NEW ATTEMPTS FOR ERROR REDUCTION IN LATTICE FIELD THEORY CALCULATIONS

Dissertation
zur Erlangung des akademischen Grades
doctor rerum naturalium
(Dr. rer. nat.)
im Fach Physik
Spezialisierung: Theoretische Physik

eingereicht an der
Mathematisch-Naturwissenschaftlichen Fakultät
der Humboldt-Universität zu Berlin

von
M.SC. JULIA LOUISA VOLMER

Präsidentin der Humboldt-Universität zu Berlin
Prof. Dr.-Ing. Dr. Sabine Kunst
Dekan der Mathematisch-Naturwissenschaftlichen Fakultät
Prof. Dr. Elmar Kulke

Gutachter: 1. Prof. Dr. Rainer Sommer
2. Dr. habil. Karl Jansen
3. PD Dr. habil. Falk Bruckmann

Tag der mündlichen Prüfung: 9. Juli 2018

ABSTRACT

Lattice quantum chromodynamics (QCD) is a very successful tool to compute QCD observables non-perturbatively from first principles. Therefore, the QCD path integral is evaluated on a discrete Euclidean 3+1-dimensional lattice.

A typical evaluation consists of two parts. First, sampling points, called *configurations*, are generated at which the path integral is evaluated. This is typically achieved by Markov chain Monte Carlo (MCMC) methods which work very well for most applications but also have some drawbacks. Typical issues of MCMC methods include their slow error scaling and the numerical *sign-problem*, where the numerical evaluation of an integral is extremely difficult due to a highly oscillatory integrand. Alternatives to MCMC are needed for these problems. The second part of the evaluation is the computation of the integrand on the configurations and includes the computation of quark connected and disconnected diagrams. Improvements of the signal-to-noise ratio have to be found since the disconnected diagrams, though their estimation being very noisy, contribute significantly to physical observables.

Methods are proposed to overcome the aforementioned difficulties in both parts of the evaluation of the lattice QCD path integral. We tested the exact eigenmode reconstruction with deflation method for the computation of quark disconnected diagrams and applied it to a $16^3 \times 32$ sites twisted mass lattice with a lattice spacing of $a = 0.079\,\mathrm{fm}$ and a pion mass of $m_\pi = 380\,\mathrm{MeV}$. The runtime of the evaluation is reduced 5.5-fold by the tested method compared to the standard method and thus promises a more efficient and accurate estimate for the observable.

In addition, we tested the recursive numerical integration method, which simplifies the evaluation of the integral to address the difficulties in MCMC. We applied the method in combination with a Gauss quadrature rule to a one-dimensional, quantum-mechanical topological osciallator model. In practice, we found that we can compute error estimates that scale exponentially to the correct result. A generalization to higher space-time dimensions can be done in the future.

Moreover, we developed the symmetrized quadrature rules to address the sign-problem. We applied them to one-dimensional QCD with a chemical potential which gives rise to the sign-problem. We found that this method is capable of overcoming the sign-problem completely and is very efficient for one variable. Improvements can be made for the efficiency of multi-variable scenarios in the future.

ZUSAMMENFASSUNG

Gitter Quantenchromodynamik (QCD) ist ein sehr erfolgreiches Instrument zur nicht-perturbativen Berechnung von QCD Observablen. Dabei wird das QCD Pfadintegral auf einem diskreten, euklidischen, 3+1-dimensionalen Gitter ausgewertet.

Eine typische Auswertung besteht aus zwei Teilen. Zuerst werden Stützstellen, sogenannte *Konfigurationen*, generiert, an denen das Pfadintegral ausgewertet wird. In der Regel werden dafür Markov chain Monte Carlo (MCMC) Methoden verwendet, die für die meisten Anwendungen sehr gute Ergebnisse liefern, aber auch Nachteile bergen. Dazu gehören die langsame Fehlerskalierung und das numerische *Vorzeichenproblem*, bei dem die numerische Auswertung eines Integrals durch einen hochoszillierenden Integranden sehr aufwendig ist. Alternativen zu MCMC Methoden werden für diese Probleme benötigt. Im zweiten Teil der Auswertung wird der Integrand auf den Konfigurationen ausgewertet. Dies beinhaltet die Berechnung von Quark zusammenhängenden und unzusammenhängenden Diagrammen. Letztere tragen maßgeblich zu physikalischen Observablen bei, jedoch leidet deren Berechnung an großen Fehlerabschätzungen, sodass Verbesserungen des Signal-Rausch-Verhältnisses benötigt werden.

In dieser Arbeit werden Methoden präsentiert, um die beschriebenen Schwierigkeiten in beiden Auswertungsteilen des QCD Pfadintegrals anzugehen. Für die Berechnung der Quark unzusammenhängenden Diagramme haben wir die Methode der exakten Eigenmodenrekonstruktion mit Deflation getestet und auf ein Gitter, berechnet mit chiral rotiertem Massenterm (Twisted-Mass Fermionen), mit $16^3 \times 32$ Punkten, einem Gitterabstand von $a = 0.079\,\mathrm{fm}$ und einer Pionenmasse von $m_\pi = 380\,\mathrm{MeV}$ angewandt. Unsere Methode braucht fast 5.5 mal weniger Laufzeit im Vergleich zur Standardmethode und verspricht somit eine effizientere beziehungsweise genauere Abschätzung von Observablen.

Außerdem haben wir die rekursive numerische Integration zur Vereinfachung von Integralauswertungen getestet, um die Probleme von MCMC Methoden zu adressieren. Wir haben die Methode in Kombination mit einer Gauß Quadraturregel auf das eindimensionale, quantenmechanische Modell des topologischen Oszillators angewandt. In der Paxis konnten wir exponentiell skalierende Fehlerabschätzungen berechnen. Der nächste Schritt ist eine Verallgemeinerung zu höheren Raumzeit Dimensionen.

Zusätzlich haben wir die symmetrisierten Quadraturregeln entwickelt, um das Vorzeichenproblem zu umgehen. Wir haben diese auf die eindimensionale QCD mit chemischem Potential, das zum Vorzeichenproblem führt, angewendet. Unsere Berechnungen zeigen, dass diese Methode dazu geeignet ist, das Vorzeichenproblem zu beseitigen und sehr effizient für eine Variable angewendet werden kann. Zukünftig kann die Effizienz für mehr Variablen verbessert werden.

ACKNOWLEDGMENTS

First and foremost, I thank my supervisor Karl Jansen for making this work possible, for all his precious advice and support and always finding time for my questions. I am very grateful for the very good atmosphere working together, the coffee rounds and the band rehearsals. It was a pleasure to work at DESY Zeuthen and to meet so many nice colleagues and friends there, especially Christian, Philipp, Debasish, Miguel, Heshou, Aurora, Attila, Tillmann, Arnd and all members of the DESY Band.

I am very thankful that Rainer Sommer agreed on being my formal supervisor and giving me useful advice on my thesis. I want to thank all my collaborators who contributed to this work. For the fruitful discussions and great support I very much thank Tobias Hartung, Hernan Leövey and Andreas Ammon. I am grateful to Constantia Alexandrou for the opportunity to work on disconnected diagram computations, visiting the Cyprus institute and being able to use the computer resources at the Swiss National Supercomputing Center. Thanks goes to Christos Kallidonis, Kyriakos Hadjiyiannakou, Giannis Kotsou and Alejandro Vaquero for their help with Quda and the supply of the presented runtime comparison of different methods with the Multigrid algorithm.

This research could not have happened without the computing resources from the John von Neumann Institute for Computing, the Swiss National Supercomputing Center, the DESY Zeuthen Computing Center and the Jülich Supercomputing Center. I also thank the ETM Collaboration for providing the necessary gauge configurations. This work has been supported by DFG GR 705/13.

Many people helped me in reading parts of this thesis and giving useful advice: Jesko Hüttenhain, Tobias Hartung, Jeremy Green, Debasish Banerjee, Mateusz Lech Koren, Alessandro Nada, Heshou Zhang and Jasmin Zohren.

Finally I especially want to thank my parents Clemens and Eva, my sister Franzi and my partner Jesko for their constant support, encouragement and motivation during the process of this thesis.

CONTENTS

1 INTRODUCTION

Our understanding of the smallest building blocks of our world is based on quantum physics. The standard model of particle physics (SM) is today's theory of all particles and interactions of visible matter in the universe. It combines Quantum Chromodynamics (QCD), describing strong interactions, with the electroweak interaction theory to form a local quantum field theory with local gauge group $\mathcal{SU}(3) \times \mathcal{SU}(2) \times \mathcal{U}(1)$. Additionally, it includes six quarks, six leptons, their corresponding antiparticles and the Higgs-field. Since the introduction of the SM all experiments confirmed the theory, the most recent and popular ones are the discovery of the top-quark [3, 7] and the Higgs-boson [2, 34].

Despite the great success of the SM there are observations which cannot be explain by it. There has been striking evidence from many different observations [1, 9, 37, 78] for non-luminous matter in the universe, called dark matter, whose nature is unknown. Moreover, the amount of CP-violation of the standard model is insufficient to account for the generation of an asymmetry between matter and antimatter in the early universe [24, 32].

There are many conceptual questions about the SM as well, such as why masses and couplings of the particles differ by orders of magnitudes, why strong interactions show no CP-violation experimentally while it is theoretically possible, how electroweak and strong interactions can be unified, if gravitation can be quantized and included in the model, and many more.

There are several experiments around the world and in space which try to answer these questions, including telescopes, particle colliders and low energy experiments. Today, the most powerful collider is the Large Hadron Collider (LHC) at the research facility CERN in Switzerland. This machine collides protons at a center-of-mass energy of order 10 TeV and measures the produced particles. To be able to achieve accurate results from these collisions, the SM has to be understood as good as possible. Because the quarks in the colliding protons interact via QCD, the understanding of QCD interactions is crucial for the correct evaluation of all experimental data.

Although QCD is included in the SM and has already been tested successfully, at least in the high energy regime, it is difficult to compute QCD observables at low energies. At energies smaller than $\Lambda_{\text{QCD}} \sim 250\,\text{MeV}$, perturbation theory breaks down. Responsible is the non-abelian nature of QCD, which results in charged gluons, the mediators of the strong interaction and allows self-interactions

among the gluons. This leads to an anti-screening effect of the strong charge and to a large coupling constant at energies smaller than Λ_{QCD}, such that perturbation theory is not applicable in this regime. Two different phenomena arise at the different energy scales: At low energies the quarks are bound, *confined*, in colorless states called hadrons. At large energies the quarks are asymptotically free. Although the energies used at the LHC are large, the evaluation of experimental measurements for physical results needs low energy input, e.g. the distribution of quarks in the colliding protons. Additionally, results for some individual processes need non-perturbative input values. It is desirable to derive these inputs directly from first principles of QCD.

Kenneth Wilson introduced lattice gauge theory in 1974 [82] which turned out to be a powerful tool for non-perturbative calculations in QCD from first principles. The lattice QCD computation of expectation values of observables is based on Feynman's path integral. In this formalism the amplitude of interacting fields Φ, e.g. a state $|\Phi_a(x, t_1)\rangle$ going to a state $|\Phi_b(y, t_2)\rangle$, is computed by integrating over all possible field configurations $[\Phi]$, weighted with $\mathrm{e}^{iS[\Phi]}$ dependent on the action $S[\Phi]$ of this field configuration. If the path integral is transformed to a discretized Euclidean space with Euclidean action $S^e[\Phi]$ defined on a discretized space-time lattice, it can be interpreted as an evaluation of a finite statistical system with Boltzmann weight $\mathrm{e}^{-S^e[\Phi]}$. Therefore, already tested numerical methods from statistical physics can be applied to evaluate the integral. The continuum QCD field theory is realized at a critical point of the statistical system. QCD describes the interaction of gluons and quarks, therefore the lattice QCD path integral integrates over all possible bosonic link field and fermionic quark field configurations. Link variables are gauge transporters that relate the color spaces between two neighboring lattice sites.

Lattice observables are computed by correlation functions between different lattice sites via the lattice QCD path integral. Therefore this path integral, involving fermions and links, needs to be evaluated. Because the fermion action is bilinear, the fermions can be integrated out analytically by taking into account all possible Wick contractions of the involved quark fields. This results in two distinct diagram types: connected diagrams propagate the quark fields between two lattice sites and disconnected diagrams propagate the quark fields to and from the same site. The quark propagator is the inverse of the large Dirac matrix which is dependent on the link fields. The Dirac matrix has to be inverted numerically for specific link configurations when the path integral is evaluated. This is numerically very demanding because the Dirac matrix has typically at least $\mathcal{O}(10^6 \times 10^6)$ entries. In contrast to the connected diagrams, the inversion of the Dirac ma-

trix for the disconnected diagrams needs stochastic input. Therefore the disconnected diagrams have usually a low signal-to-noise ratio.

The bosonic path integral, that is the QCD path integral with integrated out fermions, is approximated by using sampling points, link configurations, drawn from a complicated Boltzmann distribution. In most simulations this highly non-trivial drawing task is done by using Markov chain Monte Carlo (MCMC) methods. These methods use importance sampling to draw sampling points preferably with a large Boltzmann weight such that these points give a large contribution to the integral. Importance sampling can be done by creating a Markov chain. A Markov chain is a stochastic process that generates a sequence of link configurations, where the probability distribution of each configuration only dependents on the previous configuration. In lattice computations these Markov chains are created such that this probability distribution converges to the desired Boltzmann distribution [31]. Therefore after some events in the sequence, the generated link configurations can be used as sampling points for the bosonic path integral.

TODAY'S LATTICE COMPUTATIONS Finally, the computed lattice observables should give estimates of real world quantities. Then it is possible to compare the observables to an experimentally measured quantity to check the correctness of the implemented QCD model and to search for discrepancies which could come from new physics. Additionally, a lattice result can give new insights into physics from first principles and can give new predictions which could be tested experimentally. To result in real world estimates, today's lattice QCD simulations include the lightest four quarks, use physical quark masses and go to small lattice spacing. In this setup the computation of statistically significant results needs runtimes of the order of months to years, even on large-scale supercomputers. Additionally, at the precision of today's simulations some contributions are significant which were discarded before. This is the case for the computationally expensive disconnected diagrams where quark fields propagate from and to the same lattice site. Their computation increases the already large runtime.

The runtime of the lattice computations depend on the accuracy of the results that are needed. Using MCMC methods to evaluate the bosonic path integral gives an error scaling, which leads to an asymptotic shrinking of the error with the number of link configurations n by $1/\sqrt{n}$. This is a rather slow error scaling: to reduce the error by one order of magnitude one needs two orders of magnitude more configurations, which are time intensive to produce. Additionally, at small lattice spacing, configurations in the Markov chain are highly correlated and many configurations are needed to reach a specified error estimate. This issue is called *critical slowing-down*. For some spe-

cific systems the application of MCMC methods is especially difficult: if the integrand of the bosonic path integral is complex and therefore a highly oscillatory function, the near cancellations of positive and negative contributions to the integral cannot be achieved with importance sampled points from a Markov chain. This results in large errors which scale exponentially with the lattice volume and is called *sign-problem*. The sign-problem is for example the reason why simulations of the early universe at the quark-gluon plasma phase (for large values of the chemical potential) are not possible today.

NOVEL METHODS We applied and developed novel methods to reduce error estimates of standard path integral evaluations. On the one hand we approached the noisy quark disconnected diagram computations. On the other hand we searched for alternatives to MCMC methods for the evaluation of the bosonic path integral.

We applied the exact eigenmode reconstruction with deflation method to the computation of disconnected diagrams in order to reduce the error estimate of their computations. This method combines the ideas of using eigenvectors of the Dirac matrix in [74], and using deflation, as e.g. in the initial guess deflation which is discussed in detail in this thesis, such that less stochastic sources are needed in the compuation to reach a specified error estimate. The method inverts the large Dirac matrix by using the matrix's eigenvectors to compute some part of the inverted matrix exactly. The remaining part is computed stochastically after deflating the Dirac matrix with its eigenvectors. We also combined the method with other improved techniques which are already widely used for disconnected diagram computations: stochastic sources [27], the one-end trick [4, 49, 72] and even-odd preconditioning [39]. We implemented the method into the Quda code [22, 36], which is highly parallelizable on graphic cards. We applied the method using twisted mass fermions to a lattice with $16^3 \times 32$ sites, lattice spacing $a = 0.079\,\mathrm{fm}$ and pion mass $m_\pi = 380\,\mathrm{MeV}$ to get a first impression of its error estimates and runtime in comparison to a standard method.

We searched for alternatives to MCMC methods to improve the error scaling, avoid critical slowing-down and the sign-problem in the evaluation of the path integral. We tested two polynomially exact quadrature rules to approximate examples of bosonic integrals by choosing sampling points deterministically, in contrast to importance sampled points in Monte Carlo methods: the recursive numerical integration and the symmetrized quadrature rules. We applied both to simplified models to test their abilities.

We used the recursive numerical integration method [58, 61] to improve the error scaling and to avoid critical slowing-down. The method uses the local coupling structure in the integrands of lattice path integrals to simplify the evaluation of the corresponding inte-

grals. In combination with an efficient quadrature rule this method can give polynomially exact results. We applied the method with a Gauss quadrature rule to the topological osciallator [25], a quantum-mechanical system in one dimension, which has some similarities to gauge theories.

We constructed the symmetrized quadrature rules to avoid the sign-problem. Many methods have been developed to tackle the sign-problem. The one described in [28, 29] uses MCMC to sample points from a subgroup of the full symmetry group of the model. In contrast to this approach we used sampling points from a larger symmetry group of the model. This results in polynomially exact quadrature rules where therefore we did not need any additional Monte Carlo simulation. These quadrature rules are applicable to integrals over compact groups $\mathcal{U}(N)$ and $\mathcal{SU}(N)$ for $N \leq \{2,3\}$ and they are based on the efficient quadrature rules on spheres in [57]. We applied these rules first to the one-dimensional QCD [26] which is an over-simplified QCD model with only one variable. We also applied it to the topological oscillator, which has more integration variables and is therefore computationally more expensive. We modified the method to make it feasible for more variables by combining it again with MCMC.

THIS THESIS This thesis is divided into two parts, addressing our improvements in the computation of quark disconnected diagrams and the evaluation of the bosonic path integral.

The first part approaches the computation of observables in lattice QCD, specifically the computation of quark disconnected diagrams. Here the second chapter introduces QCD and its discretization on the lattice. It presents the path integral, its bosonic and fermionic part, describes the computation of QCD observables on the lattice and how they get contributions from quark connected and disconnected diagrams. Finally it reviews some computations of disconnected diagram contributions with twisted mass fermions to hadron structure quantities.

The third chapter describes methods to improve disconnected diagram computations. It first presents widely used and already tested improved methods. Then it explains the exact eigenmode reconstruction with deflation method, its combination with other improved methods, describes our implementation in Quda and shows error scaling and runtime results applying the method to a small lattice. Finally it compares runtimes with another recently developed and very efficient method, an implementation of the Multigrid algorithm [55].

The second part of this thesis addresses the generation of configurations to approximate the bosonic path integral in benchmark models. Here chapter four presents MCMC methods and some of their possi-

ble issues. It introduces the basic terms and concepts of approximating an integral and describes ordinary Monte Carlo sampling, using random sampling points for the integral approximation. It explains importance sampling as a variance reduction technique for Monte Carlo methods and how to draw importance sampled configurations using a Markov chain. Finally it specifies the most common issues that arise when using MCMC methods, its error scaling, the critical slowing-down and the sign-problem.

The fifth chapter reports on the recursive numerical integration method. It provides insight into the structure of typical lattice path integrals, explains how this structure is used in the method to simplify the integral evaluation, introduces the topological oscillator model and finally compares results of applying recursive numerical integration and MCMC methods to the model.

The sixth chapter explains the completely symmetrized quadrature rules for only one integration variable. It explains the idea of forming these quadrature rules, how to use them and introduces the one-dimensional QCD model with a sign problem. Then it shows results of applying the method to the model, especially for the sign-problem region and compares it with MC results.

Chapter seven addresses the application of symmetrized quadrature rules to systems with more variables. It first shows how to apply one completely symmetrized quadrature rule from chapter six to each variable of a multi-variable model. Then it explains how this rule can be combined with MCMC to make the method feasible for a larger number of variables. It introduces a complex phase to the topological oscillator and presents results for applying both the original and the combined method to the one-dimensional topological oscillator with an additional complex phase factor, leading to the sign-problem. Finally it gives some possible explanations why the combined method does not solve the sign-problem.

Part I

DISCONNECTED DIAGRAMS IN LATTICE QCD

2 QUANTUM CHROMODYNAMICS ON THE LATTICE

Quantum Chromodynamics (QCD) is the theory to describe strong interactions between quarks and gluons. In this framework hadronic observables can be calculated. In contrast to QCD at large energies, where perturbation theory can be used to compute expectation values due to the small strong coupling constant, QCD has a large coupling constant at small energies, making it impossible to use perturbation theory for computations in this limit. Unfortunately, many interesting hadronic observables belong to this limit. Therefore a non-perturbative tool to compute hadronic observables is needed. Lattice Quantum Chromodynamics (LQCD) discretizes continuum QCD and uses the path integral formalism to compute observables non-perturbatively.

This chapter gives a short introduction to the computation of hadronic observables in LQCD: It first introduces the continuum QCD action, then presents two possible discretization schemes: Wilson and twisted mass fermions. Then the chapter presents the actual computation of observables and shows how observables which include fermion fields get contributions from quark connected and disconnected diagrams. Finally it shows results of some recent hadronic observable computations, using the presented lattice QCD framework. Here the main focus are the disconnected contributions to these observables, coming from the evaluation of the quark disconnected diagrams, because they have in general a smaller signal-to-noise ratio and are subject of the next chapter.

This chapter shows that lattice QCD is a valuable tool to compute hadronic observables non-perturbatively from first principles. The computation of disconnected contributions to fermionic observables is one part of the full hadronic observable computation, but results of disconnected contributions have large uncertainties. Therefore new methods are needed for the computation of disconnected contributions and for accurate results of some LQCD observables. Chapter 3 below presents some of these improved methods.

2.1 QUANTUM CHROMODYNAMICS

QCD describes the strong interaction involving quarks and gluons. QCD is an $\mathcal{SU}(3)$ gauge invariant (Yang-Mills) theory. Its action has two parts: the fermion part describes interactions of quarks, anti-quarks and gluons while the gluon part specifies the interactions of gluons among themselves.

THE FERMION ACTION is

$$S_F = \int dx^4 \, \overline{\Psi}(i\gamma^\mu D_\mu - m)\Psi. \tag{2.1}$$

Here the fermion fields $\Psi_{\substack{a \\ \alpha}}(x)$ and $\overline{\Psi}_{\substack{a \\ \alpha}}(x)$ have mass m and are spinors with a Dirac index $\alpha \in \{0,1,2,3\}$, color index $a \in \{1,2,3\}$ and depend on the four-vector x^μ in Minkowski space. The gamma matrices are defined in appendix A. This action is $\mathcal{SU}(3)$ gauge invariant which means that it does not change under applications of local $\mathcal{SU}(3)$ transformations - local rotations among the color indices of the quarks. This is ensured by the covariant derivative,

$$D_\mu = \partial_\mu + igA_\mu. \tag{2.2}$$

$A_\mu = A^a_\mu T^a$ is the gluon field, consisting of color fields A^a_μ, $a \in \{1,...,8\}$, which belong to the eight generators T^a of $\mathcal{SU}(3)$. g is the strong coupling constant.

THE GLUON ACTION is defined by

$$S_G = -\frac{1}{4}\int dx^4 \, G^{\mu\nu,a} G^a_{\mu\nu}. \tag{2.3}$$

The gluonic field tensor is given by

$$G^a_{\mu\nu} = \partial_\mu A^a_\nu - \partial_\nu A^a_\mu - g f^{abc} A^b_\mu A^c_\nu, \tag{2.4}$$

with the structure constant f^{abc}, defined by $if^{abc}T^c = [T^a, T^b]$. The third term, involving f^{abc}, originates from the non-abelian nature of the $\mathcal{SU}(3)$ group and results in three and four gluon interactions. This influences the dependence of the renormalized coupling g_r (this is the physical in contrast to the bare coupling g) on the energy scale μ substantially: g_r is small for large μ (the quarks are asymptotically free) and large for small μ (the quarks are confined in hadrons). $g_r(\mu)$ is called running coupling. For energies smaller than $\Lambda_{QCD} \sim 250\,\text{MeV}$ the perturbatively defined coupling would diverge. Therefore the computation of low energy QCD observables is difficult.

2.2 DISCRETIZING QCD ON THE LATTICE

In 1974 Wilson introduced lattice gauge theory in [82], a Yang-Mills theory in four-dimensional Euclidean space-time on a finite four-dimensional lattice. Including fermions on the lattice results in lattice QCD (LQCD).

The lattice includes N_T sites in time direction and N_L sites in the three spatial directions, all with the same lattice spacing a. Then the full lattice is defined by

$$\Lambda = \{(n_0, n_1, n_2, n_3) | n_0 \in \{0,1,...,N_T - 1\}, \\ n_1, n_2, n_3 \in \{0,1,...,N_L - 1\}\} \tag{2.5}$$

and includes $V_{\text{lat}} = N_L^3 \times N_T$ sites. Each lattice site $n \in \Lambda$ corresponds to the Euclidean space-time point $x = an \in \mathbb{R}^4$. The physical volume of the lattice is given by $V = L^3 \times T$ with the lattice side lengths $L = aN_L$ and $T = aN_T$. Fermion fields are defined on the lattice sites $n \in \Lambda$. Link variables, $U_\mu(n)$ live on the links connecting the sites n and $n + \hat{\mu}$, the next neighbors in direction $\mu \in \{1, 2, 3, 4\}$. $U_\mu(n)$ are elements of the gauge group $\mathcal{SU}(3)$.

THE GLUON ACTION The plaquette is the simplest closed loop on the lattice and is a possible gauge invariant object which consists out of link variables,

$$U_{\mu\nu}(n) = U_\mu(n) U_\nu(n + \hat{\mu}) U_\mu(n + \hat{\nu})^\dagger U_\nu(n)^\dagger, \tag{2.6}$$

with $U_{-\mu}(n) = U_\mu(n - \hat{\mu})^\dagger$. In the naive continuum limit, $a \to 0$, $U_\mu(n)$ is the parallel gauge transporter connected to the gluon field $A_\mu(x)$ and $U_{\mu\nu}(n)$ is connected to the field strength tensor whose components are defined in (2.4),

$$U_\mu(n) \xrightarrow{a\to 0} \mathrm{e}^{iagA_\mu(x)} \quad \text{and} \quad U_{\mu\nu}(n) \xrightarrow{a\to 0} \mathrm{e}^{ia^2 g G_{\mu\nu}(x)}. \tag{2.7}$$

The plaquette can be used to build a discretized Euclidean gluon action,

$$S_G^e = \frac{1}{2g^2} \sum_{n\in\Lambda} \sum_{\substack{\mu,\nu=1 \\ \mu\neq\nu}}^{4} \Re \operatorname{Tr}[\mathbb{1} - U_{\mu\nu}(n)]. \tag{2.8}$$

THE FERMION ACTION One possibility to discretize the fermion action is the Wilson fermion action

$$S_F^e = a^4 \sum_{n\in\Lambda} \overline{\Psi}(n)\, \hat{D}_W\, \Psi(n), \tag{2.9}$$

with the Wilson Dirac operator

$$\hat{D}_W = \gamma_\mu \frac{1}{2}(\nabla_\mu + \nabla_\mu^*) + \frac{a}{2} \nabla_\mu \nabla_\mu^* + m. \tag{2.10}$$

The first term is the gauge covariant derivative, the second term with the two derivatives, also called Wilson term, assures that the action describes only one fermion and not several unphysical ones, which occur due to the discretization. The derivatives are defined by

$$\begin{aligned} \nabla_\mu \Psi(n) &= \frac{1}{a}(U_\mu(n)\Psi(n + \hat{\mu}) - \Psi(n)), \\ \nabla_\mu^* \Psi(n) &= \frac{1}{a}(\Psi(n) - U_{-\mu}(x)^\dagger \Psi(n - \hat{\mu})). \end{aligned} \tag{2.11}$$

The Wilson matrix, corresponding to the Wilson operator in (2.10), can be split into a diagonal and a non-diagonal, next-neighbor interaction term,

$$D_W = C(\mathbb{1} - \kappa H), \quad \kappa = \frac{1}{2(4 + am)}, \quad C = m + \frac{4}{a}. \tag{2.12}$$

The hopping matrix H includes all next neighbor coupling terms. The factor C can be included into the fermion field definition.

The Wilson term, the second term in (2.10), vanishes in the naive continuum limit, but only by the order of a, therefore the discretization errors of the Wilson action are of the order a. Lattice simulations cannot go to infinitely small lattice spacing, today used values are around 0.15 fm to 0.05 fm. Therefore it is preferable to use a fermion action which has cutoff effects at a larger order in a. This can be done by adding counter terms that cancel the order a terms, e.g. [71] or by automatic order a improvement using twisted mass fermions.

2.3 TWISTED MASS LATTICE QCD

One way to achieve $\mathcal{O}(a)$ improvement is using twisted mass fermions. Twisted mass fermion fields are flavor doublets of up- and down-type quarks $\chi = (u_{\mathrm{tm}}, d_{\mathrm{tm}})^T$, defined in a twisted mass basis, which is chirally rotated to the physical basis $\Psi = (u, d)^T$,

$$\Psi = \exp(i\frac{\omega}{2}\gamma_5\tau^3)\chi, \qquad \overline{\Psi} = \overline{\chi}\exp(i\frac{\omega}{2}\gamma_5\tau^3), \tag{2.13}$$

where τ^3 acts in flavor space. The twist angle is defined by $\omega = \arctan(\mu/m)$, where the mass m and the twisted mass $\mu > 0$ are connected to the quark mass via $M = \sqrt{m^2 + \mu^2}$. The twisted mass action of the light mass-degenerate doublet, consisting of up and down quark, is given by

$$S_F^{e,\mathrm{tm}}[\chi, \overline{\chi}, U] = a^4 \sum_{n\in\Lambda} \overline{\chi}(n)(\hat{D}_W \mathbb{1}_2 + i\mu\gamma_5\tau^3)\chi(n). \tag{2.14}$$

Compared to the Wilson action in (2.9), $S_F^{e,\mathrm{tm}}$ includes the additional twisted mass term. The Wilson Dirac operator, defined in (2.10), is applied to each twisted mass quark field separately. The term sandwiched between the flavor doublets $\overline{\chi}$ and χ is the Wilson twisted mass Dirac operator, a diagonal operator matrix in flavor space with entries $\hat{D}_{u/d} = \hat{D}_W \pm i\mu\gamma_5$, each acting on one entry of the flavor doublets. Writing $S_F^{e,\mathrm{tm}}$ in the physical basis gives

$$\begin{aligned} S_F^{e,\mathrm{tm}}[\Psi, \overline{\Psi}, U] = a^4 \sum_n \overline{\Psi}(n)(\gamma^\mu \frac{1}{2}(\nabla_\mu + \nabla_\mu^*) + \\ + e^{i\omega\gamma_5\tau^3} \frac{a}{2}\nabla_\mu\nabla_\mu^* + M)\Psi(n), \end{aligned} \tag{2.15}$$

where only the Wilson term, which is needed to remove fermion doublers but is also responsible for the order a discretization errors, is rotated. It can be shown that observables computed with the twisted mass action at maximal twist $\omega = \frac{\pi}{2}$ have either discretization errors of $\mathcal{O}(a^2)$ or are zero in the continuum limit, due to their transformation under discrete chiral transformations [51, 56]. In the continuum

limit the twisted mass formulation describes conventional QCD [53, 56] and can therefore be used as an alternative to the Wilson action.

For heavier quarks, like strange and charm quarks s and c, which are not approximately degenerate, the action of the flavor doublet $\chi = (s_{\text{tm}}, c_{\text{tm}})^T$ is

$$S_{F,h}^{e,\text{tm}}[\chi, \overline{\chi}, U] = a^4 \sum_x \overline{\chi}(D_W \mathbb{1}_2 + i\mu\gamma_5\tau^1 + \epsilon\tau^3)\chi, \tag{2.16}$$

with $\mu, \epsilon > 0$. The strange and charm quark masses are associated with $m_s = M - \epsilon$ and $m_c = M + \epsilon$ [52].

The twisted mass formulation is used in all simulations in chapter 3. In most calculations the physical basis is used if not written otherwise because it is more convenient to compute e.g. two-point functions.

2.4 THE PATH INTEGRAL

The QCD action can be discretized on a lattice. But a tool is needed to compute hadronic observables, such as hadronic masses and form factors. This tool is the path integral formalism, which can be used to compute amplitudes of interactions. The physical path integral in Minkowski space is difficult to evaluate numerically because it includes an highly fluctuating integrand. But the evaluation of the discretized Euclidean path integral is similar to the evaluation of a correlation function of a statistical canonical ensemble and therefore numerically possible. The continuum limit of the discretized system can be approached at a critical point of the statistical system. Therefore the Euclidean path integral can be used as a tool to compute physical expectation values. This section introduces the physical and Euclidean path integral and describes its evaluation in lattice QCD, showing all steps of a typical lattice QCD simulation.

It especially shows that the lattice QCD path integral includes two types of integrals, one over fermionic degrees of freedom, the other over the links. Due to their very different nature, both integrals are evaluated differently in the simulation. The fermionic degrees of freedom can be integrated out analytically, resulting in quark propagators. The remaining integral over the links is approximated by choosing sampling points (configurations) from a Boltzmann distribution.

2.4.1 *The Euclidean path integral*

In quantum field theory all physical information about the system is stored in an infinite set of vacuum expectation values of time-ordered products of Heisenberg field operators $\hat{\Phi}_1(x)$, $\hat{\Phi}_2(y)$, ... , called the Green's function, e.g.

$$G(x, y, ...) = \langle 0 | \, \mathbb{T}[\hat{\Phi}_1(x)\hat{\Phi}_2(y)...] \, | 0 \rangle \,, \tag{2.17}$$

which can be interpreted as the amplitude of interactions of the fields $\Phi_1(x), \Phi_2(y), ...$ in the vacuum. This amplitude can be computed with the path integral, summing over all possible field configurations $[\Phi]$, each one weighted by $e^{iS[\Phi]}$, dependent on the action S of the system,

$$G(x,y,...) = \frac{\int \mathrm{d}[\Phi]\, \Phi_1 \Phi_2 ... \, \mathrm{e}^{iS[\Phi]}}{\int \mathrm{d}[\Phi]\, \mathrm{e}^{iS[\Phi]}}. \tag{2.18}$$

This Green's function is not suited for numerical calculations. For lattice computations a Wick rotation from Minkowski-space to Euclidean space is done, sending $t \to -i\tau$ and therefore $iS \to -S^e$. In lattice QCD the interacting fields are the fermion fields Ψ and $\overline{\Psi}$ and the link variables U. In the following $O[\overline{\Psi}, \Psi, U]$ stands for any gauge invariant combinations of theses involved fields and is called observable function. Then the Euclidean Green's function gives the expectation value of the observable O,

$$\langle O \rangle \overset{\text{def}}{=} G^e(O) = \frac{1}{Z} \int \mathrm{d}[U] \int \mathrm{d}[\overline{\Psi}, \Psi]\, O[\overline{\Psi}, \Psi, U]\, \mathrm{e}^{-S^e[\overline{\Psi}, \Psi, U]}, \tag{2.19}$$

with

$$Z = \int \mathrm{d}[U] \int \mathrm{d}[\overline{\Psi}, \Psi]\ \mathrm{e}^{-S^e[\overline{\Psi}, \Psi, U]}, \tag{2.20}$$

$$\mathrm{d}[U] = \prod_{n \in \Lambda} \prod_{\mu=1}^{4} \mathrm{d}U_\mu(n) \tag{2.21}$$

$$\mathrm{d}[\overline{\Psi}, \Psi] = \prod_{n \in \Lambda} \prod_{f} \prod_{\alpha=1}^{4} \prod_{c=1}^{3} \mathrm{d}\overline{\Psi}^{(f)}(n)_{\substack{\alpha \\ c}}\, \mathrm{d}\Psi^{(f)}(n)_{\substack{\alpha \\ c}}, \tag{2.22}$$

for fermion flavors f. For a finite lattice, equation (2.19) is similar to a statistical canonical ensemble correlation function with Boltzmann distribution e^{-S^e} and Z can be called partition function. Hence, the expectation value $\langle O \rangle$ can also be described by the operator $\hat{O}$, the Hamiltonian operator $\hat{H}$ and the inverse temperature T of the system,

$$\langle O \rangle = \frac{1}{Z} \mathrm{tr}[\hat{O}\, \mathrm{e}^{-T\hat{H}}], \qquad Z = \mathrm{tr}[\mathrm{e}^{-T\hat{H}}]. \tag{2.23}$$

Statistical methods can be applied to the Euclidean path integral (2.19). The inverse temperature of the system is equivalent to the lattice extent in time direction $T = aN_T$. A zero temperature expectation value results from taking the limit $N_T \to \infty$. It can be shown that the continuum limit Green's function $G(O)$ can be realized by the expectation value $\langle O \rangle$ at a critical point of the statistical system that is described by S^e. At this critical point the longest correlation length, given by the inverse of the pion mass, diverges. This can be realized by tuning the parameters, here the bare coupling g and the bare mass m to their critical values g^*, m^*. The parameters g and m also depend

on the lattice spacing a and the expectation value $\langle O \rangle$ is dependent on a, $g(a)$ and $m(a)$. The continuum limit of $\langle O \rangle$ is reached for $a \to 0$ if m and g are tuned with a in an appropriate way, such that they reach g^* and m^* respectively,

$$\langle O \rangle(g(a), m(a), a) \xrightarrow{a \to 0} G(O), \text{ for } m(a) \xrightarrow{a \to 0} m^* \text{ and } g(a) \xrightarrow{a \to 0} g^*. \tag{2.24}$$

To reach a physical situation, the tuning has to be done such that the ratio of the pion mass over the nucleon mass is given by its physical value. To keep the physical volume V of the lattice fixed when approaching the continuum limit, the number of lattice sites is chosen according to a, such that $L = aN_L$ and $T = aN_T$ remain constant.

2.4.2 *Evaluating the path integral on the lattice*

The path integral in (2.19) includes integrals over fermionic degrees of freedom $\overline{\Psi}$ and Ψ and links U. The fermionic degrees of freedom are Grassmann numbers. Because the fermion action (2.9) is bilinear in the fermion fields, the fermion fields can therefore be integrated out analytically. Hence, the path integral can be written in the form $\langle O \rangle = \langle \langle O \rangle_F [U] \rangle_G$. Then the inner fermionic path integral is an analytic expression and dependent on the link configuration $[U]$,

$$\langle O \rangle_F [U] = \frac{1}{Z_F[U]} \int \mathrm{d}[\overline{\Psi}, \Psi] O[\overline{\Psi}, \Psi, U] \, \mathrm{e}^{-S_F^e[\overline{\Psi}, \Psi, U]}, \tag{2.25}$$

$$\text{with } Z_F[U] = \int \mathrm{d}[\overline{\Psi}, \Psi] \, \mathrm{e}^{-S_F^e[\overline{\Psi}, \Psi, U]} \, . \tag{2.26}$$

The outer link integral integrates out the links,

$$\langle O \rangle = \langle \langle O \rangle_F [U] \rangle_G = \frac{1}{Z} \int \mathrm{d}[U] \, Z_F[U] \, \langle O \rangle_F [U] \, \mathrm{e}^{-S_G^e[U]}, \tag{2.27}$$

$$\text{with } Z = \int \mathrm{d}[U] Z_F[U] \, \mathrm{e}^{-S_G^e[U]} \, . \tag{2.28}$$

There are three steps involved to compute a lattice expectation value $\langle O \rangle$, and a forth one to approach the continuum limit:

1. Generate N link field configurations $[\tilde{U}]$ from the distribution $\frac{\mathrm{e}^{-S_G^e[U]} Z_F[U]}{Z}$, compare (2.27).
2. Evaluate $\langle O \rangle_F [\tilde{U}]$ in (2.25) for each link configuration.
3. Approximate the link path integral in (2.27) by the average over all evaluated $\langle O \rangle_F [\tilde{U}]$,

$$\langle O \rangle = \langle \langle O \rangle_F [U] \rangle_G \approx \frac{1}{N} \sum_{[\tilde{U}]} \langle O \rangle_F [\tilde{U}] \tag{2.29}$$

4. Approach the continuum limit by using smaller lattice spacings while adjusting the coupling constants g and m accordingly.

The generation of the link configurations in step 1 on a finite lattice is typically done by a Markov chain Monte Carlo method, using a Markov chain to create subsequent configurations $[\tilde{U}]$ which are drawn from the normalized Boltzmann distribution $\frac{e^{-S_G[U]} Z_F[U]}{Z}$. Using these configurations results in the approximation of the link path integral in (2.29). For large number of configurations $\#[\tilde{U}]$ the error estimate of this approximation shrinks with the number of configurations as $1/\sqrt{\#[\tilde{U}]}$. Markov chain Monte Carlo methods are described in more detail in chapter 4.

The fermion fields are Grassmann numbers, that means they anticommute, e.g. $\{\overline{\Psi}, \Psi\} = 0$. Because the fermion action in (2.9) is bilinear, the fermionic integral in (2.25) can be solved analytically. The expectation value of products of Grassmann numbers is given by the Wick theorem. For two fermion fields of the same flavor, located at lattice sites m and n it is

$$\langle \Psi(n) \overline{\Psi}(m) \rangle_F = a^{-4} G(n|m), \tag{2.30}$$

where $G(n|m) = D^{-1}(n|m)$ is the inverse of the Dirac matrix and propagates the fermion from m to n. Depending on the used fermion discretization the Dirac matrix can be D_W, the matrix form of (2.10) for Wilson fermions, or $D_W \pm i\mu\gamma_5$ for twisted mass fermions. For an even and larger than two number of fermion fields the expectation value is the sum over all possible combinations of two fermion fields, called Wick contractions. This leads to different types of diagrams that contribute to the expectation value. This is elaborated in section 2.5.4.

Also the fermionic partition function, needed in the generation of the configurations in step 1, can be integrated analytically: It is the determinant of the Dirac matrix, for a fermionic doublet $\Psi = (u, d)^T$: $Z_F[U] = \det(D_u[U]) \det(D_d[U])$. In contrast to $e^{-S_G^e[U]}$, these determinants are non-local quantities and specific Markov chain Monte Carlo methods are needed to handle the Boltzmann distribution involving them.

2.5 COMPUTING OBSERVABLES ON THE LATTICE

The path integral formalism can be used to compute amplitudes of hadron field interactions on the lattice. These amplitudes include information on the involved hadrons, which can to be extracted. Therefore hadron fields need to be defined on the lattice, amplitudes of hadron interactions need to be computed and the information needs to be singled out from these amplitudes. Information, like hadronic masses or form factors, can be extracted from two-point and three

point correlation functions. This section describes how to choose fields which interpolate hadron fields on the lattice, shows how important hadronic observables are derived from two- and three-point correlation functions and finally how different types of diagrams contribute to two- and three-point functions.

This section especially shows that masses and decay constants can be deduced from two-point functions, while hadron structure observables, like form factors, charges and transition amplitudes can be derived from a combination of two- and three-point functions. Additionally, this section shows that these two- and three-point functions consists of two parts, a connected part, describing propagations of fermions from one lattice site to another, and a disconnected part, characterizing fermion loops.

2.5.1 *Interpolating fields*

A hadron state $|h(p)\rangle$ with momentum p can be simulated on the lattice through an interpolating operator $\hat{O}(n)$ at site n that creates a state $|O(n)\rangle = \hat{O}(n)|0\rangle$ with quantum numbers that match the hadron quantum numbers, such that $\hat{O}(n)$ has a non-zero overlap with $|h(p)\rangle$,

$$\sum_{\vec{n}} e^{ia\vec{p}\cdot\vec{n}} \langle h(p)|\hat{O}(n)|0\rangle \neq 0. \tag{2.31}$$

Hadronic quantum numbers, e.g. isospin I, isospin component I_z, charge Q, spin J and parity P arise by combining the quark spinors Ψ accordingly. The quark fields of up- down- and strange quarks have quantum numbers (with quark spin S)

Ψ	S	I	I_z	Q	P
u	$\frac{1}{2}$	$\frac{1}{2}$	$+\frac{1}{2}$	$\frac{2}{3}$	$-$
d	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{3}$	$-$
s	$\frac{1}{2}$	0	0	$\frac{2}{3}$	$-$

Additionally, the hadron operator $\hat{O}$ should only create color-singlets. Only these color-states are invariant under an $\mathcal{SU}(3)$ transformation and are therefore the only ones projected out of the link path integral in (2.27).

Then a meson can be simulated by the bilinear interpolating field

$$O(n) = \bar{q}_1(n)_{\substack{\alpha \\ c}} \Gamma_{\alpha\beta} q_2(n)_{\substack{\beta \\ c}} \tag{2.32}$$

with the quark fields q_1 and q_2 and Γ, see table 2.1, chosen according to the quantum numbers of the meson. Using flavor doublets $\Psi = (u,d)^T$, interpolating meson fields can be written as

$$O(n) = \overline{\Psi}(n)\Gamma\frac{\tau_a}{2}\Psi(n), \tag{2.33}$$

state	J^{PC}	Γ	particles
Scalar	0^{++}	$\mathbb{1}, \gamma_0$	$f_0, a_0, K_0^*, ...$
Pseudoscalar	0^{-+}	$\gamma_5, \gamma_0\gamma_5$	$\pi^\pm, \pi^0, \eta, K^\pm, K^0, ...$
Vector	1^{--}	$\gamma_i, \gamma_0\gamma_i$	$\rho^\pm, \rho^0, \omega, K^*, \phi, ...$
Axial vector	1^{++}	$\gamma_i\gamma_5$	$a_1, f_1, ...$
Tensor	1^{+-}	$\gamma_i\gamma_j$	$h_1, b_1, ...$

Table 2.1: A bilinear interpolating field $q_1\Gamma q_2$ simulates a meson with quark content q_1q_2 on the lattice, if it matches the meson's quantum numbers, here spin J, parity P and charge conjugation C. ($\gamma_i \in \{\gamma_1, \gamma_2, \gamma_3\}$)

where τ_a acts in flavor space, being either $\tau_1 \pm i\tau_2$, τ_3 or $\mathbb{1}$. Then the interpolating fields with $\tau_1 \pm i\tau_2$ and τ_3 form an isotriplet or isovector state with $I = 1$ and the ones with $\tau = \mathbb{1}$ form an isosinglet or isoscalar state with $I = 0$.

