Real-Time Complex Langevin - A Differential Programming Perspective

by

Daniel Alvestad

Thesis submitted in fulfilment of
the requirements for the degree of
PHILOSOPHIAE DOCTOR
(PhD)

University
of Stavanger

Faculty of Science and Technology
Department of Mathematics and Physics
2023

University of Stavanger
NO-4036 Stavanger
NORWAY
www.uis.no

ISBN: 978-82-8439-186-1
ISSN: 1890-1387
PhD: Thesis UiS No. 718

Preface

Firstly, I want to thank my supervisor, Prof. Alexander Rothkopf, for his invaluable guidance throughout my research project. His ability to motivate and challenge me, coupled with his unwavering patience, enabled me to navigate the complex aspects of this research project.

I am equally grateful to my colleagues, Gaurang and Rasmus, for their invaluable contributions to my work. Our frequent and diverse discussions have brought fresh perspectives and dimensions to my research. Furthermore, I thank Gerhard, Magdalena, and Abihijit, fellow Ph.D. students in our office, for engaging in fruitful discussions on pertinent and sometimes irrelevant topics. Additionally, I appreciate my fellow Ph.D. students and colleagues at the University of Stavanger for creating a welcoming atmosphere.

Finally, to my partner, Stine, for your tireless support, patience, and encouragement.

Daniel Alvestad
Stavanger, June 2023

Abstract

In this thesis, I aim to find solutions to the NP-hard *sign-problem* that arises when modeling strongly correlated systems in real-time. I will use the complex Langevin (CLE) method, and address its problem of runaway trajectories and incorrect convergence using an implicit solver and a novel kernel optimization scheme, respectively. The implicit solver stabilizes the numerical solution, making the runaway solution problem a thing of the past. It also acts as a regulator, allowing for simulation along the canonical Schwinger-Keldysh contour. Additionally, our investigation shows that a kernel can act as a regulator as well, resulting in an effective change in the action and integral measure while leaving the path integral measure intact.

To restore correct convergence in CLE simulations, we present a novel strategy that involves learning a kernel and utilizing functionals that encode relevant prior information, such as symmetries or Euclidean correlator data. Our approach recovers the correct convergence in the non-interacting theory on the Schwinger-Keldysh contour for any real-time extent. It achieves the correct convergence up to three times the real-time extent of the previous benchmark study for the strongly coupled quantum anharmonic oscillator.

Furthermore, we investigate the stability of the CLE by calculating the Lyapunov exponents of the CLE and uncovering that the real-time CLE behaves like a chaotic dynamical system. This has consequences for obtaining a reliable gradient of a loss function that contains a real-time CLE simulation. To address this issue, we adapt the shadowing sensitivity method to a stochastic differential equation (SDE), which allows for calculating a reliable gradient of chaotic SDEs.

Table of Contents

1 Introduction

Lattice field theory has made significant strides in shaping our understanding of quantum field theory in recent decades. It is often the only tool available to study theories with a non-perturbative behavior where conventional methods based on small coupling fail. The numerical simulation of quantum field theories on a discrete space-time lattice has allowed for precise first-principles calculations of various quantities, including the hadron spectrum [1], heavy quark physics such the spectrum of quarkonia [2] and exotics xyz states [3], as well as confinement in the form of the heavy quark potential. However, despite the success of standard lattice field theory methods in computing static and equilibrium properties, there remains an important class of dynamical phenomena that are inherently real-time and out of equilibrium. We must tame the so-called *sign-problem* to capture these phenomena.

1.1 Motivation

Many-body real-time simulations are essential for understanding dynamical processes, such as charge transport and thermalization, of systems in domains like condensed matter physics, high-energy physics, and nuclear physics. They are needed to calculate transport coefficients such as shear and bulk viscosity, charge conductivity, and diffusion constants, to name a few. In high energy physics, these coefficients play a vital role in understanding the behavior of the quark-gluon-plasma [4] produced in heavy-ion collisions [5] and its transition to the hadronic phase. This is important information for phenomenological studies at, e.g., the Large Hadron Collider located at CERN. For low-energy applications, the ability to reliably calculate transport coefficients of charged fermions [6] in complex atomic lattices is of high importance in the discovery of new high-temperature superconductivity materials [7].

Real-time quantities, such as the transport coefficient, can be formulated via quantum field theory on the so-called Schwinger-Keldysh closed time contour formalism [8, 9]. In this setup, it is real-time correlation functions that are related to these quantities. However, a non-perturbative treatment is necessary for the abovementioned examples, which correspond to strongly correlated quantum systems. This means that the dynamical quantities cannot be computed perturbatively for such systems, and hence we need a non-perturbatively method to obtain the real-time correlation functions.

