$
(5)	Compute $w := Az_j$
(6)	For $i = 1,, j$ do
(7)	$h_{i,j} := (w, v_i)$
(8)	$w := w - h_{i,j} v_i$
(9)	Compute $h_{j+1,j} =   w  _2$ and $v_{j+1} = w/h_{j+1,j}$
(10)	Compute $x_m = x_0 + Z_m y_m$ where $y_m = argmin_y   \beta e_1 - \bar{H}_m y  _2$
(11)	Compute the residual norm. If satisfied then stop, else GoTo 2

# Algorithm 12: $Cycle(v_j)$

- (1) Smooth on  $Az = v_j$  with GMRES $(\nu)$ . Restrict the residual to the intermediate grid with  $\hat{r} = P^{\dagger}(Az v_j)$
- (2) Smooth on  $\hat{A}\hat{z}_j = \hat{r}$  with GMRES $(\nu)$ .
- (3) Using  $\hat{z}$ , partially solve the system  $\hat{A}\hat{z}_j = \hat{r}$  with GMRES $(m_{int})$ . Restrict the residual to the coarse grid with  $\hat{r} = \hat{P}^{\dagger}(\hat{A}\hat{z}_j \hat{r})$
- (4) If j = 1
- (5) Solve  $\hat{A}\hat{z}_j = \hat{r}$  with GMRES-DR $(m_{coarse}, k_{coarse})$
- (6) Else
- (7) Solve  $\hat{A}\hat{z}_j = \hat{r}$  with GMRES-Proj $(m_{coarse}, k_{coarse})$
- (8) Prolong the coarse grid solution to the intermediate grid and add the error to the intermediate grid solution with  $\hat{z}_j = \hat{z}_j + \hat{P}\hat{z}$
- (9) Smooth on the system  $\hat{A}\hat{z}_j = \hat{r}$  with initial guess  $\hat{z}_j$  with GMRES( $\nu$ ).
- (10) Prolong and add the error to the solution on the fine grid:  $z_j = z_j + P\hat{z}_j$ .
- (11) Smooth on the system  $Az_j = r$  with initial guess  $z_j$  using GMRES( $\nu$ ).

# APPENDIX B

### Matlab Program for Coarsening the Wilson-Dirac Operator

In this section, I describe in detail the coarsening procedure using examples from the MATLAB code. As described in Chapter 3, forming a coarsened Wilson-Dirac operator consists of four phases:

- (1) Calculate near null vectors
- (2) Chirally double the near null vectors
- (3) Partition the chirally doubled near null vectors based on partitionings of the lattice
- (4) Form the prolongator (interpolator), P, and form the coarse grid operator

Each of these phases is performed in the function **coarsen\_parallel.m**. In this function, the creation of the interpolating matrix is parallelized using **parpool**, which significantly reduces the time for the creation of a coarsened operator for large lattices.

#### B.1 Near Null Vector Generation

The near null vectors are generated in the function **getnullvectors.m**. The solve of equation 3.12 is performed using the Conjugate Gradient method, using Matlab's built in function **pcg**.

```
% Output:
11
  % nullvec
                    : the array containing num near null vectors
13
                    : the total number of matrix vector products
  % totalmvps
14
   %
                       for the generation of near null vectors
15
16
  n = size(A, 1);
17
   nmax = 250;
18
   z = sparse(n, 1);
19
   nullvec = zeros(n,num);
20
   totalmvps = 0;
21
22
   % generate near null vectors
23
   for i = 1:num
24
        z0 = sqrt(0.5) * (randn(n,1)+1i*randn(n,1));
25
        \mathbf{b} = -\mathbf{A} \ast \mathbf{A}' \ast \mathbf{z}\mathbf{0};
26
        [z, \tilde{,}, \tilde{,}, iter] = pcg(A*A', b, tol, nmax, [], []);
27
        nullvec(:, i) = z0 + z;
28
        totalmvps = 2*iter + totalmvps;
29
   end
30
31
   % othonormalize with modified gram schmidt
32
   for i = 1:num
33
        for j = 1:i-1
34
        dot = nullvec(:, j) '* nullvec(:, i);
35
        \operatorname{nullvec}(:, i) = \operatorname{nullvec}(:, i) - \operatorname{dot} * \operatorname{nullvec}(:, j);
36
        end
37
      nullvec(:, i) = nullvec(:, i) / norm(nullvec(:, i));
38
   end
39
```

The right hand side of equation 3.12 is formed by multiplying -AA' onto a normal gaussian vector, z0. The linear equations are approximately solved to a tolerance specified by *tol*. The approximate solution, z, is then added to z0 to form the near null vector. This process is performed *num* times to form the  $n_{vec}$  near null vectors. The near null vectors are then globally orthonormalized using the modified Gram Schmidt procedure.