Baryons contain three quarks, therefore the singlet color wave function is, in contrast to mesons, antisymmetric and the baryon interpolating field has the form

$$O(n)_\delta = \epsilon_{abc}\ P_{\delta\epsilon}\ \Gamma^A_{\epsilon\alpha}\ q_1(n)_{\substack{\alpha \\ a}}\ (q_2(n)^T_{\substack{\beta \\ b}}\ \Gamma^B_{\beta\gamma}\ q_3(n)_{\substack{\gamma \\ c}}), \tag{2.34}$$

where P projects the baryon to definite parity, e.g. $P_\pm = \frac{1}{2}(\mathbb{1} \pm \gamma_0)$ for zero momentum fields. To describe baryons with $J^P = \frac{1}{2}^+$, $(\Gamma^A, \Gamma^B) = (\mathbb{1}, C\gamma_5)$ can be used.

2.5.2 *Two-point functions*

Hadronic interpolating fields can be propagated through the lattice to deduce, as a result of their propagation characteristics, some of their properties. The mass and decay constant of a hadron can be computed from the expectation value $\langle O(n)\overline{O}(0)\rangle$ of an interpolating hadron field O being created at site m (translational invariance allows to choose $m = (0, \vec{0})$) and a similar hadron field being annihilated at site $n = (n_t, \vec{n})$. The two-point function $C^{2\text{pt}}(n)$ is usually defined as the connected correlation function of two interpolating fields,

$$C^{2\text{pt}}(n) = \langle O(n)\overline{O}(0)\rangle - \langle O(n)\rangle\langle\overline{O}(0)\rangle. \tag{2.35}$$

A Fourier transformation results in a dependence on momentum $\vec{p}$ instead of spatial site vector $\vec{n}$,

$$C^{2\text{pt}}(n_t, \vec{p}) = \sum_{\vec{n}} \mathrm{e}^{-ia\vec{p}\cdot\vec{n}}\, C^{2\text{pt}}(n_t, \vec{n}). \tag{2.36}$$

For large times the two-point function decays exponentially,

$$C^{2\text{pt}}(n_t, \vec{p}) \xrightarrow{\text{large } t,T} A\, \mathrm{e}^{-t\Delta E_1}, \tag{2.37}$$

with euclidean time $t = an_t$ and extent of the lattice in time direction $T = aN_T$. At zero momentum of the hadron interpolating field, $\vec{p} = \vec{0}$, ΔE_1 is the mass of the ground state hadron and the amplitude A is proportional to the decay constant of this state. In the following this exponential decay is derived and for brevity only the time-dependence of fields are shown explicitly, $C^{2\text{pt}}(n_t) \stackrel{\text{def}}{=} C^{2\text{pt}}(n_t, \vec{p})$.

Using (2.23), the mentioned expectation value can be written as

$$\langle O(n_t)\overline{O}(0)\rangle = \frac{1}{Z}\mathrm{Tr}[\hat{O}(n_t)\hat{\overline{O}}(0)\,\mathrm{e}^{-T\hat{H}}]. \tag{2.38}$$

This expression is converted to the Schrödinger picture, where operators are time-independent. In this picture the time dependent Heisenberg-picture operators in (2.38) can be written as $\hat{O}(n_t) = \mathrm{e}^{t\hat{H}}\,\hat{O}\,\mathrm{e}^{-t\hat{H}}$. The eigenvector basis $|n\rangle$ of the Hamiltonian can be used to evaluate the trace in (2.38). Additionally, a complete set of orthonormal vectors, $\mathbb{1} = \sum_m |m\rangle\langle m|$ is inserted just before $\hat{\overline{O}}$ to give

$$\langle O(n_t)\overline{O}(0)\rangle = \frac{1}{Z}\sum_{n,m}\langle n|\,\hat{O}\,\mathrm{e}^{-t\hat{H}}\,|m\rangle\,\langle m|\,\hat{\overline{O}}\,\mathrm{e}^{-(T-t)\hat{H}}\,|n\rangle\,, \tag{2.39}$$

$$= \frac{1}{Z}\sum_{n,m}\mathrm{e}^{-tE_m}\,\mathrm{e}^{-(T-t)E_n}\,\langle n|\,\hat{O}\,|m\rangle\,\langle m|\,\hat{\overline{O}}\,|n\rangle\,. \tag{2.40}$$

If t and $T-t$ are large, the largest contributions come from the eigenstates with smallest eigenvalues, $E_0 < E_1 < ... < E_n$ and the operator $\hat{O}$ filters out the states that have quantum numbers according to $\hat{O}$ with ground state $|1\rangle$,

$$\begin{aligned}\langle O(n_t)\overline{O}(0)\rangle \underset{\text{large } T-t}{\overset{\text{large } t}{\longrightarrow}} \frac{1}{Z}\big[&|\langle 0|\,\hat{O}\,|0\rangle|^2\,\mathrm{e}^{-TE_0}\\ &+|\langle 0|\,\hat{O}\,|1\rangle|^2(\mathrm{e}^{-t(E_1-E_0)-TE_0}+\mathrm{e}^{t(E_1-E_0)-TE_1})\\ &+\mathcal{O}(\mathrm{e}^{-t(E_2-E_0)-TE_0}+\mathrm{e}^{t(E_2-E_0)-TE_2})\big],\end{aligned} \tag{2.41}$$

The factor e^{-TE_0} can be pulled out in the numerator and the denominator $Z = \sum_n \mathrm{e}^{-TE_n}$, and all energies can be converted into energy differences to the vacuum, $\Delta E_n = E_n - E_0$. The term $|\langle 0|\,\hat{O}\,|0\rangle|^2$ is the squared vacuum expectation value of the field O and just the second term in equation (2.35), which is removed in the two-point function. For large T all terms of the denominator except the first one vanish and also in the numerator some terms disappear to result in

$$\begin{aligned}C^{2\text{pt}}(n_t) \overset{\text{large } t\ll T}{\longrightarrow} &|\langle 0|\,O\,|1\rangle|^2(\mathrm{e}^{-t\Delta E_1}+\mathrm{e}^{(t-T)\Delta E_1})\\ &+\quad \mathcal{O}(\mathrm{e}^{-t\Delta E_2}+\mathrm{e}^{(t-T)\Delta E_2}).\end{aligned} \tag{2.42}$$

For $\vec{p} = \vec{0}$ it is $\Delta E_l = m_l$, the mass of lth excited state. Therefore the exponential decay of the two-point function includes the masses of the particles which are created by $\hat{\overline{O}}$, denoted by the term $\mathcal{O}(\mathrm{e}^{-t\Delta E_2}+\mathrm{e}^{(t-T)\Delta E_2})$ in (2.42). The interpolating field $\hat{\overline{O}}$ creates not

only the ground state with mass m_1, but also excited states. The ground state mass can be extracted from the exponential decay of (2.42) when using large time t.

Additionally, the decay constant of the ground state can be calculated from the amplitude of the two-point function, $|\langle 0|\hat{O}|1\rangle|^2$.

2.5.3 *Three-point functions*

Hadron structure observables, like form factors, charges or transition amplitudes, can be obtained from matrix elements of hadron interactions. The matrix element of two baryons with momentum p,p' in states $|a(p)\rangle$ and $|b(p)\rangle$ respectively, interacting through an operator $\hat{G}$ with some Dirac structure Γ, is given by

$$\langle b(p)|\hat{G}(\Gamma)\left|a(p')\right\rangle = \overline{u}(p)\,B(q,\Gamma)\,u(p'), \tag{2.43}$$

with the baryon spinors $u, \overline{u}$ and some function $B(q,\Gamma)$, dependent on the initial and final baryon states and the structure of the operator $\hat{G}$. $B(q,\Gamma)$ includes the hadron structure observables: For the same initial and final state and zero momentum transfer, $B(\Gamma)$ includes the coupling or charge g_G of the baryon to the field G created by $\hat{G}$. With finite momentum transfer, $B(q,\Gamma)$ includes form factors $f(q^2)$ of the interaction. For different initial and final states, $B(q,\Gamma)$ includes transition amplitudes between these states. Examples of these matrix elements and the concrete form of $B(q,\Gamma)$ are given in section 2.6.

On the lattice, the matrix element in (2.43) for the same initial and final state with zero momentum transfer can be derived from the expectation value $\langle O(n_t)G(i_t)\overline{O}(0)\rangle$ of a hadron field $\overline{O}(m)$ at lattice site n (choose again $m = (0,\vec{0})$), an operator $\hat{G}(i)$ at site $i = (i_t,\vec{i})$ and a hadron field $O(n)$ at site $n = (n_t,\vec{n})$. Both operators $\hat{O}$ and $\hat{\overline{O}}$ are constructed such that they only overlap with baryon states. The hadronic three-point function $C^{3\mathrm{pt}}(n_t,\vec{p},i_t)$, using Fourier transformation with momentum $\vec{p}$ of initial and final state, is defined as the connected correlation function of three interpolating fields, similar to the two-point function in (2.35), and decays exponentially for large times,

$$C^{3\mathrm{pt}}(n_t,\vec{p},i_t) \overset{\text{large } t,t_I,T}{\longrightarrow} A\,\mathrm{e}^{-t\Delta E_1}, \tag{2.44}$$

with $t = an_t$, $t_I = ai_t$ and $T = aN_T$. The amplitude A is proportional to the matrix element $\langle 1|\hat{G}|1\rangle$ of the baryon ground state $|1\rangle$ interacting with the field G. This matrix element has the form of (2.43) and therefore can give the charge g_G.

The exponential decay in (2.44) can be deduced using the mentioned expectation value and the definition in (2.23),

$$\langle O(n_t)G(i_t)\overline{O}(0)\rangle = \frac{1}{Z}\mathrm{Tr}[\hat{O}(n_t)\hat{G}(k_t)\hat{\overline{O}}(0)\,\mathrm{e}^{-T\hat{H}}]. \tag{2.45}$$

The time-dependent operators are converted to the Schrödinger picture, the trace is evaluated and two complete sets of states are inserted to give

$$\langle O(n_t)G(i_t)\overline{O}(0)\rangle = \frac{1}{Z}\sum_{l,m,n} \mathrm{e}^{t(E_l-E_m)}\,\mathrm{e}^{t_I(E_m-E_n)}\,\mathrm{e}^{-TE_l}\cdot \\ \cdot \langle l|\,\hat{O}\,|m\rangle\,\langle m|\,\hat{G}\,|n\rangle\,\langle n|\,\hat{\overline{O}}\,|l\rangle \tag{2.46}$$

For large T the largest contributions come from the vacuum state

$$\langle O(n_t)G(i_t)\overline{O}(0)\rangle \overset{\text{large } T}{\longrightarrow} \frac{1}{Z}\sum_{m,n} \mathrm{e}^{t(E_0-E_m)}\,\mathrm{e}^{t_I(E_m-E_n)}\,\mathrm{e}^{-TE_0}\cdot \\ \cdot \langle 0|\,\hat{O}\,|m\rangle\,\langle m|\,\hat{G}\,|n\rangle\,\langle n|\,\hat{\overline{O}}\,|0\rangle\,. \tag{2.47}$$

Additionally, for large t and t_I the largest contributions come from the eigenstates with the smallest eigenvalues that are filtered out by the operators $\hat{O}$ and $\hat{G}$,

$$\langle O(n_t)G(i_t)\overline{O}(0)\rangle \overset{\text{large } T,t,t_I}{\longrightarrow} (\text{VEV}) \\ + \mathrm{e}^{-t\Delta E_1}\,\langle 0|\,\hat{O}\,|1_O\rangle\,\langle 1_G|\,\hat{G}\,|1_G\rangle\,\langle 1_O|\,\hat{\overline{O}}\,|0\rangle \\ + \mathcal{O}(\mathrm{e}^{\Delta E_1(t_I-T)} + \mathrm{e}^{\Delta E_1(t-t_I-T)}). \tag{2.48}$$

For $\hat{O}$ and $\hat{G}$ having different quantum numbers, additional contributions occur in $\mathcal{O}(\ldots)$ which involve energies of different states. The vacuum expectation values is subtracted in the definition of the three-point function.

It is not straightforward to deduce the matrix element $\langle 1|\,G\,|1\rangle$ from the exponential decay of the three-point function. Therefore the ratio of three-point over two-point function is introduced,

$$R(n_t, i_t) \overset{\text{large } T,t,t_I}{\longrightarrow} \frac{C^{3\text{pt}}(n_t, i_t)}{C^{2\text{pt}}(n_t)} \tag{2.49}$$

$$\approx \frac{\mathrm{e}^{-t\Delta E_1}\,\langle 0|\,\hat{O}\,|1_O\rangle\,\langle 1_G|\,\hat{G}\,|1_G\rangle\,\langle 1_O|\,\hat{\overline{O}}\,|0_O\rangle}{\mathrm{e}^{-t\Delta E_1}\,\langle 0|\,\hat{O}\,|1_O\rangle\,\langle 1_O|\,\hat{\overline{O}}\,|0\rangle} \tag{2.50}$$

$$\approx \langle 1|\,\hat{G}\,|1\rangle + \mathcal{O}(\mathrm{e}^{-(\Delta E_2-\Delta E_1)t_I} + \mathrm{e}^{-(\Delta E_2-\Delta E_1)(t-t_I)}\,. \tag{2.51}$$

For large times this ratio cancels the contributions of the exponential decay and the amplitudes involving the operators $\hat{O}$ and $\hat{\overline{O}}$ and excited states are suppressed. The ratio gives the matrix element of the hadron ground state interacting with the operator $\hat{G}$.

For $\vec{q} \neq \vec{0}$ this ratio is more complicated, but still consists of some combination of two-point and three-point functions.

2.5.4 *Quark-connected and -disconnected diagrams*

Hadron observables can be computed by evaluating two- and three-point functions of hadron interpolating fields, that means evaluating

the path integral in (2.27) with analytically solvable fermionic integral (2.27). In general, this fermionic integral gets contributions from quark-connected and -disconnected diagrams. In the following the analytic solution of the fermionic path integral is derived for a specific two-point function, showing the two diagram types. The shown principle can be applied to any other two-point function, as well as to three-point functions.

The isoscalar interpolating meson field involving only up and down quarks is given by

$$O = \frac{1}{\sqrt{2}}(\overline{u}\Gamma u + \overline{d}\Gamma d), \tag{2.52}$$

for a general Γ-matrix.

The fermionic two-point function, that is the fermionic path integral in (2.27) over two correlating meson fields, $O\overline{O}$ is given by,

$$C = \frac{1}{2}\left(\langle \overline{u}\Gamma u\, \overline{u}\Gamma u\rangle_F + \langle \overline{d}\Gamma d\, \overline{d}\Gamma d\rangle_F + \langle \overline{u}\Gamma u\, \overline{d}\Gamma d\rangle_F + \langle \overline{d}\Gamma d\, \overline{u}\Gamma u\rangle_F\right). \tag{2.53}$$

With the hadron source $\overline{O}$ at site 0 and the sink O at site n, the first term of (2.53), ignoring a factor $1/2$, is given by

$$C_1(n) = \langle \overline{u}(n)\Gamma u(n)\, \overline{u}(0)\Gamma u(0)\rangle_F. \tag{2.54}$$

The Wick contractions, the possible combinations of $\overline{u}$ and u are here

$$C_1(n) = \langle \overline{u}(n)\Gamma u(n)\overline{u}(0)\Gamma u(0)\rangle_F + \langle \overline{u}(n)\Gamma u(n)\overline{u}(0)\Gamma u(0)\rangle_F \tag{2.55}$$
$$\qquad\qquad \textit{disconnected} \qquad\qquad \textit{connected}$$

Already here it is obvious that C_1 is composed of two very distinct objects. The second one connects the two sites n and 0, such that one u is created at n and annihilated at 0 and one u vice versa and is called *connected part*. The first summand separates n and 0, one u is created and annihilated at 0, the other at n, and is therefore called *disconnected part* or *loop*. By reordering and taking into account the anticommutation relations of fermions, it is

$$\begin{aligned} C_1(n) = &\quad \mathrm{Tr}[\Gamma\, \langle u(n)\overline{u}(n)\rangle_F]\ \mathrm{Tr}[\Gamma\, \langle u(0)\overline{u}(0)\rangle_F] \\ &- \mathrm{Tr}[\Gamma\, \langle u(n)\overline{u}(0)\rangle_u\, \Gamma\, \langle u(0)\overline{u}(n)\rangle_u] \qquad (2.56) \\ = &\quad \mathrm{Tr}[\Gamma\, G_u(n|n)]\ \mathrm{Tr}[\Gamma\, G_u(0|0)] \quad \textit{disconnected} \\ &- \mathrm{Tr}[\Gamma\, G_u(n|0)\, \Gamma\, G_u(0|n)] \qquad \textit{connected}, \qquad (2.57) \end{aligned}$$

where (2.30) with $a = 1$ is used in the second step.

C_1 depends on the Γ-matrix of the simulated meson and the up-quark propagator G_u, the inverse Dirac matrix for the up-quark[1]. The

1 The Dirac matrices for up and down quarks differ in the twisted mass formulation.

other terms in the two-point correlation function in (2.53) include similar terms, some are dependent on the down-quark propagator G_d. The Dirac matrix for a specific link configuration is known, but its inversion is computationally expensive because of its large size, for typical lattice computations at least $\mathcal{O}(10^6 \times 10^6)$. Therefore the main part of computing two-point functions is the inversion of the Dirac matrix. Equation (2.57) shows clearly that the connected part needs only the one-to-all propagator $G_u(n|0)$ with one fixed source, where the disconnected part needs the all-to-all propagator $G_u(n|m)$ with $m = n$. In contrast to the computation of $G_u(n|0)$, the evaluation of the all-to-all propagator $G(n|m)$ needs stochastic input. Therefore its estimation is much noisier than the one of $G(n|0)$.

Disconnected contributions occur only in correlation function where quark and anti-quark from the same lattice site are contracted, therefore for interpolating fields which include q and $\overline{q}$ of the same quark-flavor. This is the case for the meson isoscalar ($I = 0$) two-point function in (2.52), corresponding to (2.33) with $\tau_a = \mathbb{1}$. The meson isovector ($I = 1$) has three components: $I_z = 0$ with $\tau_a = \tau_3$ ($\sim \overline{u}\Gamma u - \overline{d}\Gamma d$) results in disconnected diagrams in the two-point function, which cancel each other in the case of exact isospin symmetry, $D_u = D_d$. The two-point functions of the other two components do not get contributions from disconnected diagrams. For baryon two-point functions also no disconnected contributions exist because one baryon interpolating field (2.34) is generated out of only quark or only anti-quark fields and therefore only quark and anti-quark from different nodes are contracted. For baryonic three-point functions the insertion operator $\hat{G}$ has to be isoscalar-like in order to produce disconnected contributions.

The values of connected and disconnected parts can vary between different fermion discretization schemes. The next section presents values of disconnected contribution to observables for twisted mass fermions.

2.6 DISCONNECTED DIAGRAMS IN NUCLEON STRUCTURE OBSERVABLES

As already discussed in section 2.5.4, disconnected diagrams can only occur if at least one interpolating field or operator in the two- or three-point function consists of a $q\overline{q}$ combination for one specific quark-flavor q. It is not clear beforehand how large the disconnected contribution to an observable is, therefore computations of connected and disconnected parts have to be done to find that out. Here some computation results of nucleon structure observables with disconnected contributions are presented for the twisted mass fermion discretization. From these results it becomes clear that the disconnected contributions vary in size from observable to observable and contribute to

some of the observables significantly. Additionally, the error estimates of some disconnected contributions are rather large. Therefore the computation of disconnected diagrams is important and optimized methods for their computation need to be developed.

The nucleon is still not fully understood. Therefore many lattice QCD calculations focus on the computation of its charges and form factors. The general matrix element for these computation is

$$\left\langle N(p,s')\right| \hat{O}_{\Gamma,q} \left|N(p,s)\right\rangle . \tag{2.58}$$

for a nucleon N with momentum p and spin s and an insertion operator $\hat{O}_{\Gamma,q}$, compare (2.43). The matrix elements that include an isoscalar operator get contributions from disconnected diagrams. Operators for the two light degenerate quarks, $O_{\Gamma,u+d}$, as well as operators for the heavier strange $O_{\Gamma,s}$ and charm $O_{\Gamma,c}$ quarks are defined by

$$O_{\Gamma,u+d} = \frac{1}{2}(\overline{u}\Gamma u + \overline{d}\Gamma d), \tag{2.59}$$

$$O_{\Gamma,s} = \overline{s}\Gamma s, \tag{2.60}$$

$$O_{\Gamma,c} = \overline{c}\Gamma c. \tag{2.61}$$

It should be noted that the matrix elements using $O_{\Gamma,s}$ or $O_{\Gamma,c}$ only get disconnected contributions because the nucleon includes no valence strange or charm quarks with which the strange or charm quarks from the insertion operator could contract.

Results of charges and form factors of some recent lattice computations for different Γ-matrices are shown in the following. These simulations use physical quark masses for two different lattice: $N_f = 2$, $48^3 \times 96$ lattice sites, $a = 0.0938(2)$ fm [11, 12] and $N_f = 2+1+1$, $32^3 \times 64$ lattice sites, $a = 0.082(4)$ fm [4]. The resulting observables in the articles are given in Table 2.2, the text below states the percentage of the error estimate of connected and disconnected part over the full value (connected part plus disconnected part). For the presented results the relative error estimates of the disconnected parts are significantly larger than the typical error estimates of the connected parts for the charges of charm and strange quark, which are purely disconnected, and the structure functions $\langle x\rangle_{u+d}$ and $B_{20}^{u+d}(0)$, both derived from the vector derivative matrix element.

SCALAR CHARGE The nucleon scalar matrix element with $\Gamma = \mathbb{1}$,

$$\left\langle N(p,s')\right| \hat{O}_{S,q} \left|N(p,s)\right\rangle = \overline{u}_N(p,s') \left[\frac{1}{2} G_S^q(0)\right] u_N(p,s) \tag{2.62}$$

gives the scalar charge $G_S^q(0) = g_S^q$. This is the coupling of the nucleon to scalar particles and, in particular, to the Higgs field, and is therefore an essential ingredient in beyond the standard model (BSM)

physics. Direct detection experiments of dark matter candidates measure the recoil energy of nuclei scattered off a dark matter candidate. In many BSM theories this candidate is a weakly interacting massive particle (WIMP), interacting with nucleons through a Higgs boson because of its large mass. At zero momentum transfer the spin-independent elastic cross section of this process is proportional to the squared scalar matrix element [44] and is therefore very sensitive to the value of the matrix element (2.62). There are no direct experimental measurements of g_s, but there are phenomenological values of the related σ-term, e.g. [10]. The error estimate for computation in [12] of the connected part of g_S^{u+d} is approximately 6% of the full g_S^{u+d} value, while the error estimate of the disconnected part is around 3%. The error of the fully disconnected g_S^s is 24% and of g_S^c is 23%.

TENSOR CHARGE The tensor matrix element uses $\Gamma = \sigma^{\mu\nu} = \frac{1}{2}[\gamma_\mu, \gamma_\nu]$ for

$$\left\langle N(p,s') \right| O_{T,q}\mu\nu \left| N(p,s) \right\rangle = \overline{u}_N(p,s') \left[\frac{1}{2} A_{T10}^q(0) \sigma^{\mu\nu} \right] u_N(p,s) \tag{2.63}$$

to get the tensor charge $A_{T10}^q(0) = g_T^q$. It gives the leading contribution of the electric dipole moment of the quarks to the neutron electric dipole moment (nEDM). This neutron electric dipole moment is CP-violating and is therefore a good indicator for BSM physics. The standard model nEDM is supposed to be around $10^{-32} e \cdot \text{cm}$, but new physics models, such as supersymmetry have larger nEDMs [84]. There are measurements of the tensor charge, e.g. [18]. In [12] the connected part of g_T^{u+d} has an error estimate of 3%, while the disconnected part has 1%. The error of the fully disconnected g_T^s is estimated by 23% and the one of g_T^c 103%.

AXIAL CHARGE The axial charge is known very well experimentally [76] and is computed by the axial-vector current with $\Gamma = \gamma_5\gamma_\mu$,

$$\left\langle N(p,s') \right| O_{A,a}^\mu \left| N(p,s) \right\rangle = i\overline{u}_N(p,s') \left[\frac{1}{2} G_A^a(0) \gamma_5 \gamma_\mu \right] u_N(p,s), \tag{2.64}$$

with $G_A^q(0) = g_A^q$. It gives the intrinsic spin carried by the quarks, which is $\frac{1}{2} g_A^q$. The isovector axial charge is measured quite accurately in the neutron beta decay [76]. In [4] the connected part error of g_A^{u+d} is 3%, the disconnected part error 2% large. The error for g_A^s is given by 15%.

QUARK MOMENTUM FRACTION AND TOTAL ANGULAR MOMENTUM The vector derivative matrix element uses the symmetrized derivative operator $\Gamma = \gamma^{\{\mu} \overleftrightarrow{D}^{\nu\}}$,

$$\left\langle N(p,s') \right| O^{\mu\nu}_{V,q} \left| N(p,s) \right\rangle = i\overline{u}_N(p,s') \left[\frac{1}{2} A^q_{20}(0) \gamma^{\{\mu} p^{\mu\}} \right] u_N(p,s), \tag{2.65}$$

for the average momentum fraction of a quark q in the nucleus, $A^q_{20}(0) = \langle x \rangle_q$. In [4] the connected part error estimate of $\langle x \rangle_{u+d}$ is 4%, the one of the disconnected part 12%.

Structure functions as $A^q_{20}(q^2)$ for $q^2 \neq 0$ give more insight into the internal structure of the nucleon. The total angular momentum of one quark is computed via two of these structure functions, $J_q = \frac{1}{2}(\langle x \rangle_q + B^q_{20}(0))$, using the $q^2 \neq 0$ vector derivative matrix element

$$\begin{aligned} \left\langle N(p',s') \right| O^{\mu\nu}_{V^a} \left| N(p,s) \right\rangle &= i\overline{u}_N(p',s') \left[\frac{1}{2} \Lambda^{\mu\nu}_q(q^2) \right] u_N(p,s), \\ \Lambda^{\mu\nu}_q(q^2) &= A^a_{20}(q^2) \gamma^{\{\mu} p^{\mu\}} + B^a_{20}(q^2) \frac{i\sigma^{\{\mu\rho} q_\rho P^{\nu\}}}{2m} + C^a_{20}(q^2) \frac{q^{\{\mu} q^{\nu\}}}{m}. \end{aligned} \tag{2.66}$$

B_{20} and C_{20} can be deduced by computing the matrix element at various momenta and do an extrapolation to $q^2 = 0$. In [4] the error estimate of the connected part of B^{u+d}_{20} is 4%, for the disconnected part 80%. The angular momentum of the quarks is computed in [4] to $J_u = 0.202(78)$ and $J_d = -0.078(78)$.

GLUON TOTAL ANGULAR MOMENTUM AND NUCLEON SPIN The forming of the nucleon spin from its constituents was unclear since in 1987 the European Muon Collaboration found that the spin of the quarks constitute only a small fraction of the nucleon spin [19, 20]. Recent experiments suggest a non-zero gluonic spin contribution [8, 45], but are very imprecise. On the lattice the gluon total angular momentum can be computed via $O^{\mu\nu}_{V,g} = F^{\{\mu\rho} F^{\nu\}}_{\rho}$ with

$$\left\langle N(p,s') \right| O^{\mu\nu}_{V,g} \left| N(p,s) \right\rangle = i\overline{u}_N(p,s') \left[\frac{1}{2} A^g_{20}(0) \gamma^{\{\mu} p^{\mu\}} \right] u_N(p,s), \tag{2.67}$$

with $\langle x \rangle_g = A^g_{20}(0)$ being the gluon form factor. The gluon's angular momentum is $J_g = \frac{1}{2}(\langle x \rangle_g + B^g_{20}(0))$. This calculation is done in [11], omitting $B^g_{20}(0)$. The computation is purely disconnected and gives an error of 6% on $\langle x \rangle_g$. The nucleon spin can be calculated, using Ji's sum rule [65] $J_N = \sum_q J_q + J_g$, which is checked in [11] with a result $J_N = 0.541(79)$ to sum up to $\frac{1}{2}$.

	conn.	$\frac{\Delta\text{conn.}}{\text{conn.+disc.}}$ [%]	disc.	$\frac{\Delta\text{disc.}}{\text{conn.+disc.}}$ [%]	ref.
g_S^{u+d}	8.221(610)	6.4	1.249(266)	2.8	[12]
g_S^s			0.329(78)	23.7	[12]
g_S^c			0.062(14)	23.0	[12]
g_T^{u+d}	0.582(16)	2.9	−0.0213(54)	1.0	[12]
g_T^s			−0.00319(72)	22.7	[12]
g_T^c			−0.00263(272)	103.2	[12]
g_A^{u+d}	0.576(13)	2.6	−0.0699(89)	1.8	[4]
g_A^s			−0.0227(34)	15.0	[4]
$\langle x \rangle_{u+d}$	0.586(22)	3.6	0.027(76)	12.4	[4]
B_{20}^{u+d}	−0.035(16)	4.4	−0.33(29)	79.5	[4]
$\langle x \rangle_g$			0.267(16)	5.9	[11]

Table 2.2: The relative error estimates of the disconnected part of nucleon charges and form factors are larger than the relative error estimates of the connected parts. Results of some recent twisted mass lattice computations.

3

IMPROVED METHODS FOR DISCONNECTED DIAGRAMS

The accurate computation of quark disconnected diagrams is crucial to precise measurements of many QCD observables, see section 2.6. A disconnected diagram results from integrating out fermions in the path integral. The quark fields in the observable, which can be contracted by some Dirac structure Γ, are combined via Wick-contractions. Contractions that include only quarks at the same lattice site give rise to disconnected diagrams, compare section 2.5.4. A disconnected diagram describes the propagation of a quark field from one lattice site $n \in \Lambda$ to itself and is therefore defined via the all-to-all propagator $G(n|m)$ with $m = n$. Its general form is

$$L(n, \Gamma) = \mathrm{tr}[G(n|n)\Gamma], \tag{3.1}$$

compare (2.57), and can be called loop as well. Projection to some definite spatial hadron momentum $L(\vec{p}, n_t, \Gamma)$, as it is needed for hadron observables, results in a sum over spatial site vector $\vec{n}$ in (3.1) and the need for $G(n|n)$ for all n. The propagator is the inverted Dirac matrix D where its corresponding operator is defined in (2.10) for Wilson fermions and in section 2.3 for twisted mass fermions. The inversion of the Dirac matrix is done by solving several linear equations

$$D\Psi = b, \tag{3.2}$$

for Ψ, with varying right-hand sides b. Finding the solution Ψ to one equation (3.2) is denoted by one *inversion*. This is also done to compute the one-to-all propagator $G(n|m_o)$ with fixed source m_0, which is needed in the connected diagrams. However, the all-to-all propagator needs the solution of orders of magnitude more equations (3.2) and is therefore computationally much more demanding. Several improved methods have been developed to invert D. These methods can be split into improvements reducing the runtime of one inversion (3.2) and improvements reducing the number of inversions needed to arrive at a given error estimate of the solution in equation (3.1).

Despite these improved methods, disconnected diagram computations still suffer from long runtimes to get meaningful results. Hence, with the given supercomputer resources, there is still demand for additional methods to be able to make statements about specific observables in a reasonable total runtime. A promising idea to reduce the runtime is to use the eigenmodes of the Dirac matrix D in the solving algorithm. We implemented the exact eigenmode reconstruction with deflation method to compute one part of the twisted mass loop in

(3.1) exactly with the help of the Dirac matrix eigenvectors and the other part stochastically via (3.2) by deflating the Dirac matrix with the same eigenvectors.

This chapter introduces different methods to improve the computation of disconnected diagrams. First, it presents two methods to reduce the number of inversions, stochastic sources and the one-end trick, and two to reduce the runtime per inversion, even-odd preconditioning and initial guess deflation. The chapter mainly focuses on the exact eigenmode reconstruction with deflation method and shows results of this method's application. Finally the performance of this method is compared to the Multigrid method, which reduces the runtime per inversion very efficiently.

In our test case of applying exact eigenmode reconstruction with deflation we found a performance gain of approximately 5.5. This gain was achieved because we found that the method does not only reduce the runtime of one inversion, but reduces also the number of inversions needed to arrive at a given error estimate. The so-gained runtime can be used to reduce the error estimate for more precise observable results. There are active developments on improving the computation of disconnected diagram contributions and the exact eigenmode reconstruction with deflation method presents one potential step to more precise computations of noisy observables on the lattice. One recent advance in this field was achieved with the Multigrid algorithm, which gives performance gains up to order 100 when applied to twisted mass. In the future this development of disconnected diagram computations can lead to lattice results with error estimates that are smaller than experimental uncertainties and therefore are very well suited as input to experimental analysis, in tests of the standard model and in the finding of new physics parameters spaces.

3.1 STOCHASTIC SOURCES

Instead of computing a one-to-all propagator value $\underset{ba}{G_{\beta\alpha}}(n|m_0)$ with point sources by solving the 12 equations

$$\underset{ab}{D_{\alpha\beta}}(m|n)\underset{ba_0}{G_{\beta\alpha_0}}(n|m_0) = \delta_{mm_0}\delta_{\alpha\alpha_0}\delta_{aa_0} \tag{3.3}$$

with the Dirac matrix D for all color and Dirac indices a_0 and α_0, the computation of disconnected diagrams needs all-to-all propagator values $\underset{ba}{G_{\beta\alpha}}(n|m)$. This means V_{lat} times more inversions of the form (3.3), which results for QCD-computations in at least $\mathcal{O}(10^6)$ inversions. Instead, R stochastic sources η_r, also called random vectors, can be used, see Appendix A in [27], to solve the R equations

$$\underset{ab}{D_{\alpha\beta}}(m|n)\underset{b}{\Psi^r_\beta}(n) = \underset{a}{\eta^r_\alpha}(m). \tag{3.4}$$

Here, R is normally of the order 10^3 but can also be of the order 10, depending on the computed observable and the desired final error estimate. Equation (3.4) can be solved efficiently with the Conjugate Gradient[1] (CG) [81] or some related algorithm. Then the loop in (3.1), omitting color and Dirac indices and changing the position of the source index for better readability, is approximated by

$$L(m,\Gamma) = \frac{1}{R}\sum_{r=1}^{R} \mathrm{tr}[\eta_r^\dagger(m)\,\Gamma\,\Psi_r(m)] + \mathcal{O}\left(\frac{1}{\sqrt{R}}\right), \tag{3.5}$$

if the stochastic sources are chosen such that

$$\lim_{R\to\infty}\frac{1}{R}\sum_{r=1}^{R}\eta^r_{\substack{\alpha\\a}}(m)\eta^{r\,\dagger}_{\substack{\beta\\b}}(n) = \delta_{mn}\delta_{\alpha\beta}\delta_{ab}, \tag{3.6}$$

$$\lim_{R\to\infty}\frac{1}{R}\sum_{r=1}^{R}\eta^r_{\substack{\alpha\\a}}(m) = 0_{\substack{\alpha\\a}}(m), \tag{3.7}$$

where $0_{\substack{\alpha\\a}}(m)$ denotes the zero vector in all indices α, a, m. Best results were found by using the Z_2 noise, more specifically $Z_2 \otimes Z_2$, such that $\eta^r_{\substack{\alpha\\a}}(x) \in \{\frac{1}{\sqrt{2}}(\pm 1 \pm i)\}$ [42, 48]. These sources introduce stochastic noise, which decreases with $1/\sqrt{R}$, dependent on the number of stochastic sources used. This noise is added to the gauge noise coming from the fact that not infinitely many configurations are used in the evaluation of the path integral in (2.29). It is important to mention that there is no summation convention used in (3.5) for m and the summation over Dirac and color indices is carried out but not shown in (3.5) explicitly.

3.2 ONE-END TRICK

For twisted mass fermions the one-end trick [4, 49, 72] combines the computation of two loops via stochastic sources into one computation. This enhances the signal-to-noise ratio of the result and therefore lead to a reduced number of inversions needed. These two loops of the form (3.1) in one computation incorporate u-quark and d-quark propagator. Depending on the observable, more specifically whether it includes an isovector or isoscalar state, this combination is either $G_u - G_d$, where the standard one-end trick can be used, or $G_u + G_d$, where the general one-end trick can be applied. It is important to mention that the Dirac- and iso-structure of states differ between their physical and twisted mass representation and this section is only concerned with twisted mass fermions. Additionally, note that for twisted mass fermions $D_u - D_d$ is nonzero.

1 In most simulations $D^\dagger D\Psi = D^\dagger b$ is solved instead of (3.4) because the solving CG algorithm needs a positive definite matrix.

STANDARD ONE-END TRICK For an observable including an isovector state, the disconnected diagram evaluation needs the computation of

$$L^{u-d}(m,\Gamma) = \mathrm{tr}[(G_u(m|m) - G_d(m|m))\Gamma]. \tag{3.8}$$

The twisted mass Dirac matrix, compare section 2.3, has the property

$$\begin{aligned} D_d(m|n) - D_u(m|n) &= -2i\mu\gamma^5\delta_{mn} && (3.9)\\ \Leftrightarrow G_u(m|n) - G_d(m|n) &= -2i\mu \sum_l G_d(m|l)\,\gamma^5\, G_u(l|n) && (3.10)\\ &= -2i\mu \sum_l G_d(m|l) {G_d}^\dagger(l|n)\,\gamma^5, && (3.11) \end{aligned}$$

where $D_u = \gamma^5 {D_d}^\dagger \gamma^5$ was used in the last step. With $G_d {G_d}^\dagger = G_u {G_u}^\dagger \stackrel{\text{def}}{=} GG^\dagger$, (3.8) becomes

$$L^{u-d}(m,\Gamma) = -2i\mu \sum_n \mathrm{tr}\Big[G(m|n)G^\dagger(n|m)\,\gamma^5\Gamma\Big]. \tag{3.12}$$

Inserting unity in form of (3.6) between G and $G^\dagger$ results in

$$\begin{aligned} L^{u-d}(m,\Gamma) = -\frac{2i\mu}{R} \sum_{n,l} \sum_{r=1}^{R} \mathrm{tr}\Big[G(m|n)\,\eta_r(n)\eta_r^\dagger(l)\,G^\dagger(l|m)\,\gamma^5\Gamma\Big] \\ + \mathcal{O}\left(\frac{1}{\sqrt{R}}\right). \end{aligned} \tag{3.13}$$

Now the equations

$$D(n|m)\Psi_r(m) = \eta_r(n) \tag{3.14}$$

for just one flavor u or d have to be solved for $\Psi_r(m)$ such that $\Psi_r(m) \approx G(m|n)\eta_r(n)$ can be plugged in (3.13),

$$L^{u-d}(m,\Gamma) = -\frac{2i\mu}{R} \sum_{r=1}^{R} \mathrm{tr}\Big[\Psi_r^\dagger(m)\,\gamma^5\Gamma\,\Psi_r(m)\Big] + \mathcal{O}\left(\frac{1}{\sqrt{R}}\right). \tag{3.15}$$

Therefore, although there are two propagators present in (3.8), the inversions in (3.14) have to be done for only one of them.

The signal-to-noise ratio of L^{u-d} is more favorable when using the one-end trick, [30]. The one-end trick uses the fact that the vector product (over Dirac and color indices) of Ψ_r and $\Psi_r^\dagger$ in (3.15) approximates the matrix product (over Dirac, color and space-time indices) of G and $G^\dagger$,

$$\begin{aligned} \frac{1}{R}\sum_r \Psi_r(m)\Psi_r^\dagger(m) &= \frac{1}{R}\sum_r \left(\sum_l G(m|l)\eta_r(l)\right)\left(\sum_i \eta_r^\dagger(i)G^\dagger(i|m)\right) \\ &= \sum_l G(m|l)G^\dagger(l|m) + \text{noise} \end{aligned} \tag{3.16}$$

to compute L^{u-d}. This has a signal of order $3 \cdot 4 \cdot V_{\text{lat}}$ due to the matrix multiplication $GG^\dagger$ in all Dirac, color and lattice indices. The error is of order $\sqrt{(3 \cdot 4 \cdot V_{\text{lat}})^2}$ because $(3 \cdot 4 \cdot V_{\text{lat}})^2$ entries of the noise vector η_r are involved in the multiplication $(G\eta_r) \cdot (\eta_r^\dagger G^\dagger)$ and therefore the signal-to-noise ratio is of order one. Without the trick the loop L^{u-d} is computed by using (3.5) for both matrices G_d and G_d separately,

$$\begin{aligned} \frac{1}{R} \sum_r \Psi_r(m) \eta_r^\dagger(m) &= \frac{1}{R} \sum_r \left(\sum_l G(m|l) \eta_r(l) \right) \cdot \eta_r^\dagger(m) \\ &= G(m|m) + \text{noise}. \end{aligned} \tag{3.17}$$

This has a signal of order one because $G(m|m)$ denotes only one value. The error is of order $\sqrt{3 \cdot 4 \cdot V_{\text{lat}}}$ because there are $3 \cdot 4 \cdot V_{\text{lat}}$ noise entries involved in the multiplication $(G\eta_r) \cdot \eta_r^\dagger$ in (3.17) and the signal-noise ratio is of order $\frac{1}{\sqrt{3 \cdot 4 \cdot V_{\text{lat}}}}$. Therefore the signal-to-noise ratio order of the one-end trick outperforms the one not using the trick.

GENERAL ONE-END TRICK For an observable including an isoscalar state, the disconnected diagram evaluation needs the computation of

$$L^{u+d}(m, \Gamma) = \text{tr}[(G_u(m|m) + G_d(m|m))\Gamma]. \tag{3.18}$$

The twisted mass Dirac matrix has the property

$$D_u(m|n) + D_d(m|n) = 2D_W, \tag{3.19}$$

with the Wilson Dirac matrix D_W, see (2.10). Applying the same steps as in the standard one-end trick results in

$$L^{u+d}(m, \Gamma) = \frac{2}{R} \sum_{r=1}^{R} \text{tr}\left[\Psi_r^\dagger(m) \, \gamma^5 \Gamma \gamma^5 \, D_W \, \Psi_r(m) \right] + \mathcal{O}\left(\frac{1}{\sqrt{R}} \right). \tag{3.20}$$

3.3 EVEN-ODD PRECONDITIONING

Most simulations in lattice QCD use even-odd preconditioning to increase the speed of one inversion. The condition number c of the matrix D quantifies how hard it is to solve (3.4) for Ψ, in fact the number of iterations for a CG algorithm is proportional to $\sqrt{c}$ [81]. The condition number is a measure for how close D is to the unit matrix, which would give a solution to (3.4) directly. The condition number of a normal matrix as $D^\dagger D$ can be computed by [81]

$$c(D^\dagger D) = \frac{\lambda_{\max}(D^\dagger D)}{\lambda_{\min}(D^\dagger D)}. \tag{3.21}$$

Preconditioning multiplies a preconditioning matrix $\mathscr{P}$ to both sides of (3.4) to arrive at

$$\mathscr{P}D\Psi = \mathscr{P}b, \tag{3.22}$$

such that the condition number of $\mathscr{P}D$ is smaller and therefore is closer to one than the condition number of D.

For even-odd preconditioning [39] the lattice is divided into two sub-lattices, one with all even lattice number sites, one with all odd ones. Because the lattice interactions are restricted to next-neighbor interactions, there are only even-odd and odd-even interaction and (3.2) can be written as

$$\begin{pmatrix} D_{ee} & D_{eo} \\ D_{oe} & D_{oo} \end{pmatrix} \begin{pmatrix} \Psi_e \\ \Psi_o \end{pmatrix} = \begin{pmatrix} b_e \\ b_o \end{pmatrix}. \tag{3.23}$$

The Wilson Dirac matrix, and therefore also the Wilson Dirac twisted mass matrix, can easily be divided into diagonal terms G_{ee}, G_{oo} and non-diagonal terms G_{eo}, G_{oe}, shown in (2.12). Multiplying both sides of (3.23) by the preconditioning matrix

$$\mathscr{P} = \begin{pmatrix} G_{ee} & 0 \\ -D_{oe}G_{ee} & 1 \end{pmatrix} \tag{3.24}$$

results in the two equations

$$(D_{oo} - D_{oe}G_{ee}D_{eo})\Psi_o = b_o - D_{eo}G_{ee}b_e, \tag{3.25}$$

$$\Psi_e = G_{ee}(b_e - D_{eo}\Psi_o). \tag{3.26}$$

The first one can be solved for Ψ_o by some inversion algorithm, e.g. CG. Then Ψ_e can be derived from the second equation and (3.23) is fully solved. The inversion of (3.25) is fast because it is done on the odd sub-lattice, which is half the size of the full lattice and it has a better condition number than the original equation (3.23) [64].

This even-odd preconditioning approach is possible because D_{ee} is diagonal and therefore easy to invert, which is just plugged into (3.25). For the Wilson Dirac matrix it is $G_{Wee} = D_{Wee} = \mathbb{1}$, compare (2.12). The even-odd form of the twisted mass Wilson Dirac matrix differs from the Wilson Dirac matrix in the diagonal terms $D_{u/d_{ee}} = D_{u/d_{oo}} = \mathbb{1} \pm i\mu\gamma^5$, but is still diagonal itself and therefore easy to invert as well,

$$G_{u/d_{ee}} = \frac{1}{1+\mu}(\mathbb{1} \mp i\mu\gamma^5). \tag{3.27}$$

3.4 INITIAL GUESS DEFLATION

The initial guess deflation method uses eigenmodes of the Dirac matrix D to increase the speed of one inversion. In most applications

the iterative CG algorithm or some related algorithm is used to solve the equation (3.2).The error of the CG algorithm after N iterations is bounded by $\Delta \leq \gamma^N$, for $\gamma > 0$ depending on the condition number of the involved matrix[2] [81]. The initial error can be reduced if the algorithm already starts with a good initial guess of the solution, Ψ_0. By default this vector is chosen to be the zero-vector. But if Ψ_0 is close to or even approximates the wanted solution Gb, it is possible that less iterations are needed to arrive at a specified error estimate. Of course the asymptotic error scaling of γ^N is not modified by choosing a good initial guess, but already few less needed iteration due to a reduced initial CG error can lead to significant runtime gains in the computation of observables, especially if many observables are computed from the same inverted Dirac matrix, e.g. to result in different hadron structure observables.

More information is needed to find this approximate solution. One possible way is to decompose the Dirac matrix into its eigenvalues and eigenvectors: Every diagonalizable Hermitian $N \times N$ matrix, here $D^\dagger D$, can be decomposed into its N eigenvectors v_i and corresponding eigenvalues λ_i.

$$D^\dagger D = \sum_{i=1}^{N} \lambda_i v_i v_i^\dagger, \quad (D^\dagger D)^{-1} = \sum_{i=1}^{N} \frac{1}{\lambda_i} v_i v_i^\dagger \tag{3.28}$$

Computing all eigenvectors and eigenvalues of $D^\dagger D$ and plugging them into (3.28) would result in a fully exact result for $(D^\dagger D)^{-1}$ without any need for an inversion algorithm. But in lattice QCD the Dirac matrix has normally $N \approx \mathcal{O}(10^8)$ and therefore the computation of all eigenvectors and eigenvalues is very time intensive. Exact deflation computes the N_{EV} eigenvectors with the smallest eigenvalues, because they give the most important contribution to $(D^\dagger D)^{-1}$ in (3.28). Then,

$$(D^\dagger D)^{-1} \approx \sum_{i=1}^{N_{\mathrm{EV}}} \frac{1}{\lambda_i} v_i v_i^\dagger. \tag{3.29}$$

With these smallest eigenvalues the starting vector can be chosen to be

$$\Psi_0 = \left(\sum_{i=1}^{N_{\mathrm{EV}}} \frac{1}{\lambda_i} v_i v_i^\dagger \right) D^\dagger b \approx Gb. \tag{3.30}$$

On the other hand the computation of the eigenvectors needs extra runtime.