Lattice field theory is the standard approach for evaluating non perturbatively the static thermodynamical properties such as its equation of state of hot nuclear matter (see, e.g., [10]). With this framework, we can numerically obtain solutions of the Feynman path-integral, which underlies the standard formulation of quantum field theory quantities in lattice field theory. However, a direct computation of a real-time correlator using the lattice field theory framework is prevented by the infamous *sign-prblem* [11, 12]. It describes the problem of having complex-valued weights in the path integral or that the sign of the underlying distribution oscillates. In general, we can say that the problem suffers from a sign-problem if the phase fluctuation increase as the size of the system increases. This prohibits the use of the otherwise efficient Mote-Carlo importance sampling method necessary for numerical computation of the highly dimensional path integral.

Lattice simulation of static systems are formulated along imaginary (Eucledian) time and can, in principle, via analytic continuation, recover real-time information of thermal systems. This is due to the relation between the imaginary time and the physical time (Minkowski) correlators, which shares a common spectral function [13]. The spectral functions, which encapsulate information about a system's dynamical properties and are linked to real-time correlation functions, serve as the most effective means to express thermal real-time physics. However, extracting spectral functions from, e.g., lattice QCD data poses a challenge, as they are concealed within an ill-posed inverse problem. As a result, the spectral reconstruction method tries to overcome this challenge by incorporating prior knowledge on the spectrum.

The ill-posed nature of the problem arises from the fact that lattice QCD only provides Euclidean-time correlation functions, which are related to the spectral functions through an integral transform. To extract the spectral functions, one must invert this transform, which is an ill-conditioned and numerically unstable procedure. Several approaches have been developed to address this issue. To name some, we have the *Bayesian Inference* [13] methods which include the *Maximum Entropy Method* (MEM)[14], *Bayesian reconstruction* [15] and *Tikhonov* [16]. We also have the Bakus-Gilbert [17], Padé method[2], Gaussian Processes reconstruction [18], and approaches directly using machine learning methods[19]. However, the method suffers from systematic uncertainties due to the ill-posed nature of the problem, which emphasizes the importance of developing alternative and more robust real-time simulation techniques.

A different area of study that faces challenges similar to those encountered in investigating dynamical QCD quantities is the computation of many-body systems in low-energy physics [20]. These simulations confront the issue of rapidly increasing complexity as the number of particles in the system grows. For example, while we can numerically solve the Schrödinger equation for systems containing a small number of particles, adding more particles to the problem exponentially increase computational cost, making it intractable without resorting to approximations. Developing new algorithms and methods that can accurately simulate these systems could have far-reaching implications for understanding many-body systems and their applications in materials science, chemistry, and nuclear physics, among others [20]. For example, we can potentially gain valuable insight into high-temperature superconductivity by simulating the Hubbard model in Minkowski time [21].

In light of these limitations and challenges, it is clear that direct real-time simulations are essential for obtaining a more accurate and robust understanding of the dynamical aspects of QCD and other quantum many-body theories. Developing new techniques and improving existing ones to perform real-time simulations will facilitate exploring a wide range of phenomena in high-energy physics, many-body systems, nuclear physics, and early universe cosmology. In addition, direct real-time simulations are crucial for uncovering the rich dynamical structure underlying the fundamental interactions in non-perturbative many-body physics.

Various approaches have been developed to tackle the sign problem arising in dynamical systems. This includes the complex Langevin framwork [22], which evolves a complexified field as a stochastic process in a fictitious time. The Lefschetz thimble methods [23, 24], which is a group of methods that seek to reduce the sign-problem by deforming the integration contour into the complex plane in such a way that the imaginary part of the Feynman weight remains constant. Tensor network techniques [25, 26] is a method that originates from condensed matter physics, which represents the quantum many-body states based on their entanglement structure.

In recent years, many of the improvements made in the field of lattice simulations and extraction of spectral function have been made by applying machine learning techniques [27]. This has seen successes such as improved sampling algorithms using normalizing flows [28] and improved spectral reconstruction methods [19, 29]. These techniques have also been used successfully in some of the Lefschetz thimbles methods, where machine learning algorithms have been used to find optimal integration contours [24]. All of these methods rely on optimization of a large number of parameters, which is usually done using a gradient-based optimization scheme. To this end, obtaining efficient and reliable gradients of quantities related to the method at hand is important.

Differential programming is a paradigm that enables easy computation of derivatives [30]. The basic building block of a differential programming system is automatic differentiation, which computes machine-precision derivatives with minimal changes to the original code. This technique can be beneficial for quick implementation of gradients, Jacobians, and Hessians, which in turn are used in optimization algorithms [30]. An example of such a case is implementing implicit differential equation solvers, where the (quasi) Newton method to solve the non-linear system needs the (gradient) Hessian, which only requires the user to provide the ODE function. Then the AD framework handles the (gradient) Hessian.

Differential programming can be seen as a generalization of machine learning implementations, as it builds on the idea of solving problems by optimizing a functional. Furthermore, this framework allows for optimizing arbitrary model structures, making it applicable to any method containing free parameters. For example, differential programming can be used to optimize the complex integration path in contour deformation methods [24] or adjust free parameters in the complex Langevin method [31].