#### B.2 Chiral Doubling of the Near Null Vectors

In the lattice Schwinger model, the Wilson-Dirac operator acts on a vector space of size  $S \otimes 1$ , where the spin degrees of freedom are given by S = 2, and the 1 arises from the U(1) gauge group of the theory. The near null vectors then contain two complex numbers for every lattice site, and thus x, the near null vector, is  $\in \mathbb{C}^{n \times 1}$ , where  $n = 2 \times N^2$  and N is the number of lattice sites:  $N = N_s \times N_t$ . The chiral projectors,  $1 \pm \sigma_3$ , then act on a lattice site by site basis. The chiral doubling is performed in the function chiral double.m

```
function [cdparray, cdmarray] = chiral_double(nullvec, N, ...
1
                                                                   num, nv, ch)
2
3
  % Inputs:
4
  % nullvec
                    : the array containing "num" near null vectors
6
                    : the dimension of the lattice
   % N
7
  % num
                    : the number of near null vectors
8
  % nv
                    : factor to retain degrees of freedom
9
   \% ch
                    : the chiral doubling factor.
10
11
   % Outputs:
12
13
   % cdparray
                    : array containing "num" positively chirally
14
   %
                       doubled near null vectors
15
                  : array containing "num" negatively chirally
   % cdmarray
16
17
                       doubled near null vectors
18
19
   % form chiral projectors
20
   sig = [1 \ 0; \ 0 \ -1];
21
   pp = (1/2) * (eye(ch, ch) + sig);
22
   pm = (1/2) * (eye(ch, ch) - sig);
23
24
25
   k = 1;
26
   for i = 1:num
27
         \operatorname{cdp} = \operatorname{zeros}(\operatorname{ch}, \operatorname{nv}, (\operatorname{N}^2));
28
         \operatorname{cdm} = \operatorname{zeros}(\operatorname{ch}, \operatorname{nv}, (\operatorname{N}^2));
29
         null = reshape(nullvec(:, i), ch, nv, (N^2));
30
         for j = 1:((N^2))
31
              \operatorname{cdp}(:,:,j) = \operatorname{pp}*\operatorname{null}(:,:,j);
32
              cdm(:,:,j) = pm * null(:,:,j);
33
         end
34
         cdp = reshape(cdp, ch*nv*N^2, 1);
35
         \operatorname{cdm} = \operatorname{reshape}(\operatorname{cdm}, \operatorname{ch*nv*N^2}, 1);
36
         cdparray(:,k) = cdp;
37
         cdmarray(:,k) = cdm;
38
         k = k + 1;
39
   end
40
```

The variable *sig* is the Pauli matrix  $\sigma_3$ . The Matlab function **reshape** is used to reshape the near null vectors to a  $2 \times N^2$  array, where the leading dimension of the array is the spin degrees of freedom. The multiplication of the chiral projector operators can then take place in the natural formalism of the  $2 \times 2$  Pauli matrix,  $\sigma_3$ . This process of multiplication of the near null vectors is repeated until each near null vector has been multiplied by both spin projectors. The output of this function is the arrays *cdparray* and *cdmarray*, which contain the right and left handed chiral vectors, respectively.

#### B.3 Partitioning the Near Null Vectors

After the chiral doubling the near null vectors, they must be partitioned based on grids within the lattice. The partitioning is performed by first creating a 2D array that corresponds to the lattice site ordering of the 2D lattice.

```
function [grids] = getsubgrids (N, nsub, gridsz, numgrids)
1
  % Input:
2
3
  % N
                 : dimension of lattice
4
  % nsub
                 : number of grids in one dim
5
  % gridsz
                 : is the size of grid
6
  % numgrids
                 : the total number of grids
7
8
  % Output:
9
10
  % grids
                 : array of vectors containing
11
  %
                   partitioned lattice points
12
13
  grids = sparse(gridsz^2, numgrids);
14
  vec = [1:N^2];
15
  vec = reshape(vec, N, N);
16
  forcell = gridsz*ones(1, nsub);
17
18
  j = N:
19
  for i = 1:N
20
       newvec(i,:) = vec(:,j);
21
       j = j - 1;
22
  end
23
24
  B = mat2cell(newvec, forcell, forcell);
25
26
  k = 1;
27
  for i = nsub: -1:1
28
       for j = 1:nsub
29
            sgrid = cell2mat(B(i,j));
30
            sgrid = sgrid';
31
           ix = [gridsz: -1:1];
32
            sgrid = sgrid(:, ix);
33
            sgrid = reshape(sgrid, gridsz^2, 1);
34
```