This approach is used e.g. in [5] to reduce the computational cost to compute disconnected quantities.

2 $\gamma = \frac{\sqrt{c}-1}{\sqrt{c}+1}$ for condition number c.

3.5 EXACT EIGENMODES RECONSTRUCTION WITH DEFLATION

The previous section 3.4 presents a method which uses eigenmodes of the operator to reduce the runtime of one inversion in the form of an initial guess (3.30) for the linear equation that has to be solve. But the computation of the eigenmodes is very time intensive, therefore it would be preferable to use the full potential of the eigenmodes not only as an initial guess but for approximating the full propagator by using equation (3.29). This approximation is further described in more detail and applied to disconnected diagrams in [74], as well as in [13]. We implemented a method that not only uses this exact reconstruction of eigenmodes to compute one part of the propagator exactly, but also computes the other part stochastically by preconditioning the Dirac matrix with the eigenmodes. This reduces the runtime of one inversion as well as the number of inversions needed to arrive at a specific statistical uncertainty.

This section presents the principle of the method in more detail, describes how it is combined with the one-end trick to decrease the runtime even more, explains the difficulty combining it with even-odd preconditioning, gives some details about our implementation and finally shows first results.

Our test of exact eigenmodes reconstruction on a $16^3 \times 32$ lattice gave an 5.5 times smaller runtime than not using the method. This shows a potential for more accurate computations of disconnected diagrams when using the method. On the other hand, the new Multigrid algorithm gives a much better speed-up rate and is therefore presently superior for computing propagators for twisted mass fermions.

3.5.1 *The Principle*

Similar to the initial guess deflation in Section 3.4 the exact eigenmodes reconstruction with deflation uses the N_{EV} lowest eigenvalues with its corresponding eigenvectors, here to approximate the propagator $(D^\dagger D)^{-1}$. With these eigenvectors, one part of the inverse matrix can be computed exactly using the eigenvectors, and one part can be computed stochastically. The projector

$$P(m|n) = \sum_{i=1}^{N_{\mathrm{EV}}} v_i(m) v_i^\dagger(n) \tag{3.31}$$

splits the matrix $(D^\dagger D)^{-1}$ into one part $\mathcal{E}$, which is projected to the eigenvector space and is computed exactly, that means without

stochastic sources and one part $\mathcal{S}$, where the eigenvector space is projected out and which needs stochastic sources,

$$(D^\dagger D)^{-1} = P(D^\dagger D)^{-1} + (\mathbb{1} - P)(D^\dagger D)^{-1} \tag{3.32}$$
$$= \mathcal{E} + \mathcal{S}. \tag{3.33}$$

Then the exact part is calculated via

$$\mathcal{E}(m|n) = \sum_l P(m|l)(D^\dagger D)^{-1}(l|n) \tag{3.34}$$
$$= \sum_{i=1}^{N_{\text{EV}}} v_i(m) v_i^\dagger(n) \sum_{j=1}^{N} \frac{1}{\lambda_j} v_j(l) v_j^\dagger(n) \tag{3.35}$$
$$= \sum_{i=1}^{N_{\text{EV}}} \frac{1}{\lambda_i} v_i(m) v_i^\dagger(n). \tag{3.36}$$

The other part is computed stochastically,

$$\mathcal{S}(m|n) = \frac{1}{R} \sum_{r=1}^{R} \Psi_r(m) \eta_r^\dagger(n) + \mathcal{O}\left(\frac{1}{\sqrt{R}}\right), \tag{3.37}$$

similar to (3.5), but by solving

$$(\mathbb{1} - P)(D^\dagger D)\Psi_r(m) = \eta_r(m), \tag{3.38}$$

such that

$$\mathcal{S} = (\mathbb{1} - P)(D^\dagger D)^{-1} + \mathcal{O}\left(\frac{1}{\sqrt{R}}\right), \tag{3.39}$$

compare (3.32). Equation (3.38) can be rewritten, using that $(\mathbb{1} - P)^2 = \mathbb{1}$, to

$$(D^\dagger D)\Psi_r(m) = (\mathbb{1} - P)\eta_r(m), \tag{3.40}$$

such that the deflation of the matrix in (3.38) is equivalent to a deflation of the sources, which can be implemented easily.

Equation (3.38) can be interpreted as a linear equation with a preconditioning matrix $\mathscr{P} = (\mathbb{1} - P)$. The deflated matrix $(\mathbb{1} - P)(D^\dagger D)$ is deflated by the N_{EV} smallest eigenvectors, that means $\lambda_{\min}\big((\mathbb{1} - P)(D^\dagger D)\big) > \lambda_{\min}(D^\dagger D)$ and therefore, using (3.21), $c\big((1 - P)(D^\dagger D)\big) < c(D^\dagger D)$. Hence, the inversion of $(1 - P)(D^\dagger D)$ in (3.38) is faster than inverting the full matrix $D^\dagger D$. We computed the minimal eigenvalue of $D^\dagger D$ in our simulation described in section (3.5.4) to be $2.4 \cdot 10^{-6}$. We found that 100 eigenvectors are enough to deflate the matrix efficiently, compare section (3.5.5) below, and then the 100^{th} eigenvalue, $1.3107 \cdot 10^{-3}$ approximates the minimal eigenvalue of $(1 - P)D^\dagger D$. The maximal eigenvalue is not changed by the deflation. Therefore the condition number c is reduced around 550 times and the number of CG iterations, proportional to $\sqrt{c}$, decreases by a factor of approximately 23.

Therefore exact eigenmodes reconstruction with deflation has two main advantages: each inversion is faster because of this preconditioning and the method reduces the stochastic errors by computing one part of the observable exactly, which means that fewer inversions are needed in total to arrive at a specified error. Of course these advantages in runtime are reduced by the additional initial runtime to compute the eigenvectors. Therefore the parameter N_{EV}, the number of eigenvectors used, has to be chosen dependent on the actual setup of the simulation, i.e. the lattice size, the gauge coupling and the quark mass, to achieve an overall gain in runtime.

3.5.2 *Application to the standard one-end trick*

The standard one-end trick can be modified to incorporate exact eigenmodes reconstruction with deflation to increase the efficiency of the loop computation. Starting from (3.12) one loop is computed by

$$L(m,\Gamma) = -2i\mu \sum_n \mathrm{tr}\Big[G(m|n)G^\dagger(n|m)\,\gamma^5\Gamma\Big]. \tag{3.41}$$

The N_{EV} lowest eigenvectors of the matrix ${D_d}^\dagger D_d = {D_u}^\dagger D_u \overset{\mathrm{def}}{=} D^\dagger D$, which is related to the propagator via $(D^\dagger D)^{-1} = GG^\dagger$ with

$$(D^\dagger D)(m|n)\, v_i(n) = \lambda_i\, v_i(m), \qquad i \in \{1,...,N_{\mathrm{EV}}\}, \tag{3.42}$$

are used to define the projector

$$P(m|n) = \sum_{i=1}^{N_{\mathrm{EV}}} v_i(m)\, v_i^\dagger(n). \tag{3.43}$$

Adding a zero to (3.41) gives

$$\begin{aligned} L(m,\Gamma) = -2i\mu \sum_{n,l} \Big(&\mathrm{tr}\Big[G(m|n)G^\dagger(n|l)\,P(l|m)\,\gamma^5\Gamma\Big] \\ &+ \mathrm{tr}\Big[G(m|n)G^\dagger(n|l)\,(\mathbb{1}-P)(l|m)\,\gamma^5\Gamma\Big]\Big) \end{aligned} \tag{3.44}$$

$$= -2i\mu\,(\mathcal{E}(m,\Gamma) + \mathcal{S}(\vec{m},t,\Gamma)). \tag{3.45}$$

It follows the evaluation of the exact part $\mathcal{E}$ and the deflated part $\mathcal{S}$.

EXACT PART Because being Hermitian, the operator $D^\dagger D$ can be diagonalized by all its N eigenvectors

$$D^\dagger D(m|n) = \sum_{i=1}^{N} \lambda_i\, v_i(m) v_i^\dagger(n) \tag{3.46}$$

$$\Leftrightarrow \left(D^\dagger D\right)^{-1}(m|n) = \sum_{i=1}^{N} \frac{1}{\lambda_i}\, v_i(m) v_i^\dagger(n) \tag{3.47}$$

Therefore,

$$\mathcal{E}(m,\Gamma) = \sum_l \mathrm{tr}\left[\left(D^\dagger D\right)^{-1}(m|l)\, P(l|m)\, \gamma^5\Gamma\right] \tag{3.48}$$

$$= \sum_l \mathrm{tr}\left[\left(\sum_{i=1}^{N} \frac{1}{\lambda_i} v_i(m) v_i^\dagger(l)\right) \cdot \left(\sum_{j=1}^{N_{\mathrm{EV}}} v_j(l) v_j^\dagger(m)\right) \gamma^5\Gamma\right] \tag{3.49}$$

$$= \sum_{j=1}^{N_{\mathrm{EV}}} \frac{1}{\lambda_j} \mathrm{tr}\left[v_j^\dagger(m)\, \gamma^5\Gamma\, v_j(m)\right], \tag{3.50}$$

using the orthogonality of the eigenvectors $\sum_l v_i^\dagger(l) v_j(l) = \delta_{ij}$.

STOCHASTIC PART In the stochastic part, $(\mathbb{1} - P)$ is a projector, therefore $(\mathbb{1} - P)(\mathbb{1} - P) = (\mathbb{1} - P)$. Additionally, $(\mathbb{1} - P)$ commutes with $GG^\dagger$, therefore

$$\mathcal{S}(m,\Gamma) = \sum_{n,l} \mathrm{tr}\left[G(m|n) G^\dagger(n|l)\, (\mathbb{1} - P)(l|m)\, \gamma^5\Gamma\right] \tag{3.51}$$

$$= \sum_{n,l,l'} \mathrm{tr}\left[(\mathbb{1} - P)(m|l')\, G(l'|n) G^\dagger(n|l)\, (\mathbb{1} - P)(l|m)\, \gamma^5\Gamma\right] \tag{3.52}$$

The stochastic sources in (3.6) can be used to insert unity between G and $G^\dagger$

$$\mathcal{S}(m,\Gamma) = \mathcal{O}\left(\frac{1}{\sqrt{R}}\right) + \frac{1}{R} \sum_r \sum_{n,l,l',l''} \cdot \tag{3.53}$$

$$\cdot\, \mathrm{tr}\left[(\mathbb{1} - P)(m|l')\, G(l'|n)\, \eta_r(n) \eta_r^\dagger(l'')\, G^\dagger(l''|l)\, (\mathbb{1} - P)(l|m)\, \gamma^5\Gamma\right]$$

The deflated propagator $(\mathbb{1} - P)G$ can be approximated by performing stochastic inversions [3],

$$D\Psi_r(m) = (\mathbb{1} - P)\eta_r(m). \tag{3.54}$$

The solutions of (3.54),

$$\Psi_r(m) = \sum_{l',n} (\mathbb{1} - P)(m|l') G(l'|n)\, \eta_r(n), \tag{3.55}$$

and its transpose can be used in the stochastic part (3.53) such that it reads

$$\mathcal{S}(m,\Gamma) = \frac{1}{R} \sum_r \mathrm{tr}\left[\Psi_r^\dagger(m)\, \gamma^5\Gamma\, \Psi_r(m)\right]. \tag{3.56}$$

A similar procedure is possible to incorporate exact eigenmode reconstruction with deflation into the general one-end trick.

3 Because the CG algorithm needs a symmetric, positive definite matrix, in practice $D^\dagger D\Psi_r(m) = (\mathbb{1} - P)D^\dagger\eta_r(m)$ is solved.

3.5.3 *Combination with Even-Odd Preconditioning*

It is preferable to combine exact eigenmodes reconstruction with deflation with even-odd preconditioning to enhance the performance. But its application is not straightforward. Although we did not use this combination in the computation of the results shown below in 3.5.5, this subsection explains the difficulty of this combination and how it can be resolved. We concluded that to combine exact eigenmodes reconstruction with deflation with even-odd preconditioning using twisted mass fermions, the eigenvectors of both, the full and the preconditioned matrix have to be computed. Here the number of eigenvectors for each of these two eigenvector sets has to be optimized, such that the larger initialization time is finally compensated by the faster inversion time, as done e.g. in [6].

Combining these two methods means to compute the exact part of the inverse Dirac matrix with the eigenvectors of the full matrix and then evaluate the deflated part by using even-odd preconditioning to solve the linear equations in (3.40). In this second step the even-odd preconditioned equation (3.25),

$$(D_{oo} - D_{oe}G_{ee}D_{eo})\Psi_o = b_o - D_{eo}G_{ee}b_e, \tag{3.57}$$

includes the matrix $D_{oo} - D_{oe}G_{ee}D_{eo}$, which should be deflated in the exact eigenmode reconstruction with deflation method. To deflate it, its eigenvectors are needed. It would be preferable if these eigenvectors do not have to be computed from scratch, but if they could be recomputed from the eigenvectors of the full matrix. This subsection shows these attempts: we did not find a way to recompute the eigenvectors of the even-odd twisted mass Dirac matrix from the eigenvectors of the full twisted mass Dirac matrix. Therefore both eigenvector sets are needed to combine exact deflation with even-odd preconditioning. Of course, an alternative is to not deflate the matrix in (3.57) and only use the eigenvectors of the full matrix to compute some part of the propagator exactly.

The eigenvectors v of the full matrix can be split in an even and an odd part by reordering its entries; then the eigenvalue equation is

$$Dv = \begin{pmatrix} D_{ee} & D_{eo} \\ D_{oe} & D_{oo} \end{pmatrix} \begin{pmatrix} v_e \\ v_o \end{pmatrix} = \lambda \begin{pmatrix} v_e \\ v_o \end{pmatrix}. \tag{3.58}$$

Combining the two equations involved in (3.58) gives

$$(D_{oe}(\lambda - D_{ee})^{-1}D_{eo} + D_{oo})v_o = \lambda v_o. \tag{3.59}$$

On the other hand, the eigenvalue equation of the even-odd preconditioned matrix in (3.57) with eigenvectors u_o and eigenvalues λ_o is

$$(D_{oo} - D_{oe}G_{ee}D_{eo})u_o = \lambda_o u_o. \tag{3.60}$$

In case of the Wilson Dirac matrix, $D_{Woo} = D_{Wee} = \mathbb{1}$, therefore $G_{Wee} = \mathbb{1}$ and equation (3.59) reads

$$(D_{Woe}D_{Weo} - \mathbb{1})v_o^W = \lambda^W(\lambda^W - 2)v_o^W. \tag{3.61}$$

Equation (3.60) gives

$$(\mathbb{1} - D_{Woe}D_{Weo})u_o^W = \lambda_o^W u_o^W. \tag{3.62}$$

Therefore the eigenvectors of the precondition odd matrix are the same as the ones of the odd part of the full matrix, $u_o^W = v_o^W$, for $\lambda_o^W = -\lambda^W(\lambda^W - 2)$. Then theoretically, the inverse Dirac matrix can be estimated using exact eigenmodes reconstruction with deflation: One part is computed exactly with the eigenvectors and eigenvalues of the full matrix $(v_e^W, v_o^W)^T$ and λ^W, while the other part is evaluated with even-odd preconditioning by deflating equation (3.57) with the eigenvectors $u_0^W = v_o^W$ and eigenvalues $-\lambda^W(\lambda^W - 2)$.

In the case of the twisted mass operator, $D_{u/d_{oo}} = D_{u/d_{ee}} = \mathbb{1} \pm i\mu\gamma^5$ and therefore here $G_{u/d_{ee}}$ is not trivial, as it is the case for the Wilson matrix. Hence we did not find a way to relate equations (3.59) and (3.60) for $\lambda^{u/d} \neq 0$, and therefore to find a relation between the eigenvectors of the precondition odd matrix and the eigenvectors of the full matrix. Therefore, for twisted mass fermions both sets of eigenvectors have to be computed separately to combine exact eigenmode reconstruction with deflation and even-odd preconditioning.

3.5.4 *Implementation*

In order to get accurate results of physical quantities in a reasonable amount of time, not only methods need to be improved, but the simulations need to run on supercomputers. These supercomputers consist of thousands of nodes, each containing $\mathcal{O}(10)$ cores and often also graphic units, enabling programs to run most of their computations in parallel. The simulation programs have to be written such that they can parallelize their work efficiently. We used the Quda library, designed for large lattice QCD calculations, and the ARPACK package to compute eigenvectors. It follows a short overview over both Quda, ARPACK and our implementation of the exact eigenmode reconstruction with deflation method. We built and used the packages and written code on the Piz Daint supercomputer of the Swiss national computing center, [33], where up to 2400 hybrid nodes can be used to highly parallelize the computation on graphic cards.

QUDA The Quda library [22, 36] is an open source software package to perform lattice QCD calculations using Graphic Processing Units (GPUs). Computations on GPUs are highly parallelizable and can be generally used since NVIDEA developed Cuda (Compute Unified Device Architecture), a general purpose language to program on GPUs

[79]. Quda combines Cuda code, written in kernels which are run on the GPUs, and C++ code, which is executed on CPUs. It uses the Message Passing Interface (MPI) [46] for inter- and intra-node multi-GPU communication.

The Quda library includes different Dirac operators and several solvers. Additionally there exist kernels and an interface to efficiently perform contractions of two and three-point functions and disconnected quark loops for several 1000 noise vectors and for Fourier transformations.

ARPACK The ARnoldi PACKage (ARPACK) [70] is a numerical software library, written in FORTRAN77, which can compute extreme eigenvalues and eigenvectors of matrices and works very efficiently for large sparse matrices. It uses a modification of the Arnoldi process, the Implicitly Restarted Arnoldi Method (IRAM). This algorithm is an iterative Krylov space method, similar to the Conjugate Gradient method and only needs the application of the matrix to a vector as an input.

The call to this package is implemented inside Quda, such that computation and usage of the eigenvectors can be performed in the same job and therefore time- and storage-intensive writing to and reading from disk is not needed. Additionally, the application of the matrix to a vector can be done on GPUs, which makes the procedure to compute the eigenvectors very fast.

In this work we implemented the full procedure to use exact eigenmodes reconstruction with deflation in Quda and the analysis of the results in R [50]. To this end we wrote two functions, one for the computation of the exact part, the other one for the deflated part. The function to compute the exact part combines the eigenvectors and eigenvalues, which are computed via ARPACK, in such a way that the already implemented one-end trick can be applied, which partly evaluates (3.50).

The function to compute the deflated part uses functions from the CBLAS library [21, 69] for the vector and matrix operations, MPI [47] to distribute the vector and matrix operations over several processes and the already implemented CG solver algorithm and one-end trick. It creates stochastic sources, deflates each source, compare (3.54), uses the conjugate gradient solver to solve the equation in (3.54) and combines the solution with its transpose using the one-end trick function, which partly evaluates (3.56). We implemented the deflation to run on CPUs such that we did not have to write additional Quda-kernels. The already implemented and runtime intensive solver algorithm and the one-end trick are running on GPUs. Because the GPUs have no shared memory, our implemented functions take care of the copying of vectors to and from GPUs.

The analysis contracts the exact part and the deflated parts of each source with a chosen gamma matrix, compare (3.50) and (3.56), averages the deflated part over all sources and combines the resulting exact and deflated part via (3.45) to L^{u-d}. It estimates the stochastic error of this loop by the variance over the single source deflated parts.

3.5.5 *Results*

Having derived exact eigenmodes reconstruction with deflation and combining it with the one-end trick to compute loops, the potential of this new setup has to be tested on a lattice configuration, both for the correctness of the implementation and whether it indeed provides an improvement as anticipated.

We used one twisted mass $16^3 \times 32$ configuration with $N_f = 2$, a twisted mass $\mu = 0.004$, coupling $g = 1.24$, tuned $\kappa_c = 0.160856$, lattice spacing $a = 0.079\,\text{fm}$ and pion mass $m_\pi = 380\,\text{MeV}$ [35]. With and without exact eigenmodes reconstruction with deflation we employed stochastic sources and the one-end trick, but did not use even-odd preconditioning. The method not using exact eigenmode reconstruction with deflation is in the following called *standard method*. We used only results for momentum $\vec{q} = 0$ and computed results of $L(n_t, \Gamma) = \sum_{\vec{n}} L(n, \Gamma)$ with $n = (\vec{n}, n_t)$. This loop is computed for $\mathcal{O}(1000)$ stochastic sources. This section shows only results of the standard one-end-trick with $\Gamma = \mathbb{1}$: $L^{u-d}(n_t) \stackrel{\text{def}}{=} L^{u-d}(n_t, \Gamma = \mathbb{1})$. This section first compares loop results, then error estimates and finally runtimes of the new and the standard method.

THE LOOP We used an already tested implementation of the computation of loops without using exact eigenmode reconstruction, called standard method, to check the implementation. We found that for enough sources both methods converge to the same loop result. The standard method needs more than 10^4 sources to give an approximately stable loop estimate, see Figure 3.1 for timeslice $n_t = 1$, and it is not clear whether even more sources are needed. The loop computed with the exact eigenmode reconstruction with deflation method using 100 eigenvectors gives a stable result for much less sources, here we used maximally 4000 sources. Plots for different timeslices are shown in Figures B.1 and B.2.

THE ERROR BEHAVIOR The error estimate using exact eigenmode reconstruction is proportional to $1/\sqrt{R}$, the expected error behavior when using stochastic sources, see Figure 3.2. Additionally, the error estimate using the new method is around four times smaller than using the standard method. Therefore, the number of sources needed for the new method to give the same error estimate as the standard method should be reduced by a factor of approximately $4^2 = 16$.

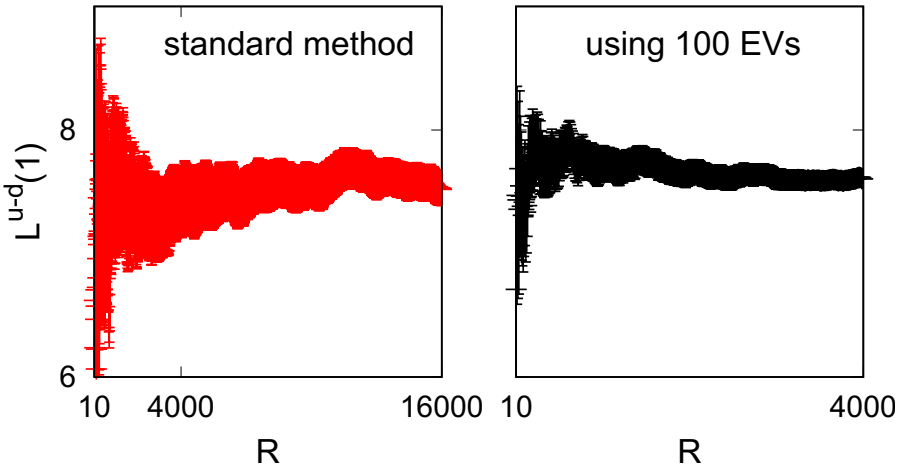


Figure 3.1: For enough sources the exact eigenmode reconstruction with deflation method converges to the same loop value as the standard method, here shown for timeslice $n_t = 1$.

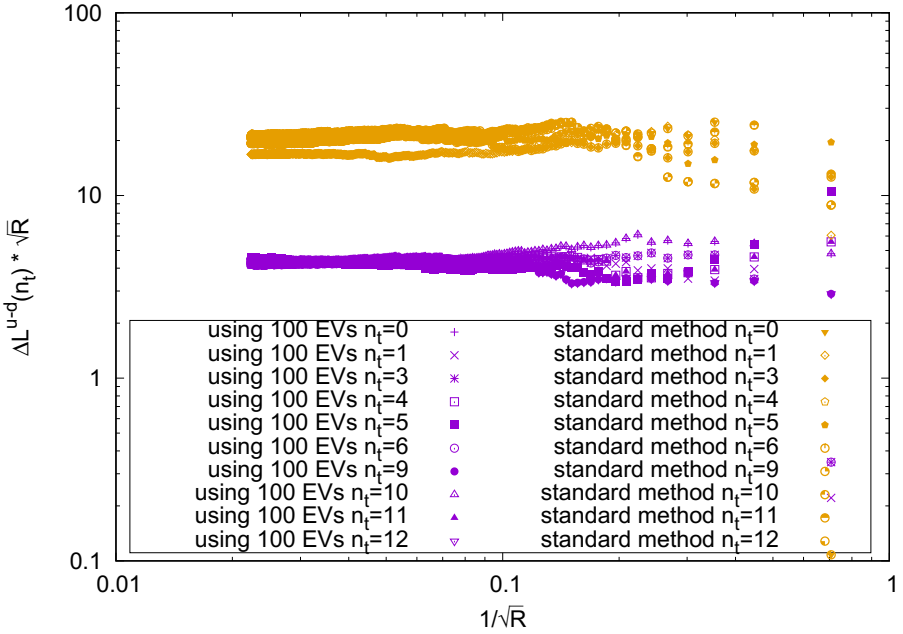


Figure 3.2: The error estimate using exact eigenmode reconstruction with deflation is proportional to $1/\sqrt{R}$ and smaller than the error estimate from the standard method.

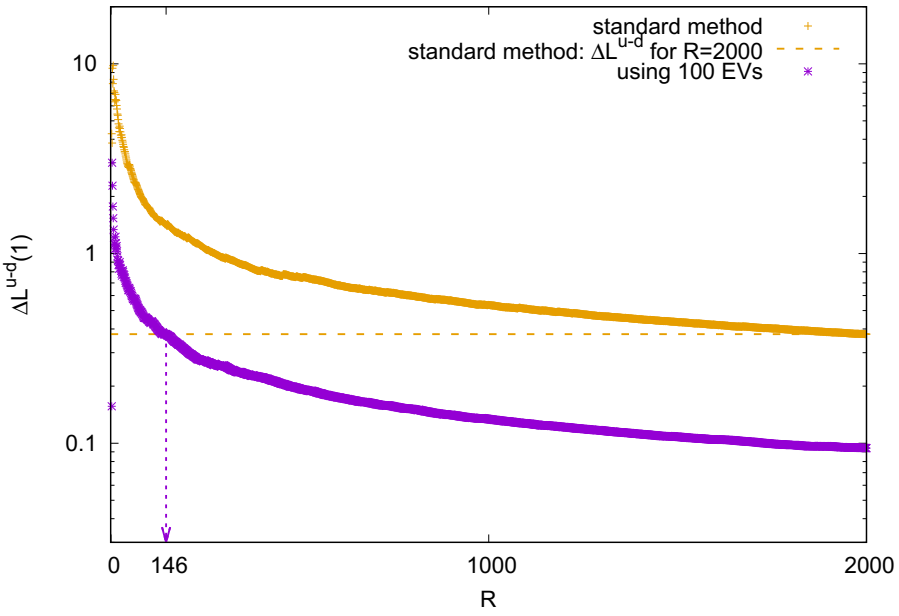


Figure 3.3: The same error estimate that is reached with the standard method for 2000 sources is reached with the new method for only 146 sources.

We checked this explicitly for timeslice one, see Figure 3.3. The error estimate from the standard method using 2000 sources is $\Delta L^{u-d}(1) \approx 0.38$. The exact eigenmode reconstruction with deflation method results in this error estimate for only 146 sources, it needs approximately 13.7 times less sources, which is a bit smaller than the factor 16 we expected before.

Additionally, the statistical uncertainty resulting from the exact eigenmode reconstruction with deflation method depends on the number of eigenvectors included. Figure 3.4 shows that the smallest eigenvalues used in the method have the most effect on the error estimate, as expected from (3.47): Using only five eigenvectors in the exact eigenmode deflation shrinks the error estimate by more than a factor of two, but the error estimate using 250 eigenvectors is only slightly smaller the the one using 100 eigenvectors.

THE RUNTIME The smaller error estimates that can be achieved with the exact eigenmode reconstruction with deflation method in comparison to the standard method are promising, but in the end the runtime to get these error estimates decides whether the method is an improvement over the standard method. For timeslice one we found a runtime gain of approximately 5.5 over the standard method for 100 used eigenvectors. This gain results out of three factors: the less sources needed for the new method, the faster inversions, but also the additional time to compute the eigenvectors, all of them visible in Figure 3.5. First, the number of sources that are needed by the new method, 146, to reach the same error estimate as the standard method

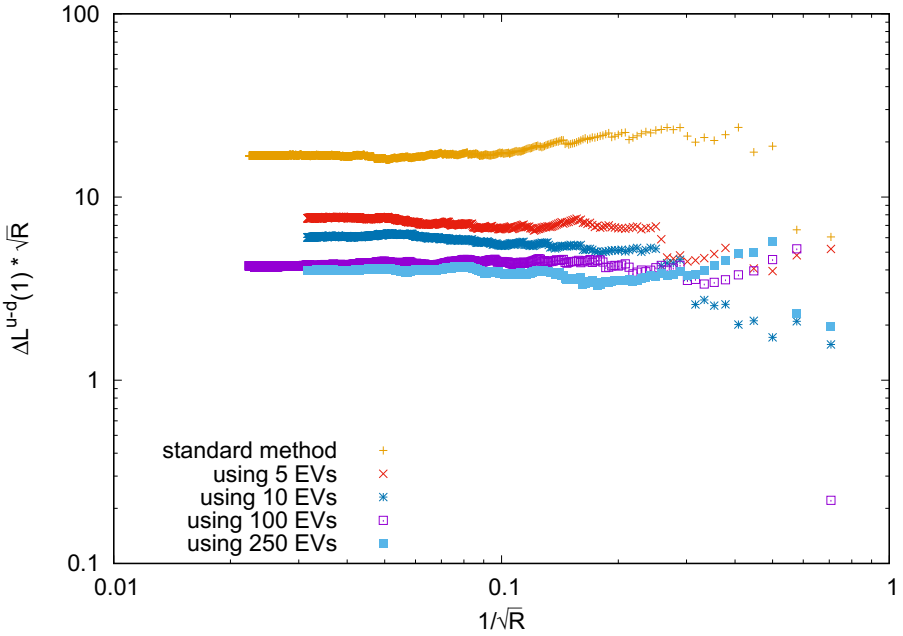


Figure 3.4: The smallest eigenvalues used in the new method shrink the error estimate the most, compared to the standard method.

with 2000 sources, is marked in Figure 3.5. Second, each inversion of the new method is a bit faster than a standard inversion because it inverts a matrix which is deflated with its eigenvectors. This is visible in the smaller slope of the new method runtime in Figure 3.5. And third, the method has to compute the needed eigenvectors at the start of its application, which is runtime intensive: For the full computation with 146 sources the computation of the 100 used eigenvectors, shown as the intercept of the runtime at $R = 0$ in Figure 3.5, takes approximately 70 % of the full runtime. In the end, the exact eigenmode reconstruction with deflation method needs 500 core − h to result in the same error estimate as the standard method using 2731 core − h, which is a runtime gain by a factor of approximately 5.5.

CONCLUDING REMARKS We showed that a runtime gain can be achieved with the exact eigenmode with deflation method on one configuration. Therefore we did not take into account the gauge noise, fluctuating results from different configurations, but which is expected to have a similar effect for our method and the standard one. The eigenvectors used in our method have to be computed for each configuration individually and a more thoroughly check for the optimal number of eigenvectors should be done with more configurations.

Additionally, the method should be tested for simulations extrapolating the continuum limit and for a physical quark mass. Especially when approaching the continuum limit the computation of eigenvectors becomes more time intensive: Extrapolating to the continuum limit means using smaller lattice spacings and for a fixed physical

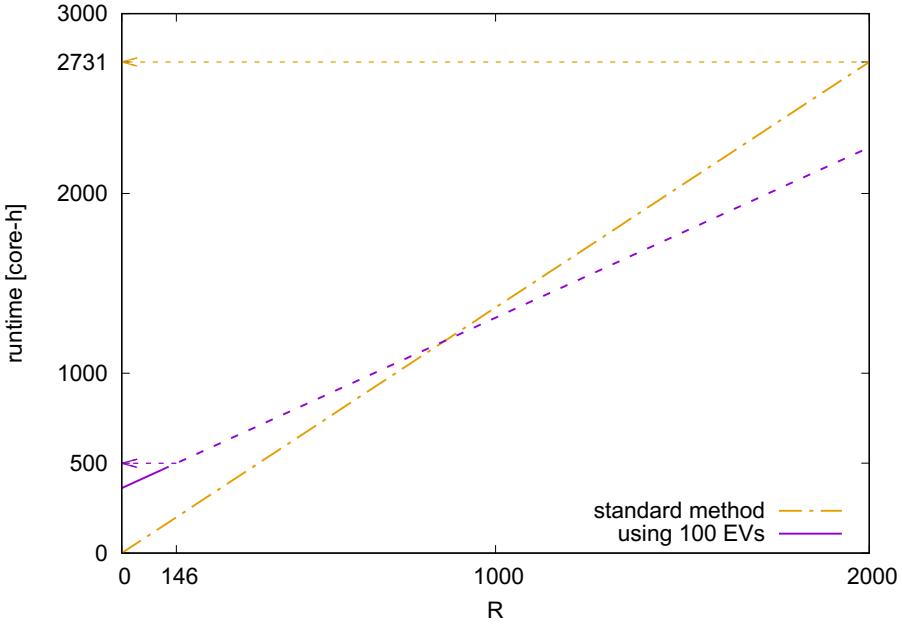


Figure 3.5: Using the exact eigenmode reconstruction with deflation method with 100 eigenvectors results in a runtime gain of approximately a factor six to result in the same error estimate as the standard method.

lattice volume the number of lattice sites has to be increased. The eigenvectors have a number of components, which is proportional to the number of lattice sites and therefore the effort to compute them grows for larger volumes. Additionally, for larger number of lattice sites the eigenvector spectrum becomes denser and the number of eigenvalues below a certain cutoff grows. The advantage of the exact eigenmode reconstruction with deflation method over the standard method has to be checked here.

All these tests were not done in this thesis, mainly because at the time of testing the exact eigenmode reconstruction with deflation method, the Multigrid method was applied to twisted mass fermion lattices and gave orders of magnitude better gains than the here presented method.

3.6 MULTIGRID

Shortly after the here presented tests of the exact eigenmodes reconstruction method, the Multigrid algorithm was applied to twisted mass fermions and resulted in very good performances. This subsection shortly introduces the idea of the Multigrid method and presents a comparison of Multigrid and exact eigenmodes reconstruction with deflation method.

The Multigrid method solves the equation $D\Psi = b$ on a coarser grid and then uses this solution as a preconditioner to the equation on

the original lattice. This preconditioning reduces the runtime of each inversion. There are several different implementations, the Adaptive Aggregation-based Domain Decomposition Multigrid method [55] was already successfully applied to twisted mass fermions [13, 23]. The method has many parameters, e.g. the size of the lattice blocks that are used as lattice sites on a coarser grid, or the number of levels of coarser grids to be used, which all have to be tuned once to the lattice ensemble the method is applied to. Additionally, the described Multigrid method needs some initialization time due to the creation of the coarser grid.

The Multigrid algorithm applied to twisted mass fermions in [13] is implemented in the tmLQCD-framework [63], the implementation of the Multigrid algorithm in QUDA is still experimental. Therefore we were not able to compare both methods directly, using the same architecture. [13] compares the Multigrid to the CG performance on a $48^3 \times 64$ lattice at the physical point with $N_f = 2$. To invert 1000 sources the Multigrid is around 220 times faster than CG method and this performance gain is approximately similar for larger numbers of sources as well. Our performance gain of approximately 5.5 of the exact eigenmode reconstruction with deflation over the standard method with CG using 2000 sources is significant but clearly smaller in comparison to that.

A direct comparison of the Multigrid method described in [13] applied to twisted mass configurations and the initial guess deflation, see section 3.4 was done by the group of Constantia Alexandrou at the Cyprus Institute and the University of Cyprus on a $48^3 \times 96$ lattice at the physical point with $N_f = 2$. It can be assumed that the performance of the initial guess deflation gives a rough approximation of the performance of the exact eigenmode reconstruction with deflation in two of three points: First, the initialization time should be comparable, which is needed to compute the eigenvectors of the Dirac matrix in both cases. Second, the scaling of the runtime dependent on the number of sources can possibly be comparable because in both cases the use of the eigenvectors speeds up the inversions. On the other side, the initial guess deflation does not use the eigenvectors to approximate the final solution directly, as done in the exact part of the exact eigenmode reconstruction and which, as seen in section 3.5.5, is responsible for a lower number of inversions needed to arrive at a given error estimate than the standard method.

Figure 3.6 shows that the Multigrid method in comparison to the standard method using CG and the initial guess deflation gives the smallest runtime for all numbers of sources R. For fewer than 100 sources the runtime of the initial guess deflation is almost constant and large, due to its large initialization time needed to compute the eigenvectors, here 1600. The initialization time of the Multigrid method is much smaller, as well as its runtime per inversion. Estimat-

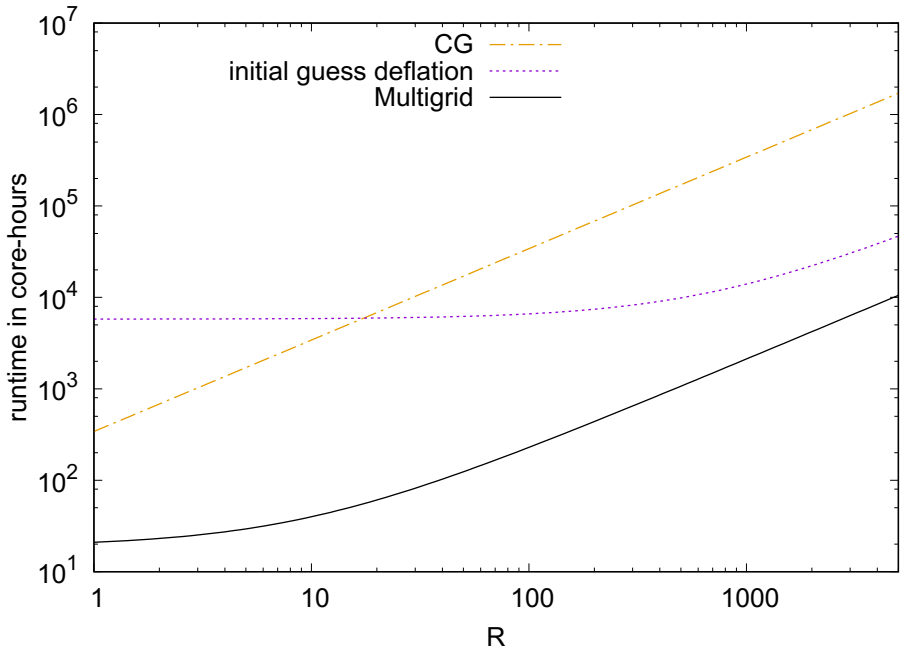


Figure 3.6: The initial guess deflation is not competitive with the Multigrid algorithm due to its very long initialization time and its larger runtime per inversion.

ing the exact eigenmode reconstruction with deflation runtime by the initial guess deflation runtime, it is probable that the exact eigenmode reconstruction with deflation does not outperform the Multigrid algorithm.

The Multigrid method is so powerful because it only depends weakly on the condition number of the Dirac matrix D since the system is preconditioned by the solution of a linear equation on a coarser grid. Especially for simulations at the physical point this is very important: there the twisted mass is very small and increases the condition number of the Dirac matrix. Additionally, in the Multigrid method no time intensive eigenmodes computation is needed.

Part II

GOING BEYOND MARKOV-CHAIN MONTE CARLO

4

MARKOV CHAIN MONTE CARLO INTEGRATION

To compute observables in lattice QCD, high-dimensional integrals have to be solved. The path integral in (2.27) is an integral over all link variables on the lattice. Typical lattices have more than $\mathcal{O}(10^5)$ link and a numerical evaluation of the integral can be demanding. Monte Carlo (MC) methods use random numbers to approximate an integral and have an asymptotic error scaling with the number of sampling points which does not depend on the number of integration variables. This makes them very attractive for high-dimensional integral evaluations. In the method of importance sampling these random numbers are chosen according to the normalized Boltzmann distribution. This leads to a variance reduction compared to ordinary MC sampling. In general any probability distribution which approximates the integrand reasonably well can be used for importance sampling. If importance sampling is done by using a Markov chain, a specific stochastic process, it is called Markov chain Monte Carlo (MCMC) sampling. This chapter describes how these methods work and what their drawbacks are: The chapter introduces how to numerically approximate integrals in general by choosing sampling points and weights of the integrand. It explains MC methods, importance sampling and how to choose sampling points from a probability distribution using Markov chains. Finally, the chapter discusses some issues that arise when using MCMC methods.

This chapter shows that MC methods and more specifically MCMC methods are a very efficient way to evaluate the high-dimensional path integral. On the other hand the application of these methods can lead to some issues: the MC error scaling is quite slow, for a specified error estimate the runtime of the MCMC algorithm grows substantially when approaching the continuum limit and the MCMC method has difficulties in giving reasonable error estimates when applied to complex integrands. Hence these issues have to be investigated and new methods have to be developed to approach these problems.

4.1 APPROXIMATING INTEGRALS

An integrable function $f : \mathbb{R} \to \mathbb{R}$ can be numerically integrated by an n-point *quadrature rule* [38],

$$\int_a^b \mathrm{d}x\, f(x) \approx \sum_{i=1}^{n} w_i f(t_i), \tag{4.1}$$

with *sampling points* $t_1, \ldots, t_n \in \mathbb{R}$ and *weights* $w_1, \ldots, w_n \in \mathbb{R}$.

Common integrals in lattice gauge theory have many more than only one integration variable. The number of integration variables is in the following called d. The integration variables $x_1, \dots, x_d \in D$ are combined in a vector $\boldsymbol{x} = (x_1, \dots, x_d) \in D^d$ for some phase space D. Then an integral of a function $f : D^d \to \mathbb{R}$ or $\mathbb{C}$ is given by

$$I_d(f) = \int_D \mathrm{d}x_1 \dots \int_D \mathrm{d}x_d \, f(x_1, \dots, x_d) = \int_{D^d} \mathrm{d}\boldsymbol{x} \, f(\boldsymbol{x}). \tag{4.2}$$

The number d is also called the dimension of the integral. For the lattice QCD path integral with out-integrated fermions in (2.27) the integration variables are the link variables of the lattice with integration measure given in (2.21), $D = \mathcal{SU}(3)$ and $d = 4V_{\text{lat}}$.

Integrals of the form (4.2) can be evaluated numerically by a *cubature rule* [41],

$$I_d(f) \approx Q_{n,d}(f) = \sum_{i=1}^{n} w_i f(\boldsymbol{t}_i), \tag{4.3}$$

with sampling points $\boldsymbol{t}_1, \dots, \boldsymbol{t}_n \in D^d$ and corresponding weights $w_1, \dots, w_n \in \mathbb{R}$.

One special cubature rule for a high-dimensional integral is the application of quadrature rules to each integral over one variable x_ℓ in (4.2), $\ell \in \{1, \dots, d\}$,

$$\int_D \mathrm{d}x_\ell \, f(x_1, \dots, x_\ell, \dots, x_d) \approx \sum_{j=1}^{m} u_j f(x_1, \dots, x_{\ell-1}, t_j, x_{\ell+1} \dots, x_d).$$

Here $u_j \in \mathbb{R}$ are the weights of this specific quadrature rule and $t_j \in D$ are the sampling points . Applying this quadrature rule to all integrals in (4.2) gives a cubature rule which approximates $I_d(f)$,

$$Q_{m^d,d}(f) = \sum_{j_1=1}^{m} \dots \sum_{j_d=1}^{m} u_{j_1} \cdots u_{j_d} f(t_{j_1}, \dots, t_{j_d}). \tag{4.4}$$

By setting $n \stackrel{\text{def}}{=} m^d$, equation (4.4) can be viewed as a summation of the form (4.3), where each w_i corresponds to some $u_{j_1} \cdots u_{j_d}$ and the vector $\boldsymbol{t}_i$ is given by $(t_{j_1}, \dots, t_{j_d})$. This special cubature rule is called *product rule*. The product rule involves m^d sampling points $\boldsymbol{t}_i$. For lattice QCD applications d is given by $4V_{\text{lat}}$ and is therefore at least $d \sim \mathcal{O}(10^5)$ for typical lattice QCD computations. With this large d already for $m = 2$ the number of terms in (4.4) are astronomically high ($m^d > 2^{10^5}$). Therefore using the product rule is not an option for lattice QCD applications.

In the following only d-dimensional integrals are discussed and $I \stackrel{\text{def}}{=} I_d$ and $Q \stackrel{\text{def}}{=} Q_{n,d}$ is used for brevity.

4.2 ORDINARY MONTE CARLO SAMPLING

The integral $I(f)$ in (4.2) can be estimated by MC sampling. Given a sequence $\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots$ of independent, identically distributed elements in the phase space D^d then also $f(\boldsymbol{t}_1), f(\boldsymbol{t}_2), \ldots$ are independent, identically distributed elements which have variance

$$\mathrm{var}\{f(\boldsymbol{t}_1)\} = \mathrm{var}\{f(\boldsymbol{t}_2)\} = \ldots \stackrel{\mathrm{def}}{=} \mathrm{var}\{f(\boldsymbol{t})\} = I(f^2) - (I(f))^2.$$

Then an estimator for $I(f)$ is the MC cubature rule

$$Q^{\mathrm{MC}}(f) = \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{t}_i), \tag{4.5}$$

a cubature rule in the form of (4.3) with equal weights $w_i = \frac{1}{n}$. The strong law of large numbers states that $Q^{\mathrm{MC}}(f)$ converges almost surely (a.s.), that means with probability one, to $I(f)$ if the number of sampling points n goes to infinity [68],

$$Q^{\mathrm{MC}}(f) \underset{\text{a.s.}}{\overset{n\to\infty}{\longrightarrow}} I(f). \tag{4.6}$$

The set of possible exceptions from this law does not need to be empty but has probability zero. If $\sigma \stackrel{\mathrm{def}}{=} \mathrm{var}\{f(\boldsymbol{t})\}$ is finite, the Central Limit Theorem states that for large n the probability, Pr, that the MC estimate $Q^{\mathrm{MC}}(f)$ lies in between $I(f) - z\frac{\sigma}{\sqrt{n}} \leq Q^{\mathrm{MC}}(f) \leq I(f) + z\frac{\sigma}{\sqrt{n}}$ for $z \in \mathbb{R}$ is given by [54]

$$\lim_{n\to\infty} \Pr[|(Q^{\mathrm{MC}}(f) - I(f))| \leq z\frac{\sigma}{\sqrt{n}}] = \int_{-z}^{z} \mathrm{d}x\, \frac{1}{\sqrt{2\pi}}\, \mathrm{e}^{-\frac{x^2}{2}}. \tag{4.7}$$

From (4.7) it is clear that the MC error scales asymptotically with $1/\sqrt{n}$.