It has been shown that the sign problem belongs to the class of NP-hard computational problems [32], which indicates that an efficient and universally applicable solution may not exist. Differential programming can aid in exploring different methodologies and optimizing the freedom inherent in these methods. As long as the gradient is defined, the ability to compute gradients for any user-defined function allows for a wide range of applications and problem-solving techniques to be applied in tackling the sign problem. It also simplifies code, increases flexibility, reduces errors when implementing gradient-based algorithms, and makes it possible to implement new methods that have not been tested in the past.

1.2 Short introduction to essential concepts

In this section, I will introduce some basic concepts to motivate the following chapters. This is not a complete introduction to the topics, as the following section will only introduce the concepts without derivations. For further reading on thermal field theory and lattice field theory, see. [33, 10, 34].

1.2.1 Real-time formulation

Real-time QFT evolves a system in a mixed initial state (ρ) in Minkowski time. The expectation value of an observable O is defined as $\langle O \rangle = \mathrm{Tr}[\rho O]$. To study such a system, we use the Schwinger-Keldysh closed time-path formalism, which formulates the system's time evolution in terms of two contour paths. This can be derived by the insertion of a complete set of field states in the forward path (ϕ^+) and the same for the backward branch (ϕ^-), such that the trace is written as an integral [34]

$$\langle O(\phi)\rangle = \frac{1}{Z}\int d\phi_1 d\phi_2\ \rho(\phi_1,\phi_2)\int_{\phi_2}^{\phi_1} D\phi^+ D\phi^-\ O(\phi)\ e^{iS[\phi_+]-iS[\phi_-]}. \tag{1.1}$$

The initial density matrix ρ connects to the starting point ϕ_1 and endpoint ϕ_2 of the closed-time contour.

In the thermal equilibrium setting with inverse temperature $\beta = 1/T$, the density matrix ρ takes the form of the standard Boltzmann weight $\rho = \exp[-\beta H]$ [33]. By identifying this with the Eucledian time evolution operator in compactified imaginary time with time length $-i\beta$, we can rewrite the initial density matrix ρ in terms of S_E

$$\langle O(\phi)\rangle = \frac{1}{Z}\int D\phi_E e^{-S_E[\phi_E]}\int_{\phi_E(\beta)}^{\phi_E(0)} D\phi^+ D\phi^-\ O(\phi)\ e^{iS[\phi_+]-iS[\phi_-]}. \tag{1.2}$$

We see that for a time contour with a real-time extent of zero, i.e., only sampling from the initial density matrix, we recover the usual Euclidean simulation, defined in the next section. This is why we have noted the action inserted from the density matrix as S_E. We can combine the actions into $iS[\phi] = iS[\phi_+] - iS[\phi_-] - S_E[\phi_E]$, such that the path integral can be written in the concise notation

$$\langle O(\phi)\rangle = \frac{1}{Z}\int D\phi\ O(\phi)e^{iS[\phi]}, \tag{1.3}$$

where the integration contour is part of the action.

1.2.2 Euclidean quantum field theory

It is not generally possible to directly treat numerically the path integral defined in the previous section (eq. (1.3)). This is normally circumvented by doing a Wick rotation from Minkiowski time to Eucledian time, which is a continuation from t to $i\tau$, where τ is the Euclidean time. The resulting expression for the path integral of an observable O is in the Eucledian formalism

$$\langle O(\phi)\rangle = \frac{1}{Z}\int D\phi\; O(\phi)e^{-S_E[\phi]}, \quad S_E[\phi] = \int_0^\beta d\tau L_E[\phi] \tag{1.4}$$

which now creates exponential damping in the Boltzmann weight instead of the highly oscillating complex weight. The Eucledian time is compact, restricted to the compact interval $\tau \in [0, \beta]$, such that $\phi(0) = \phi(\beta)$. The trace in the partition function demands this periodicity. We can compute static thermodynamical quantities using the Euclidean path integral, but it does not give access to dynamical quantities. In the view of eq. (1.2), we can think of it as the initial condition of a thermal system, where the dynamics of the system are described in the path integral over the ϕ^+ and ϕ^- states.

1.2.3 Monte-Carlo simulations

Monte Carlo simulations represent a powerful and versatile method to explore non-perturbative systems numerically and are notably employed in lattice field theory in Euclidean time. The lattice field theory formulation discretizes spacetime into a lattice, effectively transforming the continuous infinite-sized spacetime of quantum field theory into a finite, computationally tractable system. The challenge is to compute the path integral over all allowed field configurations on the lattice, which is where Monte Carlo simulations come in.