The output is a sparse array of vectors whose elements correspond to the lattice points belonging to a particular grid within the lattice. The indexing of this array corresponds to a sequential ordering of the grids within the lattice. It is these vectors that can be used to index the chirally doubled near null vectors to partition them according to the grid within the lattice.

#### B.4 Forming the Coarsened Operator

The function **coarsened\_parallel.m** calls the full coarsening of the Wilson-Dirac operator. Sections B.1-B.3 outline the first three steps of forming the coarsened Wilson-Dirac operator. These steps are called in lines 43, 45 and 47 of the following function.

```
function [Ahat, P, finemvps] = coarsen parallel(A, N, 1, ...
1
                                    nsub, gridsz, numi, num, ...
2
                                    tol, numprocs)
3
4
  % Inputs:
\mathbf{5}
6
  % A
                   : l-level Wilson-Dirac operator
\overline{7}
  % N
                    lattice dim
8
                   1
  % 1
                   : the level of coarsening
9
  % nsub
                   : number of subgrids in one dim. Since the
10
                     lattice is isotropic , nsub = N/\,{\tt gridsz}
  %
11
  % gridsz
                   : the size of the subgrids in one dim
12
  % num
                    the number of desired near nullvectors for
13
  %
                     this coarsening
14
  % numi
                   : the number of near null vectors for the
15
  %
                     previous coarsening if l > 1
16
  % tol
                    the tolerance for the null vectors
                   :
17
  % numprocs
                   : the number of processors desired for
18
  %
                     assembling the prolongator, limited
19
  %
                     to number of cores per node
20
21
  % Outputs:
22
23
  \% Ahat
                 : l+1-level coarse operator
24
25 % P
                 : prolongator matrix from l-1 to l grid
```

```
% finemvps
               : matrix vector product counter
26
27
  % below retains appropriate degrees of freedom for
28
  % l-level coarsening
29
  if (1 = 1)
30
       nv = 1;
31
  elseif (l > 1)
32
       nv = numi;
33
  end
34
35
  % preallocations
36
  ch = 2; %chirality factor, corresponds to spin d.o.f
37
  cdparray = zeros(ch*nv*N^2,num);
38
  cdmarray = zeros(ch*nv*N^2,num);
39
  numgrids = N^2/gridsz^2;
40
  rowrest = num*numgrids*ch;
41
  R = sparse(size(A, 1), rowrest);
42
43
   [nullvec, finemvps] = getnullvectors (A, num, tol);
44
45
   [cdparray, cdmarray] = chiral double(nullvec, N, num, nv, ch);
46
47
   [grids] = getsubgrids(N, nsub, gridsz, numgrids);
48
49
  parpool ('local', numprocs)
50
51
  spmd(numprocs)
52
53
       ij = (numgrids*num*ch)/numprocs;
54
       jk = numgrids / numprocs;
55
56
       k = (labindex - 1)*ij + 1;
57
58
  tic()
59
  for j = (labindex - 1)*jk+1:(labindex*jk)
60
61
       for i = 1:num
62
63
       nullp = reshape(cdparray(:, i), ch*nv*N^2, 1);
64
       nullp = reshape(nullp, ch, nv, N^2);
65
       nullm = reshape(cdmarray(:, i), ch*nv*N^2, 1);
66
       nullm = reshape(nullm, ch, nv, N^2);
67
68
69
       rp = zeros(ch, nv, N^2);
70
       rm = zeros(ch, nv, N^2);
71
       rp(:,:,grids(:,j)) = nullp(:,:,grids(:,j));
72
       rm(:,:,grids(:,j)) = nullm(:,:,grids(:,j));
73
       rp = reshape(rp, ch * nv * N^2, 1);
74
       rm = reshape(rm, ch * nv * N^2, 1);
75
76
       R(:,k) = rp;
77
       R(:, k+1) = rm;
78
79
       if (mod(k+1,2*num) = 0)
80
         for ii = ((k+1)-2*num)+1:(k+1)
81
```

```
for jj = ((k+1)-2*num)+1:(ii-1)
82
                 83
84
             end
85
                 R(:, ii) = R(:, ii) / norm(R(:, ii));
86
          end
87
       end
88
89
           k = k + 2;
90
       end %i
91
   end %j
92
93
   end %spmd
^{94}
   Rout = R(:);
95
96
   P = sparse(size(A, 1), rowrest);
97
   for i = 1:numprocs
98
       P = P + cell2mat(Rout(i)):
99
   end
100
101
   delete(gcp('nocreate'))
102
   Ahat = P' * A * P;
103
104
   clear cdp cdm nullvec rp rm nullp nullm grids Rout R
105
```