In practice σ is not known. An unbiased estimator $\hat{\sigma}$ with $\hat{\sigma} \underset{\text{a.s.}}{\overset{n\to\infty}{\longrightarrow}} \sigma$ is given by

$$\hat{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^{n} \Big(f(\boldsymbol{t}_i) - Q^{\mathrm{MC}}(f)\Big)^2. \tag{4.8}$$

In MC simulations the MC standard error, $\hat{\sigma}/\sqrt{n}$, is reported.

The MC error scaling with n is independent of the number of integration variables d, which makes it very attractive for applications to high-dimensional integrals. On the other hand, the asymptotic MC error scaling of $\mathcal{O}(1/\sqrt{n})$ is quite slow: to decrease the error estimate by one order of magnitude, the number of samples has to be increased by two orders of magnitude. There are variance reduction methods to reduce the error estimate of the MC method and therefore the number of samples needed for a given error estimate. These reduction methods can reduce the error estimate by an n-independent factor but do not change the overall error scaling.

4.3 IMPORTANCE SAMPLING

One of the variance reduction techniques for MC methods is importance sampling. The integral in (4.2) can be written as [77]

$$I_d(f) = \int_{D^d} \mathrm{d}\boldsymbol{x}\, f(\boldsymbol{x}) = \int_{D^d} \mathrm{d}\boldsymbol{x}\, p(\boldsymbol{x}) \frac{f(\boldsymbol{x})}{p(\boldsymbol{x})}. \tag{4.9}$$

If $p(\boldsymbol{x})$ is positive and normalized, $\int_{D^d} \mathrm{d}\boldsymbol{x}\, p(\boldsymbol{x}) = 1$, it can be interpreted as a probability density function. If random vectors $\boldsymbol{t}_i$ can be generated according to the distribution $\pi(\boldsymbol{x}) = \mathrm{d}\boldsymbol{x} p(\boldsymbol{x})$, then an estimator of the integral is given by[1]

$$Q^{\mathrm{MCMC}}(f) = \frac{1}{n} \sum_{\substack{i=1 \\ \boldsymbol{t}_i \text{ with} \\ \text{probability} \\ \pi(\boldsymbol{t}_i)}}^{n} \frac{f(\boldsymbol{t}_i)}{p(\boldsymbol{t}_i)}. \tag{4.10}$$

The function $p(\boldsymbol{x})$ is chosen such that it approximates $f(\boldsymbol{x})$ reasonably well in shape and such that random numbers can be generated according to the distribution $\pi(\boldsymbol{x})$. Then important contributions $f(\boldsymbol{t})$ to the integral $I(f)$ are considered with a larger probability in $Q^{\mathrm{MCMC}}(f)$ than in $Q^{\mathrm{MC}}(f)$ in (4.5).

The Euclidean path integral in (2.27) strongly suggests to use the normalized Boltzmann-factor in the distribution $\pi(\boldsymbol{x})$. The path integral to compute the expectation value of an observable O has the generic form

$$\langle O \rangle = I(O, \rho) = \frac{\int \mathrm{d}\boldsymbol{x}\, O(\boldsymbol{x}) \rho(\boldsymbol{x})}{\int \mathrm{d}\boldsymbol{x}\, \rho(\boldsymbol{x})}, \tag{4.11}$$

with weight ρ. In lattice QCD the weight is given by $\rho(\boldsymbol{x}) = Z_F(\boldsymbol{x})\, \mathrm{e}^{-S_G^e(\boldsymbol{x})}$ with Z_F defined in (2.26) and S_G^e in (2.8) and the integration variables are the links, see (2.21). With $\pi(\boldsymbol{x}) = \frac{\mathrm{d}\boldsymbol{x}\rho(\boldsymbol{x})}{\int \mathrm{d}\boldsymbol{y}\, \rho(\boldsymbol{y})} \geq 0$ an estimator for $I(O, \rho)$ is given by

$$Q^{\mathrm{MCMC}}(O, \rho) = \frac{1}{n} \sum_{\substack{i=1 \\ \boldsymbol{t}_i \text{ with} \\ \text{probability} \\ \pi(\boldsymbol{t}_i)}}^{n} O(\boldsymbol{t}_i). \tag{4.12}$$

The difficult part is to choose the sampling points $\boldsymbol{t}$ according to the probability distribution $\pi(\boldsymbol{x})$. This can be done by creating a Markov chain.

1 This estimate is called $Q^{MCMC}(f)$, a Markov chain Monte Carlo estimate, because for most applications the vectors $\boldsymbol{t}_i$ are chosen by a Markov chain, discussed in the next section.

4.4 MARKOV CHAINS

A stochastic process in discrete time is a sequences $\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots$ of random elements of a countable or non-countable set, the state space. This sequence is a Markov chain if the conditional distribution of $\boldsymbol{t}_{n+1}$ given $\boldsymbol{t}_1, ..., \boldsymbol{t}_n$ depends on $\boldsymbol{t}_n$ only and the discrete time is called Markov time. A Markov chain is specified by two ingredients, the *initial distribution* of $\boldsymbol{t}_1$ and the *transition probability distributions* which specify the conditional distribution of $\boldsymbol{t}_{n+1}$ given $\boldsymbol{t}_n$.

In the following only stationary Markov chains are considered. A Markov chain is stationary if the distribution of $\boldsymbol{t}_n$ does not depend on n. This implies stationary transition probabilities, that means that the conditional distribution of $\boldsymbol{t}_{n+1}$ given $\boldsymbol{t}_n$ does not depend on n. A probability distribution π is called *invariant* or *equilibrium* for specified transition probabilities if the Markov chain that results from using that distribution as the initial distribution is stationary. To create a Markov chain with random elements $\boldsymbol{t}$ that are distributed according to π one has to find the transition probability P such that π is invariant, $\pi = P\pi$. For more details on Markov chains the reader is referred to [31].

If the Markov chain with transition probability P is chosen such that it leaves the distribution $\pi(\boldsymbol{x}) = \frac{\mathrm{d}\boldsymbol{x}\rho(\boldsymbol{x})}{\int \mathrm{d}\boldsymbol{y}\,\rho(\boldsymbol{y})}$ is invariant, the Markov chain can be used to create the sampling points for the cubature rule $Q^{\mathrm{MCMC}}(O,\rho)$ in (4.12) which is then called MCMC cubature rule. If a stationary Markov chain has a unique invariant distribution, then the strong law of large numbers also applies to Markov chains [31],

$$Q^{\mathrm{MCMC}}(O,\rho) \overset{n\to\infty}{\underset{\text{a.s.}}{\longrightarrow}} I(O,\rho). \tag{4.13}$$

Because this statement involves almost sure convergence, that means the convergence happens with probability one[2], the convergence happens from almost all starting points $\boldsymbol{t}_1$.

Under certain conditions also the Central Limit Theorem holds,

$$\lim_{n\to\infty} \Pr[|(Q^{\mathrm{MCMC}}(O,\rho) - I(O,\rho))| \leq z\frac{\sigma}{\sqrt{n}}] = \int_{-z}^{z} \mathrm{d}x\, \frac{1}{\sqrt{2\pi}}\, \mathrm{e}^{-\frac{x^2}{2}}, \tag{4.14}$$

for $z \in \mathbb{R}$. This is similar to the Central Limit Theorem stated in the MC case (4.7) but in contrast to the MC sampling points, the MCMC sampling points $\boldsymbol{t}_i$ are not independent. Therefore σ^2 is not simply given by $\mathrm{var}\{O(\boldsymbol{t})\}$, but by [31]

$$\sigma^2 = \mathrm{var}\{O(\boldsymbol{t})\} + 2\sum_{k=1}^{\infty} \mathrm{cov}\{O(\boldsymbol{t}_i), O(\boldsymbol{t}_{i+k})\}. \tag{4.15}$$

2 Compare description around (4.6).

The Central Limit Theorem in (4.14) states that the MCMC error scales asymptotically with $1/\sqrt{n}$. The conditions such that the Central Limit Theorem applies are more complicated than in the ordinary MC case in (4.7), where only a finite variance is required. A discussion on these conditions can for example be found in [31], section 7.7.

For stationary Markov chains σ^2 does not depend on i and can be written as

$$\sigma^2 = \sigma_0 + 2\sum_{k=1}^{\infty}\sigma_k, \tag{4.16}$$

with the variance of uncorrelated values $\sigma_0 = \mathrm{var}\{O(\boldsymbol{t})\}$ and their correlation $\sigma_k = \mathrm{cov}\{O(\boldsymbol{t}_i), O(\boldsymbol{t}_{i+k}\}$ which is independent of i. In practice σ is not known. An estimator $\hat{\sigma}$ with $\hat{\sigma} \overset{n\to\infty}{\underset{\text{a.s.}}{\longrightarrow}} \sigma$ is given by [80]

$$\hat{\sigma}^2 = \hat{\sigma}_0 + 2\sum_{k=1}^{W-1}\hat{\sigma}_k, \tag{4.17}$$

with W chosen to balance the systematic error due to the truncation, with the statistical error and

$$\hat{\sigma}_k = \frac{1}{K}\sum_{i=0}^{K}(O(\boldsymbol{t}_i) - Q^{\mathrm{MCMC}}(O,\rho))(O(\boldsymbol{t}_{i+k}) - Q^{\mathrm{MCMC}}(O,\rho)),$$

for a large enough K. In MCMC simulations the MCMC standard error $\hat{\sigma}/\sqrt{n}$ is reported.

The Metropolis algorithm in its original form [73] was the first practically used MCMC algorithm and was developed further by different people, e.g. [60]. For an unnormalized equilibrium probability distribution h, it works as follows:

1. When the current state of the Markov chain is $\boldsymbol{x}$, propose a new state $\boldsymbol{y}$ which has a conditional *proposal* probability density given $\boldsymbol{x}$, denoted by $q(\boldsymbol{x}, \cdot)$.

2. Compute the ratio

$$r(\boldsymbol{x}, \boldsymbol{y}) = \frac{h(\boldsymbol{y})q(\boldsymbol{y}, \boldsymbol{x})}{h(\boldsymbol{x})q(\boldsymbol{x}, \boldsymbol{y})}, \tag{4.18}$$

   using the unnormalized equilibrium distribution h.

3. Accept the proposed move $\boldsymbol{y}$ with *acceptance* probability

$$\min(1, r(\boldsymbol{x}, \boldsymbol{y})). \tag{4.19}$$

The algorithm works only if $h(\boldsymbol{x}) > 0$ and $q(\boldsymbol{x}, \boldsymbol{y}) > 0$ for state $\boldsymbol{x}$. In lattice simulations the Metropolis algorithm is often used for pure gauge theory with $h(\boldsymbol{x}) = \mathrm{e}^{-S_{\mathrm{G}}^{\mathrm{e}}(\boldsymbol{x})}$ and a proposal probability

which is symmetric, $q(\boldsymbol{x}, \boldsymbol{y}) = q(\boldsymbol{y}, \boldsymbol{x})$, and local. Local means that for a Markov state $\boldsymbol{x} = (x_1, \ldots, x_d)$ with $d = 4V_{\text{lat}}$ a new Markov state $\boldsymbol{y}$ is proposed by changing only one variable x_i to y_i, $\boldsymbol{y} = (x_1, \ldots x_{i-1}, y_i, x_{i+1}, \ldots, x_d)$. Because this local change in the variable $\boldsymbol{x}$ results in a local change of the action $S^e_G(\boldsymbol{x})$, $r(\boldsymbol{x}, \boldsymbol{y})$ is only dependent on the local action change in the vicinity of lattice site i. Updating the full lattice, that is all entries in $\boldsymbol{x}$, needs d Metropolis-steps. Therefore the cost of the local Metropolis algorithm scales linearly with the lattice volume V_{lat}.

Using non-local changes possibly of all d variables in the proposal probability would in general result in large changes in the action S^e_G, the acceptance rate for configurations where the new action $S^e_G(\boldsymbol{y})$ has increased in comparison to $S^e_G(\boldsymbol{x})$ would be very small and therefore the system would move only slowly through the phase space D^d. The algorithm becomes also very slow when local changes in the phase space result in non-local changes of $h(\boldsymbol{x})$, e.g. in full QCD for $h(\boldsymbol{x}) = Z_F(\boldsymbol{x})\, \mathrm{e}^{-S^e_G(\boldsymbol{x})}$. Here $Z_F(\boldsymbol{x})$ is dependent on the determinant of the Dirac matrix, which is a non-local quantity. An MCMC algorithm which can efficiently be applied to actions including fermions is the Hybrid Monte Carlo method [43].

4.5 ISSUES OF MARKOV CHAIN MONTE CARLO METHODS

MCMC is used in many physics applications to numerically evaluate the involved integrals. It is well suited for high-dimensional integrations and there are efficient methods to apply it to the lattice QCD path integral. But the method has also some issues: Despite the already mentioned slow error scaling of $\mathcal{O}(1/\sqrt{n})$ with the number of sampling points n, there are some situations where the method results in very large MCMC error estimates which can make it impossible to get any significant outcome. On the one hand this is happening when approaching the continuum limit of the discretized Euclidean path integral, where the sampling points are highly correlated. On the other hand the application of MCMC is difficult for complex integrands. This section describes both issues in more detail.

This section shows that the MCMC algorithms can be improved with some techniques to reduce the described issues but in order to overcome them completely, different alternative methods have to be developed. Some of these alternative methods are presented in the next chapters.

4.5.1 *Autocorrelations*

Sampling points created through a Markov chain are correlated. Each sampling point is created out of the previous one and therefore they

are not independent. This is reflected by the covariance terms σ_k in (4.16).

The integrated autocorrelation time is given by

$$\tau_{\text{int}} \overset{\text{def}}{=} \frac{\sigma^2}{\sigma_0} = 1 + 2\sum_{k=1}^{\infty} \frac{\sigma_k}{\sigma_0}. \tag{4.20}$$

Therefore σ^2 is given by the variance of uncorrelated data σ_0 times the autocorrelation time τ_{int}. The the number of samples n to reach a specific MCMC error estimate $\hat{\sigma}/\sqrt{n}$ is proportional to τ_{int}.

The autocorrelation time depends on the specific algorithm used to create the Markov chain and on the observable O. In the continuum limit the longest correlation ξ of the system diverges. In QCD ξ is given by the inverse of the pion mass. The autocorrelation time depends on this correlation length via [56],

$$\tau_{\text{int}} \propto \xi^z. \tag{4.21}$$

with dynamical critical exponent $z \geq 0$, which depends on the updating algorithm. Therefore τ_{int} and also the MCMC standard error estimate grow very large when approaching the continuum limit. This behavior is called critical slowing-down.

There are algorithms which have very small exponents z, such that the critical slowing-down is not problematic. One example is the Cluster algorithm, [75, 83], an MCMC algorithm which creates clusters of lattice points with similar characteristics and updates all points in this cluster together. Unfortunately, this algorithm is not applicable to all models, especially not to gauge theories.

4.5.2 *The sign-problem*

If the weight function $\rho(\boldsymbol{x})$ in the path integral in (4.11) is non-positive, it is impossible to interpret it as a probability density function for importance sampling.

This is the case for an important set of problems, namely for QCD systems with a non-vanishing quark chemical potential[3] μ. In contrast to systems in the QCD vacuum with zero quark chemical potential, systems with a non-zero background density of quarks have $\mu > 0$. One possibility to include the chemical potential in the lattice Dirac operator is the redefinition of the temporal derivatives in (2.11) [56]

$$\begin{aligned} \nabla_0 \Psi(n) &= \frac{1}{a}(\mathrm{e}^{a\mu} U_0(n)\Psi(n+\hat{0}) - \Psi(n)), \\ \nabla_0^* \Psi(n) &= \frac{1}{a}(\Psi(n) - \mathrm{e}^{-a\mu} U_{-0}(n)^\dagger \Psi(n-\hat{0})). \end{aligned} \tag{4.22}$$

Therefore the term in the Dirac operator in (2.10) with γ_0 (and one part of the Wilson term) depends on the chemical potential. This effects the determinant of the Dirac matrix, it can be complex because

3 In the following, μ only denotes the chemical potential and not the twisted mass.

the Dirac matrix is no longer γ_5-Hermitian, $\gamma_5 D_W \gamma_5 = D_W^\dagger$. Now it is $\gamma_5 D_W(\mu)\gamma_5 = D_W^\dagger(-\mu)$ and therefore $\det[D_W(\mu)] = \det[D_W(-\mu)]^*$, which means that the determinant can be complex for $\mu \neq 0$. This determinant occurs in the fermionic partition function, e.g. $Z_F(\boldsymbol{x}) = \det(D_W(\boldsymbol{x}))$ for a single fermion flavor (compare the description after (2.30)), which is part of the weight function $\rho(\boldsymbol{x}) = Z_F(\boldsymbol{x})\, \mathrm{e}^{-S_G^e(\boldsymbol{x})}$ in (2.27). The complex weight function cannot be used to create a Markov chain because it is non-positive and results in oscillatory integrands in the path integral (4.11).

Using ordinary MC sampling without any importance sampling described in section (4.2) instead of MCMC for oscillatory integrands with complex ρ is possible but very inefficient. Evaluating this integral numerically shows a *sign-problem*: Positive and negative contributions that cancel each other in an exact computation are not chosen symmetrically in the numerical evaluation of the integral and only cancel for an infinite number of sampling points. Therefore these types of integrals result normally in large error estimates. The sign-problem scales with the lattice volume because ρ depends on the action which involves a sum over all lattice sites.

If ρ is complex, the sign-problem can occur and ρ cannot be used to create points in a Markov chain. Despite the alternative to use ordinary MC without any importance sampling, MCMC methods can be applied anyway if the weight function is redefined: By decomposing $\rho = \omega\varrho$ into its real modulus $\varrho \in \mathbb{R}$ and complex phase factor $\omega(\boldsymbol{x}) = \mathrm{e}^{i\theta(\boldsymbol{x})} \in \mathbb{C}$ for $\theta(\boldsymbol{x}) \in \mathbb{R}$, only ϱ can be used as the new weight function and the complex phase factor is subjoined to the observable O,

$$\begin{aligned}\langle O \rangle = I(O, \omega\varrho) &= \frac{\int \mathrm{d}\boldsymbol{x}\, O(\boldsymbol{x})\omega(\boldsymbol{x})\varrho(\boldsymbol{x})}{\int \mathrm{d}\boldsymbol{x}\, \omega(\boldsymbol{x})\varrho(\boldsymbol{x})} \\ &= \frac{\int \mathrm{d}\boldsymbol{x}\, \left(O(\boldsymbol{x})\omega(\boldsymbol{x})\right)\varrho(\boldsymbol{x})}{\int \mathrm{d}\boldsymbol{x}\, \varrho(\boldsymbol{x})} \cdot \frac{\int \mathrm{d}\boldsymbol{x}\, \rho(\boldsymbol{x})}{\int \mathrm{d}\boldsymbol{x}\, \omega(\boldsymbol{x})\varrho(\boldsymbol{x})} \\ &= I(O\omega, \varrho) \cdot \frac{1}{I(\omega, \varrho)}. \end{aligned} \tag{4.23}$$

An estimator of this expectation value can be computed via MCMC using the real probability distribution $\frac{\mathrm{d}\boldsymbol{x}\varrho(\boldsymbol{x})}{\int \mathrm{d}\boldsymbol{y}\, \varrho(\boldsymbol{y})}$ to estimate $I(O\omega, \varrho)$ and $I(\omega, \varrho)$ separately. Still, the complex phase factor is highly fluctuating, especially for larger θ and therefore with this reweighting technique it is still challenging to compute significant results for large θ values.

5

RECURSIVE NUMERICAL INTEGRATION

MCMC is the method of choice for most lattice QCD simulations. Typical lattice QCD simulations have to approximate integrals with at least 10^5 integration variables, but the MCMC error scaling does not depend on this number of variables, compare chapter 4. Chapter 4 also demonstrated that using MCMC can lead to some issues. If the integrand is real, there are two main ones: First, the configurations that are created by a Markov chain are correlated. This correlation diverges near the critical point of the model and therefore leads to large error estimates of observables when approaching the continuum limit. Second, the slow error scaling of MCMC methods results in a large effort to reduce the error estimate of an observable. New methods are needed to attack these problems. But these methods still need to be competitive for large numbers of variables. The recursive numerical integration (RNI) described in [58, 61] is such a method. It exploits the local structure of typical integrands of lattice path integrals to simplify the corresponding integrals, such that each full integral can be approximated by recursively applied quadrature rules. For these quadrature rules we used the efficient Gaussian quadrature rule, applied the method to a one-dimensional $O(2)$-model and compared its efficiency with MCMC methods. The method as presented here is only applicable to one-dimensional models. Results are published in [14] and [16].

This chapter first analyzes the structure of typical integrands, then describes the RNI method, introduces the $O(2)$ model of the topological oscillator and shows numerical results of a topological oscillator observable, computed with the RNI method.

We found that the method gives accurate results near the critical point, computations of errors are possible in a region where the error scales exponentially and the method needs less runtime than an optimal MCMC algorithm to reach a specified error estimate. Therefore the recursive numerical integration method is a very promising alternative to MCMC, at least when applied to a one-dimensional problem. A generalization to $\mathcal{U}(N)$ and $\mathcal{SU}(N)$ variables with appropriate quadrature rules is possible. In the future this method should be generalized such that it is also efficiently applicable to models with higher dimensions. It is not clear yet, whether the method can finally be applied to QCD.

5.1 STRUCTURE OF INTEGRANDS

RNI uses the structure of an integrand to simplify the corresponding integral evaluation. In lattice field theory, expectation values of observables are computed by the path integral, compare (4.11). The involved integrals in numerator and denominator have integrands that include the weight function ρ and an observable O. RNI uses the fact that most physical models have only local interactions. These local interactions are simulated with low-order couplings and in most simulations only next-neighbor couplings are considered. Then it can be possible to decompose ρ and O into factors which are only dependent on nearest-neighbor variables. For example, in pure gauge theory with configurations $\boldsymbol{U}$ consisting of entries $U_\mu(n)$ with $\mu \in \{1,2,3,4\}$ and $n \in \Lambda$, the weight function $\rho(\boldsymbol{U}) = \mathrm{e}^{-S_{\mathrm{G}}^{e}(\boldsymbol{U})}$ can be split by using the definition of the action in (2.27),

$$\mathrm{e}^{-S_{\mathrm{G}}^{e}(\boldsymbol{U})} = \prod_{n\in\Lambda} \prod_{\substack{\mu,\nu=1 \\ \mu\neq\nu}}^{4} \mathrm{e}^{-\frac{a^4}{2g^2}\Re(\mathrm{Tr}[1-U_{\mu\nu}(n)])}. \tag{5.1}$$

Recall that $U_{\mu\nu}(n) = U_\mu(n)U_\nu(n+\hat{\mu})U_\mu(n+\hat{\nu})^\dagger U_\nu(n)^\dagger$ (2.6) and define

$$\rho(A,B,C,D) \overset{\mathrm{def}}{=} \mathrm{e}^{-\frac{a^4}{2g^2}\Re(\mathrm{Tr}[1-ABCD])}. \tag{5.2}$$

Then the weight function can be written as

$$\mathrm{e}^{-S_{\mathrm{G}}^{e}(\boldsymbol{U})} = \prod_{n\in\Lambda} \prod_{\substack{\mu,\nu=1 \\ \mu\neq\nu}}^{4} \rho(U_\mu(n), U_\nu(n+\hat{\mu}), U_\mu(n+\hat{\nu}), U_\nu(n)). \tag{5.3}$$

Each ρ-term is only dependent on four link variables, which are all part of the smallest possible, non-trivial, closed loop. This factorization of the weight can be used to simplify the evaluation of the integral over this weight function $\int \mathrm{d}\boldsymbol{U}\ \mathrm{e}^{-S_{\mathrm{G}}^{e}(\boldsymbol{U})}$.

If this factorization of an integrand is possible, it can be used to simplify the integral evaluation. We tested this method with a simpler model, where both numerator and denominator integral of the expectation value can be factorized. We considered a *one-dimensional* quantum-mechanical system with d lattice sites, periodic boundary conditions, $\mathcal{U}(1)$ variables and only two next-neighbor couplings per site. In contrast to QCD, we used $\mathcal{U}(1)$ variables that are located at the lattice sites. For example, the variable $U_j = \mathrm{e}^{i\alpha_j}$ is located at lattice site j with $\alpha_j \in [0,\pi)$. In the following, only the α_j variables are shown and $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_d)$. The weight function of the model is given by $\rho(\boldsymbol{\alpha})$. The expectation value of an observable O is then given by

$$\langle O \rangle = \frac{\int \mathrm{d}\boldsymbol{\alpha}\, O(\boldsymbol{\alpha})\rho(\boldsymbol{\alpha})}{\int \mathrm{d}\boldsymbol{\alpha}\, \rho(\boldsymbol{\alpha})}. \tag{5.4}$$

We used a weight function form that is similar to the form in (5.3) but applicable to the described one-dimensional $\mathcal{U}(1)$ model,

$$\rho(\boldsymbol{\alpha}) = \prod_{i=1}^{d} \rho_i(\alpha_{i+1}, \alpha_i). \tag{5.5}$$

This function consists of local weight functions ρ_i that are only dependent on two variables α_{i+1} and α_i.

We considered the observable $\sum_{i=1}^{d} O_i(\alpha_{i+1}, \alpha_i)$ and correlations of it, therefore the general observable function is given by

$$O(\boldsymbol{\alpha}) = \left(\sum_{i=1}^{d} O_i(\alpha_{i+1}, \alpha_i) \right)^k, \tag{5.6}$$

where $k \in \mathbb{N}, k \leq d$ defines the number of correlations. The full integrand of the numerator in (5.4), $f(\boldsymbol{\alpha}) = O(\boldsymbol{\alpha}) \cdot \rho(\boldsymbol{\alpha})$, is then given by

$$f(\boldsymbol{\alpha}) = \sum_{i_1=1}^{d} \cdots \sum_{i_k=1}^{d} \left(\prod_{\ell=1}^{k} O_{i_\ell}(\alpha_{i_\ell+1}, \alpha_{i_\ell}) \right) \left(\prod_{j=1}^{d} \rho_j(\alpha_{j+1}, \alpha_j) \right) \tag{5.7}$$

$$= \sum_{i_1=1}^{d} \cdots \sum_{i_k=1}^{d} \prod_{j=1}^{d} f_j^{i_1 \ldots i_k}(\alpha_{j+1}, \alpha_j). \tag{5.8}$$

Here, we have collected all O_j and ρ_j terms in $f_j^{i_1 \ldots i_k}$. Consider for example $k = 5$ and $d = 20$. Then e.g. for the multi-index $(i_1, \ldots, i_5) = (1, 1, 2, 9, 1)$ there are factors

$$\begin{aligned} f_1^{11291} &= O_1^3 \rho_1, \quad f_4^{11291} = \rho_4, \quad f_7^{11291} = \rho_7, \\ f_2^{11291} &= O_2 \rho_2, \quad f_5^{11291} = \rho_5, \quad f_8^{11291} = \rho_8, \\ f_3^{11291} &= \rho_3, \quad f_6^{11291} = \rho_6, \quad f_9^{11291} = O_9 \rho_9, \end{aligned} \tag{5.9}$$

and all other $f_j^{11291} = \rho_j$ for $j > 9$. Because integration is linear, the following considerations use a fixed multi-index $(i_1, \ldots, i_k)$ and consequently omit it if not stated otherwise. Then both integrands in (5.4), $f(\boldsymbol{\alpha}) = O(\boldsymbol{\alpha}) \cdot \rho(\boldsymbol{\alpha})$ in the numerator and $f(\boldsymbol{\alpha}) = \rho(\boldsymbol{\alpha})$ in the denominator, have the same structure,

$$f(\boldsymbol{\alpha}) = \prod_{i=1}^{d} f_i(\alpha_{i+1}, \alpha_i). \tag{5.10}$$

5.2 RECURSIVE NUMERICAL INTEGRATION

RNI uses the structure of the integrand f in (5.10) to simplify the computation of the integral

$$I(f) = \int_{[0,2\pi)^d} \mathrm{d}\boldsymbol{\alpha}\, f(\boldsymbol{\alpha}). \tag{5.11}$$

Here f can be $O \cdot \rho$ or ρ to compute numerator or denominator of the expectation value $\langle O \rangle$ in (5.4), respectively. We used the integrand structure to split the integral into d nested one-variable integrals. This means that the cubature rule of the full integral consists of d recursive one-variable quadrature rules, which can be solved very efficiently. This section explains the three main ingredients to an estimate of (5.4) with RNI: the cubature rule for an integral with the structure (5.10), the specific cubature rules for numerator and denominator of (5.4) and the efficient Gaussian quadrature rule that can be used inside the cubature rule.

CREATING THE CUBATURE RULE The integral of (5.10) can be rewritten with recursive integration as described in [58, 61]. Because of next-neighbor couplings each lattice point α_i appears only twice in $f(\boldsymbol{\alpha})$, in f_i and f_{i-1} and therefore the integral can be written as d nested one-variable integrals I_i,

$$\begin{aligned} I(f) &= \int_D \mathrm{d}\alpha_1 \dots \int_D \mathrm{d}\alpha_d \prod_{i=1}^{d} f_i(\alpha_i, \alpha_{i+1}) \qquad (5.12) \\ &= \underbrace{\int_D \mathrm{d}\alpha_1 \dots \underbrace{\left(\int_D \mathrm{d}\alpha_{d-1}\, f_{d-2}(\alpha_{d-2}, \alpha_{d-1}) \cdot \underbrace{\left(\int_D \mathrm{d}\alpha_d\, f_{d-1}(\alpha_{d-1}, \alpha_d) \cdot f_d(\alpha_d, \alpha_{d+1}) \right)}_{I_d} \right)}_{I_{d-1}}}_{I_1}. \end{aligned}$$

This full integral can be computed recursively: I_d integrates out α_d first, then I_{d-1} integrates out α_{d-1} and so on until finally $I_1 = I(f)$ integrates out α_1. These integrations are approximated by using an n-point quadrature rule for each integral.

To avoid under- and overflow of the single quadrature rule results we actually used quadrature rules to approximate $I_i^* = \frac{1}{c_i} I_i$ with $c_i > 0$ chosen adaptively. Then the final integral is computed via $I = \left(\prod_{i=1}^{d} c_i \right) I^*$. For brevity the method is described in the following without this trick.

A quadrature rule of the form[1] (4.1) is used for each one-variable integral in (5.12). The integrand of I_d depends on three variables α_{d-1}, α_d and α_{d+1}. The variable α_d is integrated out, therefore the quadrature rule $Q_d(f_{d-1} \cdot f_d) \overset{\text{def}}{=} Q_d$ of I_d depends on two variables,

$$Q_d[\alpha_{d-1}, \alpha_{d+1}] = \sum_{r=1}^{n} w^r f_{d-1}(\alpha_{d-1}, t^r) \cdot f_d(t^r, \alpha_{d+1}). \qquad (5.13)$$

For better readability the quadrature indices of the sampling points t^r and weights w^r are written as superscripts. There are other integrals which integrate over α_{d-1} and α_{d+1} and each integral is approximated

1 In this chapter the indices i of quadrature rule Q_i and integral I_i should not be confused with the indices d or n of I_d and $Q_{n,d}$ in section 4.1, which are not shown here.

by a similar quadrature rule than (5.13). These quadrature rules include sums over sampling points t^m for α_{d-1} and t^k for α_{d+1} with $m, k \in \{1, \ldots, r\}$. We used the same set of sampling points for each quadrature rule. Therefore the quadrature rule for I_d can be written as

$$Q_d[t^m, t^k] = \sum_{r=1}^{n} w^r f_{d-1}(t^m, t^r) \cdot f_d(t^r, t^k) \stackrel{\text{def}}{=} Q_d^{m,k}. \tag{5.14}$$

Each $f_i(t^m, t^k)$ can be interpreted as a matrix entry of a matrix M_i, and with $\text{diag}(w) \stackrel{\text{def}}{=} \text{diag}(w^1, w^2, \ldots, w^n)$ equation (5.14) can be written in matrix notation,

$$Q_d = M_{d-1} \cdot \text{diag}(w) \cdot M_d, \tag{5.15}$$

where all matrices have $n \times n$ entries. The next integral to be approximated in the recursive approach of (5.12) is

$$I_{d-1}(\alpha_{d-2}, \alpha_{d+1}) = \int_D \mathrm{d}\alpha_{d-1}\, f_{d-2}(\alpha_{d-2}, \alpha_{d-1}) \cdot I_d(\alpha_{d-1}, \alpha_{d+1}). \tag{5.16}$$

Using a similar quadrature rule as in (5.14) it is

$$\begin{aligned} Q_{d-1} &= M_{d-2} \cdot \text{diag}(w) \cdot Q_d && (5.17)\\ &= M_{d-2} \cdot \text{diag}(w) \cdot M_{d-1} \cdot \text{diag}(w) \cdot M_d, && (5.18) \end{aligned}$$

where (5.15) is inserted. The quadrature rule is used for all other integrals in (5.12) recursively until $I_2 \approx Q_2 = \left(\prod_{i=1}^{d-1} M_i \cdot \text{diag}(w)\right) M_d$. Due to periodic boundary conditions with $\alpha_{d+1} = \alpha_1$ the last integral is then

$$I = I_1 = \int_D \mathrm{d}\alpha_1\, I_2(\alpha_1, \alpha_1), \tag{5.19}$$

$$Q = Q_1 = \sum_{r=1}^{n} w^r I_2(t^r, t^r) = \text{tr}\left[\prod_{i=1}^{d} (M_i \cdot \text{diag}(w))\right]. \tag{5.20}$$

THE NUMERATOR AND DENOMINATOR CUBATURE RULE The derived cubature rule in (5.20) is applicable to integrands of the form (5.10). This rule has to be applied to numerator $I(\rho) = \int \mathrm{d}\boldsymbol{\alpha}\, \rho(\boldsymbol{\alpha})$ and denominator $I(O\rho) = \int \mathrm{d}\boldsymbol{\alpha}\, O(\boldsymbol{\alpha})\rho(\boldsymbol{\alpha})$ of (5.4) separately. Therefore, the specific integrand form of the numerator in (5.8) with summation over the multi-index has to be taken into account. Additionally, the cubature rules can be simplified when assuming isotropy for the local weights in (5.5), $\rho_1 = \rho_2 = \ldots = \rho_d \stackrel{\text{def}}{=} \rho$.

Then, for the denominator $f(\boldsymbol{\alpha}) = \rho(\boldsymbol{\alpha})$ it is $M_1 = \ldots = M_d \stackrel{\text{def}}{=} M$. This gives

$$Q(\rho) = \text{tr}[(M\,\text{diag}(w))^d] = \text{tr}[(\text{diag}(\sqrt{w})\; M\,\text{diag}(\sqrt{w}))^d]. \tag{5.21}$$

A real symmetric matrix can be diagonalized, resulting in a diagonal matrix D which includes eigenvalues λ_i of $M\text{diag}(w)$ and it is

$$Q(\rho) = \text{tr}[D^d] = \sum_i \lambda_i^d. \tag{5.22}$$

Either (5.21) or (5.22) can be used to approximate the denominator integral in (5.4). For both it is valid that the smaller the matrix M, the faster Q is computed. Because M has $n \times n$ entries, a quadrature rule is needed that uses few sampling points n to result in a good approximation of each integral I_i.

The integrand of the numerator is given by $f(\boldsymbol{\alpha}) = O(\boldsymbol{\alpha}) \cdot \rho(\boldsymbol{\alpha})$ and it's form is given in (5.8). Because of the sum over the multi-index $(i_1, \ldots, i_k)$ in (5.8) the matrices M_i have a multi-index. From (5.7) it is clear that the $f_j^{i_1 \ldots i_k}$ are in general not equal to ρ_j, but also include factors of the observable O. Therefore the $M_j^{i_1,\ldots,i_k}$ can all be different from each other. The quadrature rule of the numerator is given by

$$Q(O\rho) = \sum_{i_1=1}^{d} \ldots \sum_{i_k=1}^{d} \text{tr}\left[\prod_{j=0}^{d-1} (M_j^{i_1,\ldots,i_k} \cdot \text{diag}(w))\right]. \tag{5.23}$$

Here, the trace of a product of d matrices $\left(M_j^{i_1,\ldots,i_k}\,\text{diag}(w)\right)$ has to be computed for each multi-index. Note that in practice $k \ll d$ and therefore it happens that adjacent M_i coincide. This reduces some of the products to matrix powers which can be computed slightly faster.

THE GAUSSIAN QUADRATURE RULE The main question becomes how to choose the sampling points and weights which determine the matrices M_i. More specifically, our goal is to produce the best error estimate for the quadrature rules Q_i with as few sampling points as possible, because the number of sampling points affects the number of entries of the M_i quadratically. We used the Gaussian n-point quadrature rule, see [81]. It approximates integrals of the form $\int_a^b \mathrm{d}x\, W(x) f(x)$ with a function $W(x)$ that is positive and continuous on the interval $[a, b]$. The quadrature rule is given by

$$\int_a^b \mathrm{d}x\, W(x) f(x) \approx \sum_{r=1}^{n} w_r f(t_r) = Q. \tag{5.24}$$

Note that in our case, $W(x) = 1$. The rule is exact when $f(x)$ is a polynomial of degree $2n - 1$. This is archived by choosing the sampling points t_r and the weights w_r through orthogonal polynomials $p_n(x)$, associated with the weight function $W(x)$. For $W(x) = 1$ these are the Legendre polynomials. The t_r are the roots of the nth polynomials $p_n(x)$ and the w_r are chosen by the condition

$$\sum_{r=1}^{n} p_k(t_r) w_r = \begin{cases} \langle p_0 | p_0 \rangle, & k = 0 \\ 0, & k \in \{1, \ldots, n-1\} \end{cases}, \tag{5.25}$$

see [81]. If f is not a polynomial but continuously differentiable on $[a,b]$, the quadrature error of (5.24) is given by [81]

$$\sigma \stackrel{\text{def}}{=} I - Q = \frac{f^{(2n)}(\xi)}{(2n)!} \langle p_n | p_n \rangle \,, \tag{5.26}$$

for some $\xi \in [a,b]$. The error scales asymptotically (for large n) as[2] $\mathcal{O}\left(\frac{1}{(2n)!}\right)$. The Stirling formula ($n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$ asymptotically) approximates the factorial to give

$$\sigma \sim \mathcal{O}\left(\exp(-2n\ln n)\frac{1}{\sqrt{n}}\right) \tag{5.27}$$

asymptotically. One drawback using Gaussian integration is that all sampling points change when using the $(n+1)$-point quadrature rule instead of the n-point quadrature rule in (5.24). This is very different from MCMC integration, where one additional sampling point from the Markov chain is added to the already existing sampling points of the n-point quadrature rule to form a $(n+1)$-point rule.

5.3 THE TOPOLOGICAL OSCILLATOR

We applied the RNI to the topological oscillator, described for example in [25]. We chose this model to test RNI on a simple one-dimensional model, but one that also has non-trivial characteristics, such as a topological charge. The topological oscillator describes a particle with mass M moving along a circle with radius R and therefore $I = MR^2$ moment of inertia. The Lagrangian of the system is given by

$$\mathscr{L} = \frac{I}{2}[(\partial_t x)^2 + (\partial_t y)^2], \tag{5.28}$$

with $x^2 + y^2 = 1$. This is analogous to the $O(2)$ sigma-model in quantum field theory if the coordinates (x,y) of the mass are interpreted as scalar fields (ϕ_1, ϕ_2). Then $\mathscr{L}$ is invariant under the transformation $\phi_i \to T_{ij}\phi_j$ with $T \in O(2)$. This is generalizable to $O(N)$ sigma-models.

With polar coordinates and inverse finite temperature T the action of the topological osciallator is given by

$$\mathrm{S} = \frac{I}{2}\int_0^T \mathrm{d}t\,(\partial_t \varphi)^2. \tag{5.29}$$

To solve this model on a computer, the time can be discretized by d timesteps with distance a. The inverse finite temperature is then interpreted as the time extent of the lattice with $T = a \cdot d$. We used

2 For Legendre polynomials the correct asymptotic error scaling is $\frac{(n!)^4}{((2n)!)^3}$, [66], which is slightly improved over $\frac{1}{(2n)!}$.

a particular form for the discretized derivative, a non-linear term $\frac{1}{2}(\partial_t \varphi)^2 \approx \frac{1}{a}(1 - \cos(\varphi_{i+1} - \varphi_i))$,

$$S^e = \frac{I}{a} \sum_{i=1}^{d} (1 - \cos(\varphi_{i+1} - \varphi_i)) \,. \tag{5.30}$$

One characteristic observable of this model is its topological charge,

$$Q_{\text{top}} = \frac{1}{2\pi} \sum_{i=1}^{d} (\varphi_{i+1} - \varphi_i) \mod 2\pi, \tag{5.31}$$

which describes here the number of complete revolutions of the rotor in the time period T and is therefore an integer number. The topological charge is also an important quantity in QCD, where it is connected to chiral symmetry and the mass of the η'-meson. The width of the topological charge distribution is the topological charge susceptibility,

$$\chi_{\text{top}} = \frac{Q_{\text{top}}^2}{T}. \tag{5.32}$$

This χ_{top} is an observable function, which is of the form (5.6) with $k = 2$.

The model approaches the continuum limit for $\frac{I}{a} \to \infty$. The continuum limit of the topological susceptibility is $\langle \chi_{\text{top}} \rangle \overset{I/a \to \infty}{\longrightarrow} \frac{1}{4\pi^2 I}$, [25].

5.4 NUMERICAL RESULTS

Section (5.2) describes the RNI method and its error scaling. Whether the asymptotic error scaling can be reached has to be tested in practice to check the advantage of RNI over MCMC methods. We computed the topological susceptibility in the topological oscillator model and compared it to result from an optimal MCMC simulation. This section shows results approaching the continuum limit of the topological osciallator, first computed with MCMC, then with the RNI method. After that it demonstrates the error scaling of the RNI method and finally compares the runtime of both, RNI and MCMC, methods.

We found that the Cluster algorithm is an optimal MCMC algorithm for the application to the topological oscillator and therefore a challenging comparison algorithm for the RNI method. We saw that the RNI method gives correct results near the continuum limit and that its error decays at least exponentially. We finally measured that the RNI method needs orders of magnitude less runtime than the cluster algorithm. All in all the RNI method gives better results than the optimal cluster MCMC algorithm when applied to the topological oscillator. The next step is to develop the method further to be applicable to larger dimensional models and check the advantage of the method over MCMC methods there.

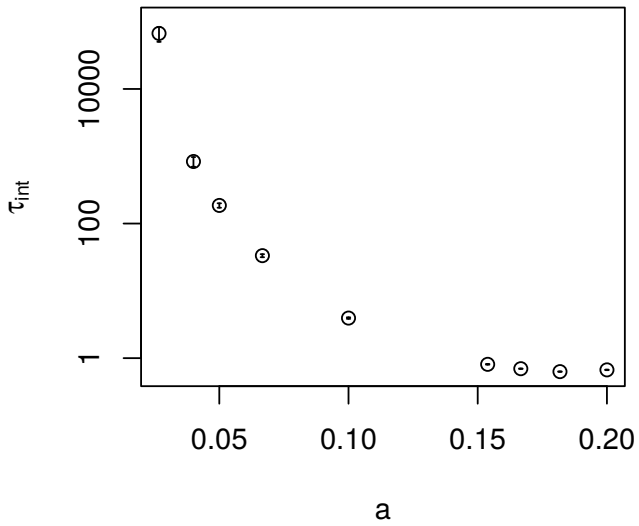


Figure 5.1: Computing the topological susceptibility with the Metropolis algorithm is difficult when approaching the continuum limit because the autocorrelation times grows substantially.

MCMC RESULTS APPROACHING THE CONTINUUM LIMIT The Cluster algorithm described in [83] can directly be applied to the topological oscillator. It is an optimal MCMC algorithm for the topological oscillator and therefore a good comparison method to the RNI method. Approaching the continuum limit of the model, $\frac{I}{a} \to \infty$, critical slowing-down described in section 4.5 is happening, which can be reduced significantly by the Cluster algorithm. We demonstrated this by applying the Metropolis and the Cluster algorithm to the model and comparing their results. We approached the continuum limit by keeping the moment of inertia fixed, $I = 0.25$, and decreasing the lattice constant a while keeping the full lattice extent $T = a \cdot d = 20$ constant. In the following this limit is only denoted by $a \to 0$. For $I = 0.25$ the continuum topological susceptibility is given by π^{-2}, compare below (5.32). We used a constant number of configurations, $N = 10^5$.

The Metropolis algorithm shows the expected critical slowing down behavior towards the continuum limit. The integrated autocorrelation time for the topological susceptibility grows rapidly towards the continuum limit, see Fig. 5.1, as it is expected from equation (4.21). Therefore, the error estimate for the topological susceptibility grows to large values in the continuum limit as well. For the Cluster algorithm we found autocorrelation times that do not exceed 10, also for as small a-values as $a = 0.002$.

Reasonable simulations approaching the continuum limit are only possible with the Cluster algorithm. This becomes especially clear from the direct comparison of the topological susceptibility behavior towards the continuum limit, computed with both algorithms in Fig. 5.2. The error estimate from the Metropolis algorithm becomes very

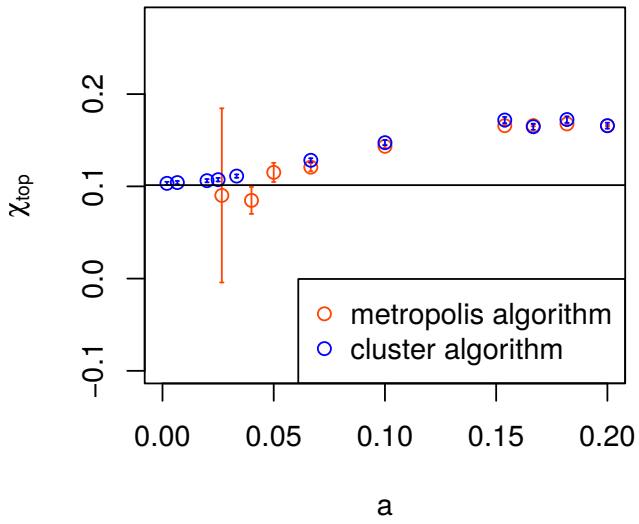


Figure 5.2: Approaching the continuum limit, reasonable error estimates are only possible with the Cluster algorithm, whose results come close to the continuum value of χ_{top}, the black line, for very small a-values.

large for $a < 0.05$. The result from the Cluster algorithm at $a = 0.002$ is close to the analytically computed continuum value $\frac{1}{\pi^2}$ and has a relatively small error estimate.