Monte-Carlo integration is a way of numerically computing integrals by stochastically evaluating the integrand. In lattice field theory, this method is applied to the path integral. Even though we could sample the field variables uniformly, this leads to a highly inefficient sampling, as most field configurations contribute only a small part to the total integral. The idea of importance sampling solves this problem by sampling the configurations weighted according to the value of the Boltzmann weight; e^{-S_E}, i.e., the higher the value of the Boltzmann weight, the more of the corresponding configuration should appear in the ensemble. Two examples of implementations of importance sampling are the Metropolis algorithm [35] and the Hybrid Monte-Carlo (HMC) [36] method.

Given that we have sampled N configurations ϕ_i, distributed according to the Boltzmann weight, observables can be estimated by the sample expectation value of the configurations, i.e.,

$$\langle O \rangle = \frac{1}{N} \sum_i^N O(\phi_i). \tag{1.5}$$

Here we have denoted each configuration by ϕ_i, where i denotes the configuration number. Even though expectation value notation is used, this is only an estimated expectation value. The statistical error of the estimated expectation value can be estimated by

$$\sigma_O = \sqrt{\frac{1}{N(N-1)} \sum_i^N (O(\phi_i) - \langle O \rangle)^2} \tag{1.6}$$

where we have assumed the configurations to be independent. For correlated configurations, which is the case for most importance sampling algorithms, if the configurations are sampled at every update, the autocorrelation must be calculated and then add a correction factor to N in the equation above.

1.2.4 Sign problem

The most common way to numerically solve the path integral with a real Feynman weight is to use Monte-Carlo sampling, as described in the previous section. On the other hand, the distribution is highly oscillatory for a complex Feynman weight $\exp[iS]$, such as in the real-time scenario section 1.2.1. To obtain an accurate result using importance sampling, cancellation of contributions of the same order of magnitude must be taken care of [11, 12]. This also includes the contributions from the exponentially suppressed tails, which becomes equally important. This means that the amount of sampling necessary to obtain an accurate result increases exponentially. This prohibits the use of Monte-Carlo methods for problems with even a small sign problem [11, 12].

For example, we can investigate a simple model given by the partition function $Z = \int dx \exp[-x^2 + i\lambda x]$, a Gaussian with an additional complex

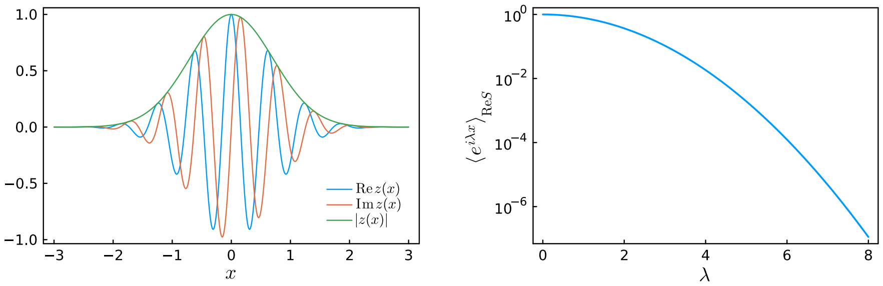

Figure 1.1: (left) Distribution of the Feynman weight $z(x) = e^{-x^2+i\lambda x}$ for $\lambda = 10$, showing the real and imaginary part, including the absolute value which yields a Gaussian distribution with mean 0 and unit variance. (right) The average phase $\langle e^{i\lambda x}\rangle_{\mathrm{Re}S}$ with changing λ.

phase. This is an instructive example, as discussed in ref. [37], which shows the oscillatory behavior of a complex-valued Feynman weight. In fig. 1.1, we show the integrand for $\lambda = 0$, i.e., the real case, and $\lambda = 10$. We see that both $\mathrm{Re}z(x)$ and $\mathrm{Im}z(x)$ oscillate heavily, which means that to obtain a convergent expectation value we need precise cancelations between the negative and positive parts.

To quantify this behavior, we need to calculate the average phase, which is given by $\frac{\int dx O e^S}{\int dx|e^S|} = \frac{Z(\lambda)}{Z(0)} = \langle e^{i\lambda x}\rangle_{\mathrm{Re}S}$, where we have used that $|e^{i\lambda x}| = 1$. For this simple model, we can obtain the average phase analytically by integrating it; this is given by $e^{-\lambda^2/4}$ [37]. We see that large cancellations between the positive and negative parts of the integrand are needed for the expectation value to converge correctly, this requires a large amount of sampled configurations to get the statistic error smaller than the average phase. For a small λ, reweighting is possible

$$\langle O\rangle_S = \frac{\int dx O(x) e^{-x^2+i\lambda x}}{\int dx e^{-x^2+i\lambda x}} = \frac{\int dx\ O(x)\frac{e^{-x^2+i\lambda x}}{e^{-x^2}}e^{-x^2}}{\int dx\ \frac{e^{-x^2+i\lambda x}}{e^{-x^2}}e^{-x^2}} = \frac{\langle O e^{i\lambda x}\rangle_{\mathrm{Re}S}}{\langle e^{i\lambda x}\rangle_{\mathrm{Re}S}} \tag{1.7}$$

where the denominator becomes the average phase. For λ close to zero, using the reweighting method eq. (1.7) is possible as the amount of