It is prudent to mention the degrees of freedom parameter of the coarse operator, given by the variable nv in lines 29 - 33. Keeping track of the number of near null vectors used in a previous coarsening is necessary when using a different number of near null vectors to create coarsened operators. This is due to the fact that there are  $2 \times n_{vec}$  degrees of freedom per coarsened lattice site. If a coarsening is performed at l = 1, then nv = 1 as there are no previous near nullvectors. If l > 1, then nv must be equal to the previous amount of calculated near null vectors in order to account for the appropriate degrees of freedom for the coarsened lattice. This strategy is necessary when creating coarsened operators of l > 2, where the degrees of freedom on the coarse lattice may exceed those of the lattice correspond to the level above it unless the number of near null vectors is lowered.

The 3D arrays rp and rm are the partitioned chirally doubled near null vectors, who have been partitioned by using the *grids* vectors. Lines 71 - 72 has the effect of zeroing the elements of the chirally doubled near null vectors, if those elements of the vector do not correspond to the lattice site labelled by the elements of the vector grids. These are then assigned to the columns of the variable R. Lines 79 - 87 performs the orthonormalization of the columns of R if the columns of are belong to the same grid

Because the columns of R are being formed in parallel, the array R is being worked on by a different CPU core. In order to form the whole prolongator, the result of each CPU core calculation is added together to form the final prolongation matrix. The coarsened operator can then be formed using the simple relation  $\hat{D} = P'DP$ , given by line 102.

# REFERENCES

- [1] Multigrid methods. https://ocw.mit.edu/courses/mathematics/ 18-086-mathematical-methods-for-engineers-ii-spring-2006/ readings/am63.pdf.
- [2] A. Abdel-Rehim, C. Alexandrou, M. Constantinou, K. Hadjiyiannakou, K. Jansen, Ch. Kallidonis, G. Koutsou, and A. Vaquero Avilés-Casco. Direct evaluation of the quark content of nucleons from lattice qcd at the physical point. *Physical Review Letters*, 116(25), Jun 2016.
- [3] Abdou M. Abdel-Rehim, Ronald B. Morgan, Dywayne A. Nicely, and Walter Wilcox. Deflated and restarted symmetric lanczos methods for eigenvalues and linear equations with multiple right-hand sides. SIAM Journal on Scientific Computing, 32(1):129–149, 2010.
- [4] C. Alexandrou, S. Bacchio, M. Constantinou, J. Finkenrath,
   K. Hadjiyiannakou, K. Jansen, G. Koutsou, and A. Vaquero Aviles-Casco.
   Proton and neutron electromagnetic form factors from lattice qcd. *Physical Review D*, 100(1), Jul 2019.
- [5] C. Alexandrou, K. Hadjiyiannakou, G. Koutsou, A. OCais, and A. Strelchenko. Evaluation of fermion loops applied to the calculation of the mass and the nucleon scalar and electromagnetic form factors. *Computer Physics Communications*, 183(6):1215 – 1224, 2012.
- [6] W.E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. Quart. Appl. Math., 9:17–29, 1951.
- [7] R. Babich, J. Brannick, R. C. Brower, M. A. Clark, T. A. Manteuffel, S. F. McCormick, J. C. Osborn, and C. Rebbi. Adaptive multigrid algorithm for the lattice wilson-dirac operator. *Physical Review Letters*, 105(20), Nov 2010.
- [8] R. Babich, M. A. Clark, B. Joó, G. Shi, R. C. Brower, and S. Gottlieb. Scaling lattice qcd beyond 100 gpus. Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis on -SC '11, 2011.
- [9] Ronald Babich, James Brannick, Richard C. Brower, Michael A. Clark, Saul D. Cohen, James C. Osborn, and Claudio Rebbi. The role of multigrid algorithms for lqcd, 2009.