RNI RESULTS APPROACHING THE CONTINUUM LIMIT We found that the topological susceptibility computed with RNI as described in section 5.2 approaches the continuum limit expectation value. The results using RNI and the Cluster algorithm behave similarly towards the continuum limit, see Fig. 5.3, using $I = 0.25$ and $n = 120$ sampling points for RNI and $N = 10^5$ sampling points for the Cluster algorithm.

ERROR SCALING OF RNI We found that for $n \gtrsim 200$ the error resulting from using n instead of infinitely many sampling points in the RNI method decays exponentially. Because RNI is a deterministic method, the error of an observable for a given number of sampling points n cannot be computed by the statistical fluctuations as done for MCMC methods, but can be computed as the difference to the exact result, which is normally not available. We estimated the error by choosing a large value n_g where we assumed that χ_{top} is approximated quite well. We computed the difference to $\chi_{\text{top}}(n_g)$ for $n < n_g$,

$$\Delta\chi_{\text{top}}(n) = |\chi_{\text{top}}(n) - \chi_{\text{top}}(n_g)|. \tag{5.33}$$

This truncation error behaves exponentially for $m \gtrsim 200$, see Fig 5.4, where we used $a = 0.4$, $I = 0.25$ and $n_g = 560$. The blue line indicates an exponential fit to the data points $n > 200$. This means that for $n \gtrsim 200$ the exponential scaling of the estimation of the asymptotic

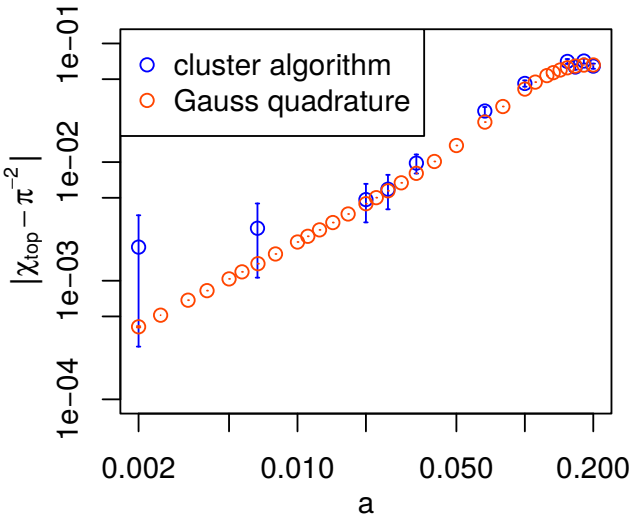


Figure 5.3: The topological susceptibilities, computed with MCMC (cluster algorithm) and RNI (Gauss quadrature), behave similarly towards the continuum limit value π^{-2}.

scaling in (5.27) is reached. Theoretically, for even larger n one would however expect that the error scales with $\mathcal{O}\left(\frac{1}{(2n)!}\right)$, compare (5.26).

RUNTIME COMPARISON OF MCMC AND RNI Our simulations showed that the RNI method needs orders of magnitude less runtime than the Cluster algorithm to result in a specified error estimate on an observable. Because of its error scaling it is clear that the RNI method can give smaller errors than the Cluster algorithm if enough sampling points are used. But is the method also advantageous for lower n-values, where the error scaling is not yet in the exponential error scaling regime? Cluster algorithm and RNI method work very differently and therefore the runtime to compute an observable with a specified error is the best direct comparison of the efficiency of both methods.

With fixed $a = 0.1$ and $I = 0.25$, our Cluster algorithm measurements resulted in an error estimate that decreases proportional to $t^{-1/2}$ for runtime t, see Figure 5.5, consistent with the typical MCMC error scaling in section 4.4. We used between 10^2 and 10^6 sampling points and repeated the measurements for each number of sampling points several times to get an error estimate on both, t and $\Delta\chi_{top}$. The error estimate on t arises due to a fluctuating workload on the computer because of other processes that are running on it, as well as changing Cluster sizes and distributions of the Cluster algorithm. The error estimate on $\Delta\chi_{top}$ originates from the stochastic nature of the algorithm.

The RNI method, using between 10 and 300 sampling points with $n_g = 400$, resulted in orders of magnitude smaller errors, see

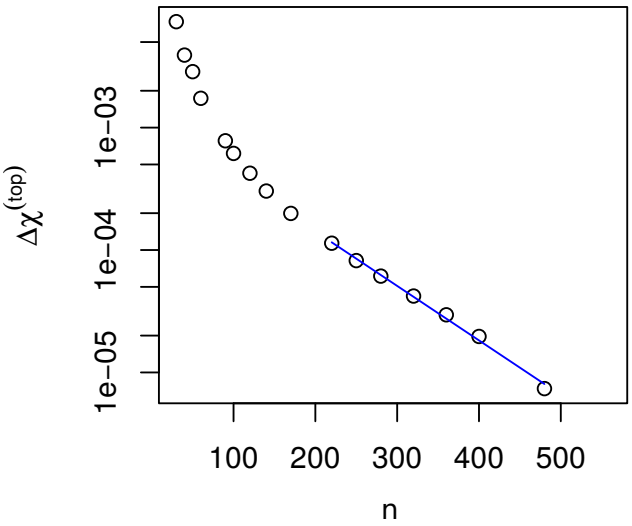


Figure 5.4: The error of the RNI method scales at least exponentially when using enough sampling points n. The blue line indicates an exponential fit to the data points $n > 200$.

Figure 5.5. The exponential error scaling is not visible here, the asymptotic regime of the method is not yet reached with the used numbers of sampling points.

All in all, the RNI method results in orders of magnitude smaller errors than the Cluster algorithm for a fixed runtime or equivalently, the RNI method needs orders of magnitude less runtime than the Cluster algorithm to arrive at a fixed error estimate, already for a number of sampling points where the RNI error does not yet scale exponentially.

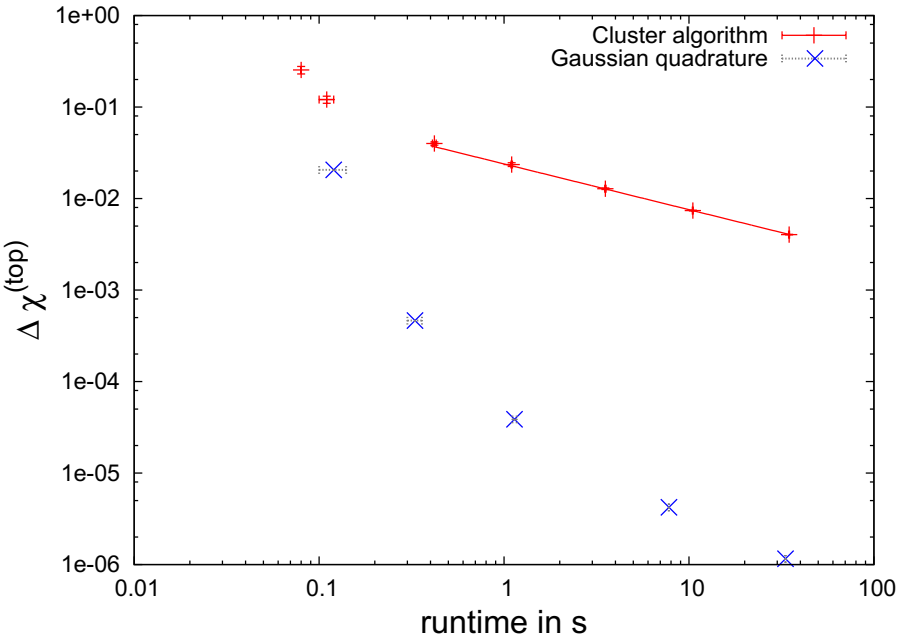


Figure 5.5: The runtime to arrive at a given topological susceptibility error estimate is orders of magnitudes smaller when using the RNI method than using the cluster algorithm.

6

COMPLETELY SYMMETRIZED QUADRADURE RULES

MCMC methods are used for most lattice QCD computations. Section 4.5.2 showed that it is not possible to use MCMC methods in their original form if the integrand is complex, e.g. when introducing a chemical potential μ. Furthermore, variations of the method do not give satisfactory results for specific parameters, e.g. large μ. As discussed in section 4.5.2, this is called the sign-problem. For example, it is the reason why we cannot simulate QCD in the early universe and therefore a big obstacle to understanding the transition from the quark gluon plasma region to the confinement region at lower temperature. The RNI method in combination with the Gaussian quadrature rule as described in chapter 5 is a polynomially exact numerical integration method and solves some issues of MCMC integration. However, the Gaussian quadrature rule is only defined in real space and therefore cannot be applied to complex integrals. In chapter 5 the $\mathcal{U}(1)$ variables of the topological osciallator can be converted to real space, using only angles instead of complex phases, whereas this is not possible for QCD link variables in $\mathcal{SU}(3)$. In this chapter we show that there are alternative quadrature rules that are applicable to $\mathcal{SU}(3)$ variables and can be used to avoid the sign-problem.

We developed polynomially exact quadrature rules which are applicable to complex integrands with one variable in a compact group. They give exact results up to machine precision for polynomial integrands and are based on fully symmetric quadrature rules on spheres from [57]. We applied these *symmetrized quadrature rules* for $\mathcal{U}(N)$ and $\mathcal{SU}(N)$ with $N \leq 3$ to the one-dimensional QCD model with a chemical potential. This is an over-simplified model of QCD with only one variable. In the sign-problem parameter region we compared the results from symmetrized quadrature rules with results from MC simulations. Results are published in [15, 17].

This chapter first explains how the symmetrized quadrature rules are constructed, then introduces the one-dimensional QCD model and finally shows and analyzes numerical results from applying the symmetrized quadrature rules to the model.

In practice, we found that the symmetrized quadrature rule error estimates are orders of magnitude smaller than MC error estimates, especially in the sign-problem parameter region, and are only limited by the machine precision used, in contrast to the MC results. Hence the method is able to avoid the sign-problem for the compact variables, e.g. $\mathcal{SU}(3)$. However, to be able to apply the method to full

QCD, it has to be generalized to more than one variable. A first attempt of an application to a one-dimensional model with more variables is presented in chapter 7 below.

6.1 POLYNOMIALLY EXACT QUADRATURE RULES OVER COMPACT GROUPS

A recently developed method to solve the sign-problem is described in [28, 29] and applied to the one-dimensional QCD with one link variable $U \in \mathcal{SU}(3)$. This model has a sign-problem, the function $\rho(U)$ in the path integral (4.11) is complex and therefore standard MCMC methods cannot be applied. In the articles [28, 29], the variables are grouped into subsets Ω with $|\Omega|$ number of elements, such that the sum of their individual weights $\rho(U)$ in each subset is real and positive, $\sigma_\Omega = \frac{1}{|\Omega|}\sum_{V\in\Omega}\rho(V)$, and therefore MCMC can be applied.

If the subsets consist of $\mathbb{Z}_3$ rotations and complex conjugation, $\Omega_U = \{e^{\frac{2\pi ik}{3}}U, e^{\frac{2\pi ik}{3}}U^\dagger : k \in \{1,2,3\}\}$ for $U \in \mathcal{SU}(3)$ whose elements are all in $\mathcal{SU}(3)$ again, σ_{Ω_U} is real and positive for maximally five fermion flavors [28]. Then the integral

$$I(O,\rho) = \frac{\int_{\mathcal{SU}(3)} dU\, O(U)\rho(U)}{\int_{\mathcal{SU}(3)} dU\, \rho(U)} \tag{6.1}$$

$$= \int_{\mathcal{SU}(3)} dU\, \frac{\sigma_{\Omega_U}}{\int_{\mathcal{SU}(3)} dU\, \sigma_{\Omega_U}} \left(\frac{1}{\sigma_{\Omega_U}|\Omega_U|} \sum_{V\in\Omega_U} O(V)\rho(V) \right), \tag{6.2}$$

can be approximated by MCMC choosing variables according to the distribution $\sigma_{\Omega_U} / \int_{\mathcal{SU}(3)} dU\, \sigma_{\Omega_U}$.

The approach discussed in this chapter generalizes the ideas underlying [28, 29] to larger symmetry groups than the cyclic group $\mathbb{Z}_3$. It is an interesting question which finite subgroups of $\mathcal{SU}(3)$ are best suited for this technique. Note that it is not straightforward to find such symmetry groups where the Ω_U satisfy the condition that σ_{Ω_U} is real and positive. This is also true for other types of interaction, in general $\mathcal{U}(N)$ and $\mathcal{SU}(N)$ with $N > 1$. Since fully symmetric quadrature rules are know on spheres [57], we derived fully symmetrized quadrature rules on the compact groups $\mathcal{SU}(N)$ and $\mathcal{U}(N)$ with $N \leq 3$ from these rules on spheres. Since the resulting rules on compact groups are polynomially exact, an additional MCMC simulation is unnecessary and the involved subsets Ω_U do not have to be chosen such that σ_{Ω_U} is real and positive.

This section first presents the simple quadrature rule for $\mathcal{U}(1)$ and explains fully symmetric quadrature rules on spheres of [57]. It shows how integrals over spheres can be transformed into integrals over compact groups $\mathcal{SU}(N)$ and $\mathcal{U}(N)$ with $N \in \{2,3\}$. Based on this

transformation and the quadrature rules on spheres it presents the completely symmetrized quadrature rules on these compact groups, which are polynomially exact.

6.1.1 *Symmetric quadrature rules on $\mathcal{U}(1)$*

It is easy to choose a symmetrized quadrature rule on $\mathcal{U}(1)$. The integration can be approximated by using $\mathfrak{m}$ equidistant sampling points on the complex unit-circle with equal weights $1/\mathfrak{m}$. The quadrature rule is called a spherical design [40],

$$\int_{\mathcal{U}(1)} \mathrm{d}U\, f(U) \approx Q_{\mathcal{U}(1)} \stackrel{\text{def}}{=} \frac{1}{\mathfrak{m}} \sum_{k=1}^{\mathfrak{m}} f(\mathrm{e}^{\frac{2\pi i k}{\mathfrak{m}}}), \tag{6.3}$$

for some function f. Note that if f is a polynomial of maximal degree $(\mathfrak{m}-1)$, equation (6.3) is an equality on machine precision.

In general, spherical designs are defined on unit n-spheres S^n, embedded in an $(n+1)$-dimensional space, $S^n = \{\boldsymbol{x} \in \mathbb{R}^{n+1} : |\boldsymbol{x}| = 1\}$. Here, $\mathcal{U}(1)$ corresponds to S^1, the unit-circle. To apply this to larger groups, like $\mathcal{U}(N)$ or $\mathcal{SU}(N)$, these groups have to be expressed in terms of spheres first and then a spherical design has to be found for these unit spheres. Since it is difficult to find spherical designs for high-dimensional spheres, we used weighted spherical designs, i.e. polynomially exact quadrature rules that may not be of equal weight and transformed them to polynomially exact quadrature rules for $\mathcal{U}(N)$ and $\mathcal{SU}(N)$ with $N \in \{2,3\}$. In the following the weighted spherical designs are introduced and the quadrature rules for $\mathcal{U}(N)$ and $\mathcal{SU}(N)$ are deduced from them.

6.1.2 *Symmetric quadrature rules on spheres*

We used the polynomially exact quadrature rules on S^n given in [57] which are described in the following. They are based on the Lagrange interpolating polynomials. A function $f : \mathbb{R} \to \mathbb{R}$ with $(\mathfrak{m}+1)$ known support points $(x_k, f(x_k))$ can be approximated by a polynomial P of degree $\mathfrak{m}$,

$$P(x) = \sum_{k=0}^{\mathfrak{m}} f(x_k) L_k(x) = \sum_{k=0}^{\mathfrak{m}} f(x_k) \prod_{\substack{l=0 \\ l\neq k}}^{\mathfrak{m}} \frac{x - x_l}{x_k - x_l}, \tag{6.4}$$

with the Lagrange polynomials L_k [81]. Then a possible quadrature rule of degree $\mathfrak{m}$ for $\int_{\mathbb{R}} \mathrm{d}x\, f(x)$ is the integral over the interpolating polynomial,

$$Q(f) = \int_{\mathbb{R}} \mathrm{d}x\, P(x). \tag{6.5}$$

This section presents the generalization of this quadrature rule to polynomials on spheres, gives an example on how the quadrature

rule chooses points on the 2-sphere in $\mathbb{R}^3$, and shows a possible error estimate computation for these quadrature rules.

QUADRATURE RULE The quadrature rule in (6.5) can be generalized to integrations of functions $f : S^{n-1} \to \mathbb{R}$. In what follows, such a function is approximated by a polynomial, which can be integrated straightforwardly. Similarly to (6.4), the interpolating polynomial is defined by sampling points $\boldsymbol{t} \in S^{n-1}$ and the corresponding function values $f(\boldsymbol{t})$. To obtain a polynomial which is symmetric in $\boldsymbol{t}$ one replaces $f(\boldsymbol{t})$ with $f\{\boldsymbol{t}\}$ that denotes the average over a set of points symmetric to $\boldsymbol{t}$,

$$f\{\boldsymbol{t}\} = \frac{1}{2^{c(\boldsymbol{t})}} \sum_{\boldsymbol{s}} f(s_1 t_1, s_2 t_2, ..., s_n t_n). \tag{6.6}$$

Here $c(\boldsymbol{t})$ is the number of non-zero entries in $\boldsymbol{t}$ and the sum runs over all possible sign combinations with $s_i = \pm 1$ for $t_i \neq 0$.

The sampling points are chosen in the following way: $\boldsymbol{t} \in S^{n-1}$ means that $|\boldsymbol{t}| = t_1^2 + t_2^2 + ... + t_n^2 = 1$. This can be achieved by $\boldsymbol{t} = (u_{p_1}, \ldots, u_{p_n})$ with[1] $u_i = \sqrt{\frac{i}{\mathfrak{m}}}$ for $p_i \in \mathbb{N}_0$ and $\sum_{i=1}^n p_i = \mathfrak{m} \in \mathbb{N}_{>0}$. All such points $\boldsymbol{t}$ can be labeled by the vector $\boldsymbol{p} = (p_1, ..., p_n)$ and we refer to the corresponding point as $\boldsymbol{t}_{\boldsymbol{p}}$.

Then the interpolating polynomial for f is given by

$$P_{S^{n-1}}(\boldsymbol{x}) = \sum_{|\boldsymbol{p}|=\mathfrak{m}} f\{t_{\boldsymbol{p}}\} \prod_{l=1}^{n} \prod_{i=0}^{p_l - 1} \frac{x_l^2 - u_i^2}{u_{p_l}^2 - u_i^2}. \tag{6.7}$$

Here $|\boldsymbol{p}| = \mathfrak{m}$ is the abbreviation for $\sum_{i=1}^n p_i = \mathfrak{m}$. The integral over S^{n-1} of this polynomial is the quadrature rule for $\int_{S^{n-1}} \mathrm{d}\boldsymbol{x}\, f(\boldsymbol{x})$:

$$\begin{aligned} Q_{S^{n-1}}(f) &= \int_{S^{n-1}} \mathrm{d}\boldsymbol{x}\, P_{S^{n-1}}(\boldsymbol{x}) && (6.8)\\ &= \sum_{|\boldsymbol{p}|=\mathfrak{m}} f\{t_{\boldsymbol{p}}\} \int_{S^{n-1}} \mathrm{d}\boldsymbol{x} \prod_{l=1}^{n} \prod_{i=0}^{p_l - 1} \frac{x_l^2 - u_i^2}{u_{p_l}^2 - u_i^2}. && (6.9)\\ &\stackrel{\text{def}}{=} \sum_{|\boldsymbol{p}|=\mathfrak{m}} f\{t_{\boldsymbol{p}}\}\, w_{\boldsymbol{p}}. && (6.10) \end{aligned}$$

It has polynomial degree $2\mathfrak{m} + 1$ with weights $w_{\boldsymbol{p}}$ defined by the integral over S^{n-1} in (6.9). For a specified $\mathfrak{m}$ the weights can be computed, some are shown in [57].

EXAMPLE The surface of a ball is a 2-sphere in three-dimensional space. Sampling points on this sphere have three entries because $\boldsymbol{t} \in S^2$ and also $\boldsymbol{p} \in \mathbb{N}_{\geq 0}^3$. Choosing $\mathfrak{m} = 1$ means that $p_1 + p_2 + p_3 = \mathfrak{m} = 1$, which is possible for three district $\boldsymbol{p} \in \{(1,0,0), (0,1,0), (0,0,1)\}$.

1 More generally $u_i = \sqrt{\frac{i+\mu}{\mathfrak{m}+\mu n}}$ for some $0 \leq \mu \leq 1$.

With $\mathfrak{m} = 1$ there are two distinct sampling point entries $u_0 = \sqrt{0/1} = 0$ and $u_1 = \sqrt{1/1} = 1$. The sampling points are chosen with respect to $\boldsymbol{p}$, therefore there is one sampling point per $\boldsymbol{p}$, $\boldsymbol{t}_{\boldsymbol{p}} = (u_{p_1}, u_{p_2}, u_{p_3})$:

$$\begin{aligned}
\boldsymbol{t}_{(1,0,0)} &= (u_1, u_0, u_0) = (1,0,0),\\
\boldsymbol{t}_{(0,1,0)} &= (u_0, u_1, u_0) = (0,1,0),\\
\boldsymbol{t}_{(0,0,1)} &= (u_0, u_0, u_1) = (0,0,1).
\end{aligned} \tag{6.11}$$

In general $\boldsymbol{t}_{\boldsymbol{p}}$ is not equal to $\boldsymbol{p}$. The number of non-zero entries in each $\boldsymbol{t}$ is $c(\boldsymbol{t}) = 1$ (for $\mathfrak{m} \neq 1$ these $c(\boldsymbol{t})$ are in general not the same for every $\boldsymbol{t}$), therefore it is

$$\begin{aligned}
f\{\boldsymbol{t}_{(1,0,0)}\} &= \frac{1}{2}\left(f(+1,0,0) + f(-1,0,0)\right),\\
f\{\boldsymbol{t}_{(0,1,0)}\} &= \frac{1}{2}\left(f(0,+1,0) + f(0,-1,0)\right),\\
f\{\boldsymbol{t}_{(0,0,1)}\} &= \frac{1}{2}\left(f(0,0,+1) + f(0,0,-1)\right),
\end{aligned} \tag{6.12}$$

with the same weights for all three functions, $w_{\boldsymbol{p}} = \frac{4\pi}{3}$, [57]. By construction, points whose coordinates are permutations of each other have the same weight. Thus, the quadrature rule $Q_{S^2}(f)$ in (6.10) has to satisfy $3w = Q_{S^2}(1)$ and $Q_{S^2}(1)$ is the volume of the full surface, $Q_{S^2}(1) = \mathrm{vol}(S^2) = 4\pi$, therefore $w = 4\pi/3$. For larger $\mathfrak{m}$ the weights in general differ among each other. The quadrature rule sums over all possible $\boldsymbol{p}$ vectors:

$$\begin{aligned}
Q_{S^2}(f) = \frac{4\pi}{3} \cdot \frac{1}{2}\big(&f(1,0,0) + f(-1,0,0) + f(0,1,0) + f(0,-1,0)\\
&+ f(0,0,1) + f(0,0,-1)\big).
\end{aligned} \tag{6.13}$$

This quadrature rule uses six symmetric points on the sphere, all with the same weight. This is a polynomially exact rule for integrals of polynomials of degree $2\mathfrak{m} + 1 = 3$ for $\mathfrak{m} = 1$.

$f(\boldsymbol{x}) = x_3^2$ is a polynomial of degree 2 and therefore should be integrated exactly by Q_{S^2}. Additionally, the integral over the 2-sphere of $f(\boldsymbol{x})$ is not too complicated to solve analytically. In spherical coordinates with $x_3 = \cos\theta$ this integral is

$$I_{S^2}(f) = \int_0^{2\pi} \mathrm{d}\phi \int_0^{\pi} \mathrm{d}\theta \, \sin\theta \cos^2\theta = \frac{4\pi}{3}. \tag{6.14}$$

In the quadrature rule (6.13) only the values $f(0,0,1) = 1$ and $f(0,0,-1) = 1$ are non-zero. Then $Q_{S^2}(f) = \frac{4\pi}{3}$ gives the same value as the analytic calculation (6.14).

For $\mathfrak{m} = 1$ the symmetrized quadrature rule on S^n has $2(n+1)$ sampling points.

ERROR ESTIMATE To compute an error estimate for $Q_{S^{n-1}}(f)$, the quadrature rule in (6.10) can be randomized by applying random orthogonal $n \times n$ matrices Z, $Z^T Z = \mathbb{1}$, to the vectors $t_{\boldsymbol{p}}$ [57],

$$Q_{S^{n-1}}(f, Z) = \sum_{|\boldsymbol{p}|=\mathfrak{m}} f\{Z t_{\boldsymbol{p}}\}\, w_{\boldsymbol{p}}. \tag{6.15}$$

If M matrices Z_i are chosen randomly according to the Haar measure from the set of all matrices in the orthogonal group, $\overline{Q}_{S^{n-1}}(f, Z)$ is an unbiased estimator for $\int_{S^{n-1}} \mathrm{d}\boldsymbol{x}\, f(\boldsymbol{x})$,

$$\overline{Q}_{S^{n-1}}(f, Z) = \frac{1}{M} \sum_{i=1}^{N} Q_{S^{n-1}}(f, Z_i) \tag{6.16}$$

with the error estimate

$$\Delta Q_{S^{n-1}} = \sqrt{\frac{1}{M(M-1)} \sum_{i=1}^{N} \left(Q_{S^{n-1}}(f, Z_i) - \overline{Q}_{S^{n-1}}(f, Z)\right)^2}. \tag{6.17}$$

6.1.3 *Connection between compact groups and spheres*

The groups $\mathcal{U}(N)$ and $\mathcal{SU}(N)$ with $N \in \{2, 3\}$ can be connected to products of spheres S^n via

$$\mathcal{SU}(N) \simeq S^3 \times S^5 \times \ldots \times S^{2N-1}, \tag{6.18}$$

$$\mathcal{U}(N) \simeq S^1 \times S^3 \times \ldots \times S^{2N-1}. \tag{6.19}$$

For an isomorphism which preserves the group structure, $\Phi : \bigtimes_j S^{2j-1} \to G$ with $G \in \{\mathcal{U}(N), \mathcal{SU}(N)\}$, the integral over the Haar-measure of G can be written as the integral over the products of spheres,

$$\begin{aligned}\int_G \mathrm{d}U\, f(U) = \int_{S^{2N-1}} \mathrm{d}\boldsymbol{x}_{S^{2N-1}} \int_{S^{2N-3}} \mathrm{d}\boldsymbol{x}_{S^{2N-3}} \cdots \int_{S^{n+2}} \mathrm{d}\boldsymbol{x}_{S^{n+2}} \cdot \\ \cdot \int_{S^n} \mathrm{d}\boldsymbol{x}_{S^n} f(\Phi(\boldsymbol{x}_{S^{2N-1}}, \boldsymbol{x}_{S^{2N-3}}, \ldots, \boldsymbol{x}_{S^{n+2}}, \boldsymbol{x}_{S^n})),\end{aligned} \tag{6.20}$$

with $n = 1$ for $\mathcal{U}(N)$ and $n = 3$ for $\mathcal{SU}(N)$ [17]. Here $\boldsymbol{x}_{S^k}$ is an element on the k-sphere. The polynomially exact quadrature rules on spheres in section 6.1.2 can be used to estimate integrals over compact groups, if the isomorphism Φ for the given group is known. In the following, Φ is given for $\mathcal{SU}(N)$ and $\mathcal{U}(N)$ with $N \in \{2, 3\}$, [15].

$\mathcal{SU}(2)$ The isomorphism for $\mathcal{SU}(2)$ is given by

$$\begin{aligned}\Phi_{\mathcal{SU}(2)} : S^3 &\to \mathcal{SU}(2), \\ \boldsymbol{x} &\mapsto \begin{pmatrix} x_1 + i x_2 & -(x_3 + i x_4)^* \\ x_3 + i x_4 & (x_1 + i x_2)^* \end{pmatrix}.\end{aligned} \tag{6.21}$$

$\mathcal{SU}(3)$ For $\mathcal{SU}(3)$ spherical coordinates of S^5 are needed,

$$\Psi : [0, 2\pi)^3 \times [0, \frac{\pi}{2}) \to S^5,$$

$$(\alpha_1, \alpha_2, \alpha_3, \phi_1, \phi_2) \mapsto \begin{pmatrix} \cos\alpha_1 \sin\phi_1 \\ \sin\alpha_1 \sin\phi_1 \\ \sin\alpha_2 \cos\phi_2 \sin\phi_2 \\ \cos\alpha_2 \cos\phi_2 \sin\phi_2 \\ \sin\alpha_3 \cos\phi_1 \cos\phi_2 \\ \cos\alpha_3 \cos\phi_1 \cos\phi_2 \end{pmatrix}. \tag{6.22}$$

Then the seeked isomorphism is

$$\Phi_{\mathcal{SU}(3)} : S_1^5 \times S^3 \to \mathcal{SU}(3), \tag{6.23}$$

$$(\boldsymbol{x}, \boldsymbol{y}) \mapsto A(\Psi^{-1}(\boldsymbol{x})) \cdot B(\boldsymbol{y}). \tag{6.24}$$

with the matrices

$$A(\Psi^{-1}(\boldsymbol{x})) = \begin{pmatrix} e^{i\alpha_1}\cos\phi_1 & 0 & e^{i\alpha_1}\sin\phi_1 \\ -e^{i\alpha_2}\sin\phi_1\sin\phi_2 & e^{-i(\alpha_1+\alpha_3)}\cos\phi_2 & e^{i\alpha_2}\cos\phi_1\sin\phi_2 \\ -e^{i\alpha_3}\sin\phi_1\cos\phi_2 & -e^{-i(\alpha_1+\alpha_2)}\sin\phi_2 & e^{i\alpha_3}\cos\phi_1\cos\phi_2 \end{pmatrix}, \tag{6.25}$$

$$B(\boldsymbol{y}) = \begin{pmatrix} x_1 + ix_2 & -(x_3 + ix_4)^* & 0 \\ x_3 + ix_4 & (x_1 + ix_2)^* & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{6.26}$$

$\Psi^{-1}(\boldsymbol{x})$ is the inverse transformation of (6.22) from Euclidean to spherical coordinates. S_1^5 denotes S^5 without its poles, $\phi_1 = 0$ or $\phi_2 = 0$ because there the inverse transformation is not unique. The therefore excluded set is a null set, thus $\Phi_{\mathcal{SU}(3)}$ can still be used in (6.20).

$\mathcal{U}(2)$ The isomorphism for $\mathcal{U}(2)$ is

$$\Phi_{\mathcal{U}(2)} : S^3 \times S^1 \to \mathcal{U}(2), \tag{6.27}$$

$$(\boldsymbol{x}, \alpha) \mapsto \Phi_{\mathcal{SU}(2)}(\boldsymbol{x}) \cdot e^{i\alpha}. \tag{6.28}$$

The vector $\boldsymbol{y} \in S^1$ is written as a complex phase $e^{i\alpha}$.

$\mathcal{U}(3)$ The isomorphism for $\mathcal{U}(3)$ is then given by

$$\Phi_{\mathcal{SU}(3)} : S_1^5 \times S^3 \times S^1 \to \mathcal{U}(3), \tag{6.29}$$

$$(\boldsymbol{x}, \boldsymbol{y}, \alpha) \mapsto \Phi_{\mathcal{SU}(3)}(\boldsymbol{x}, \boldsymbol{y})\, e^{i\alpha}. \tag{6.30}$$

6.1.4 *Symmetrized quadrature rules on compact groups*

An integral over compact groups $\mathcal{U}(N)$, $\mathcal{SU}(N)$ for $N \in \{2, 3\}$ can be written as integrals over spheres using equation (6.20) and the shown isomorphisms Φ in section 6.1.3. These integrals over spheres can be approximated by the quadrature rule given in (6.10). In the following, the derived symmetrized quadrature rules over compact groups are shown.

$\mathcal{SU}(2)$

$$Q_{\mathcal{SU}(2)}(f) = \sum_{|\boldsymbol{p}|=\mathfrak{m}} w_{\boldsymbol{p}} f\left\{\begin{pmatrix} t_{p_1}+it_{p_2} & -(t_{p_3}+it_{p_4})^* \\ t_{p_3}+it_{p_4} & (t_{p_1}+it_{p_2})^* \end{pmatrix}\right\} \tag{6.31}$$

$$= \sum_{|\boldsymbol{p}|=\mathfrak{m}} w_{\boldsymbol{p}} f\{\Phi_{\mathcal{SU}(2)}(\boldsymbol{t}_{\boldsymbol{p}})\}, \tag{6.32}$$

with $f\{\Phi_{\mathcal{SU}(2)}(\boldsymbol{t}_{\boldsymbol{p}})\} = 2^{-c(t)} \sum_s f(\Phi_{\mathcal{SU}(2)}(s_1 t_1, ..., s_n t_n))$, sampling points $\boldsymbol{t}_{\boldsymbol{p}}$ and weights $w_{\boldsymbol{p}}$ on S^3.

$\mathcal{SU}(3)$

$$Q_{\mathcal{SU}(3)}(f) = \sum_{|\boldsymbol{q}|=\mathfrak{m}} v_{\boldsymbol{q}} \sum_{|\boldsymbol{p}|=\mathfrak{m}} w_{\boldsymbol{p}} f\{\Phi_{\mathcal{SU}(3)}(\boldsymbol{s}_{\boldsymbol{q}}, \boldsymbol{t}_{\boldsymbol{p}})\}, \tag{6.33}$$

where $\boldsymbol{s}_{\boldsymbol{q}}$ and $v_{\boldsymbol{q}}$ are the sampling points and weights on S^5 and $\boldsymbol{t}_{\boldsymbol{p}}$ and $w_{\boldsymbol{p}}$ are the sampling points and weights on S^3.

$\mathcal{U}(1)$

$$Q_{\mathcal{U}(1)}(f) = \frac{1}{\mathfrak{m}_{\mathcal{U}(1)}} \sum_{k=1}^{\mathfrak{m}_{\mathcal{U}(1)}} f\{e^{\frac{2\pi i k}{\mathfrak{m}_{\mathcal{U}(1)}}}\}. \tag{6.34}$$

This is the spherical design in (6.1.1). Its number of sampling points is denoted by $\mathfrak{m}_{\mathcal{U}(1)}$ to distinguish it from $\mathfrak{m}$ in the weighted spherical designs for $\mathcal{U}(N)$ and $\mathcal{SU}(N)$.

$\mathcal{U}(2)$

$$Q_{\mathcal{U}(2)}(f) = \frac{1}{\mathfrak{m}_{\mathcal{U}(1)}} \sum_{|\boldsymbol{p}|=\mathfrak{m}} \sum_{k=1}^{\mathfrak{m}_{\mathcal{U}(1)}} w_{\boldsymbol{p}} f\{\Phi_{\mathcal{SU}(2)}(t_{\boldsymbol{p}}) \cdot e^{\frac{2\pi i k}{\mathfrak{m}_{\mathcal{U}(1)}}}\}, \tag{6.35}$$

where $\boldsymbol{t}_{\boldsymbol{p}}$ and $w_{\boldsymbol{p}}$ are the sampling points and weights on S^3.

$\mathcal{U}(3)$

$$Q_{\mathcal{SU}(3)}(f) = \frac{1}{\mathfrak{m}_{\mathcal{U}(1)}} \sum_{k=1}^{\mathfrak{m}_{\mathcal{U}(1)}} \sum_{|\boldsymbol{q}|=\mathfrak{m}} v_{\boldsymbol{q}} \sum_{|\boldsymbol{p}|=\mathfrak{m}} w_{\boldsymbol{p}} \cdot \cdot f\{\Phi_{\mathcal{SU}(3)}(\boldsymbol{s}_{\boldsymbol{q}}, \boldsymbol{t}_{\boldsymbol{p}}) \cdot e^{\frac{2\pi i k}{\mathfrak{m}_{\mathcal{U}(1)}}}\}, \tag{6.36}$$

where $\boldsymbol{s}_{\boldsymbol{q}}$ and $v_{\boldsymbol{q}}$ are the sampling points and weights on S^5 and $\boldsymbol{t}_{\boldsymbol{p}}$ and $w_{\boldsymbol{p}}$ are the sampling points and weights on S^3.

6.2 ONE-DIMENSIONAL LATTICE QCD

Because the presented symmetrized quadrature rules are polynomially exact, they can possibly avoid the sign-problem. To test this, we applied the rules to a model with a sign-problem where the MC simulations become unfeasible. Preferably, analytic results are known for the model to directly compare the quadrature results to them. And to apply the symmetrized quadrature rules in section 6.1.4 directly, a model with only one variable is needed.

The one-dimensional QCD model with chemical potential [26] has all these desired properties. It describes fermions on a string, interacting via compact links. It is an oversimplified version of the four-dimensional QCD and has some similar characteristics. On the one hand, a chemical potential in QCD leads to non-vanishing baryon density and is an important parameter in the dense early universe. The sign-problem for large chemical potential, compare section 4.5.2, prevents the understanding of the early universe. On the other hand, both QCD models have the chiral condensate observable, which is a measure for chiral symmetry breaking in the model. In four-dimensional QCD this breaking is responsible for fundamental characteristics of our world, like the meson masses, especially the light pion masses.

This section first presents the discretized Euclidean action of the model and its chiral condensate, which can be derived from the partition function of the model. Then it shows that by rewriting the partition function, the model is only dependent on one variable and finally gives analytic results for the partition function using different compact groups.

THE MODEL The one-dimensional lattice contains L discretized time-steps, with lattice spacing a. The quark field (here just one flavor is used) with mass m is described by L Grassmann-variables Ψ_i, each one associated to one lattice point. The interaction between the quark field components is described by L compact link-variables U_i. We investigated the model using different groups, $U_i \in \{\mathcal{U}(1), \mathcal{U}(N), \mathcal{SU}(N) : N \in \{2,3\}\}$. A chemical potential $\mu > 0$ describes a non-vanishing baryon density. The discretized Euclidean action is given by

$$\begin{aligned} S^e(\boldsymbol{U}, \boldsymbol{\Psi}, \overline{\boldsymbol{\Psi}}) &= a \sum_{i=1}^{L} \left(m \overline{\Psi}_i \Psi_i + \frac{e^{\mu}}{2a} \overline{\Psi}_{i+1} U_i \Psi_i - \frac{e^{-\mu}}{2a} \overline{\Psi}_{i-1} U_i^{\dagger} \Psi_i \right) \\ &= a \sum_{i=1}^{L} \overline{\Psi}_i \left(m \delta_{i,j} + \frac{e^{\mu}}{2a} \delta_{i,j+1} U_j - \frac{e^{-\mu}}{2a} \delta_{i,j-1} U_j^{\dagger} \right) \Psi_j, \end{aligned} \tag{6.37}$$

with $\overline{\boldsymbol{\Psi}} = (\overline{\Psi_1}, ..., \overline{\Psi_L})$, $\boldsymbol{\Psi} = (\Psi_1, ..., \Psi_L)^T$ and $\boldsymbol{U} = (U_1, ..., U_L)^T$. This can be written in terms of the Dirac matrix $\mathfrak{D}(\boldsymbol{U})$,

$$S^e(\boldsymbol{U}, \boldsymbol{\Psi}, \overline{\boldsymbol{\Psi}}) = a\, \overline{\boldsymbol{\Psi}}\, \mathfrak{D}(\boldsymbol{U})\, \boldsymbol{\Psi}, \tag{6.38}$$

with

$$\mathfrak{D}(\boldsymbol{U}) = \begin{pmatrix} m\mathbb{1}_N & \frac{e^{\mu}}{2}U_1 & & & & \frac{e^{-\mu}}{2}U_L^\dagger \\ -\frac{e^{-\mu}}{2}U_1^\dagger & m\mathbb{1}_N & \frac{e^{\mu}}{2}U_2 & & & \\ & -\frac{e^{-\mu}}{2}U_2^\dagger & m\mathbb{1}_N & \frac{e^{\mu}}{2}U_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -\frac{e^{-\mu}}{2}U_{L-2}^\dagger & m\mathbb{1}_N & \frac{e^{\mu}}{2}U_{L-1} \\ -\frac{e^{\mu}}{2}U_L & & & & -\frac{e^{-\mu}}{2}U_{L-1}^\dagger & m\mathbb{1}_N \end{pmatrix}, \tag{6.39}$$

where all empty entries are zero. This is the oversimplified one-dimensional version of the QCD fermion action in (2.9). The following calculations use $a = 1$. The expectation value of the chiral condensate is

$$\chi = \langle \overline{\boldsymbol{\Psi}} \boldsymbol{\Psi} \rangle = \frac{1}{Z} \int \mathrm{d}\boldsymbol{\Psi} \int \mathrm{d}\overline{\boldsymbol{\Psi}} \int \mathrm{d}\boldsymbol{U}\ \overline{\boldsymbol{\Psi}} \boldsymbol{\Psi}\, \mathrm{e}^{-S^e(\overline{\boldsymbol{\Psi}}, \boldsymbol{\Psi}, \boldsymbol{U})}, \tag{6.40}$$

with the partition function $Z = \int \mathrm{d}\boldsymbol{\Psi} \int \mathrm{d}\overline{\boldsymbol{\Psi}} \int \mathrm{d}\boldsymbol{U}\ \mathrm{e}^{-S^e(\overline{\boldsymbol{\Psi}}, \boldsymbol{\Psi}, \boldsymbol{U})}$. The chiral condensate is a measure for the chiral symmetry breaking in the model, which results here from the non-zero quark mass m. Using the definition of the action in (6.37), χ can be written as

$$\chi = \frac{\partial_m Z}{Z}. \tag{6.41}$$

Because the action in (6.38) is bilinear in the fermion fields, the integration over them can be done analytically, such that the partition function is

$$Z = \int \mathrm{d}\boldsymbol{U}\ \det(\mathfrak{D}(\boldsymbol{U})). \tag{6.42}$$

REDUCING THE DIMENSIONS The dimension of this integration (over L variables U_i) can be reduced by using the structure of $\mathfrak{D}(\boldsymbol{U})$ and gauge invariance. The determinant of a block decomposed matrix

$$X = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \text{ is } \det(X) = \det(A)\det(D - CA^{-1}B).$$

Starting with $A = m\mathbb{1}_N$ and applying this block decomposition to $\det \mathfrak{D}(\boldsymbol{U})$ iteratively gives [15]

$$\det \mathfrak{D}(\boldsymbol{U}) = \det\left(\prod_{j=1}^{L} \tilde{m}_j + 2^{-L}\,\mathrm{e}^{-L\mu}\left(\prod_{j=1}^{L} U_j\right)^{\dagger} + \right.$$
$$\left. + (-1)^L 2^{-L}\,\mathrm{e}^{L\mu}\prod_{j=1}^{L} U_j\right), \tag{6.43}$$

$$\text{with } \tilde{m}_1 = m,$$
$$\tilde{m}_j = m + \frac{1}{4\tilde{m}_{j-1}} \quad \forall j \in \{2, 3, ..., L-1\},$$
$$\tilde{m}_L = m + \frac{1}{4\tilde{m}_{L-1}} + \sum_{j=1}^{L-1} \frac{(-1)^{j+1}2^{-2j}}{\tilde{m}_j \prod_{k=1}^{j-1} \tilde{m}_k^2}.$$

The determinant in (6.43) is only dependent on the product of all U_i's, $U \overset{\text{def}}{=} \prod_{j=1}^{L} U_j$. Because the Haar measure is independent of multiplication with other group elements, the partition function Z does not change if the gauge $U_j = \mathbb{1}$ for all $j \in \{1, ..., L-1\}$ and $U_L = U$ is used. Therefore the determinant is only dependent on one variable,

$$\det \mathfrak{D}(U) = \det\left(c_1 + c_2 U^{\dagger} + c_3 U\right),$$
$$\text{with } c_1 = \prod_{j=1}^{L} \tilde{m}_j, \tag{6.44}$$
$$c_2 = 2^{-L}\,\mathrm{e}^{-L\mu},$$
$$c_3 = (-1)^L 2^{-L}\,\mathrm{e}^{L\mu}.$$

For $U \in \{\mathcal{U}(N), \mathcal{SU}(N)\}$ the determinant in (6.44) is a polynomial of degree N in the entries of U, which is also the case for its derivative $\partial_m \det \mathfrak{D}(U)$.

The determinant in (6.44) is complex. Therefore it cannot be used as a weight function for MCMC methods. Especially for $c_1 \ll c_2, c_3$, which means $m \ll L\mu$, the determinant is dominated by the complex matrix U and the sign-problem arises when using ordinary MC methods without any importance sampling described in section 4.2.

ANALYTIC RESULTS The integral over the determinant in (6.42) can be computed analytically,

$$\begin{aligned} Z(\mathcal{U}(1)) &= c_1, \\ Z(\mathcal{U}(2)) &= c_1^2 - c_2 c_3, \\ Z(\mathcal{SU}(2)) &= c_1^2 - c_2 c_3 + c_2^2 + c_3^2, \\ Z(\mathcal{U}(3)) &= c_1^3 - 2c_1 c_2 c_3, \\ Z(\mathcal{SU}(3)) &= c_1^3 - 2c_1 c_2 c_3 + c_2^3 + c_3^3. \end{aligned} \tag{6.45}$$

From (6.45) is follows that the partition functions for $\mathcal{U}(N)$ and $\mathcal{SU}(N)$ with $N \in \{2,3\}$ are related via

$$Z(\mathcal{SU}(N)) = Z(\mathcal{U}(N)) + c_2^N + c_3^N. \tag{6.46}$$

6.3 NUMERICAL RESULTS

We tested whether the symmetrized quadrature rules given in section 6.1.4 avoid the sign-problem in one-dimensional QCD. We computed the chiral condensate with ordinary MC and symmetrized quadrature rules and compared the results from both methods. We used ordinary MC without importance sampling to circumvent the problem of $\det \mathfrak{D} \in \mathbb{C}$ which cannot be used as a weight function for importance sampling. We investigated the chiral condensate using link variables $U \in \{\mathcal{U}(1), \mathcal{U}(2), \mathcal{U}(3), \mathcal{SU}(1), \mathcal{SU}(2)\}$. In the following $\mathcal{U}(N)$ and $\mathcal{SU}(N)$ denote the compact groups with $N \leq 3$. This section first visualizes the sign-problem in MC simulations. Then it presents results, first for the partition function as a first step towards the chiral condensate, and then for the chiral condensate itself. For both, partition function and chiral condensate, first the analytic solutions are discussed and then the numerical results are presented.