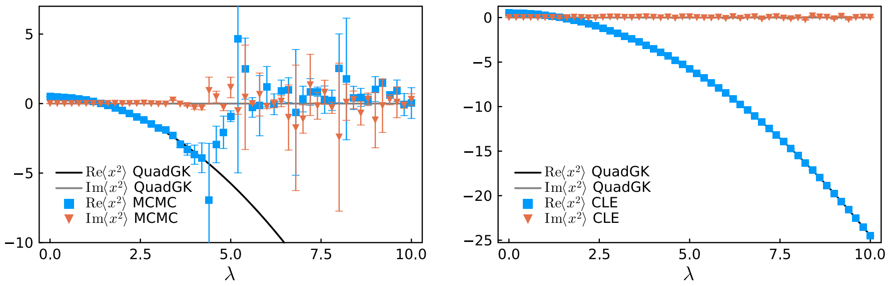

Figure 1.2: (Left) Using the reweighting method (eq. (1.7)) to calculate the $\langle x^2\rangle$ observable for an increasing λ, shows as the colored data points. The method fails for a λ around 3.5. The black and gray lines show the solution to the integral using a numerical quadrature integration; this is done using the Gauss-Konrad quadrature (QuadGK). (Right) Solving the same problem on the left using the complex Langevin equation (CLE). This method has no problems with the increasingly complex phase of this system.

statistics required for the expectation values to converge is with standard Monte-Carlo sampling of the real part of the action. The average phase becomes too small for $\lambda \gg 1$. Hence we need a large amount of configurations to resolve the expectation value of the observable.

In the left panel of fig. 1.2, we have plotted the result of the reweighting technique using the HMC importance sampling scheme. We see that when we get close to $\lambda = 4$, which corresponds to an average phase of $\langle e^{i\lambda x}\rangle_{\mathrm{Re}S} \sim 0.1$ in fig. 1.1, the errorbars increase due to lack of statistics, and after $\lambda = 5.0$ we are unable to get any useful information out of the configurations. To obtain a reliable result for $\lambda > 5.0$, we need to sample exponentially many configurations for increasing λ. This makes it infeasible to use Monte-Carlo simulation in combination with reweighting when the average phase becomes small.

As a precursor of the next chapter, we also demonstrate how the complex Langevin behaves for this problem; this is shown in the right panel of fig. 1.2. The complex Langevin method has no problem with this added phase in the Gaussian Feynman weight. Details on the method will be discussed in the next chapter.

1.3 Contribution of this thesis

This thesis focuses on real-time simulation using the complex Langevin method. The goal has been to stabilize and correct the convergence problem of the complex Langevin equation, with a focus on the application to real-time simulation. The main contribution of this thesis can be summarised in the following points,

- Implementation of an implicit scheme to solve the real-time complex Langevin equation, part of [38] and content of chapter 4.
  - Understanding how an implicit scheme effectively introduces a regulator in the real-time action
  - Simulating, for the first time, the anharmonic oscillator directly on the canonical real-time contour up to a short real-time with a perfect agreement with the Schrödinger solution both for a thermal action and a non-equilibrium action.
- Extending the original correctness criterion [39] to the kernel-controlled complex Langevin (KCLE) and comparing KCLE to the Lefschetz Thimbles. This is the main part of chapter 5, which is published in the appendix of [31], and also the main content of the proceeding [40].
  - Boundary terms are calculated for different toy models using different kernels. We obtain zero boundary terms for some positive definite kernels while the complex Langevin equation converges incorrectly.
- Application and optimization of a kernel-controlled complex Langevin equation in the context of real-time simulations. This is the main content of [31], which is discussed in chapter 6. Part of this work is also published in the proceeding [41].
  - First simulation of the anharmonic oscillator free theory at arbitrary real-time extent using a kernel in the complex Langevin equation.

  - I have shown analytically for a simple toy model and numerically for larger systems that a kernel in the complex Langevin equation can effectively introduce a regulator in real-time actions.
  - Introduced a machine learning-inspired scheme to optimize a kernel in the complex Langevin equations to recover correct convergence.
  - Introduced a low-cost loss function that can be used to find a suitable kernel. It allowed us to extend the range of reliability three times the previously feasible real-time extent of the anharmonic oscillator.

The remaining part of this thesis is unpublished and work in progress related to applying adjoint sensitivity methods to optimize kernels of the complex Langevin equation.

- Introduced and applied the adjoint sensitivity method to the complex Langevin equation, using a simple toy model as an example. This can be found in section 3.3.
- I show that the complex Langevin dynamics for real-time systems behave chaotically for real-time extent above the point where the complex Langevin starts to converge to the wrong solution. This is the first part of chapter 7.
- I have obtained a reliable gradient of the stochastic Lorenz system using the NILSAS scheme, which has not been achieved before, to my knowledge. This is the first step in obtaining a reliable adjoint sensitivity method for the KCLE. This is the last part of chapter 7.