- [10] Suman Baral. Thomas Fermi model for mesons and new noise subtraction techniques in lattice QCD. PhD thesis, 2018.
- [11] Suman Baral, Travis Whyte, Walter Wilcox, and Ronald B. Morgan. Disconnected loop subtraction methods in lattice qcd. Computer Physics Communications, 241:64 – 79, 2019.
- [12] Suman Baral, Walter Wilcox, and Ronald B. Morgan. New noise subtraction methods in lattice qcd, Nov 04 2016.
- [13] A. Bazavov et al. Lattice QCD ensembles with four flavors of highly improved staggered quarks. *Phys. Rev. D*, 87:054505, 2013.
- [14] Khalil M. Bitar, T. DeGrand, R. Edwards, Steven Gottlieb, U. M. Heller, A. D. Kennedy, J. B. Kogut, A. Krasnitz, W. Liu, Michael C. Ogilvie, and et al. Hadron spectrum in qcd with valence wilson fermions and dynamical staggered fermions at6g2=5.6. *Physical Review D*, 46(5):2169–2178, Sep 1992.
- [15] Artan Borici. QCDLAB: Designing Lattice QCD Algorithms with MATLAB. 2006.
- [16] Achi Brandt. Algebraic multigrid theory: The symmetric case. Applied Mathematics and Computation, 19(1):23 – 56, 1986.
- [17] J. Brannick, R. C. Brower, M. A. Clark, J. C. Osborn, and C. Rebbi. Adaptive Multigrid Algorithm for the QCD Dirac-Wilson Operator. *PoS*, LATTICE2007:029, 2007.
- [18] James Brannick, Andreas Frommer, Karsten Kahl, Björn Leder, Matthias Rottmann, and Artur Strebel. Multigrid preconditioning for the overlap operator in lattice qcd. *Numer. Math.*, 132(3):463–490, 2016.
- [19] David Brayford, Momme Allalen, and Volker Weinberg. Extreme scaling of lattice quantum chromodynamics, 2013.
- [20] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge. Adaptive algebraic multigrid. SIAM Journal on Scientific Computing, 27(4):1261–1286, 2006.
- [21] Marian Brezina, Robert D. Falgout, S. Maclachlan, Thomas A. Manteuffel, Scott McCormick, and John W. Ruge. Adaptive smoothed aggregation (  $\alpha$  sa ) multigrid. 2005.
- [22] William Briggs, Van Henson, and Steve McCormick. A Multigrid Tutorial, 2nd Edition. 01 2000.

- [23] Richard C. Brower, Evan Weinberg, M. A. Clark, and Alexei Strelchenko. Multigrid algorithm for staggered lattice fermions.
- [24] J. et. al Bulava. Multi-hadron operators with all-to-all quark propagators. *PoS*, LATTICE2008:100, 2008.
- [25] John Burkardt. Multigrid poisson 1d: Multigrid solver for 1d poisson problem. https://people.sc.fsu.edu/~jburkardt/m\_src/multigrid\_poisson\_ 1d/multigrid\_poisson\_1d.html.
- [26] S. Cabasino et al. Beta=6.0 quenched Wilson fermions. Phys. Lett B, 258:195, 1991.
- [27] Saul D. Cohen, R. C. Brower, M. A. Clark, and J. C. Osborn. Multigrid Algorithms for Domain-Wall Fermions. PoS, LATTICE2011:030, 2011.
- [28] E. Follana, Q. Mason, C. Davies, K. Hornbostel, G. P. Lepage, J. Shigemitsu,
   H. Trottier, and K. Wong. Highly improved staggered quarks on the lattice with applications to charm physics. *Physical Review D*, 75(5), Mar 2007.
- [29] Andreas Frommer, Karsten Kahl, Francesco Knechtli, Matthias Rottmann, Artur Strebel, and Ian Zwaan. A multigrid accelerated eigensolver for the hermitian wilson-dirac operator in lattice qcd, 2020.
- [30] Christof Gattringer and Christian B. Lang. Quantum Chromodynamics on the Lattice, An Introductory Presentation. Springer, Germany, 1988.
- [31] Johannes Gerer. Gitter-qcd-test der hypothese einer effektiven restaurierung der chiralen symmetrie bei hadronenresonanzen, 2009.
- [32] M. Gong, A. Alexandru, Y. Chen, T. Doi, S. J. Dong, T. Draper, W. Freeman, M. Glatzmaier, A. Li, K. F. Liu, and Z. Liu. Strangeness and charmness content of the nucleon from overlap fermions on 2+1-flavor domain-wall fermion configurations. *Phys. Rev. D*, 88:014503, Jul 2013.
- [33] Victor Guerrero. Electric neutron polarizability and eigenspectrum subtraction techniques for disconnected quark loops. PhD thesis, 2011.
- [34] Victor Guerrero, Ronald B. Morgan, and Walter Wilcox. Eigenspectrum noise subtraction methods in lattice qcd, Jan 25 2010.
- [35] M.R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. Journal of Research of the National Bureau of Standards, 49(6), Dec 1952.