We found large MC error estimates for the partition function, especially for $\mathcal{U}(N)$ variables in the region $L\mu \gg m$. In this region $Z(\mathcal{U}(N))$ is very small. Each MC step gives a value which can be much larger than $Z(\mathcal{U}(N))$ and the average over all MC step values cannot resolve these small $Z(\mathcal{U}(N))$ values. By applying symmetrized quadrature rules, large values seem to cancel each other and give error estimates that are smaller by orders of magnitude. This means that the presented rules can be used to avoid the sign-problem. Here the tested integrands are polynomials of maximal degree three, but the rules should also work for integrands with larger polynomial degrees or integrands that can be approximated by polynomials. The efficiency of the application of the method to models with more variables has to be tested and the method possibly has to be developed further, see chapter 7 for such an attempt.

In the application of the symmetrized quadrature rules to the computation of the chiral condensate we used that the integrands, $\det \mathfrak{D}$ and $\partial_m \det \mathfrak{D}$, are polynomials of degree N. Therefore the rules are exact on machine precision if $\mathfrak{m}_{\mathcal{U}(1)} - 1 = N$, compare section 6.1.1, and $2\mathfrak{m} + 1 = N$, compare section 6.1.2. For simplicity we used for all rules the numbers for $N = 3$, therefore $\mathfrak{m}_{\mathcal{U}(1)} = 4$ and $\mathfrak{m} = 1$. As given in section 6.1.2, for $\mathfrak{m} = 1$ the symmetrized quadrature rule on

S^3 includes eight sampling points, while the one on S^5 includes 12 points. This results in

$$\#Q_G = \begin{cases} 8, & G = \mathcal{SU}(2) \\ 96, & G = \mathcal{SU}(3) \\ 4, & G = \mathcal{U}(1) \\ 32, & G = \mathcal{U}(2) \\ 384, & G = \mathcal{U}(3) \end{cases} \tag{6.47}$$

numbers of sampling points for the symmetrized quadrature rules on compact groups given in section 6.1.4.

For ordinary MC sampling we used the quadrature rule given in (4.5). For the partition function this rule is given by

$$Z(G) \approx \frac{1}{N_{\text{MC}}} \sum_{\substack{k=1 \\ U_k \in G}}^{\text{MC}} \det(c_1 + c_2 U_k^\dagger + c_3 U_k), \tag{6.48}$$

for $G \in \{\mathcal{U}(1), \mathcal{U}(2), \mathcal{U}(3), \mathcal{SU}(2), \mathcal{SU}(3)\}$. When comparing MC results with results from the symmetrized quadrature rules we used the same numbers of sampling points as the symmetrized quadrature rule for the corresponding group G, $N_{\text{MC}} = \#Q_G$, compare (6.47). The U_k's are uniformly distributed random matrices.[2]

For both MC and symmetrized quadrature rules we computed the error estimate by the relative deviation from the analytic value,

$$\Delta O = \frac{|O_{\text{numerical}} - O_{\text{analytic}}|}{|O_{\text{analytic}}|}, \tag{6.49}$$

for $O \in \{Z, \chi\}$. We computed the standard deviation from this error by repeatedly using on the one hand the MC quadrature rules with different seeds and on the other hand the symmetrized quadrature rules with different random orthogonal matrices, compare (6.17).

6.3.1 *Visualizing the sign-problem*

MCMC methods cannot be applied straightforwardly to the model because of its complex "weight" factor $\det \mathfrak{D}(U)$. Therefore, we used ordinary MC by choosing random matrices $U \in \mathcal{U}(N)$ and $\mathcal{SU}(N)$ and computed numerator and denominator for χ in (6.41) separately. For example, for $\mathcal{SU}(2)$ with $m = 0.25$, $L = 8$ and $\mu = 1$ the MC error estimate in (6.49) stays almost constant over a large range of numbers of sampling points N_{MC} and is of order one, see Fig. 6.1. Here the system is in the situation $L\mu \gg m$ where the sign-problem occurs. Due

2 We chose uniformly distributed points on spheres (using a vector with normally distributed coordinates) and used the isomorphisms described in section 6.1.3 to convert them to $\mathcal{SU}(N)$ and $\mathcal{U}(N)$ matrices.

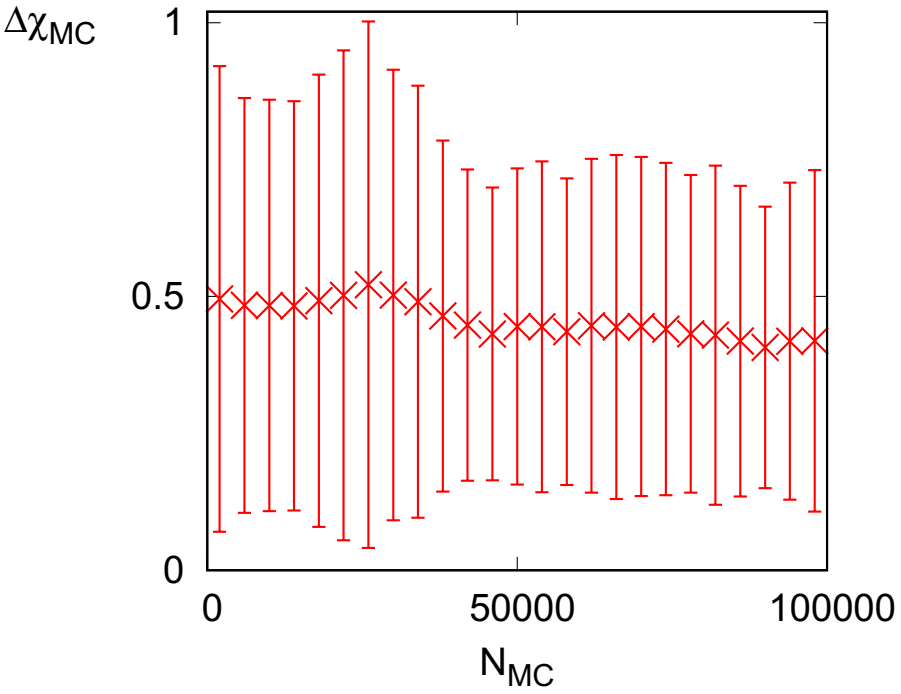


Figure 6.1: The chiral condensate MC error stays almost constant over a large range of numbers of sampling points N_{MC} for small quark masses m with $L\mu \gg m$ and is of order one.

to the error estimate being of order one, it is not possible to obtain a statistically significant result for χ. In the following, the parameter-space of the mass m is analyzed, while the chemical potential is fixed to a constant $\mu = 1$. Then the sign-problem region should occur for $m \ll 1$.

6.3.2 *The partition function*

Computing the partition function is the first step towards the computation of the chiral condensate. This section first presents the analytic partition function, analyzes where the sign-problem can occur and how this problem can be solved. Then it shows the numerical results, compares the MC error estimates with the expectations from the analytic partition functions and contrasts the symmetrized quadrature rule error estimates with the MC error estimates. Finally is presents results of the partition function error estimates where we used a much larger machine precision in the computation than double precision to check whether in principle arbitrary machine precision can be reached. We used constant $\mu = 1$ and $L = 20$.

ANALYTIC RESULTS For all tested groups $\mathcal{SU}(N)$ and $\mathcal{U}(N)$, the partition function is small for small mass m and large for large m, see Figure 6.2 for $\mathcal{SU}(3)$ and $\mathcal{U}(3)$ exemplary. For small m the partition

function of $\mathcal{U}(N)$ is very small, for $\mathcal{U}(3)$ of $\mathcal{O}(10^{-20})$. In the same region the partition function of $\mathcal{SU}(N)$ is approximately $c_2^N + c_3^N$, due to the additional summand in $Z(\mathcal{SU}(N))$ in (6.46). Due to our choice of parameters, it is $c_2 \ll c_3$, compare their definitions in (6.44), and therefore in the following $c_2^N + c_3^N \approx c_3^N$ is used. In contrast to c_1 and c_2, the parameter c_3 is dependent on m, compare (6.44). For $m \gtrsim 1$, the additional summand of $Z(\mathcal{SU}(N))$, c_3^N, is not relevant in comparison to the summand c_1^N in the partition functions of all groups in (6.45). Therefore the partition functions for $\mathcal{SU}(N)$ and $\mathcal{U}(N)$ behave similarly in this region.

For the MC method it is almost impossible to result in the very small values of $Z(\mathcal{U}(N))$ for small m. This is due to the large order of magnitude of the values computed with MC. An MC value is computed by averaging over the integrand $\det(c_1 + c_2 U^\dagger + c_3 U)$, evaluated at different sampling points U, compare (6.48). Each integrand evaluation at one MC-step can result in a relatively large value. This value can be estimated by $|c_1|^N + |c_2|^N + |c_3|^N$ because the integrand, the determinant, is a polynomial of degree N and can be estimated by its term with the largest exponent which is $|c_1|^N + |c_2|^N + |c_3|^N \approx |c_1|^N + |c_3|^N$ for $U \in \mathcal{U}(N)$ or $\mathcal{SU}(N)$, since $|U| < 1$. This estimate of one integrand evaluation, $|c_1|^N + |c_3|^N$, is in the following called *integrand evaluation scale*. For small m this evaluation scale is approximately $|c_3|^N$ (because then $|c_1| \ll |c_3|$), which is shown in figure 6.2. For the average in (6.48) over many such single integrand evaluations of the determinant at $m \ll 1$ to give a result much smaller than the integrand evaluation scale $|c_3|^N$, as for $Z(\mathcal{U}(N))$, values of different sampling points need to cancel each other. For MC methods, where the matrices are chosen randomly, this is very unlikely to happen. On the other side, MC methods should have no problems with analytic values around or above the integrand evaluation scale $|c_1|^N + |c_3|^N$ and therefore for computations of $Z(\mathcal{U}(N))$ for $m \gtrsim 1$ and for $Z(\mathcal{SU}(N))$ in the full m-range.

The symmetrized quadrature rules in section 6.1.4 have the same integrand evaluation scale as MC, for each sampling point the determinant is evaluated. But because the sampling points are chosen symmetrically, evaluations of the determinant for different sampling points cancel each other and can, in contrast to MC, result in smaller averages over single evaluations than the integrand evaluation scale. For example, for $\mathcal{U}(1)$ it is $Z(\mathcal{U}(1)) = \int_{\mathcal{U}(1)} \mathrm{d}U \, (c_1 + c_2 U^\dagger + c_3 U)$ with analytic solution $Z(\mathcal{U}(1)) = c_1$, because $\int_{\mathcal{U}(1)} \mathrm{d}U \, c_2 U^\dagger = 0$ and $\int_{\mathcal{U}(1)} \mathrm{d}U \, c_3 U = 0$. In an MC computations with $m \ll 1$ and therefore $c_1 \ll c_3$ (as well as $c_2 \ll c_3$ as states before), each evaluation of the integrand $c_1 + c_2 U^\dagger + c_3 U$ gives a value in the vicinity $|c_3|$, which is large due to the factor $\mathrm{e}^{L\mu}$ inside it. A symmetrized quadrature rule, for simplicity only with two sampling points $\mathrm{e}^0 = 1$ and $\mathrm{e}^{i\pi} = -1$, would include evaluations at both c_3 and $-c_3$. These two evaluations

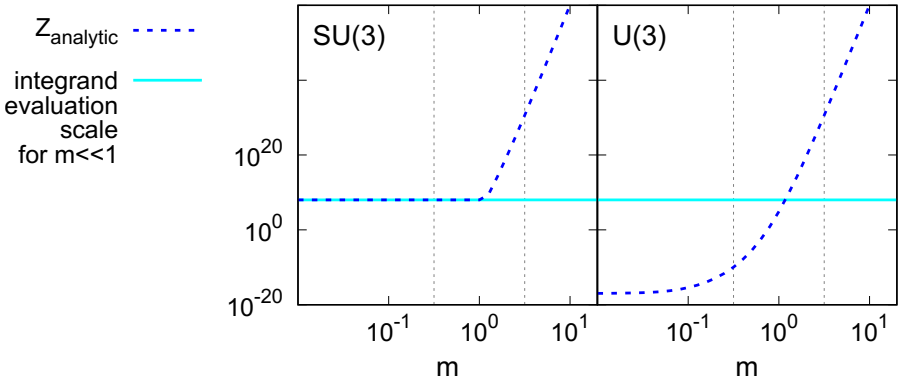


Figure 6.2: The partition functions of $\mathcal{SU}(N)$ and $\mathcal{U}(N)$ for constant μ and L differ in the small mass region $m \lesssim 1$, where $Z(\mathcal{U}(N))$ shrinks to very small values, while $Z(\mathcal{SU}(N))$ stays around the order of the integrand evaluation scale of the quadrature rules.

cancel each other if they are considered with the same weight, and therefore this symmetrized quadrature rule gives a much better estimate for $Z(\mathcal{U}(1))$ than MC methods.

NUMERICAL RESULTS We compared MC and symmetrized quadrature rule errors ΔZ, both computed via (6.49) and averaged over 50 independent computations for a standard deviation estimate. Our obtained error estimates can be explained by the analytic partition function shapes analyzed above and can be roughly split into a small m ($m < 10^{-0.5}$), a large m ($m > 10^{0.5}$) and a transition region, which are visualized in figures 6.3 and 6.2.

For small m, MC cannot simulate the very small partition function results of $\mathcal{U}(N)$, the error estimates are at least $\mathcal{O}(10^8)$, see bottom row in figure 6.3. In contrast, in the same region the error estimates for $\mathcal{SU}(N)$ partition functions, top row in figure 6.3, are much smaller, smaller than $\mathcal{O}(10^{-9})$, because in comparison to $Z(\mathcal{U}(N))$, the $\mathcal{SU}(N)$ partition function has an additional summand $|c_3|^N$, which is of the order of each MC-step evaluation.

For all tested groups the error estimates at large m are almost at machine precision, less than $\mathcal{O}(10^{-12})$. Here the partition function value is very large and therefore easy to approximate with integrand evaluations of the order $|c_1|^N + |c_3|^N$.

The $\mathcal{U}(N)$ error results show a monotone behavior. For $\mathcal{SU}(N)$ an error peak occurs around $m \sim 1$. At this mass the variance of $Z(\mathcal{SU}(N))$ is probably largest, but further investigations are needed here to fully understand this behavior [17].

The symmetrized quadrature rule error estimates are orders of magnitude smaller than the MC ones in the small m region, see figure 6.3. For $\mathcal{SU}(N)$ the error estimates are at machine precision over the full shown mass range and do not show a peak around $m \sim 1$. This is due to the symmetric and not fully random choice of the integration

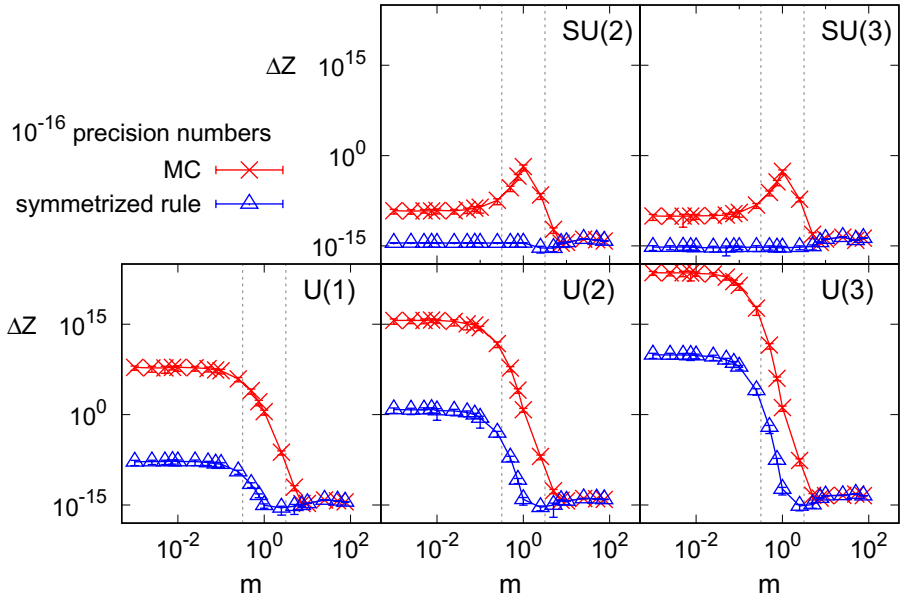


Figure 6.3: In the region $m \ll 1$ where the sign-problem occurs, the symmetrized quadrature rules result in much better partition function error estimates than MC quadrature rules for all shown groups.

points. In the small m region the errors of $Z(\mathcal{U}(N))$ are still orders of magnitude larger than 10^{-15}. This results from the fact that the symmetrized quadrature rules still sum values of the order of magnitude $|c_3|^N$ on machine precision, here double precision. The cancellation of values cannot resolve values below double precision times the order of magnitude of each integrand evaluation, $10^{-16} \cdot |c_3|^N$. Figure 6.3 shows that in the small m region the difference between MC and symmetrized quadrature rule error estimates for $\mathcal{U}(N)$ is around 10^{-16}.

USING 1024 BIT EXTENDED PRECISION NUMBERS We also used 1024-bit extended precision numbers, which have around $2^{-1024} \approx 10^{-310}$ machine precision, to check numerically that the symmetrized quadrature rules can in principle give arbitrary results.

For $\Delta Z(\mathcal{SU}(3))$ and $\Delta Z(\mathcal{U}(3))$ we used 10 independent measurements to estimate the standard deviation, due to the much larger computational effort for these two groups, for all other groups we used 50 independent measurements. The MC error estimates do not differ significantly from the double precision results, see Figure 6.4, because the integrand evaluation of order $|c_1|^N + |c_3|^N$ is responsible for the error. Only for large m the error decreases to much smaller values than possible with double precision. Here, the computed MC results get closer and closer to the analytic value because of the small machine precision and the large and growing analytic partition function.

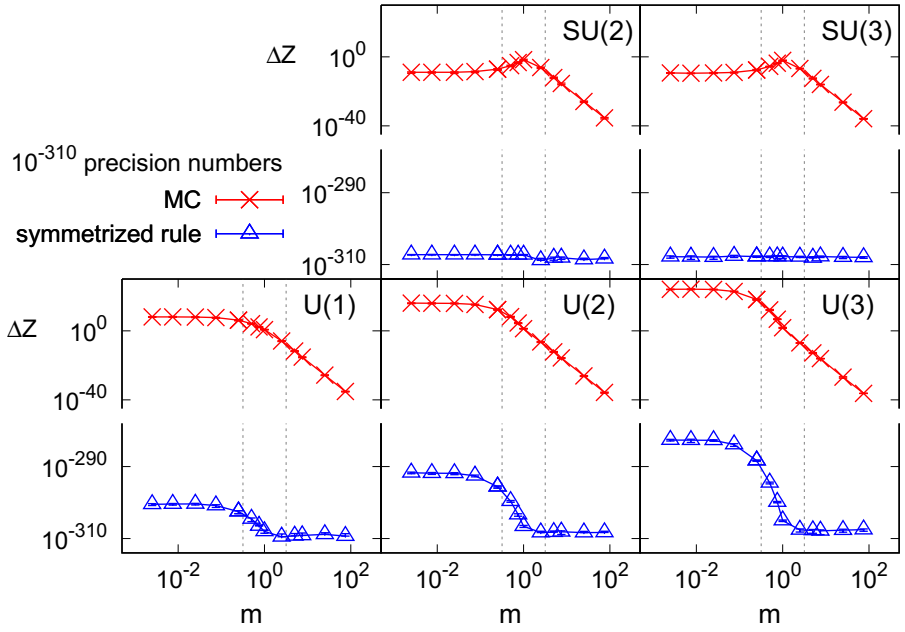


Figure 6.4: Also for 1024-bit extended precision numbers the error estimates from the symmetrized quadrature rules are around machine precision while the MC error estimates do not differ significantly from the MC error estimates computed with double precision.

The error estimates of the symmetrized quadrature rules are for all m and groups around machine precision, 10^{-310}. For $\mathcal{U}(N)$ and small masses, there is a slightly larger error estimate of maximally 10^{-290} measurable, which is again due to the fact that the polynomially exact quadrature rules cannot resolve values smaller than 10^{-310} times the integrand evaluation scale.

6.3.3 *The chiral condensate*

The chiral condensate is defined by $\chi = \partial_m Z / Z$. In all calculations we computed both numerator and denominator separately and finally divided them. This section first shows analytic results, where both ingredients, $\partial_m Z$ and Z, are analyzed and compared to each other to estimate possible problematic regions in MC simulations. Then it presents numerical results, here directly using 1024-bit precision numbers, compares the MC error estimates with the expectations from the analysis of the analytic results and compares the symmetrized quadrature rule error estimates with the MC error estimates. We used constant $\mu = 1$ and $L = 8$.

ANALYTIC RESULTS Because in the numerical experiments we computed both numerator and denominator of the chiral condensate separately, we analyzed both ingredients, $\partial_m Z$ and Z, analytically. We analytically calculated $\partial_m Z$ by symbolic differentiation of the analytic

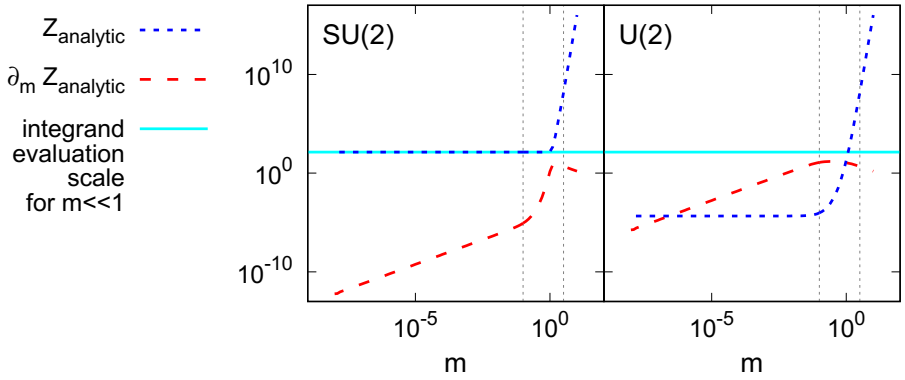


Figure 6.5: For $m \ll 1$ the $\mathcal{SU}(N)$ chiral condensate, $\chi = \partial_m Z/Z$, is mostly determined by its very small numerator, while for $\mathcal{U}(N)$ both numerator and denominator are smaller than the integrand evaluation scale of the quadrature rules.

forms in (6.45). The integrand evaluation of $\partial_m Z$ is of the same order as of Z, $|c_1|^N + |c_3|^N$.

For $m \ll 1$ and $\mathcal{SU}(N)$ we found that the numerator $\partial_m Z$ is much smaller, $\mathcal{O}(10^{-10})$, than the integrand evaluation scale, see figure 6.5 for $\mathcal{SU}(2)$ and $\mathcal{U}(2)$ exemplary, while Z, as seen before, is of the same order as the evaluation scale. Therefore in this region for MC it is very hard to compute $\partial_m Z$ and therefore also χ accurately. In the same region for $\mathcal{U}(N)$ both numerator and denominator are small compared to the integrand evaluation scale. It is likely that MC systematically overestimates both values and hence in the ratio these systematic effects can cancel. Therefore it can be possible to compute χ values with MC that are more accurate than the MC estimates of numerator and denominator by themselves.

For $m > 1$, $Z(\mathcal{U}(N))$ and $Z(\mathcal{SU}(N))$ grow very large and are therefore easy to simulate, while $\partial_m Z$ is smaller and decreases for larger m. Therefore, the accuracy of numerically computing χ depends here mostly on how accurate the numerator $\partial_m Z$ can be estimated.

NUMERICAL RESULTS We used again 1024-bit precision numbers and symbolically differentiated equation (6.44) to arrive at a formula for $\partial_m Z$, which is again dependent on U and can be integrated numerically. The numerical error estimates can be explained by the analytic values of Z and $\partial_m Z$ and can again be roughly split into a small m ($m < 10^{-1}$), a large m ($m > 10^{0.5}$) and a transition region, plotted in figures 6.6 and 6.5.

For MC computations in the small m region, $\Delta\chi(\mathcal{SU}(N))$ are larger than one, see figure 6.6, due to the very small $\partial_m Z$ values in this region, which are difficult to compute via MC. For $\mathcal{U}(N)$ the error estimates in this region are all of order one. Here analytic values of both numerator and denominator of χ are smaller than the integrand

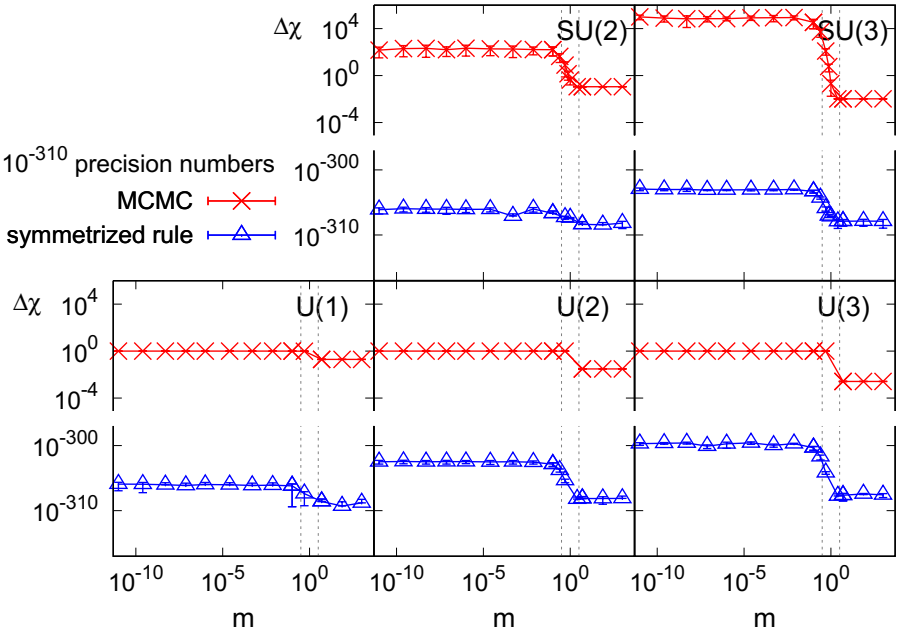


Figure 6.6: The symmetrized quadrature rule error estimates of the chiral condensate are approximately at machine precision for all shown groups, in contrast to the MC error estimates, which are of order one or even larger for $m \ll 1$ and are a result of the sign-problem.

evaluation scale but an overestimation of both with MC can possibly cancel in χ, as states above.

For large m all groups have a smaller MC error estimate than in the small m region, due to the very large and therefore easy to simulate analytic Z values. In this region the error estimates seem to stay constant for larger m, opposed to the ΔZ behaviors in figure 6.4, probably because of the small and decreasing $\partial_m Z$ values here.

Here again the symmetrized quadrature rules give error estimates approximately at machine precision up to very small m values, see figure 6.6.

6.4 CONCLUDING REMARKS

The numerical results show that the symmetrized quadrature rules are not only polynomially exact quadrature rules theoretically, but also give significant results in the sign-problem region in practice, where MC simulations have error estimates of the order one. In the sign-problem region MC cannot dissolve the small analytic values of the partition function and the chiral condensate. It is possible for the symmetrized quadrature rules to result in these small values because it uses sampling points which are symmetrically distributed on spheres and lead to cancellations of large values.

The symmetrized quadrature rules are rules for one integration variable and we applied them to the one-dimensional QCD model

that has only one independent variable. The next step is to generalize the method such that it is applicable to models with many variables.

7

SYMMETRIZED CUBATURE RULES FOR MORE-DIMENSIONAL INTEGRALS

In lattice field theory, parameter regions of specific models cannot be reached due to the sign-problem of the applied MCMC methods, as discussed in section 4.5.2. For example, this is the reason why the early universe cannot be simulated and therefore the transition between a quark gluon plasma and the confinement of quarks in hadrons is not fully understood. Chapter 6 presents polynomially exact, completely symmetrized quadrature rules, which avoid the sign-problem entirely and are applicable to gauge theories, among others also to $\mathcal{SU}(3)$, the gauge group of QCD. But the presented quadrature rules approximate only integrals with one integration variable. It is important to generalize these rules to models with more variables to avoid the sign-problem also there. We investigated two different cubature rules which generalize the completely symmetrized quadrature rules from chapter 6 to more variables. We restricted these two rules to $\mathcal{U}(1)$ variables even though a generalization to $\mathcal{U}(N)$ and $\mathcal{SU}(N)$ variables for $N \in \{2,3\}$ is in principle possible.

Our object of study was the topological osciallator, a one-dimensional model with $\mathcal{U}(1)$ variables and an artificially enforced sign-problem. The first cubature rule we tested, is a straightforward generalization of the completely symmetrized $\mathcal{U}(1)$ quadrature rule in chapter 6 and therefore called completely symmetrized cubature rule (CSCR). We developed a second cubature rule, which combines the sampling points of the completely symmetrized quadrature rule, called *symmetrization points*, with an MCMC simulation to make its computation feasible for a large number of variables in the model. Results are published in [59].

This chapter first explains both methods. It then presents an altered topological oscillator model with a complex phase which is dependent on the parameter θ and introduces a sign-problem. It shows numerical results of applying the two cubature rules to this model and finally discusses possible explanations for why they behave very differently.

For small numbers of variables we were able to compute errors with the CSCR that shrink exponentially with the number of symmetrization points. The method yields small error estimates for all values of $\theta > 0$, where MCMC has shortcomings due to the sign-problem. But the rule is disappointingly inefficient when applied to a model with many variables. Therefore, the CSCR avoids the sign-problem and gives a similar fast error scaling than obtained for the

one-variable completely symmetrized quadrature rule in chapter 6, but it is not feasible for applications to models with many variables.

The results of the combined method agree with MCMC results. Conversely, the error estimate grows for larger numbers of symmetrization points used. Additionally, for $\theta > 0$ the combined method gives even larger errors than already obtained for $\theta = 0$. It does not improve the MCMC error estimate and does not circumvent the sign-problem. A possible reason is, that the method uses for efficiency reasons only a portion of all possible combinations of symmetrization points, such that most desired cancellations of large fluctuations cannot occur. Therefore, other methods have to be developed to generalize the results of the polynomially exact, completely symmetrized quadrature rules of chapter 6 to more variables.

7.1 SYMMETRIZED CUBATURE RULES

In a model with d variables $\boldsymbol{U} = (U_1, \ldots, U_d)$ with $U_1, \ldots, U_d \in \mathcal{U}(1)$ and weight function $\rho(\boldsymbol{U})$ the expectation value of an observable $O(\boldsymbol{U})$ is the integral

$$\langle O \rangle = I(O, \rho) = \frac{I(O\rho)}{I(\rho)} = \frac{\int_{\mathcal{U}(1)^d} \mathrm{d}\boldsymbol{U}\, O(\boldsymbol{U})\rho(\boldsymbol{U})}{\int_{\mathcal{U}(1)^d} \mathrm{d}\boldsymbol{U}\, \rho(\boldsymbol{U})}. \tag{7.1}$$

In lattice field theory this integral is evaluated numerically. Chapter 6 presents a method to compute the integrals $I(O\rho)$ and $I(\rho)$ in a polynomially exact way, but only for models with one variable U. This can be interpreted as a special case of (7.1) with $d = 1$. In this case, the integral of the numerator in (7.1) can be approximated by the completely symmetrized quadrature rule

$$I(O\rho) = \int_{\mathcal{U}(1)} \mathrm{d}U\, O(U)\rho(U) \approx \frac{1}{\mathfrak{m}} \sum_{k=1}^{\mathfrak{m}} O(s_k)\rho(s_k) \tag{7.2}$$

with sampling points $s_k = \mathrm{e}^{\frac{2\pi i k}{\mathfrak{m}}}$, compare (6.3). Because these sampling points lie symmetric on the unit-circle and are used in chapter 6 to cancel large fluctuations, they are called *symmetrization points*. The denominator can be computed similarly. Almost all interesting models have more than one variable and therefore the completely symmetrized quadrature rule needs to be generalized to $d > 1$. This section presents such generalized cubature rules.

The quadrature rule in (7.2) can be generalized to more variables in a straightforward manner, resulting in the CSCR. Unfortunately, its number of required sampling points depends exponentially on the number of variables and is therefore unfeasible for a large number of variables. We developed an alternative method to be more efficient by combining the symmetrization points with an MCMC simulation. The MCMC simulation is used to approximate a slightly different

integral than the one that gives the expectation value. This slightly different integral involves the symmetrization points and its MCMC approximation gives therefore possibly smaller error estimates than standard MCMC simulations. The expectation value we seek can be recomputed from this *symmetrized integral* if the observable is a polynomial.

7.1.1 *The completely symmetrized cubature rule*

The one-variable quadrature rule of chapter 6, e.g. in (7.2), can be directly applied to the general case

$$I(f) = \int_{\mathcal{U}(1)^d} \mathrm{d}\boldsymbol{U}\, f(\boldsymbol{U}), \tag{7.3}$$

for $f = O\rho$ or $f = \rho$. A possible CSCR $Q(f)$ is the product rule, discussed in section 4.1: The integral in (7.3) is split into one-dimensional integrals

$$\int_{\mathcal{U}(1)} \mathrm{d}U_1 \int_{\mathcal{U}(1)} \mathrm{d}U_2 \ldots \int_{\mathcal{U}(1)} \mathrm{d}U_d$$

and the corresponding quadrature rule of (7.2) for $O\rho = f$ is applied to each of them:

$$Q^{\mathrm{CSCR}}(f) = \frac{1}{\mathfrak{m}^d} \sum_{k_1=1}^{\mathfrak{m}} \cdots \sum_{k_d=1}^{\mathfrak{m}} f(s_{k_1}, \ldots, s_{k_d}), \tag{7.4}$$

$$= \frac{1}{\mathfrak{m}^d} \sum_{\ell=1}^{\mathfrak{m}^d} f(\boldsymbol{t}_\ell). \tag{7.5}$$

This cubature rule needs $\mathfrak{m}^d$ sampling points $\boldsymbol{t}_\ell = (s_{k_1}, \ldots, s_{k_d})$, where $s_k = \mathrm{e}^{\frac{2\pi i k}{\mathfrak{m}}}$. Since the one-variable quadrature rule in (7.2) is polynomially exact, the CSCR in (7.5) is polynomially exact as well. To compute expectation values as shown in (7.1), CSCR is applied to numerator $I(O\rho)$ and denominator $I(\rho)$ separately.

As already mentioned in section 4.1, for integrals with $d \gg 0$, for example in the case of QCD, the number of required sampling points is $\mathfrak{m}^d$ and therefore the product rule in (7.5) cannot be efficiently computed this way.

7.1.2 *Combining symmetrization with MCMC*

We developed a different approach than the product rule to be able to apply the symmetrization points more efficiently to a large number of integration variables d. As discussed in chapter 4 MCMC methods are efficient for high-dimensional integrals, that means for a large number of integration variables. Therefore we combined the symmetrization points with an MCMC step.

A combination of symmetrization points and MCMC is used in [28, 29] and shortly explained in section 6.1. This method can be applied to a model with one variable U and a sign-problem. Our combined method differs from this approach and can best be explained in comparison to it. A sign-problem arises if the "weight" function ρ in (7.1) is complex. MCMC cannot be used with a complex weight function, therefore the method in [28, 29] chooses subsets Ω_U of the full phase space such that their combined weight $\sigma_{\Omega_U}(\rho) = \frac{1}{|\Omega_U|}\sum_{V\in\Omega_U}\rho(V)$ is real and positive. Each MCMC step generates one subsets with the probability distribution $\sigma_{\Omega_U}(\rho)/(\int \mathrm{d}U\,\sigma_{\Omega_U}(\rho))$ and the expectation value of an observable O is computed via the integral

$$I(O,\rho) = \int \mathrm{d}U\,\frac{\sigma_{\Omega_U}(\rho)}{\int \mathrm{d}U\,\sigma_{\Omega_U}(\rho)}\left(\frac{1}{\sigma_{\Omega_U}(\rho)|\Omega_U|}\sum_{V\in\Omega_U} O(V)\rho(V)\right). \tag{7.6}$$

We applied a slightly modified version to a model with more than one variable $\boldsymbol{U} = (U_1,\ldots,U_d) \in \mathcal{U}(1)^d$. We also used subsets $\Omega_{\boldsymbol{U}}$ of the phase space and applied MCMC to generate these subsets. But in contrast to the above described method we did not search for specific subsets for which the combined weight σ_{Ω_U} is positive. We used the method of reweighting discussed in section 4.5.2 to be able to apply MCMC to an integral with a complex weight $\rho = \varrho\omega \in \mathbb{C}$ with $\varrho \in \mathbb{R}$ and $\omega \in \mathbb{C}$. Here the complex phase factor ω is an oscillatory function. In reweighting, the complex part ω of the weight is handled as part of the observable and the expectation value of O is computed via $I(O,\rho) = \frac{I(O\omega,\varrho)}{I(\omega,\varrho)}$, compare (4.23). Therefore two integrals with real weight ϱ, one with observable function $O\omega$, the other one with ω, have to be evaluated. This means that the highly oscillatory part of the integrands of both integrals are the new observables $O\omega$ and ω. We averaged observables over subsets $\Omega_{\boldsymbol{U}}$, such that large fluctuations can cancel each other, e.g. $\frac{1}{|\Omega_{\boldsymbol{U}}|}\sum_{\boldsymbol{V}\in\Omega_{\boldsymbol{U}}} O(\boldsymbol{V})\omega(\boldsymbol{V})$. The specific subsets are drawn according to the real subset weight $\sigma_{\Omega_{\boldsymbol{U}}}(\varrho) = \frac{1}{|\Omega_{\boldsymbol{U}}|}\sum_{\boldsymbol{V}\in\Omega_{\boldsymbol{U}}}\varrho(\boldsymbol{V})$. This subset weight and averaged observable can be used to define the integral

$$I^{\mathrm{sym}}(O\omega,\varrho) \overset{\mathrm{def}}{=} \int \mathrm{d}\boldsymbol{U}\,\frac{\sigma_{\Omega_{\boldsymbol{U}}}(\varrho)}{\int \mathrm{d}\boldsymbol{U}\,\sigma_{\Omega_{\boldsymbol{U}}}(\varrho)}\left(\frac{1}{|\Omega_{\boldsymbol{U}}|}\sum_{\boldsymbol{V}\in\Omega_{\boldsymbol{U}}} O(\boldsymbol{V})\omega(\boldsymbol{V})\right), \tag{7.7}$$

which has a similar form as (7.6). In contrast to (7.6), the two factors of ρ, ϱ and ω, are separated such that each one occurs in a different part of (7.7). This means that even by setting $\omega = 1$ (and therefore $\rho = \varrho$) (7.7) differs from (7.6). We found that if $O\omega$ is a polynomial in the U_i variables, $I(O\omega,\varrho)$ can be recomputed from $I^{\mathrm{sym}}(O\omega,\varrho)$. Similarly $I(\omega,\varrho)$ can be recomputed from $I^{\mathrm{sym}}(\omega,\varrho)$. Therefore in the combined method we first estimated $I^{\mathrm{sym}}(O\omega,\varrho)$ and $I^{\mathrm{sym}}(\omega,\varrho)$ by

drawing subsets Ω_U from σ_{Ω_U} with MCMC. From these estimates we derive an estimate for $I(O\omega, \varrho)$ and $I(\omega, \varrho)$ to finally arrive at an estimate for the expectation value $\langle O \rangle = \frac{I(O\omega,\varrho)}{I(\omega,\varrho)}$. We used subsets Ω_U that include the symmetrization points of the completely symmetrized quadrature rule. Unfortunately, we found that this ansatz did not yield the desired precision and clearly still suffered from the MCMC typical sign-problem. This chapter proceeds to describe our findings in detail and leave it as an open question whether this method can be modified such that the sign-problem can be eliminated.

In the following (7.7) is written as as $I^{\text{sym}}(\tilde{O}, \varrho)$, where $\tilde{O}$ can be $O\omega$ or ω,

$$I^{\text{sym}}(\tilde{O}, \varrho) = \int \mathrm{d}\boldsymbol{U} \, \frac{\sigma_{\Omega_U}}{\int \mathrm{d}\boldsymbol{U} \, \sigma_{\Omega_U}} \tilde{O}_{\Omega_U} \tag{7.8}$$

with subset weight $\sigma_{\Omega_U} \overset{\text{def}}{=} \sigma_{\Omega_U}(\varrho)$,

$$\sigma_{\Omega_U} = \frac{1}{|\Omega_U|} \sum_{\boldsymbol{V} \in \Omega_U} \varrho(\boldsymbol{V}), \tag{7.9}$$

and subset observable

$$\tilde{O}_{\Omega_U} = \frac{1}{|\Omega_U|} \sum_{\boldsymbol{V} \in \Omega_U} \tilde{O}(\boldsymbol{V}). \tag{7.10}$$

This section describes in the following first how Ω_U is chosen in detail, then how $I(\tilde{O}, \varrho)$ can be derived from $I^{\text{sym}}(\tilde{O}, \varrho)$ and finally how a MCMC step is performed with the subset weight $\frac{\sigma_{\Omega_U}(\varrho)}{\int \mathrm{d}\boldsymbol{U} \, \sigma_{\Omega_U}(\varrho)}$.