1.4 Outline of thesis

The thesis is structured as follows; In chapter 2, the stochastic quantization and the complex Langevin method are introduced along with kernel-controlled complex Langevin. Chapter 3 delves into differential programming concepts, highlighting the adjoint sensitivity method and presenting a use case with the complex Langevin equation of a simple toy model.

Chapter 4 explores the application of implicit solvers for numerically solving the complex Langevin. This chapter demonstrates how the implicit scheme can achieve unconditionally stable real-time simulation. In chapter 5, the correctness criterion for the complex Langevin equation with a kernel is derived, and connections to Lefschitz thimbles are made.

The content of chapter 6, introduces a novel scheme for optimizing kernels, and a low-cost method for addressing the real-time problem is developed. This chapter also investigates the use of a kernel as a regulator for the real-time complex Langevin equation.

Lastly, chapter 7 discusses the limitations of the standard adjoint sensitivity method for the real-time complex Langevin method. It then explores a possible way to obtain a reliable gradient for the complex Langevin, which can be used in the optimization scheme presented in chapter 6.

In the appendices, I have included a discussion and example of implicit solvers for the complex Langevin applied to Gauge theories (appendix A), and in chapter D an example of the use of non-linear adjoint sensitivity method is applied to a Bayesian spectral reconstruction inference problem using a Generalized Thikonov regulator.

2 The Langevin Framework

Quantum expectation values in Eucledian QFT can be expressed by Feynman path integral

$$\langle O(\phi)\rangle = \frac{1}{Z}\int D\phi \; O(\phi)\; e^{-S(\phi)}, \tag{2.1}$$

where $Z = \int D\phi \;\; e^{-S(\phi)}$ is the partion funciton. In *Stochastic Quantization*, we consider the path integral measure $\frac{1}{Z}e^{-S(\phi)}$ as a probability distribution and construct a stochastic process with this distribution as its stationary distribution. This is similar to obtaining the Boltzmann distribution from the stochastic process of a Brownian particle from statistical physics. The path integral can be formulated in terms of a stochastic variable ϕ, which was proposed by Parisi and Wu [22] to evolve in a fictitious time τ_{L}, which we shall refer to as Langevin time. The degrees of freedom in the system, therefore, depend on this additional time dimension such that

$$\phi(x) \longrightarrow \phi(x, \tau_{\mathrm{L}}). \tag{2.2}$$

where x donates a vector in n-dimensional Euclidean space. Note that the Langevin time τ_{L} should not be confused by the Eucledian time. The fields evolve in Langevin time via the Langevin equation

$$\frac{\partial\phi(x,\tau_{\mathrm{L}})}{\partial\tau_{\mathrm{L}}} = -\frac{\delta S(x)}{\delta\phi(x,\tau_{\mathrm{L}})} + \eta(x,\tau_{\mathrm{L}}) \tag{2.3}$$

with the stochastic noise term following a Gaussian distribution;

$$\langle\eta(x,\tau_{\mathrm{L}})\rangle = 0, \quad \langle\eta(x,\tau_{\mathrm{L}})\eta(x,\tau'_{\mathrm{L}})\rangle = 2\delta^n(x-x')\delta(\tau_{\mathrm{L}}-\tau'_{\mathrm{L}}). \tag{2.4}$$

At this point, it is instructive to write the Langevin equation as a mathematically well-defined stochastic differential equation[1]. We rewrite the integral as a stochastic integral in either Ito or Stratonovich form. A stochastic integral is an extension of the standard integral and is defined

[1] η is non-differentiable, hence eq. (2.3) is not a well-defined equation as it contains a derivative of the stochastic process

as the limit of a sum of random variables multiplied by the corresponding Wiener increments dW. This is defined as the difference between two Wiener processes with an infinitesimal time change. A Wiener process $W(\tau) = \int \eta(\tau)d\tau$ can be defined via $\langle W(\tau_\mathrm{L})\rangle$ and $\langle W(\tau_\mathrm{L})^2\rangle = t$, with independent increments that are normally distributed with zero mean and a variance of $d\tau_\mathrm{L}$, i.e., $W(\tau_\mathrm{L} + d\tau_\mathrm{L}) - W(\tau_\mathrm{L}) \sim \mathcal{N}(0, d\tau_\mathrm{L})$. The forward Wiener process difference gives Ito's version of the stochastic integral

$$\int f(\tau_\mathrm{L})\, dW(\tau_\mathrm{L}) = \lim_{n\to\infty} \sum_{i=0}^{n-1} f(\tau_\mathrm{L}^i)(W(\tau_\mathrm{L}^{i+1}) - W(\tau_\mathrm{L}^i)) \tag{2.5}$$