- [36] Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of* the National Bureau of Standards, 45, Oct 1950.
- [37] Randy Lewis, W. Wilcox, and R. M. Woloshyn. Nucleon's strange electromagnetic and scalar matrix elements. *Phys. Rev. D*, 67:013003, Jan 2003.
- [38] Quan Liu. Thomas-Fermi quark model and techniques to improve lattice QCD calculation. PhD thesis, 2014.
- [39] Jennifer A. Loe and Ronald B. Morgan. New polynomial preconditioned gmres.
- [40] Martin Lüscher. Local coherence and deflation of the low quark modes in lattice QCD. Journal of High Energy Physics, 2007(07):081–081.
- [41] Ronald B. Morgan. A restarted gmres method augmented with eigenvectors. SIAM Journal on Matrix Analysis and Applications, 16(4):1154–1171, 1995.
- [42] Ronald B. Morgan. Gmres with deflated restarting. SIAM Journal on Scientific Computing, 24(1):20–37, 2002.
- [43] Ronald B. Morgan, Travis Whyte, Walter Wilcox, and Zhao Yang. Two-grid deflated krylov methods for linear equations, 2020.
- [44] Ronald B. Morgan and Walter Wilcox. Deflated iterative methods for linear equations with multiple right-hand sides, 2007.
- [45] Flavio S. Nogueira and Hagen Kleinert. Compact quantum electrodynamics in 2+1 dimensions and spinor deconfinement: A renormalization group analysis. *Physical Review B*, 77(4), Jan 2008.
- [46] Eloy Romero, Andreas Stathopoulos, and Kostas Orginos. Multigrid deflation for lattice qcd. 2019.
- [47] Youcef Saad. A flexible inner-outer preconditioned gmres algorithm. SIAM Journal on Scientific Computing, 14(2):461–469, 1993.
- [48] Youcef Saad and Martin H. Schultz. Gmres: a generalized minimal residual algorithm for solving nonsymmetric linear systems. 1986.
- [49] Yousef Saad. Iterative Methods for Sparse Linear Systems. Society for Industrial and Applied Mathematics, 2003.
- [50] Julian S. Schwinger. Gauge Invariance and Mass. 2. Phys. Rev., 128:2425–2429, 1962.

- [51] Raza Sabbir Sufian, Yi-Bo Yang, Andrei Alexandru, Terrence Draper, Jian Liang, and Keh-Fei Liu. Strange quark magnetic moment of the nucleon at the physical point. *Phys. Rev. Lett.*, 118:042001, Jan 2017.
- [52] M. et. al Tanabashi. Review of particle physics. Phys. Rev. D, 98:030001, Aug 2018.
- [53] Evan Weinberg. Leadership-class multigrid algorithms for hisq fermions on gpus. 37th International Symposium on Lattice Field Theory, Jun 2019.
- [54] Travis Whyte, Suman Baral, Paul Lashomb, Walter Wilcox, and Ronald B. Morgan. Disconnected loop subtraction methods in lattice qcd, Dec 05 2019.
- [55] Travis Whyte, Walter Wilcox, and Ronald B. Morgan. Deflated gmres with multigrid for lattice qcd. *Physics Letters B*, 803:135281, 2020.
- [56] G. C. Wick. The evaluation of the collision matrix. Phys. Rev., 80:268–272, Oct 1950.
- [57] Walter Wilcox. https://blogs.baylor.edu/walter\_wilcox/.
- [58] Walter Wilcox. Noise methods for flavor singlet quantities. Interdisciplinary Workshop on Numerical Challenges in Lattice QCD, pages 127–141, 8 1999.
- [59] Kenneth G. Wilson. Confinement of quarks. Phys. Rev. D, 10:2445–2459, Oct 1974.