CHOOSING SUBSETS The approach (7.6) for one variable $U \in \mathcal{SU}(3)$, explained in section 6.1, uses the subset $\Omega_U = \{\mathrm{e}^{\frac{2\pi i k}{3}} U, \mathrm{e}^{\frac{2\pi i k}{3}} U^\dagger : k \in \{1,2,3\}\}$. Generalized to more - here $\mathcal{U}(1)$ - variables $\boldsymbol{U} \in \mathcal{U}(1)^d$ and number of symmetrization points $\mathfrak{m}$ the subset is given by

$$\Omega_U = \{(\mathrm{e}^{\frac{2\pi i k_1}{\mathfrak{m}}} U_1, \mathrm{e}^{\frac{2\pi i k_2}{\mathfrak{m}}} U_2, \ldots, \mathrm{e}^{\frac{2\pi i k_d}{\mathfrak{m}}} U_d) : k_j \in \{1, \ldots, \mathfrak{m}\}\}. \tag{7.11}$$

This set includes $\mathfrak{m}^d$ vectors. We already saw in the completely symmetrized cubature rule in section 7.1.1 that it is not feasible to use $\mathfrak{m}^d$ points for large d. However, it is also plausible that every symmetrization point $s_k = \mathrm{e}^{\frac{2\pi i k}{\mathfrak{m}}}$ should be combined with every variable U_j at least once. Therefore, we chose $\mathfrak{m}$ different vectors $(s_{k_1} U_1, \ldots, s_{k_d} U_d)$ in such a way that every $s_k, k \in \{1, \ldots, \mathfrak{m}\}$ is assigned to some U_j exactly once: inspired by [62], we chose at each lattice point j a random permutation $P_j : (1, \ldots, \mathfrak{m}) \to (P_j(1), \ldots, P_j(\mathfrak{m}))$. To each variable U_j

we applied the symmetrization points $s_{P_j(\ell)}$, $\ell \in \{1, \dots \mathfrak{m}\}$. The resulting subset is dependent on the choice of the permutations at all lattice points $\mathscr{P} = (P_1, \dots, P_d)$,

$$\Omega_{\boldsymbol{U}}^{\mathscr{P}} = \{(s_{P_1(\ell)}U_1, s_{P_2(\ell)}U_2, \dots, s_{P_d(\ell)}U_d) : \ell \in \{1, \dots, \mathfrak{m}\}\}. \quad (7.12)$$

This means that at each lattice point all $\mathfrak{m}$ symmetrization points are taken into account, but lattice-wide only $\mathfrak{m}$ of all $\mathfrak{m}^d$ possible combinations of symmetrization points are used. To reduce the variance of the choice of permutations $\mathscr{P}$ we averaged over $N_{\text{sets}} \in \mathbb{N}$ different choices $\mathscr{P}_a$ for $\mathscr{P}$. Then, if subsets $\Omega_{\boldsymbol{U}}^{\mathscr{P}_a}$ are chosen according to the probability distribution $\pi_{\Omega_{\boldsymbol{U}}^{\mathscr{P}_a}} = \frac{\mathrm{d}\boldsymbol{U}\sigma_{\Omega_{\boldsymbol{U}}^{\mathscr{P}_a}}}{\int \mathrm{d}\boldsymbol{U}\,\sigma_{\Omega_{\boldsymbol{U}}^{\mathscr{P}_a}}}$, the MCMC cubature rule for $I^{\text{sym}}(\tilde{O}, \varrho)$ in (7.8) is given by

$$Q^{\text{sym}}(\tilde{O}, \varrho) = \frac{1}{N_{\text{sets}}} \sum_{a=1}^{N_{\text{sets}}} \frac{1}{n} \sum_{\substack{i=1 \\ \Omega_{\boldsymbol{U}_i}^{\mathscr{P}_a} \text{ with} \\ \text{probability} \\ \pi_{\Omega_{\boldsymbol{U}_i}^{\mathscr{P}_a}}}}^{n} \tilde{O}_{\Omega_{\boldsymbol{U}_i}^{\mathscr{P}_a}}. \quad (7.13)$$

In the following, the elements in the subset $\Omega_{\boldsymbol{U}}^{\mathscr{P}}$ in (7.12) are written using the vectors $\boldsymbol{s}_\ell^{\mathscr{P}} = (s_{P_1(\ell)}, s_{P_2(\ell)}, \dots, s_{P_d(\ell)})$ and $\boldsymbol{U} = (U_1, \dots, U_d)$ via

$$\Omega_{\boldsymbol{U}}^{\mathscr{P}} = \{\boldsymbol{s}_\ell^{\mathscr{P}} \boldsymbol{U} : \ell \in \{1, \dots, \mathfrak{m}\}\}. \quad (7.14)$$

Then $|\Omega_{\boldsymbol{U}}| = \mathfrak{m}$ and (7.9) and (7.10) can be written as

$$\sigma_{\Omega_{\boldsymbol{U}}^{\mathscr{P}}} = \frac{1}{\mathfrak{m}} \sum_{\ell=1}^{\mathfrak{m}} \varrho(\boldsymbol{s}_\ell^{\mathscr{P}} \boldsymbol{U}), \quad (7.15)$$

$$\tilde{O}_{\Omega_{\boldsymbol{U}}^{\mathscr{P}}} = \frac{1}{\mathfrak{m}} \sum_{\ell=1}^{\mathfrak{m}} \tilde{O}(\boldsymbol{s}_\ell^{\mathscr{P}} \boldsymbol{U}). \quad (7.16)$$

RELATE I_d WITH I_d^{SYM} Equation (7.13) is a possible cubature rule to approximate the symmetrized integral $I^{\text{sym}}(\tilde{O}, \varrho)$, but eventually the expectation value of some observable O should be computed, $\langle O \rangle = I(O, \rho) = \frac{I(O\omega, \varrho)}{I(\omega, \varrho)}$. Therefore, $I(\tilde{O}, \varrho)$ for $\tilde{O} \in \{O\omega, \omega\}$ is needed. We found that if $\tilde{O}$ is a monomial in the variables U_j, $j \in \{1, \dots, d\}$, the integrals $I^{\text{sym}}(\tilde{O}, \varrho)$ and $I(\tilde{O}, \varrho)$ can be related via

$$I^{\text{sym}}(\tilde{O}, \varrho) = c\, I(\tilde{O}, \varrho), \quad (7.17)$$

with some constant $c \in \mathbb{C}$. Indeed, consider the monomial

$$\tilde{O}(\boldsymbol{U}) = \prod_{t=1}^{d} U_t^{\alpha_t}, \quad (7.18)$$

where $\alpha_i \in \mathbb{N}_0$ determines the exponent of each U_i. We also view the exponents as a vector $\boldsymbol{\alpha} \in \mathbb{N}_0^d$. By using the definitions (7.15) and (7.16), the integral $I^{\mathrm{sym}}(\tilde{O}, \varrho)$ in (7.8) is given by

$$I^{\mathrm{sym}} = \frac{\int \mathrm{d}\boldsymbol{U}\, \sigma_{\Omega_U} \tilde{O}_{\Omega_U}}{\int \mathrm{d}\boldsymbol{U}\, \sigma_{\Omega_U}} \tag{7.19}$$

$$= \frac{\int \mathrm{d}\boldsymbol{U}\, \left(\sum_{\ell=1}^{\mathfrak{m}} \varrho(\boldsymbol{s}_\ell^{\mathscr{P}} \boldsymbol{U})\right) \left(\sum_{k=1}^{\mathfrak{m}} \tilde{O}(\boldsymbol{s}_k^{\mathscr{P}} \boldsymbol{U})\right)}{\mathfrak{m} \int \mathrm{d}\boldsymbol{U}\, \left(\sum_{r=1}^{\mathfrak{m}} \varrho(\boldsymbol{s}_r^{\mathscr{P}} \boldsymbol{U})\right)} \tag{7.20}$$

For brevity $\mathscr{P}$ is in the following not shown. Numerator and denominator can be manipulated separately by using the properties of the Haar measure. In the numerator the product of two sums can be split into two parts, one part with the same index in both sums, $k = l$, and the other part with the rest,

$$N(I^{\mathrm{sym}}) = \int \mathrm{d}\boldsymbol{U} \sum_{\ell=1}^{\mathfrak{m}} \tilde{O}(\boldsymbol{s}_\ell \boldsymbol{U}) \varrho(\boldsymbol{s}_\ell \boldsymbol{U}) + \int \mathrm{d}\boldsymbol{U} \sum_{\substack{\ell,k=1 \\ \ell \neq k}}^{\mathfrak{m}} \tilde{O}(\boldsymbol{s}_\ell \boldsymbol{U}) \varrho(\boldsymbol{s}_k \boldsymbol{U}). \tag{7.21}$$

Due to the properties of the Haar-measure it is

$$N(I^{\mathrm{sym}}) = \mathfrak{m} \int \mathrm{d}\boldsymbol{U}\, \tilde{O}(\boldsymbol{U}) \varrho(\boldsymbol{U}) + \int \mathrm{d}\boldsymbol{U} \sum_{\substack{\ell,k=1 \\ \ell \neq k}}^{\mathfrak{m}} \tilde{O}(\boldsymbol{s}_\ell \boldsymbol{s}_k^{-1} \boldsymbol{U}) \varrho(\boldsymbol{U}). \tag{7.22}$$

The last step can be applied to the denominator as well,

$$D(I^{\mathrm{sym}}) = \mathfrak{m}^2 \int \mathrm{d}\boldsymbol{U}\, \varrho(\boldsymbol{U}). \tag{7.23}$$

Then numerator and denominator combine to

$$I^{\mathrm{sym}} = \frac{N(I^{\mathrm{sym}})}{D(I^{\mathrm{sym}})} = \frac{1}{\mathfrak{m}} \frac{\int \mathrm{d}\boldsymbol{U}\, \tilde{O}(\boldsymbol{U})\, \varrho(\boldsymbol{U})}{\int \mathrm{d}\boldsymbol{U}\, \varrho(\boldsymbol{U})} + \frac{1}{\mathfrak{m}^2} \sum_{\substack{\ell,k=1 \\ \ell \neq k}}^{\mathfrak{m}} \frac{\int \mathrm{d}\boldsymbol{U}\, \tilde{O}(\boldsymbol{s}_\ell \boldsymbol{s}_k^{-1} \boldsymbol{U}) \varrho(\boldsymbol{U})}{\int \mathrm{d}\boldsymbol{U}\, \varrho(\boldsymbol{U})}. \tag{7.24}$$

The first term is $I(\tilde{O}, \varrho)$ times $1/\mathfrak{m}$, compare (7.1). The second term can be manipulated using the monomial form of the observable defined in (7.18),

$$I^{\mathrm{sym}} = \frac{1}{\mathfrak{m}} I(\tilde{O}, \varrho) + \frac{1}{\mathfrak{m}^2} \sum_{\substack{\ell,k=1 \\ \ell \neq k}}^{\mathfrak{m}} \frac{\int \mathrm{d}\boldsymbol{U} \prod_{t=1}^{d} \left(s_{P_t(\ell)} s_{P_t(k)}^{-1} U_l\right)^{\alpha_t} \varrho(\boldsymbol{U})}{\int \mathrm{d}\boldsymbol{U}\, \varrho(\boldsymbol{U})}. \tag{7.25}$$

This can be simplified further, splitting first the product over t in two products, then applying again the definition of the field O in (7.18) and finally recognizing another $I(\tilde{O}, \varrho)$,

$$I^{\text{sym}} = \frac{1}{\mathfrak{m}} I(\tilde{O}, \varrho) + \frac{1}{\mathfrak{m}^2} \sum_{\substack{\ell,k=1 \\ \ell \neq k}}^{\mathfrak{m}} \prod_{t=1}^{d} \left(s_{P_t(\ell)} s_{P_t(k)}^{-1} \right)^{\alpha_t} \frac{\int \mathrm{d}\boldsymbol{U} \prod_{p=1}^{d} U_p^{\alpha_p} \varrho(\boldsymbol{U})}{\int \mathrm{d}\boldsymbol{U}\, \varrho(\boldsymbol{U})} \tag{7.26}$$

$$= \frac{1}{\mathfrak{m}} I(\tilde{O}, \varrho) + \frac{1}{\mathfrak{m}^2} \sum_{\substack{\ell,k=1 \\ \ell \neq k}}^{\mathfrak{m}} \prod_{t=1}^{d} \left(s_{P_t(\ell)} s_{P_t(k)}^{-1} \right)^{\alpha_t} \frac{\int \mathrm{d}\boldsymbol{U}\, \tilde{O}(\boldsymbol{U}) \varrho(\boldsymbol{U})}{\int \mathrm{d}\boldsymbol{U}\, \varrho(\boldsymbol{U})} \tag{7.27}$$

$$= I(\tilde{O}, \varrho) \left(\frac{1}{\mathfrak{m}} + \frac{1}{\mathfrak{m}^2} \sum_{\substack{\ell,k=1 \\ \ell \neq k}}^{\mathfrak{m}} \prod_{t=1}^{d} \left(s_{P_t(\ell)} s_{P_t(k)}^{-1} \right)^{\alpha_t} \right). \tag{7.28}$$

If the factor

$$c_{\mathscr{P}}(\tilde{O}) \stackrel{\text{def}}{=} \frac{1}{\mathfrak{m}} + \frac{1}{\mathfrak{m}^2} \sum_{\substack{\ell,k=1 \\ \ell \neq k}}^{\mathfrak{m}} \prod_{t=1}^{d} \left(s_{P_t(\ell)} s_{P_t(k)}^{-1} \right)^{\alpha_t}, \tag{7.29}$$

is non-zero, $I(\tilde{O}, \varrho)$ can be derived from the symmetrized integral $I^{\text{sym}}(\tilde{O}, \varrho)$ via

$$I(\tilde{O}, \varrho) = \frac{1}{c_{\mathscr{P}}(\tilde{O})} I^{\text{sym}}(\tilde{O}, \varrho). \tag{7.30}$$

Then based on (7.13) the cubature rule to approximate $I(\tilde{O}, \varrho)$ is given by

$$Q(\tilde{O}, \varrho) = \frac{1}{N_{\text{sets}}} \sum_{a=1}^{N_{\text{sets}}} \frac{1}{n} \sum_{\substack{i=1 \\ \Omega_{U_i}^{\mathscr{P}_a} \text{ with} \\ \text{probability} \\ \pi_{\Omega_{U_i}^{\mathscr{P}_a}}}}^{n} \frac{\tilde{O}_{\Omega_{U_i}^{\mathscr{P}_a}}}{c_{\mathscr{P}_a}(\tilde{O})}. \tag{7.31}$$

This deviation is also valid if the exponents in (7.18) are real numbers, $\alpha_i \in \mathbb{R}$ and if $\tilde{O}$ includes an additional constant factor. It should be noted that the correction factor $c_{\mathscr{P}}(\tilde{O})$ is independent of the lattice variables U_i. Additionally, $c_{\mathscr{P}}(\tilde{O})$ in (7.29) can be redefined without a factor $\frac{1}{\mathfrak{m}}$ if also the subset weight and observable in (7.15) and (7.16) are defined without the $\frac{1}{\mathfrak{m}}$ factor. This factor cancels in (7.31).

It is not only possible to compute with this method expectation values of observables which are monomials in the involved variables, but also polynomial observables are possible: if $\tilde{P}$ is a polynomial of the form

$$\tilde{P}(\boldsymbol{U}) = \sum_{v=1}^{d} \tilde{O}_v(\boldsymbol{U}) = \sum_{v=1}^{d} \prod_{t=1}^{d} U_t^{\alpha(v)_t}, \tag{7.32}$$

with different exponent vectors $\boldsymbol{\alpha}(v)$ for each summand, each monomial $\tilde{O}_v$ of the form (7.18) has its own correction factor $c_{\mathscr{P}_a}(\tilde{O}_v)$. Then the combined cubature rule for an integral $I(\tilde{P}, \varrho)$ is given by

$$Q^{\text{poly}}(\tilde{P}, \varrho) = \frac{1}{N_{\text{sets}}} \sum_{a=1}^{N_{\text{sets}}} \frac{1}{n} \sum_{\substack{i=1 \\ \Omega_{U_i}^{\mathscr{P}_a} \text{ with} \\ \text{probability} \\ \pi_{\Omega_{U_i}^{\mathscr{P}_a}}}}^{n} \sum_{v=l}^{d} \frac{\tilde{O}_{v,\Omega_{U_i}^{\mathscr{P}_a}}}{c_{\mathscr{P}_a}(\tilde{O}_v)}. \tag{7.33}$$

USING MCMC TO COMPUTE Q^{SYM} The cubature rule in (7.31) approximates the integral $I(\tilde{O}, \varrho)$ by using MCMC. Here subsets $\Omega_{\boldsymbol{U}}^{\mathscr{P}}$ are chosen from the probability distribution

$$\pi_{\Omega_{\boldsymbol{U}}^{\mathscr{P}}} = \frac{\mathrm{d}\boldsymbol{U}\, \sigma_{\Omega_{\boldsymbol{U}}^{\mathscr{P}}}}{\int \mathrm{d}\boldsymbol{U}\, \sigma_{\Omega_{\boldsymbol{U}}^{\mathscr{P}}}},$$

dependent on the permutation set $\mathscr{P}$. We used the Metropolis MCMC algorithm, discussed in section 4.4, to draw sampling points from the probability distribution $\pi_{\Omega_{\boldsymbol{U}}^{\mathscr{P}}}$. One Metropolis step updates one link variable U_j at lattice point j and consists of three sub-steps:

a. At lattice point j we chose

$$U_j^{\text{new}} = U_j^{\text{old}} \cdot \mathrm{e}^{i\pi r}, \tag{7.34}$$

with a uniform random number $r \in [-1, 1)$.

b. Then we computed the symmetrized weight of the old and new variable,

$$\sigma_{\Omega_U}^{\mathscr{P}\text{old}} = \frac{1}{\mathfrak{m}} \sum_{\ell=1}^{\mathfrak{m}} \varrho(s_{P_1(\ell)} U_1, \ldots, s_{P_j(\ell)} U_j^{\text{old}}, \ldots, s_{P_d(\ell)} U_d), \tag{7.35}$$

$$\sigma_{\Omega_U}^{\mathscr{P}\text{new}} = \frac{1}{\mathfrak{m}} \sum_{\ell=1}^{\mathfrak{m}} \varrho(s_{P_1(\ell)} U_1, \ldots, s_{P_j(\ell)} U_j^{\text{new}}, \ldots, s_{P_d(\ell)} U_d). \tag{7.36}$$

c. Finally we accepted U_j^{new} if

$$\frac{\sigma_{\Omega_U}^{\mathscr{P}\text{new}}}{\sigma_{\Omega_U}^{\mathscr{P}\text{old}}} > r, \tag{7.37}$$

with a uniform random number $r \in [0, 1)$.

Step b seems very time intensive. This is problematic because the Metropolis step is repeated very often in the simulation. But for a local real weight ϱ the subset weight $\sigma_{\Omega_{\boldsymbol{U}}^{\mathscr{P}}}$ does not need to be fully recomputed for every U_j^{new}. In a one-dimensional pure gauge model

one variable U_j couples only to its nearest neighbors U_{j+1} and U_{j-1}. Therefore the weight has the form $\varrho(\boldsymbol{U}) = \mathrm{e}^{-S(\boldsymbol{U})} = \mathrm{e}^{-\sum_{k=1}^{d} S_k(U_k, U_{k+1})}$ with action S. If one U_j changes, only two terms, S_j and S_{j-1}, are changing. Therefore the new action can be computed using the old action and two local action changes

$$S(\boldsymbol{U})^{\text{new}} = S(\boldsymbol{U})^{\text{old}} - (S_j^{\text{old}} + S_{j-1}^{\text{old}}) + (S_j^{\text{new}} + S_{j-1}^{\text{new}}) \,. \tag{7.38}$$

The same is true for the symmetrized actions $S(\boldsymbol{s}_\ell \boldsymbol{U})^{\text{new}}, \ell \in \{1, ..., \mathfrak{m}\}$,

$$\begin{aligned} S(\boldsymbol{s}_\ell \boldsymbol{U})^{\text{new}} = S(\boldsymbol{s}_\ell \boldsymbol{U})^{\text{old}} & - S_j(s_{P_j(\ell)} U_j^{\text{old}}, s_{P_{j+1}(\ell)} U_{j+1}) \\ & - S_{j-1}(s_{P_{j-1}(\ell)} U_{j-1}, s_{P_j(\ell)} U_j^{\text{old}}) \\ & + S_j(s_{P_j(\ell)} U_j^{\text{new}}, s_{P_{j+1}(\ell)} U_{j+1}) \\ & + S_{j-1}(s_{P_{j-1}(\ell)} U_{j-1}, s_{P_j(\ell)} U_j^{\text{new}}). \end{aligned} \tag{7.39}$$

The symmetrized weight is given by

$$\sigma_{\Omega_U}^{\mathscr{P}\text{new}} = \frac{1}{\mathfrak{m}} \sum_{\ell=1}^{\mathfrak{m}} \exp\big(- S(\boldsymbol{s}_\ell \boldsymbol{U})^{\text{new}} \big). \tag{7.40}$$

For all symmetrization indices ℓ in (7.39) we computed and saved $S(\boldsymbol{s}_\ell \boldsymbol{U})^{\text{old}}$ once at the start. Then for each Metropolis step we computed $S(\boldsymbol{s}_\ell \boldsymbol{U})^{\text{new}}$ using (7.39) for all ℓ, saved them in an array and computed $\sigma_{\Omega_U}^{\mathscr{P}\text{new}}$ via (7.40). If U_j^{new} is accepted, all $S(\boldsymbol{s}_\ell \boldsymbol{U})^{\text{old}}$ are overwritten by $S(\boldsymbol{s}_\ell \boldsymbol{U})^{\text{new}}$.

7.2 THE TOPOLOGICAL OSCIALLATOR WITH A COMPLEX PHASE

The topological oscillator describes a particle moving along a circle in time and is discussed in section 5.3. Its Euclidean action in (5.30) depends on the angle ϕ_t at timestep or lattice site t of the particle and can also be written in terms of $\mathcal{U}(1)$ variables $U_t = \mathrm{e}^{i\phi_t}$,

$$S^e(\phi) = \frac{I}{a} \sum_{t=1}^{d} (1 - \cos(\phi_{t+1} - \phi_t)), \tag{7.41}$$

$$S^e(\boldsymbol{U}) = \frac{I}{a} \sum_{t=1}^{d} \Re(1 - U_{t+1} U_t^*). \tag{7.42}$$

The number of integration variables is d, the number of lattice sites. The weight function is given by $\varrho(\boldsymbol{U}) = \mathrm{e}^{-S^e(\boldsymbol{U})}$.

We investigated the link correlation observable,

$$\mathscr{L}(\boldsymbol{U}) = \frac{1}{d} \sum_{t=1}^{d} U_{t+1} U_t^* \tag{7.43}$$

by computing its expectation value $\langle \mathscr{L} \rangle = I(\mathscr{L}, \rho)$. This function $\mathscr{L}$ is a polynomial in the U_t's. Each of its summands is therefore a

monomial of the form (7.18), required to relate the expectation value to symmetrized integrals.

We introduced a complex phase factor $\omega \in \mathbb{C}$ to the model such that integrals involving $\rho = \varrho\omega$ are complex and a sign-problem can occur in the evaluation of these integrals. In reweighting, ω is part of the observable. Therefore, to apply the combined method, ω has to be a polynomial. We chose a particular form that is similar to (7.18),

$$\omega(\boldsymbol{U}) = \mathrm{e}^{-i\theta\sum_{t=1}^{d}\phi_t} = \prod_{t=1}^{d} U_t^{-\theta}, \tag{7.44}$$

with an angle $\theta \in [0, 2\pi)$, a parameter of the model. This form is a slightly modified version of the complex phase factor used in the topological oscillator model in [25].

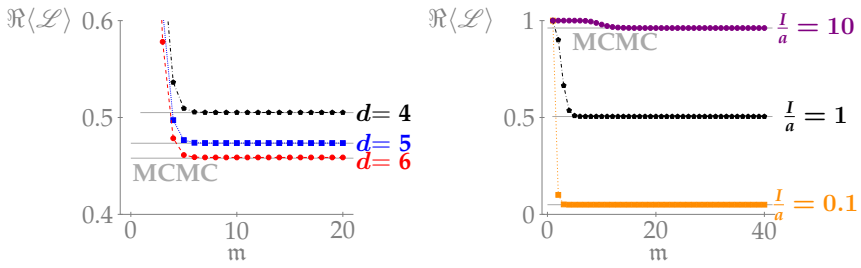
7.3 NUMERICAL RESULTS

Section 7.1 describes two methods that use symmetrization points in order to improve the error scaling and to avoid the sign-problem for models with more than one variable. Their applicability, error scaling and the occurrence of the sign-problem have to be tested in practice. We computed the link correlation expectation value of the topological oscillator model, with and without the complex phase factor. This section presents the numerical results of applying both methods, first the CSCR, then the method combining symmetrization points and MCMC, to the topological oscillator.

For small number of variables we were able to compute errors with CSCR that shrink exponentially when increasing the number of symmetrization points $\mathfrak{m}$. Without a complex phase factor our computations resulted in machine precision error estimates. For a complex phase factor with $\theta \in [0.0001, 1] \cdot 2\pi$ the method resulted in orders of magnitude smaller error estimates than the standard MCMC method. On the other hand we used only small lattices with up to six lattice points and the method is unfeasible when going to larger number of points.

We found that the combined method, using symmetrization points in combination with MCMC steps, results in values comparable to MCMC results, but with very large error estimates. The error estimate even rises with $\sqrt{\mathfrak{m}}$ and therefore much more statistics is needed here.

All in all, the CSCR is very well suited to avoid the sign-problem, but only for very small lattices, and therefore is unfeasible when applied to larger dimensional models. The combined method does not improve the error estimates of MCMC and therefore cannot be used to avoid the sign-problem. Possible explanation why the method performs poorly are given in section 7.4.



(a) $I/a = 1$ (b) $d = 4$

Figure 7.1: Not many symmetrization points $\mathfrak{m}$ are needed for the CSCR to get results consistent with MCMC simulations for small lattice sizes d, some different couplings I/a and without a complex phase factor.

7.3.1 *Applying the completely symmetrized cubature rule*

We applied the CSCR described in section 7.1.1 to the topological oscillator. We computed the expectation value of the link correlation by applying the rule to numerator and denominator separately, $\langle \mathscr{L} \rangle = I(\mathscr{L}, \rho) = \frac{I(\mathscr{L}\rho)}{I(\rho)} \approx \frac{Q^{\text{CSCR}}(\mathscr{L}\rho)}{Q^{\text{CSCR}}(\rho)}$, for different numbers of symmetrization points $\mathfrak{m}$ used. The error of the rule is estimated by a truncation error, comparing the results for different $\mathfrak{m}$ to a result with a large, constant $\mathfrak{m}_g$,

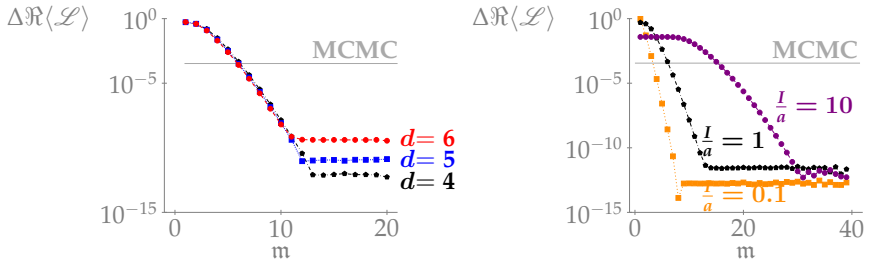
$$\Delta\langle \mathscr{L} \rangle(\mathfrak{m}) = |\langle \mathscr{L} \rangle(\mathfrak{m}) - \langle \mathscr{L} \rangle(\mathfrak{m}_g)|. \tag{7.45}$$

We compared the CSCR results with results from the Cluster MCMC algorithm [83], using 10^6 sampling points. We used reweighting to apply the Cluster algorithm to a complex integrand and applied the statistical bootstrap resampling method describe e.g. in [56] to compute error estimates for the MCMC results.

This section shows results first without and then with a complex phase factor.

WITHOUT COMPLEX PHASE FACTOR The CSCR gives link correlation results which are comparable to MCMC results for $\mathfrak{m} \gtrsim 7$. We used lattices with $d \in \{4, 5, 6\}$ sites, coupling $I/a = 1$ on the left of Figure 7.1, and $d = 4$, $I/a = \{0.1, 1, 10\}$ on the right of Figure 7.1.

The corresponding truncation errors are computed via (7.45) for $m_g = 30$ for constant $I/a = 1$ and $m_g = 40$ for constant $d = 4$ and are shown in Figure 7.2. For $\mathfrak{m} \sim \mathcal{O}(10)$ the truncation error shrinks exponentially until it reaches machine precision. CSCR needs less sampling points to reach the same error estimate as the MCMC method with 10^6 sampling points: with $I/a = 1$ the lattice with six sites needs $\mathfrak{m} \approx 6$ and therefore around $6^6 \approx 5 \cdot 10^4$ sampling points. Even a large factor of $I/a = 10$ with four lattice sites needs $\mathfrak{m} \approx 15$ and therefore around $15^4 \approx 5.1 \cdot 10^4$ sampling points. Both numbers are



(a) $I/a = 1$ (b) $d = 4$

Figure 7.2: The truncation error of the CSCR shrinks exponentially until it reaches a plateau at machine precision when applied to the topological oscillator without a complex phase factor.

smaller than 10^6 MCMC sampling points. This advantage of CSCR over MCMC is of course only valid for very small number of lattice sites used.

WITH COMPLEX PHASE FACTOR Applying the CSCR to the topological oscillator with a complex phase factor and $d = 4$, $I/a = 1$ gives much more precise results than MCMC, especially for $\theta \geq 0.3 \cdot 2\pi$, see Figure 7.3 which includes error bars for both methods. For $\theta \gtrsim 0.3 \cdot 2\pi$ the MCMC algorithm results in large error estimates, showing the sign-problem. For the combined method we used more θ-values and $m = 80$. The resulting $\Re\langle\mathscr{L}\rangle$ values in figure 7.3 show large fluctuations for different θ-values due to the choice of the complex phase factor ω in (7.44). CSCR and standard MCMC results agree with each other but the truncation errors of the CSCR are much smaller such that they are not visible in the figure.

The truncation errors are computed via (7.45) with $m_g = 100$ and their behaviors dependent on $\mathfrak{m}$ are shown in Figure 7.4. Similar to the case with $\theta = 0$, compare figure 7.2, also here the error shrinks exponentially for $\mathfrak{m} \gtrsim 30$ but with a smaller exponent, such that for $\mathfrak{m} < 90$ machine precision is not yet reached. The size of the error for the different θ-values depends on the value of $\Re\langle\mathscr{L}\rangle$ shown in figure 7.3.

Figure 7.5 shows that this scaling behavior is already apparent for very small θ-values around $\theta \geq 10^{-4} \cdot 2\pi$.

7.3.2 *Applying the combined cubature rule*

In the combined method the expectation value of the link correlation is computed via

$$\langle\mathscr{L}\rangle = I(\mathscr{L}, \omega\varrho) = \frac{I(\mathscr{L}\omega, \varrho)}{I(\omega, \varrho)} \approx \frac{Q^{\text{poly}}(\mathscr{L}\omega, \varrho)}{Q(\omega, \varrho)}. \tag{7.46}$$

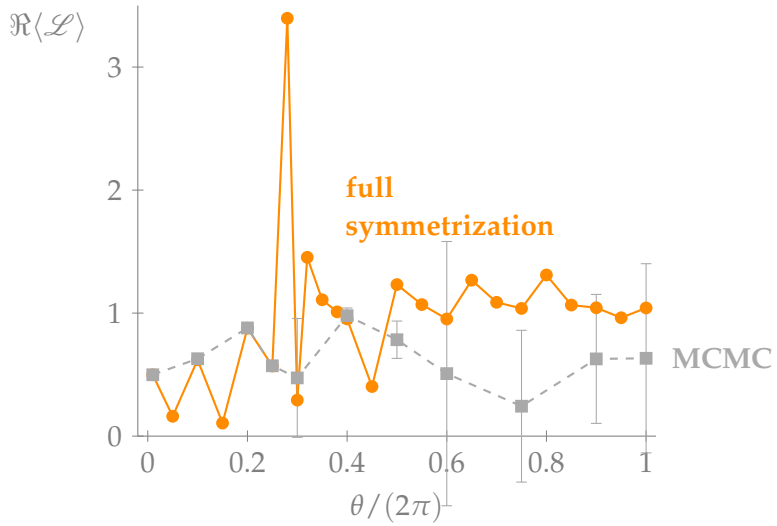


Figure 7.3: The CSCR gives much more precise results than MCMC simulations when including a complex phase factor such that the CSCR truncation errors are not visible in this figure. The fluctuations of $\Re\langle\mathscr{L}\rangle$ come from the choice of the complex phase factor.

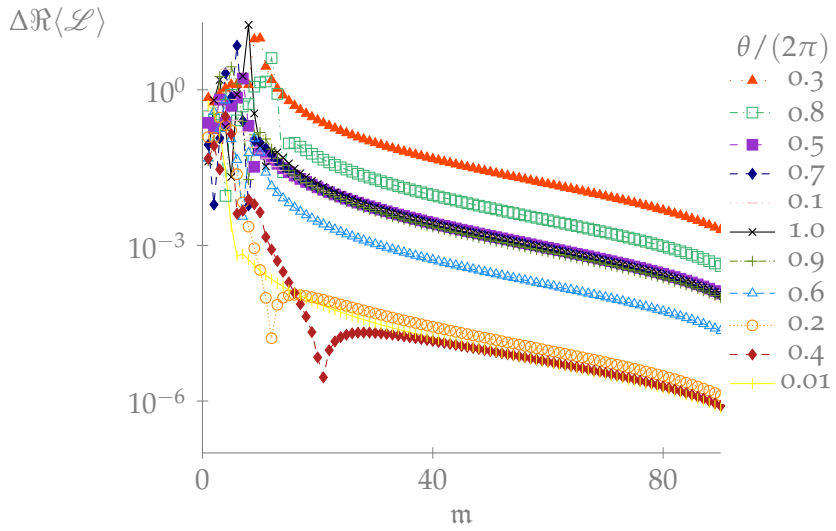


Figure 7.4: The truncation error of the CSCR applied to the topological oscillator with a complex phase factor scales exponentially for large enough $\mathfrak{m}$. Note that the θ-values in the legend appear in the order of the error size at $\mathfrak{m} = 80$.

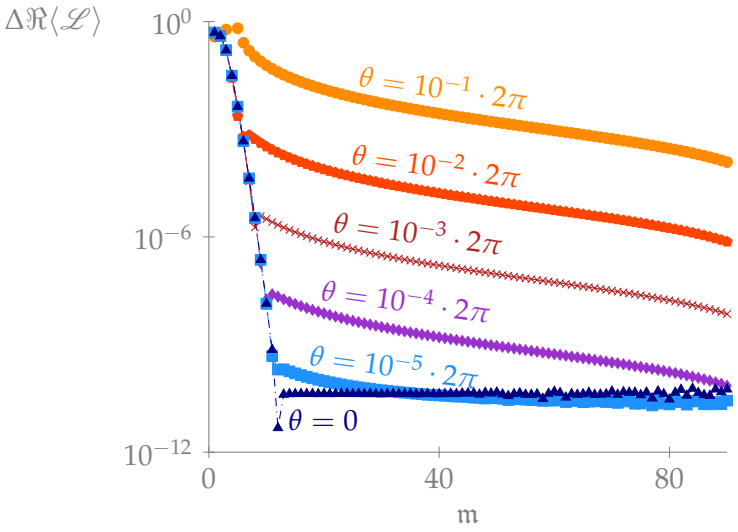


Figure 7.5: Already for very small θ-values, around $\theta \geq 10^{-4} \cdot 2\pi$, the different error scaling behavior in comparison to the case $\theta = 0$ is visible for CSCR.

The link correlation observable in (7.43) is a polynomial, therefore in the nominator (7.33) is used, while in the denominator (7.31) is used. An error estimate is computed via the statistical bootstrap resampling method.

The correction factors involved are dependent on the exponent vector α of the observable monomials, compare (7.18) and (7.32), here $\mathscr{L}_v\omega$ and ω, with $\mathscr{L}_v = \frac{1}{d}U_{v+1}U_v^*$. For $\omega = 1$ the exponent vector of $\mathscr{L}_v$ is given by $\alpha(v)_v = -1$, $\alpha(v)_{v+1} = 1$ and $\alpha(v)_k = 0$ for $k = \{m \in \{1,\dots,d\} : m \neq t, t+1\}$. Plugging this in (7.29) gives the correction factor

$$c_{\mathscr{P}}(\mathscr{L}_v) = \frac{1}{\mathfrak{m}} + \frac{1}{\mathfrak{m}^2} \sum_{\substack{\ell,k=1 \\ \ell \neq k}}^{\mathfrak{m}} \left(s_{P_v(\ell)}(s_{P_v(k)})^{-1}\right)^{-1} \left(s_{P_{v+1}(\ell)}(s_{P_{v+1}(k)})^{-1}\right). \tag{7.47}$$

For general ω the exponents of $\mathscr{L}_v\omega$ are given by $\alpha(v)_v = -1-\theta$, $\alpha(v)_{v+1} = 1-\theta$ and $\alpha(v)_k = -\theta$ for $k = \{m \in \{1,\dots,d\} : m \neq t, t+1\}$. Therefore the correction factor is given by

$$c_{\mathscr{P}}(\mathscr{L}_v\omega) = \frac{1}{\mathfrak{m}} + \frac{1}{\mathfrak{m}^2} \sum_{\substack{\ell,k=1 \\ \ell \neq k}}^{\mathfrak{m}} \left(s_{P_v(\ell)}(s_{P_v(k)})^{-1}\right)^{-1} \left(s_{P_{v+1}(\ell)}(s_{P_{v+1}(k)})^{-1}\right) \cdot \cdot \prod_{t=1}^{d} \left(s_{P_t(\ell)}(s_{P_t(k)})^{-1}\right)^{-\theta}. \tag{7.48}$$

ω is a monomial with $\alpha_t = -\theta, t \in \{1, \dots, d\}$ and the correction factor is given by

$$c_{\mathscr{P}}(\omega) = \frac{1}{\mathfrak{m}} + \frac{1}{\mathfrak{m}^2} \sum_{\substack{\ell,k=1 \\ \ell \neq k}}^{\mathfrak{m}} \prod_{t=1}^{d} \left(s_{P_t(\ell)} (s_{P_t(k)})^{-1} \right)^{-\theta}. \tag{7.49}$$

We chose random permutation forming permutation sets $\mathscr{P}$ with the shuffling algorithm described in [67].

This section presents first the distribution of the correction factors $c_{\mathscr{P}}(\tilde{O})$ for randomly chosen $\mathscr{P}$ and possible cuts on this correction factor to assure that $c_{\mathscr{P}}(\tilde{O}) \neq 0$ and therefore that (7.30) is valid. Then it shows results of the link correlation from the combined method, first without, then with a complex phase factor. In this section a lattice with $d = 4$ lattice points and coupling $I/a = 1$ is used. In our implementation we redefined the correction factor in (7.29), as well as the subset weight and observable in (7.15) and (7.16) without a factor $\frac{1}{\mathfrak{m}}$, as mentioned in section 7.1.2 after (7.31).

THE CORRECTION FACTOR The relation of $I(\tilde{O}, \varrho)$ and $I^{\text{sym}}(\tilde{O}, \varrho)$ in (7.30) is only applicable if $c_{\mathscr{P}}(\tilde{O}) \neq 0$. Very small correction factors would afflict our sample by introducing unnaturally large contributions to the targeted expectation value. But there is no reason why the correction factor in (7.29), which is dependent on the permutation set $\mathscr{P}$ used, should not be small. Therefore we investigated the distribution of the corrections factor for different permutation sets and a possible cut on this factor. More specifically we investigated the distribution of $c_{\mathscr{P}}(\mathscr{L}_v) \cdot \mathfrak{m}, v \in \{1, \dots, d\}$ with $\omega = 1$ exemplary.

We found that more than 99% of the computed values of $c_{\mathscr{P}}(\mathscr{L}_v) \cdot \mathfrak{m}$ are equal or larger than 10^{-2} for all tested $\mathfrak{m} \in \{1, 10, 50, 100\}$, see Figure 7.6. Most values (over 90%) lie in the region $[0.1, 10)$. Here we computed the correction factors of 10^4 different permutation sets $\mathscr{P}$ and repeated the computation ten times for an error estimate.

This means that there are some but few factors that are much smaller than one and can give unnaturally large contributions. To avoid these small factors we applied a cut to the correction factor, which rejects all sets which include at least one $c_{\mathscr{P}}(\mathscr{L}_v)$ value smaller than the cut value. We checked the amount of rejected sets for different cut values $c_{\min}$. We found that for $c_{\min} \cdot \mathfrak{m} = 10^{-2}$ only maximally 5% of the sets are rejected, see Figure 7.7. This percentage grows rapidly when using $c_{\min} \cdot \mathfrak{m} = 10^{-1}$, as expected from the distribution in Figure 7.6, then more than 30% of the proposed sets are rejected. Here the simulation ran until 100 sets were accepted and this experiment was repeated ten times for an error estimate.

For the further experiments we chose the cut value to be $c_{\min} = 10^{-2}$ to reject only few sets because this cut is not physically motivated but computationally necessary and we did not want to induce a systematic bias by rejecting too many sets. Additionally, a

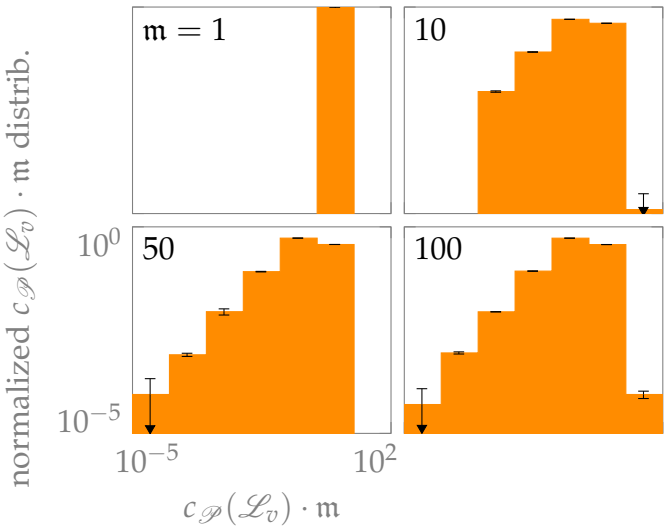


Figure 7.6: Less than 1% of the correction factor values $c_{\mathscr{P}}(\mathscr{L}_v) \cdot \mathfrak{m}, v \in \{1, \ldots, d\}$ are smaller than 10^{-2} for the shown $\mathfrak{m}$ values. All plots have the same logarithmic scale.

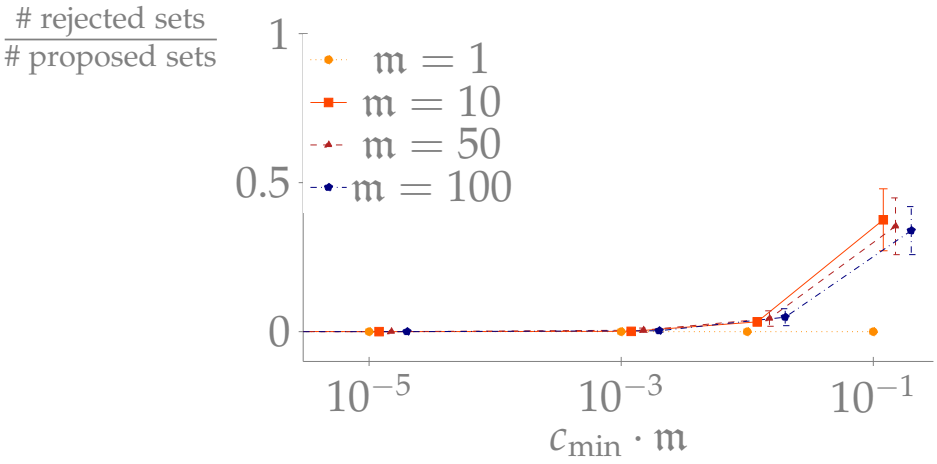


Figure 7.7: For the lower bound $c_{\min} \cdot \mathfrak{m} = 10^{-2}$ on the correction factors less than 5% of the proposed sets are rejected. This percentage increases significantly to more than 30% (for all $\mathfrak{m} > 1$) when using $c_{\min} \cdot \mathfrak{m} = 10^{-1}$.

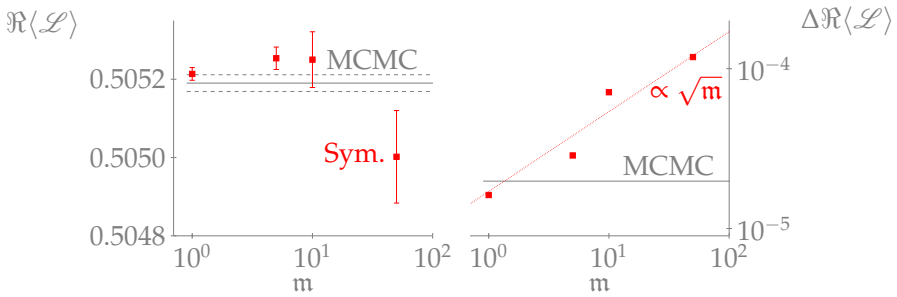


Figure 7.8: Results of the combined method (MCMC + symmetrization points) agree with pure MCMC results, but the error estimate of the combined method grows for larger number of symmetrizations $\mathfrak{m}$ to much larger values than the MCMC error estimates.

larger amount of rejected sets leads to longer runtimes because new sets have to be created and checked again. We also checked that for $c_{\min} = 10^{-2}$ the result of the link correlation expectation value, also for $\theta > 0$, is compatible with the corresponding standard-MCMC value.

WITHOUT COMPLEX PHASE FACTOR With $\omega = 1$ we approximated $\langle\mathscr{L}\rangle = I(\mathscr{L}, \varrho)$ with the cubature rule $Q^{\text{poly}}(\mathscr{L}, \varrho)$ in (7.33). We found good agreement with the MCMC Cluster algorithm results in general, but observed growing error estimates depending on $\mathfrak{m}$, see Figure 7.8. We used 10^8 configurations with 10 permutation sets each and compared the results to the Cluster algorithm using 10^9 configurations. The growing error suggests that for larger $\mathfrak{m}$ significantly larger statistics is needed to reach the precision of the Cluster algorithm. Possibly the error estimate increases proportional to $\sqrt{\mathfrak{m}}$ because for larger $\mathfrak{m}$ more term are included in the sum of $\mathscr{L}_{\Omega_U^{\mathscr{P}}}$, compare (7.16), which can lead to larger variations of $\mathscr{L}_{\Omega_U^{\mathscr{P}}}$ for different subsets $\Omega_U^{\mathscr{P}}$ and therefore to a larger error estimate of $\langle\mathscr{L}\rangle$.

WITH COMPLEX PHASE FACTOR We found that already for as small θ-values as $\theta = 0.01 \cdot 2\pi$ the combined method results in significantly larger errors than found with $\theta = 0$ for $\mathfrak{m} > 1$, while the values agree with MCMC results in the error bars, see Figure 7.9. Again the error estimate of the combined method increases with $\sqrt{\mathfrak{m}}$. For the combined method we used 100 configurations with 10^4 permutation sets each and compared it to the Cluster algorithm with 10^6 configurations.

For larger θ-values the situation becomes worse and, similarly to pure MCMC simulations, the presented combined method cannot reliably compute expectation values of the link correlation variable. The combined method is not able to beat MCMC or even solve the sign-problem of MCMC. Possible explanations, why the combination of

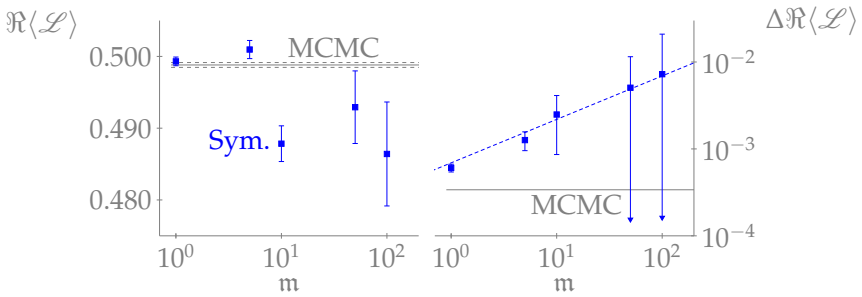


Figure 7.9: Results of the combined method (MCMC + symmetrized quadrature rule) for $\theta = 0.01 \cdot 2\pi$ agree with pure MCMC results, but the error estimate of the combined method grows for larger number of symmetrizations $\mathfrak{m}$ to much larger values than the MCMC error estimates.

the symmetrized quadrature rules with MCMC does not work well, are discussed in the next section.

7.4 POSSIBLE EXPLANATIONS

Section 7.3.2 shows that the proposed combination of the MCMC method with symmetrization points does not result in improved error estimates in comparison to standard MCMC. The method does not solve the sign-problem for the topological osciallator with a complex phase factor. On the other hand section 7.3.1 shows that the CSCR gives a superior error scaling than standard MCMC methods and results in low error estimates, especially for an application to the topological osciallator with a complex phase factor. We wanted to find an explanation for the large difference between the error estimates of the two presented methods. There are mainly two behaviors of the error estimate of the combined method that differ from the error estimate behavior of the CSCR: The error estimate grows with $\sqrt{\mathfrak{m}}$ and the error estimate is very large for $\theta > 0$. In the following both behaviors are discussed.

THE ERROR ESTIMATE GROWS WITH $\mathfrak{m}$ The increase of the error estimate with the number of symmetrizations $\mathfrak{m}$ used is probably due to an increased number of summands in the symmetrized observable and weight in (7.15) and (7.16) for larger $\mathfrak{m}$. This leads to larger fluctuations of $\tilde{O}_{\Omega_U^{\mathscr{P}}}$ between different subsets $\Omega_U^{\mathscr{P}}$ in the cubature rule (7.31) that are not compensated by cancellations because only $\mathfrak{m}$ of all $\mathfrak{m}^d$ combinations of symmetrization points are used in the symmetrized observable and weight, see explanation below. This results in a large error estimate of the cubature rule.

THE ERROR ESTIMATE GROWS SIGNIFICANTLY FOR $\theta > 0$ The large error estimates for simulations with $\theta > 0$ possibly result from the fact that the combined method as we used it includes $\mathfrak{m}$ points in the subset $\Omega_U^{\mathscr{P}}$, dependent on the permutation set $\mathscr{P} = (P_1, \ldots, P_d)$ and does not use Ω_U in (7.11) with $\mathfrak{m}^d$ entries. $\mathfrak{m}^d$ sampling points are also used in the CSCR, (7.13). We investigated the distribution of the summands in the CSCR to check how the cancellations for $\theta > 0$ are happening and whether cancellations can also occur if not all sampling points used in the CSCR are taken into account.

The CSCR computes both numerator and denominator of an expectation value in (7.1) separately. Using the CSCR, compare (7.5), the partition function (the denominator) with a complex phase factor is computed via

$$Z = Q^{\mathrm{CSCR}}(\rho) = \frac{1}{\mathfrak{m}^d} \sum_{k=1}^{\mathfrak{m}^d} \rho(\boldsymbol{t}_k), \tag{7.50}$$

for $\rho = \omega\varrho$ with $\boldsymbol{t}_k = (s_{j_1}, \ldots, s_{j_d})$ and $s_j = \mathrm{e}^{\frac{2\pi i j}{\mathfrak{m}}}$. Equation (7.50) is the sum over all possible combinations of symmetrization entries s_j.

We investigated the distributions of the real part of the partition function summands,

$$\mathscr{F}_{\Re(Z)}(x) = \frac{1}{\mathfrak{m}^d} \sum_{k=1}^{\mathfrak{m}^d} \delta\left(\Re(\rho(\boldsymbol{t}_k)) - x\right), \tag{7.51}$$

for $x \in [0, 1]$. We found that for larger θ values more negative summands occur, see Figure 7.10 for $d = 4$, $I/a = 1$ and $\mathfrak{m} = 30$. For large θ, e.g. $\theta = 0.5 \cdot 2\pi$, the $\Re(Z)$ distribution of positive and negative entries is very similar, and therefore a large part of the summands are canceling each other. We found similar results for the numerator of the expectation value in (7.1).