while the Stratonovich version is given by evaluating $f(\tau_\mathrm{L})$ at the midpoint in each subinterval. As the noise term in the stochastic process (eq. (2.3)) does not depend on the Langevin time τ_L or the field ϕ, the two flavors of the stochastic integral are equivalent. Equation (2.3) is now expressed as

$$d\phi(x, \tau_\mathrm{L}) = -\frac{\delta S}{\delta\phi(x, \tau_\mathrm{L})} d\tau_\mathrm{L} + dW(x, \tau_\mathrm{L}) \tag{2.6}$$

where dW is called a *Wiener increments* following

$$\langle dW(x, \tau_\mathrm{L})\rangle = 0, \quad \langle dW(x, \tau_\mathrm{L}) dW(x', \tau_\mathrm{L})\rangle = 2\delta(x - x')d\tau_\mathrm{L}. \tag{2.7}$$

To show that we obtain the correct unique stationary distribution, i.e., $\exp[-S]$, we need to find the equation describing the evolution of the probability distribution $P(\phi, \tau_\mathrm{L})$. This equation is called the Fokker-Planck equation. The stationary distribution should, by definition, then be proportional to $\exp[-S]$, i.e., $\lim_{\tau_\mathrm{L}\to\infty} P[\phi, \tau_\mathrm{L}] \sim \exp[-S]$. In the following, we will follow closely [42] while simplifying the notation, where possible, such that $\phi = \phi(x, \tau_\mathrm{L})$.

The evolution equation of the distribution $P[\phi, \tau_\mathrm{L}]$ can be derived by investigating the change in an observable due to a change in the Langevin time τ_L. The expectation value of an observable can be calculated from the distribution P by

$$\langle O(\phi, \tau_\mathrm{L})\rangle = \int D\phi P[\phi, \tau_\mathrm{L}] O(\phi, \tau_\mathrm{L}), \tag{2.8}$$

and the change is then given by

$$\frac{d\langle O(\phi\rangle}{d\tau_{\mathrm{L}}} = \int D\phi \frac{dP[\phi, \tau_{\mathrm{L}}]}{d\tau_{\mathrm{L}}} O(\phi). \tag{2.9}$$

Note that the observable is assumed to not depend explicitly on the Lagenvin time. We must establish an expression for $\frac{dP}{d\tau_{\mathrm{L}}}$ to obtain the Fokker-Planck equation. This can be achieved by deriving the expression for the Langevin time derivative of the observable in an alternative way, using the definition of the Langevin equation. To this end, we expand

$$dO(\phi) = \frac{\delta O(\phi)}{\delta\phi} d\phi + \frac{1}{2}\frac{\delta^2 O(\phi)}{\delta\phi^2} d\phi^2 \tag{2.10}$$

where we have truncated the expansion at $d\phi^2$ since $d\phi^2 \approx dW^2 = 2d\tau_{\mathrm{L}}$. Higher orders will, at most, be proportional to $d\tau_{\mathrm{L}}^{3/2}$, which can be neglected if we treat $d\tau_{\mathrm{L}}$ as infinitesimal. Using eq. (2.6), we can write the expectation value as

$$\langle dO(\phi)\rangle = \left\langle -\frac{\delta O(\phi)}{\delta\phi}\frac{\delta S}{\delta\phi} + \frac{\delta^2 O(\phi)}{\delta\phi^2} \right\rangle d\tau_{\mathrm{L}}. \tag{2.11}$$

Now the change in the expectation value of the observable in an infinitesimal Langevin time interval ($d\tau_{\mathrm{L}}$) is given by

$$\frac{d\,\langle O(\phi)\rangle}{dt} = \left\langle -\frac{\delta O(\phi)}{\delta\phi}\frac{\delta S}{\delta\phi} + \frac{\delta^2 O(\phi)}{\delta\phi^2} \right\rangle = \langle L_r O\rangle. \tag{2.12}$$

where we have defined the Langevin operator $L_r = \int d^n x \left(\frac{\delta}{\delta\phi} - \frac{\delta S[\phi]}{\delta\phi}\right)\frac{\delta}{\delta\phi}$. Rewriting the above expression as an integral over the field degree of freedoms, using eq. (2.8), and the probability distribution $P[\phi, \tau_{\mathrm{L}}]$ yields

$$\begin{aligned} \frac{d\,\langle O(\phi)\rangle}{d\tau_{\mathrm{L}}} &= \int D\phi \left[-\frac{\delta O(\phi)}{\delta\phi}\frac{\delta S[\phi]}{\delta\phi} + \frac{\delta^2 O(\phi)}{\delta\phi^2} \right] P[\phi, \tau_{\mathrm{L}}] \\ &= \int D\phi\, O(\phi) \left[\frac{\delta}{\delta\phi}\frac{\delta S[\phi]}{\delta\phi} + \frac{\delta^2}{\delta\phi^2} \right] P[\phi, \tau_{\mathrm{L}}]. \end{aligned} \tag{2.13}$$

In the second equation, we have used integration by parts, assuming the probability distribution to vanish at the boundaries. This is an important step that plays a critical role in our understanding of the validity, and we will return to it when discussing the validity of the complex Langevin.