The combined method uses subsets that only involve $\mathfrak{m}$ summands out of the $\mathfrak{m}^d$ summands in the complete symmetrization. The choice of the summands depends on the randomly chosen permutation set $\mathscr{P}$. For most randomly chosen $\mathscr{P}$ the cancellations that are happening with $\mathfrak{m}^d$ summands cannot happening with $\mathfrak{m}$ summands, resulting in a much worse result than obtained for the CSCR. To check this theory we truncated the sum in (7.50) up to some maximal value,

$$Z_{\mathrm{trunc}}(x) = \frac{1}{\mathfrak{m}^d} \sum_{k=1}^{\mathfrak{m}^d} \Theta\left[(\rho(\boldsymbol{t}_k)) - x\right], \tag{7.52}$$

where $\Theta[\cdot]$ denotes the Heaviside step function. We also computed the truncated numerator $A_{\mathrm{trunc}}(x)$ of the expectation value, $A = Q^{\mathrm{CSCR}}(\mathscr{L}\rho)$, and computed the truncated link correlation,

$$\mathscr{L}_{\mathrm{trunc}}(x) = \frac{A_{\mathrm{trunc}}(x)}{Z_{\mathrm{trunc}}(x)}. \tag{7.53}$$

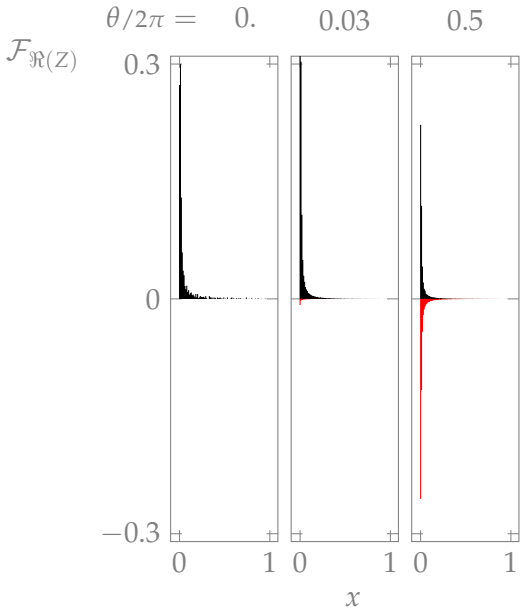
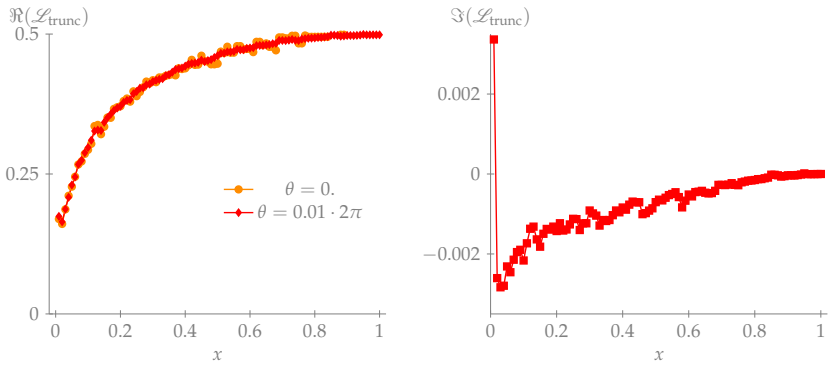


Figure 7.10: For larger θ more negative summands appear in Z. For $\theta = 0.5 \cdot 2\pi$ the negative and positive summand distributions are similar and therefore many cancellations are happening in Z.

For $x < 1$ not all $\mathfrak{m}^d$ summands or sampling points are taken into account. We investigated which x-value is needed such that this truncated link correlation gives the physical value $\langle \mathscr{L} \rangle$. $\Re(\mathscr{L}_{\text{trunc}})$ shows a similar behavior for $\theta = 0$ and $\theta = 0.01 \cdot 2\pi$: the larger the value of the truncation value x, the less $\Re(\mathscr{L}_{\text{trunc}})$ is changing, shown in Figure 7.11a. $\Re(\mathscr{L})$ is reached for $x \gtrsim 0.8$. $\Im(\mathscr{L}_{\text{trunc}})$ for $\theta = 0.01 \cdot 2\pi$ differs slightly from zero for small x, where few summands are taken into account and is approximately zero for $x \gtrsim 0.8$, compare Figure 7.11b. These figures demonstrate that to arrive at the final physical result for $\mathscr{L}$, most of the $\mathfrak{m}^d$ summands have to be taken into account. Of course, here we chose a specific order of the summands which affects the truncation result and the behavior shown in Figure 7.11 could look differently when taking a different order into account. Anyway, these Figures make it plausible that the $\mathfrak{m}$ summands we used in the combined method, chosen in a random manner, are not enough to arrive at the final $\langle \mathscr{L} \rangle$ value and lead to large error estimates when averaging over different $\mathscr{L}_{\text{trunc}}$ estimates, already for small $\theta = 0.01 \cdot 2\pi$.

For $\theta = 0.5 \cdot 2\pi$ the behavior of both real and imaginary part of $\mathscr{L}_{\text{trunc}}$ becomes unpredictable, see Figure 7.12. There is no hierarchy visible between values at small and large x and the final physical value is only reached if almost all summands, $x \approx 1$ are taken into account. Truncating the sum leads to a highly irregular behavior of the link correlation with results that can be far away from the final value.



(a) $\Re(\mathscr{L}_{\text{trunc}})$ (b) $\Im(\mathscr{L}_{\text{trunc}})$

Figure 7.11: For small θ ($\theta = 0, \theta = 0.01 \cdot 2\pi$): The more summands are taken into account (for larger x) the more are $\Re(\mathscr{L}_{\text{trunc}})$ and $\Im(\mathscr{L}_{\text{trunc}})$ approaching the final physical value at $x = 1$.

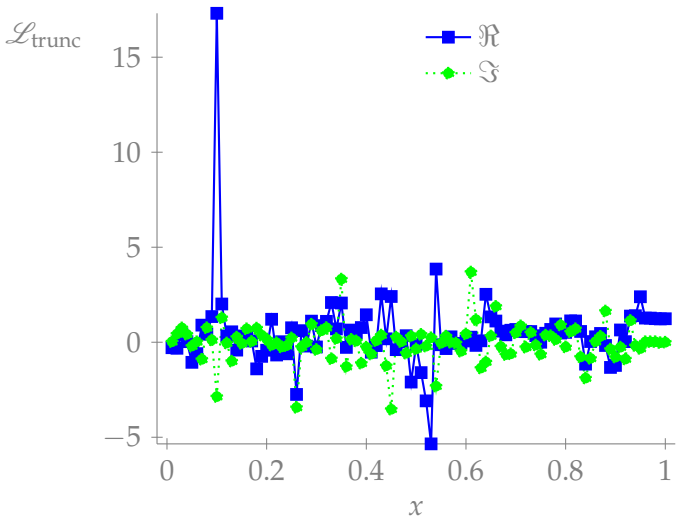


Figure 7.12: For large $\theta = 0.5 \cdot 2\pi$: There is no hierarchy visible between different truncation values x. Truncating the sum leads to highly irregular behavior of the link correlation with results which can be far away from the final value.

We suspect that the combined method failed to overcome the sign-problem because we did not sample sufficiently many relevant sampling points. Depending on which permutation sets $\mathscr{P}$ we have chosen, the results of the individual sets can fluctuate strongly, leading to large errors, which we observed. Thus, although the combined method is a possible method to compute the expectation value of an observable, it does not seem to solve the sign-problem of MCMC methods.

8

SUMMARY

This thesis presents methods to reduce error estimates of the numerical evaluation of the path integral to get in the end significant results for QCD observables, comparable with the real world. The QCD path integral is an integration over fermionic and bosonic degrees of freedom, which are very different objects. The thesis deals with both types separately, the evaluation of quark connected and disconnected diagrams, which result from integrating out the fermions of the fermionic path integral, and the integration over the bosonic degrees of freedom.

The error estimates from the numerical evaluations, on the one hand of quark diagrams an on the other hand of the bosonic path integral with integrated out fermions, depend both on the error scaling of the used evaluation method and the number of evaluation points. These evaluation points are stochastic sources for the quark diagrams and sampling points for the bosonic path integral. We tested a method that reduces the error estimate of the evaluation of quark disconnected diagrams in QCD. For the approximation of the bosonic path integral we developed methods to improve the error estimates for low-dimensional benchmark models.

COMPUTING DISCONNECTED DIAGRAMS Lattice QCD observables get contributions from quark connected and disconnected diagrams. Evaluating these diagrams means to compute the one-to-all and all-to-all propagator, respectively by inverting the large Dirac matrix. The signal-to-noise ratio of the disconnected diagrams is usually low because, in contrast to the one-to-all propagator that can be evaluated via point sources, the all-to-all propagator can only be efficiently computed using stochastic sources. Therefore methods have been developed to improve this ratio.

We applied the exact eigenmode reconstruction with deflation method to lattice QCD to improve the computation of quark disconnected diagrams. This method uses eigenmodes of the Dirac matrix with low-lying eigenvalues to compute one part of the all-to-all propagator, corresponding to these eigenmodes, exactly. For the other part of the propagator, the method deflates the Dirac matrix with the eigenmodes and inverts this deflated matrix stochastically by solving linear equations, each including one stochastic source. We applied this method to a $16^3 \times 32$ twisted mass fermion lattice with a lattice spacing of $a = 0.085\,\mathrm{fm}$ and a pion mass of $m_\pi = 370\,\mathrm{MeV}$.

We found that the exact eigenmode reconstruction with deflation method can result in approximately 5.5 times less runtime on the Piz Daint supercomputer [33], compared to a standard computation. The method has a relatively long initialization time to compute the eigenmodes, but results in faster numerical computations of solutions to the linear equations due to a deflation of the Dirac matrix. Additionally, the method results in fewer stochastic sources that are needed to reach a specific error estimate, due to computing one part of the propagator exactly. On the other hand another very efficient method was recently applied to twisted mass fermion lattices: We estimated that the Multigrid algorithm needs around 220 times less runtime than standard methods.

All in all the exact eigenmode reconstruction with deflation method, as developed and tested in this thesis, turned out to need less runtime than the standard methods used so far and its saved runtime can be used to reduce the error estimate of disconnected diagrams by including more stochastic sources in the computation. The resulting propagator can be used to compute many different observables, such that the large initialization time of the method becomes negligible. The method was eventually outperformed by the Multigrid algorithm. As a consequence, the exact eigenmode reconstruction with deflation method was, unfortunately, not used for final production runs.

It would be interesting and important to compare different methods of evaluating disconnected diagrams in the future when, e.g. smaller values of the lattice spacing of larger volumes are simulated. This would allow to employ the best available algorithm for a given simulation setup which would lead to a large reduction of the error of these quark diagrams. This in turn will provide a precise computation of many hadronic quantities which can be compared to experimental or phenomenological determinations.

EVALUATING THE BOSONIC PATH INTEGRAL MCMC methods are usually used to evaluate the bosonic path integral. We tested two alternative methods to overcome some problems of the MCMC methods: Its slow error scaling, critical slowing-down when approaching the continuum limit and the sign-problem for highly oscillatory integrands.

The recursive numerical integration uses the local coupling structure of integrands. The next-neighbor coupling in pure gauge lattice simulations allows to factorize the integrand in the path integral. Recursive numerical integration restructures the integral according to this factorization such that one-variable quadrature rules can be applied efficiently to each of the factors separately to approximate the full integral. We applied this method to the topological oscillator, using the efficient Gaussian quadrature rule. We found accurate results when approaching the continuum limit with no critical slowing-down

by construction and were able to compute error estimates which scale exponentially and therefore much faster than the MCMC error estimates. The method needs orders of magnitude less runtime than an optimal MCMC algorithm, here the Cluster algorithm [83], to reach a specified error estimate.

Therefore the recursive numerical integration is a promising alternative to the MCMC methods, can be applied at least to one-dimensional problems and is then much more efficient than MCMC methods. The next step is to generalize the method to larger dimensions. A naive generalization results in time-intensive tensor multiplications. But perhaps it is possible to use nested integrations to generalize the method at least to $1+1$ dimensions.

We developed the symmetrized quadrature rules to approximate the bosonic path integral, especially in case of a possible sign-problem. These are polynomially exact quadrature rules for integrals over compact groups $\mathcal{U}(N)$ and $\mathcal{SU}(N)$ with $N \leq 3$ and one integration variable. For gauge-theories the bosonic path integral with one integration variable is such an integral. These symmetrized quadrature rules are based on transforming efficient quadrature rules over spheres. We applied the method to the one-dimensional QCD model that depends on only one variable and shows a severe sign-problem in certain regions of the parameter space. We found error estimates which are orders of magnitude smaller than MC errors, especially in the sign-problem regions where MC simulations do not give any significant result. Also in these regions the error estimates of the symmetrized quadrature rules are only limited by the machine precision used. Therefore this method is another promising alternative to MCMC methods, especially for models with a possible sign-problem, which it avoids completely.

We also investigated two different cubature rules that are based on the one-variable symmetrized $\mathcal{U}(1)$ quadrature rule but applicable to models with more than one integration variable. The first cubature rule, the CSCR, applies the symmetrized quadrature rule to each integration variable and is therefore a polynomially exact method. We developed a second cubature rule, which combines the sampling points of the symmetrized quadrature rule with an MCMC simulation. We applied both methods to the topological osciallator with an additional complex phase factor that gives rise to a possible sign-problem. Applying the CSCR resulted in practice in error estimates which decreased exponentially. But this application is inefficient when using many more integration variables. Unfortunately, combining the method with MCMC to overcome this inefficiency resulted in very large error estimates. We showed that it is almost impossible to get small error estimates if not all sampling points of the CSCR are taken into account. This is the main reason why the combined method gives worse error estimates than the CSCR. Therefore

another method is needed to generalize the symmetrized quadrature rules to more integration variables.

A very promising idea for this generalization, at least to one dimensional models, is to combine the symmetrized quadrature rules and the recursive numerical integration. The recursive numerical integration is directly applicable to a one-dimensional model. If this model includes gauge links, the one-variable quadrature rule needed for the recursive numerical integration can be a symmetrized quadrature rule, instead of the Gaussian rule used before. Only if this avoids the sign problem, this would provide a very interesting alternative to MCMC methods. Of course, the generalization of the recursive numerical integration method to higher dimensions is still an open problem and a lot of developments, including new ideas, also on how to include fermions which induce non-local couplings, will presumably be needed to evaluate the QCD path integral with polynomial exactness.

THE END The methods presented here promise more precise computations of QCD observables in the future. The exact eigenmode reconstruction with deflation makes more precise disconnected diagram computations possible, the polynomially exact quadrature rules give a better error scaling, leading to smaller error estimates of observables in benchmark models and make simulations in sign-problem regions possible. A generalization to larger space-time dimensions is still an open question which can hopefully be answered in the future.

Part III

APPENDIX

A

CONVENTIONS

PAULI MATRICES

$$\tau_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \ \tau_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \ \tau_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{A.1}$$

GAMMA MATRICES To use the gamma matrices in LQCD, they have to be Wick-rotated to satisfy

$$\{\gamma_\mu, \gamma_\nu\} = 2\delta_{\mu\nu}\mathbb{1}_4. \tag{A.2}$$

In the chiral presentation they are given by

$$\gamma_4 = \begin{pmatrix} 0 & \mathbb{1}_2 \\ \mathbb{1}_2 & 0 \end{pmatrix}, \ \gamma_{1,2,3} = \begin{pmatrix} 0 & -i\tau_{1,2,3} \\ i\tau_{1,2,3} & 0 \end{pmatrix}, \tag{A.3}$$

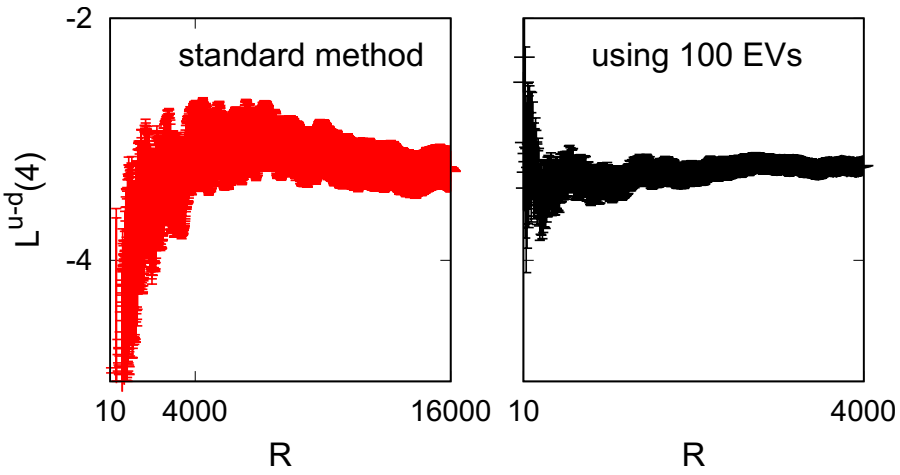
$$\gamma_5 = \gamma_1\gamma_2\gamma_3\gamma_4 = \begin{pmatrix} \mathbb{1}_2 & 0 \\ 0 & -\mathbb{1}_2 \end{pmatrix}. \tag{A.4}$$

CHARGE CONJUGATION

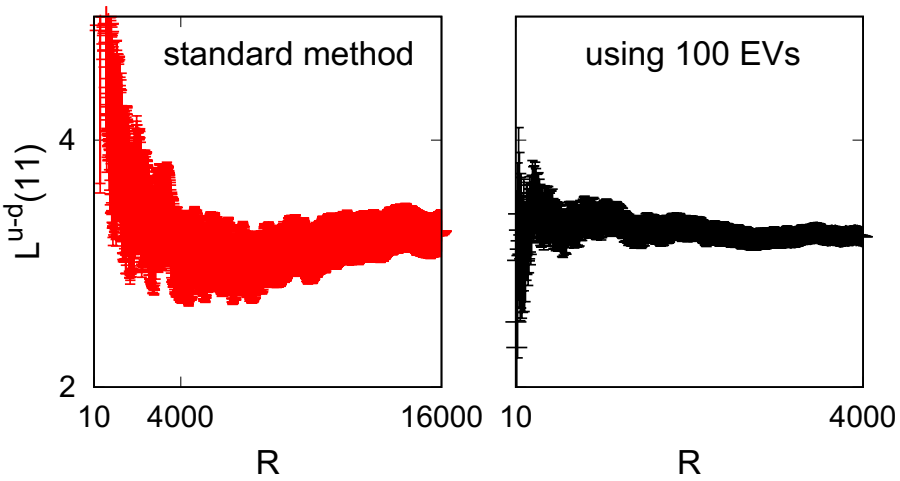
$$C = i\gamma_2\gamma_4. \tag{A.5}$$

B

MORE DISCONNECTED DIAGRAM RESULTS

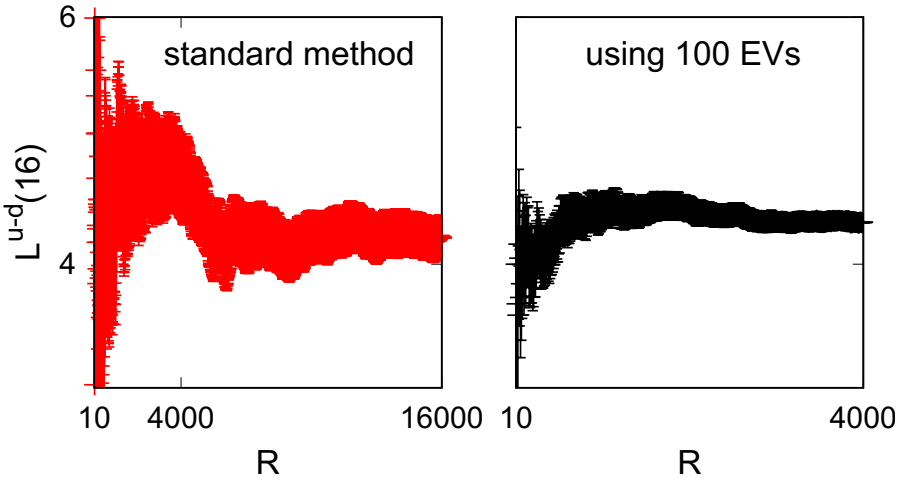


(a) $n_t = 4$

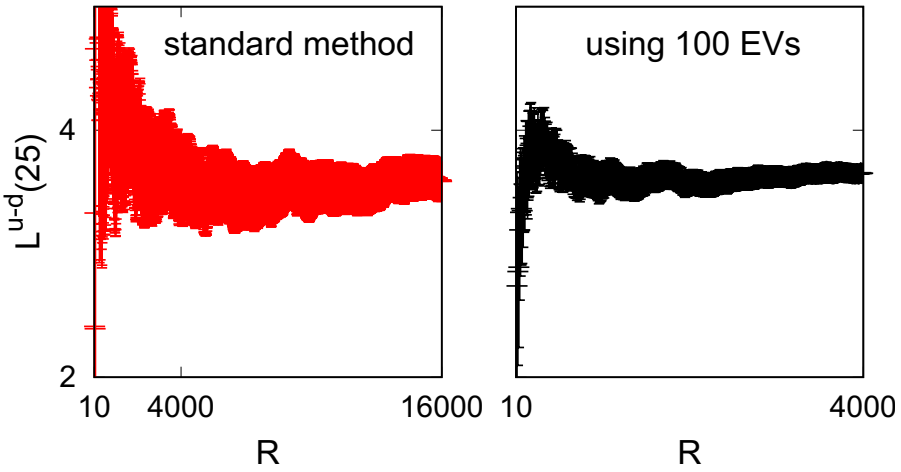


(b) $n_t = 11$

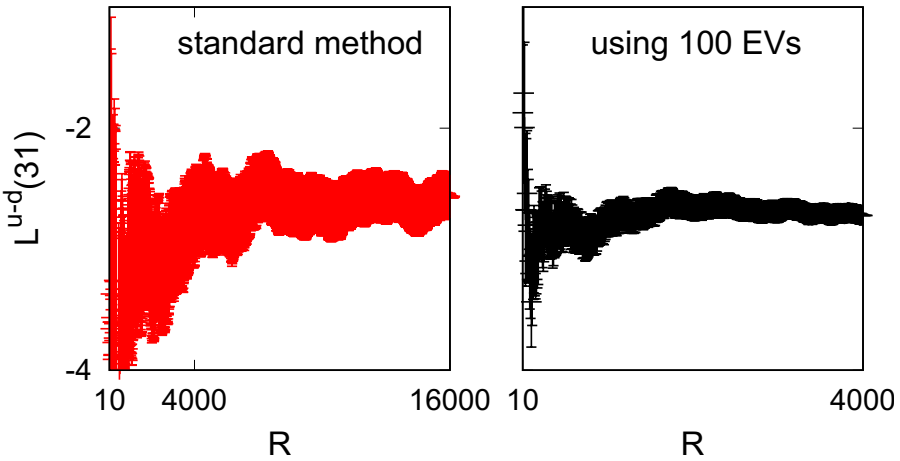
Figure B.1: Comparison of loop result computed with the standard method without deflation and with the exact eigenmode reconstruction with deflation method using 100 eigenvectors.



(a) $n_t = 16$



(b) $n_t = 25$



(c) $n_t = 31$

Figure B.2: Comparison of loop result computed with the standard method without deflation and with the exact eigenmode reconstruction with deflation method using 100 eigenvectors.

BIBLIOGRAPHY

[1] V. Petrosian A.G. Bergmann and R. Lynds. "Graviational lens models of arcs in clusters." In: *The Astrophysical Journal* 350 (1990). DOI: 10.1086/168359.

[2] Georges Aad et al. "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC." In: *Phys. Lett.* B716 (2012), pp. 1–29. DOI: 10.1016/j.physletb.2012.08.020. arXiv: 1207.7214 [hep-ex].

[3] S. Abachi et al. "Observation of the top quark." In: *Phys. Rev. Lett.* 74 (1995), pp. 2632–2637. DOI: 10.1103/PhysRevLett.74.2632. arXiv: hep-ex/9503003 [hep-ex].

[4] A. Abdel-Rehim, C. Alexandrou, M. Constantinou, V. Drach, K. Hadjiyiannakou, K. Jansen, G. Koutsou, and A. Vaquero. "Disconnected quark loop contributions to nucleon observables in lattice QCD." In: *Phys. Rev.* D89.3 (2014), p. 034501. DOI: 10.1103/PhysRevD.89.034501. arXiv: 1310.6339 [hep-lat].

[5] A. Abdel-Rehim, C. Alexandrou, M. Constantinou, K. Hadjiyiannakou, K. Jansen, Ch. Kallidonis, G. Koutsou, and A. Vaquero Aviles-Casco. "Direct Evaluation of the Quark Content of Nucleons from Lattice QCD at the Physical Point." In: *Phys. Rev. Lett.* 116.25 (2016), p. 252001. DOI: 10.1103/PhysRevLett.116.252001. arXiv: 1601.01624 [hep-lat].

[6] Abdou Abdel-Rehim, Constantia Alexandrou, Martha Constantinou, Jacob Finkenrath, Kyriakos Hadjiyiannakou, Karl Jansen, Christos Kallidonis, Giannis Koutsou, Alejandro Vaquero Avilés-Casco, and Julia Volmer. "Disconnected diagrams with twisted-mass fermions." In: *PoS* LATTICE2016 (2016), p. 155. arXiv: 1611.03802 [hep-lat].

[7] F. Abe et al. "Observation of top quark production in $\bar{p}p$ collisions." In: *Phys. Rev. Lett.* 74 (1995), pp. 2626–2631. DOI: 10.1103/PhysRevLett.74.2626. arXiv: hep-ex/9503002 [hep-ex].

[8] A. Adare et al. "Inclusive double-helicity asymmetries in neutral-pion and eta-meson production in $\vec{p}+\vec{p}$ collisions at $\sqrt{s}=200$ GeV." In: *Phys. Rev.* D90.1 (2014), p. 012007. DOI: 10.1103/PhysRevD.90.012007. arXiv: 1402.6296 [hep-ex].

[9] P.A.R. Ade et al. "Planck 2013 results. I. Overview of products and scientific results." In: (2013). arXiv: 1303.5062 [astro-ph.CO].

[10] J. M. Alarcon, J. Martin Camalich, and J. A. Oller. "The chiral representation of the πN scattering amplitude and the pion-nucleon sigma term." In: *Phys. Rev.* D85 (2012), p. 051503. DOI: 10.1103/PhysRevD.85.051503. arXiv: 1110.3797 [hep-ph].

[11] C. Alexandrou, M. Constantinou, K. Hadjiyiannakou, K. Jansen, C. Kallidonis, G. Koutsou, A. Vaquero Avilés-Casco, and C. Wiese. "Nucleon Spin and Momentum Decomposition Using Lattice QCD Simulations." In: *Phys. Rev. Lett.* 119.14 (2017), p. 142002. DOI: 10.1103/PhysRevLett.119.142002. arXiv: 1706.02973 [hep-lat].

[12] C. Alexandrou et al. "Nucleon scalar and tensor charges using lattice QCD simulations at the physical value of the pion mass." In: *Phys. Rev.* D95.11 (2017). [Erratum: Phys. Rev.D96,no.9,099906(2017)], p. 114514. DOI: 10.1103/PhysRevD.96.099906,10.1103/PhysRevD.95.114514. arXiv: 1703.08788 [hep-lat].

[13] Constantia Alexandrou, Simone Bacchio, Jacob Finkenrath, Andreas Frommer, Karsten Kahl, and Matthias Rottmann. "Adaptive Aggregation-based Domain Decomposition Multigrid for Twisted Mass Fermions." In: *Phys. Rev.* D94.11 (2016), p. 114509. DOI: 10.1103/PhysRevD.94.114509. arXiv: 1610.02370 [hep-lat].

[14] A. Ammon, A. Genz, T. Hartung, K. Jansen, H. Leövey, and J. Volmer. "On the efficient numerical solution of lattice systems with low-order couplings." In: *Comput. Phys. Commun.* 198 (2016), pp. 71–81. DOI: 10.1016/j.cpc.2015.09.004. arXiv: 1503.05088 [hep-lat].

[15] A. Ammon, T. Hartung, K. Jansen, H. Leövey, and J. Volmer. "Overcoming the sign problem in one-dimensional QCD by new integration rules with polynomial exactness." In: *Phys. Rev.* D94.11 (2016), p. 114508. DOI: 10.1103/PhysRevD.94.114508. arXiv: 1607.05027 [hep-lat].

[16] Andreas Ammon, Alan Genz, Tobias Hartung, Karl Jansen, Hernan Leövey, and Julia Volmer. "Applying recursive numerical integration techniques for solving high dimensional integrals." In: *PoS* LATTICE2016 (2016), p. 335. arXiv: 1611.08628 [hep-lat].

[17] Andreas Ammon, Tobias Hartung, Karl Jansen, Hernan Leövey, and Julia Volmer. "New polynomially exact integration rules on U(N) and SU(N)." In: 2016. arXiv: 1610.01931 [hep-lat]. URL: https://inspirehep.net/record/1490030/files/arXiv:1610.01931.pdf.

[18] M. Anselmino, M. Boglione, U. D'Alesio, A. Kotzinian, F. Murgia, A. Prokudin, and S. Melis. "Update on transversity and Collins functions from SIDIS and e+ e- data." In: *Nucl. Phys. Proc. Suppl.* 191 (2009), pp. 98–107. DOI: 10.1016/j.nuclphysbps.2009.03.117. arXiv: 0812.4366 [hep-ph].

[19] J. Ashman et al. "A Measurement of the Spin Asymmetry and Determination of the Structure Function g(1) in Deep Inelastic Muon-Proton Scattering." In: *Phys. Lett.* B206 (1988), p. 364. DOI: 10.1016/0370-2693(88)91523-7.

[20] J. Ashman et al. "An Investigation of the Spin Structure of the Proton in Deep Inelastic Scattering of Polarized Muons on Polarized Protons." In: *Nucl. Phys.* B328 (1989), p. 1. DOI: 10.1016/0550-3213(89)90089-8.

[21] *BLAS*. 2017. URL: www.netlib.org/blas/ (visited on 04/08/2018).

[22] R. Babich, M. A. Clark, B. Joo, G. Shi, R. C. Brower, and S. Gottlieb. "Scaling Lattice QCD beyond 100 GPUs." In: *SC11 International Conference for High Performance Computing, Networking, Storage and Analysis Seattle, Washington, November 12-18, 2011*. 2011. DOI: 10.1145/2063384.2063478. arXiv: 1109.2935 [hep-lat]. URL: https://inspirehep.net/record/927455/files/arXiv:1109.2935.pdf.

[23] Simone Bacchio, Constantia Alexandrou, and Jacob Finkerath. "Multigrid accelerated simulations for Twisted Mass fermions." In: *35th International Symposium on Lattice Field Theory (Lattice 2017) Granada, Spain, June 18-24, 2017*. 2017. arXiv: 1710.06198 [hep-lat]. URL: http://inspirehep.net/record/1631185/files/arXiv:1710.06198.pdf.

[24] Csaba Balazs. "Baryogenesis: A small review of the big picture." In: (2014). arXiv: 1411.3398 [hep-ph].

[25] W. Bietenholz, U. Gerber, M. Pepe, and U.-J. Wiese. "Topological Lattice Actions." In: *JHEP* 1012 (2010), p. 020. DOI: 10.1007/JHEP12(2010)020. arXiv: 1009.2146 [hep-lat].

[26] Neven Bilic and Kresimir Demeterfi. "One-dimensional QCD With Finite Chemical Potential." In: *Phys. Lett.* B212 (1988), pp. 83–87. DOI: 10.1016/0370-2693(88)91240-3.

[27] Khalil Bitar, A.D. Kennedy, Roger Horsley, Steffen Meyer, and Pietro Rossi. "The QCD finite temperature transition and hybrid Monte Carlo." In: *Nuclear Physics B* 313.2 (1989), pp. 348 –376. ISSN: 0550-3213. DOI: https://doi.org/10.1016/0550-3213(89)90323-4. URL: http://www.sciencedirect.com/science/article/pii/0550321389903234.

[28] Jacques Bloch, Falk Bruckmann, and Tilo Wettig. "Subset method for one-dimensional QCD." In: *JHEP* 10 (2013), p. 140. DOI: 10.1007/JHEP10(2013)140. arXiv: 1307.1416 [hep-lat].

[29] Jacques Bloch, Falk Bruckmann, and Tilo Wettig. "Sign problem and subsets in one-dimensional QCD." In: *PoS* LATTICE2013 (2014), p. 194. arXiv: 1310.6645 [hep-lat].

[30] Philippe Boucaud et al. "Dynamical Twisted Mass Fermions with Light Quarks: Simulation and Analysis Details." In: *Comput. Phys. Commun.* 179 (2008), pp. 695–715. DOI: 10.1016/j.cpc.2008.06.013. arXiv: 0803.0224 [hep-lat].

[31] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, eds. *Handbook of Markov Chain Monte Carlo*. Handbooks of Modern Statistical Methods. Chapman and Hall,CRC, 2011. ISBN: 1420079417.

[32] Laurent Canetti, Marco Drewes, and Mikhail Shaposhnikov. "Matter and Antimatter in the Universe." In: *New J. Phys.* 14 (2012), p. 095012. DOI: 10.1088/1367-2630/14/9/095012. arXiv: 1204.4186 [hep-ph].

[33] Swiss National Supercomputer Center. *Piz Daint*. 2018. URL: www.cscs.ch/computers/piz-daint/ (visited on 02/20/2018).

[34] Serguei Chatrchyan et al. "Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC." In: *Phys. Lett.* B716 (2012), pp. 30–61. DOI: 10.1016/j.physletb.2012.08.021. arXiv: 1207.7235 [hep-ex].

[35] Krzysztof Cichy, Gregorio Herdoiza, and Karl Jansen. "Continuum Limit of Overlap Valence Quarks on a Twisted Mass Sea." In: *Nucl. Phys.* B847 (2011), pp. 179–196. DOI: 10.1016/j.nuclphysb.2011.01.021. arXiv: 1012.4412 [hep-lat].

[36] M. A. Clark, R. Babich, K. Barros, R. C. Brower, and C. Rebbi. "Solving Lattice QCD systems of equations using mixed precision solvers on GPUs." In: *Comput. Phys. Commun.* 181 (2010), pp. 1517–1528. DOI: 10.1016/j.cpc.2010.05.002. arXiv: 0911.3191 [hep-lat].

[37] D. Clowe et al. "A direct emperical proof of the existence of dark matter." In: *ApJ* 648.2 (2006). DOI: 10.1086/508162. arXiv: 0608407 [astro-ph].

[38] P.J. Davis and P. Rabinowitz. *Methods of Numerical Integration*. Computer Science and Applied Mathematics. Academic Press, 1984. ISBN: 9781483264288.

[39] Thomas A. Degrand and Pietro Rossi. "Conditioning techniques for dynamical fermions." In: *Computer Physics Communications* 60.2 (1990), pp. 211 –214. ISSN: 0010-4655. DOI: https://doi.org/10.1016/0010-4655(90)90006-M. URL: http://www.sciencedirect.com/science/article/pii/001046559090006M.

[40] Ph. Delsarte, J.M. Goethals, and J.J. Seidel. "Spherical codes and designs." English. In: *Geometriae Dedicata* 6.3 (1977), pp. 363–388. ISSN: 0046-5755. DOI: 10.1007/BF03187604.

[41] Josef Dick, Frances Y. Kuo, and Ian H. Sloan. "High-dimensional integration: The quasi-Monte Carlo way." In: *Acta Numerica* 22 (2013), 133–288. DOI: 10.1017/S0962492913000044.

[42] S.J. Dong and K.F. Liu. "Stochastic Estimation with Z2 Noise." In: *Phys.Lett. B* 328 (1994), pp. 130–136. DOI: 10.1016/0370-2693(94)90440-5. arXiv: 9308015 [hep-lat].

[43] Simon Duane, A.D. Kennedy, Brian J. Pendleton, and Duncan Roweth. "Hybrid Monte Carlo." In: *Physics Letters B* 195.2 (1987), pp. 216 –222. ISSN: 0370-2693. DOI: https://doi.org/10.1016/0370-2693(87)91197-X. URL: http://www.sciencedirect.com/science/article/pii/037026938791197X.

[44] John R. Ellis, Keith A. Olive, and Christopher Savage. "Hadronic Uncertainties in the Elastic Scattering of Supersymmetric Dark Matter." In: *Phys. Rev.* D77 (2008), p. 065026. DOI: 10.1103/PhysRevD.77.065026. arXiv: 0801.3656 [hep-ph].

[45] Daniel de Florian, Rodolfo Sassot, Marco Stratmann, and Werner Vogelsang. "Evidence for polarization of gluons in the proton." In: *Phys. Rev. Lett.* 113.1 (2014), p. 012001. DOI: 10.1103/PhysRevLett.113.012001. arXiv: 1404.4293 [hep-ph].

[46] Message Passing Interface Forum. *MPI: A Message-Passing Interface Standard, Version 3.1*. Hight Performance Computing Center Stuttgart (HLRS), 2012. URL: mpi-forum.org/docs/mpi-3.0/mpi30-report.pdf.

[47] Message Passing Interface Forum. *MPI: A Message-Passing Interface Standard Version 3.1*. 2015. URL: mpi-forum.org.

[48] M. Foster and C. Michael. "Quark mass dependence of hadron masses from lattice QCD." In: *Phys.Rev.D* 59 (1999), p. 074503. DOI: 10.1103/PhysRevD.59.074503. arXiv: 9810021 [hep-lat].

[49] M. Foster and Christopher Michael. "Quark mass dependence of hadron masses from lattice QCD." In: *Phys. Rev.* D59 (1999), p. 074503. DOI: 10.1103/PhysRevD.59.074503. arXiv: hep-lat/9810021 [hep-lat].

[50] The R Foundation. *The R Project for Statistical Computing*. URL: www.r-project.org (visited on 02/20/2018).

[51] R. Frezzotti and G. C. Rossi. "Chirally improving Wilson fermions. 1. O(a) improvement." In: *JHEP* 08 (2004), p. 007. DOI: 10.1088/1126-6708/2004/08/007. arXiv: hep-lat/0306014 [hep-lat].

[52] R. Frezzotti and G. C. Rossi. "Twisted mass lattice QCD with mass nondegenerate quarks." In: *Nucl. Phys. Proc. Suppl.* 128 (2004). [,193(2003)], pp. 193–202. DOI: 10.1016/S0920-5632(03)02477-0. arXiv: hep-lat/0311008 [hep-lat].

[53] Roberto Frezzotti, Pietro Antonio Grassi, Stefan Sint, and Peter Weisz. "Lattice QCD with a chirally twisted mass term." In: *JHEP* 08 (2001), p. 058. arXiv: hep-lat/0101001 [hep-lat].

[54] B.E. Fristedt and L.F. Gray. *A Modern Approach to Probability Theory*. Probability and Its Applications. Birkhäuser Basel, 1997. ISBN: 978-0-8176-3807-8.

[55] Andreas Frommer, Karsten Kahl, Stefan Krieg, Björn Leder, and Matthias Rottmann. "Adaptive Aggregation Based Domain Decomposition Multigrid for the Lattice Wilson Dirac Operator." In: *SIAM J. Sci. Comput.* 36 (2014), A1581–A1608. DOI: 10.1137/130919507. arXiv: 1303.1377 [hep-lat].

[56] Christof Gattringer and Christian B. Lang. "Quantum chromodynamics on the lattice." In: *Lect. Notes Phys.* 788 (2010), pp. 1–343. DOI: 10.1007/978-3-642-01850-3.

[57] A. Genz. "Fully Symmetric Interpolatory Rules for Multiple Integrals over Hyper-Spherical Surfaces." In: *Journal of Computational and Applied Mathematics* 157 (2003), 187–195. DOI: 10.1016/S0377-0427(03)00413-8.

[58] Alan Genz and David K. Kahaner. "The numerical evaluation of certain multivariate normal integrals." In: *J. Comput. Appl. Math.* 16.2 (1986), pp. 255–258. ISSN: 0377-0427. DOI: 10.1016/0377-0427(86)90100-7. URL: http://www.sciencedirect.com/science/article/pii/0377042786901007.

[59] Tobias Hartung, Karl Jansen, Hernan Leövey, and Julia Volmer. "Improving Monte Carlo integration by symmetrization." In: *The Diversity and Beauty of Applied Operator Theory*. Ed. by Albrecht Böttcher, Daniel Potts, Peter Stollmann, and David Wenzel. Cham: Springer International Publishing, 2018, pp. 291–317. ISBN: 978-3-319-75996-8.

[60] W. K. Hastings. "Monte Carlo Sampling Methods Using Markov Chains and Their Applications." In: *Biometrika* 57.1 (1970), pp. 97–109. ISSN: 00063444. URL: http://www.jstor.org/stable/2334940.

[61] A.J. Hayter. "Recursive integration methodologies with statistical applications." In: *Journal of Statistical Planning and Inference* 136.7 (2006). In Memory of Dr. Shanti Swarup Gupta, pp. 2284–2296. ISSN: 0378-3758. DOI: 10.1016/j.jspi.2005.08.024. URL: http://www.sciencedirect.com/science/article/pii/S0378375805002223.

[62] Kerstin Hesse, Frances Y. Kuo, and Ian H. Sloan. "A component-by-component approach to efficient numerical integration over products of spheres." In: *Journal of Complexity* 23.1 (2007), pp. 25–51. ISSN: 0885-064X. DOI: https://doi.org/10.1016/j.jco.2006.08.001. URL: http://www.sciencedirect.com/science/article/pii/S0885064X06000793.

[63] K. Jansen and C. Urbach. "tmLQCD: A Program suite to simulate Wilson Twisted mass Lattice QCD." In: *Comput. Phys. Commun.* 180 (2009), pp. 2717–2738. DOI: 10.1016/j.cpc.2009.05.016. arXiv: 0905.3331 [hep-lat].

[64] Karl Jansen and Chuan Liu. "Kramers equation algorithm for simulations of QCD with two flavors of Wilson fermions and gauge group SU(2)." In: *Nucl. Phys.* B453 (1995). [Erratum: Nucl. Phys.B459,437(1996)], pp. 375–394. DOI: 10.1016/0550-3213(95)00427-T,10.1016/0550-3213(95)00619-2. arXiv: hep-lat/9506020 [hep-lat].

[65] Xiang-Dong Ji. "Hunting for the remaining spin in the nucleon." In: *High-energy spin physics. Proceedings, 12th International Symposium, SPIN 96, Amsterdam, Netherlands, September 10-14, 1996*. 1996, pp. 68–74. arXiv: hep-ph/9610369 [hep-ph].

[66] D. Kahaner, C.B. Moler, S. Nash, and G.E. Forsythe. *Numerical methods and software*. Prentice-Hall series in computational mathematics. Prentice Hall, 1989.

[67] D. E. Knuth. *Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Pearson Education, 1997. ISBN: 978-0201896848.

[68] D.P. Kroese, T. Taimre, and Z.I. Botev. *Handbook of Monte Carlo Methods*. Wiley Series in probability and statistics. Wiley, 2011. ISBN: 978-0-470-17793-8.

[69] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh. "Basic Linear Algebra Subprograms for Fortran Usage." In: *ACM Trans. Math. Softw.* 5.3 (Sept. 1979), pp. 308–323. ISSN: 0098-3500. DOI: 10.1145/355841.355847. URL: http://doi.acm.org/10.1145/355841.355847.

[70] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK Users' Guide: Solution of Large Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*. 1997. URL: www.caam.rice.edu/software/ARPACK/UG/ug.html#ARPACKUsers'Guide.

[71] Martin Luscher, Stefan Sint, Rainer Sommer, and Peter Weisz. "Chiral symmetry and O(a) improvement in lattice QCD." In: *Nucl. Phys.* B478 (1996), pp. 365–400. DOI: 10.1016/0550-3213(96)00378-1. arXiv: hep-lat/9605038 [hep-lat].

[72] C. McNeile and Christopher Michael. "Decay width of light quark hybrid meson from the lattice." In: *Phys. Rev.* D73 (2006), p. 074506. DOI: 10.1103/PhysRevD.73.074506. arXiv: hep-lat/0603007 [hep-lat].

[73] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. "Equation of State Calculations by Fast Computing Machines." In: *J.Chem.Phys.* 21 (June 1953), pp. 1087–1092. DOI: 10.1063/1.1699114.

[74] H. Neff, N. Eicker, Th. Lippert, J. W. Negele, and K. Schilling. "Low fermionic eigenmode dominance in QCD on the lattice." In: *Phys. Rev. D* 64 (11 2001), p. 114509. DOI: 10.1103/PhysRevD.64.114509. URL: https://link.aps.org/doi/10.1103/PhysRevD.64.114509.

[75] Ferenc Niedermayer. "Cluster algorithms." In: (1996). [Lect. Notes Phys.501,36(1998)]. DOI: 10.1007/BFb0105458. arXiv: hep-lat/9704009 [hep-lat].

[76] C. Patrignani et al. "Review of Particle Physics." In: *Chin. Phys.* C40.10 (2016), p. 100001. DOI: 10.1088/1674-1137/40/10/100001.

[77] C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer-Verlag New York, 2004. ISBN: 978-0-387-21239-5.

[78] V.C. Rubin and W.K. Ford. "Rotation of the andromeda nebula from a spectroscopic survey of emission regions." In: *The Astrophysical Journal* 159.3 (1970). DOI: 10.1086/150317.

[79] Jason Sanders and Edward Kandrot. *CUDA by Example: An Introduction to General-Purpose GPU Programming*. 1st. Addison-Wesley Professional, 2010. ISBN: 0131387685, 9780131387683.

[80] Stefan Schaefer, Rainer Sommer, and Francesco Virotta. "Critical slowing down and error analysis in lattice QCD simulations." In: *Nucl. Phys.* B845 (2011), pp. 93–119. DOI: 10.1016/j.nuclphysb.2010.11.020. arXiv: 1009.5228 [hep-lat].

[81] J. Stoer, R. Bartels, W. Gautschi, R. Bulirsch, and C. Witzgall. *Introduction to Numerical Analysis*. Texts in Applied Mathematics. Springer New York, 2013. ISBN: 9780387217383.

[82] Kenneth G. Wilson. "Confinement of quarks." In: *Phys. Rev. D* 10 (8 1974), pp. 2445–2459. DOI: 10.1103/PhysRevD.10.2445. URL: https://link.aps.org/doi/10.1103/PhysRevD.10.2445.

[83] Ulli Wolff. "Collective Monte Carlo Updating for Spin Systems." In: *Phys. Rev. Lett.* 62 (1989), p. 361. DOI: 10.1103/PhysRevLett.62.361.

[84] Boram Yoon, Tanmoy Bhattacharya, and Rajan Gupta. "Neutron Electric Dipole Moment on the Lattice." In: *35th International Symposium on Lattice Field Theory (Lattice 2017) Granada, Spain, June 18-24, 2017*. 2017. arXiv: 1712.08557 [hep-lat]. URL: https://inspirehep.net/record/1644798/files/arXiv:1712.08557.pdf.