We can now read off the Fokker-Planck equation by comparing the equation above to eq. (2.9). Hence the Fokker-Planck equation is given by

$$\frac{dP[\phi,\tau_\mathrm{L}]}{d\tau_\mathrm{L}} = \int d^n x \frac{\delta}{\delta\phi}\left[\frac{\delta}{\delta\phi} + \frac{\delta S[\phi]}{\delta\phi}\right] P[\phi,\tau_\mathrm{L}] = L^T P[\phi,\tau_\mathrm{L}]. \tag{2.14}$$

The Fokker-Planck operator, L^T, is the formal adjoint operator to the Langevin operator L_r. At this point, we will discretize the action, going from an infinite-dimensional Fokker-Planck equation (infinite due to the integral over all space-time x) to a discrete space-time noted by x_i. This replaces the integral in eq. (2.14) to a sum, partial derivatives replace the functional derivatives, and the fields are written as $\phi(x_i) = \phi_i$. The action-discretized Fokker-Planck equation is now written as

$$\frac{dP[\phi,\tau_\mathrm{L}]}{d\tau_\mathrm{L}} = \sum_i \Delta x \frac{\partial}{\partial\phi_i}\left[\frac{\partial}{\partial\phi_i} + \frac{\partial S[\phi]}{\partial\phi_i}\right] P[\phi,\tau_\mathrm{L}] = L^T P[\phi,\tau_\mathrm{L}], \tag{2.15}$$

where we note that we have not discretized the Langevin time.

We can see directly from the evolution equation above that the stationary distribution, which follows from $\frac{dP}{dt} = 0$, is satisfied if

$$\left[\frac{\partial}{\partial\phi} + \frac{\partial S}{\partial\phi}\right] P[\phi,\tau_\mathrm{L}] = 0 \tag{2.16}$$

which is true for $P(\phi,\tau_\mathrm{L}) = \exp[-S]$. We can show that this stationary distribution is unique by performing a similarity transform

$$\tilde{P}[\phi,\tau_\mathrm{L}] = e^{S[\phi]/2} P[\phi,\tau_\mathrm{L}] \tag{2.17}$$

such that the Fokker-Planck (FP) operator is transformed into a self-adjoint and positive semi-definite operator

$$\tilde{L}^T = e^{S[\phi]/2} L^T e^{-S[\phi]/2} = \sum_i \left(\frac{\partial}{\partial \phi_i} - \frac{1}{2} \frac{\partial S[\phi]}{\partial \phi_i} \right) \left(\frac{\partial}{\partial \phi_i} + \frac{1}{2} \frac{\partial S[\phi]}{\partial \phi_i} \right). \tag{2.18}$$

We can now see that the modified FP operator has a unique ground state $\psi_0 = e^{-S[\phi]/2}$, and since it is negative semi-definite, all the eigenvalues, except for the zero mode, are negative. This can be seen by multiplying out the parentheses, obtaining two terms; a second derivative operator $\frac{\partial^2}{\partial \phi^2}$ which is negative semi-definite, and $-\frac{1}{4}\left(\frac{\partial S[\phi]}{\partial \phi}\right)^2$ which is negative definite. By expanding the distribution $\tilde{P}$ in the complete set of eigenfunctions, we get

$$\tilde{P}[\phi, \tau_{\mathrm{L}}] = a_0 e^{-S[\phi]/2} + \sum_{n=1}^{\infty} a_n \psi_n e^{\lambda_n \tau_{\mathrm{L}}}. \tag{2.19}$$

In the limit $\tau_{\mathrm{L}} \longrightarrow \infty$, the stationary distribution $a_0 e^{-S[\phi]/2}$ is ensured due to the exponential decay of the non-stationary contributions. By a similarity transform back again, we obtain

$$\lim_{\tau_{\mathrm{L}} \to \infty} P[\phi, \tau_{\mathrm{L}}] \sim e^{-S[\phi]} \tag{2.20}$$

which is the desired stationary distribution, i.e., the Boltzmann weight.

This means that since the Langevin process converges towards the distribution of the Fokker-Planck equation, expectation values of an observable O can be obtained by the integral

$$\langle O \rangle = \lim_{T \to \infty} \frac{1}{T} \int_0^T d\tau_{\mathrm{L}} O(\phi(\tau_{\mathrm{L}})). \tag{2.21}$$

The field $\phi(\tau)$ is the solution to the Langevin equation. In practice, the Langevin equation needs to be solved numerically. This means that we must discretize the Langevin time $\tau_{\mathrm{L}} \longrightarrow \Delta\tau_{\mathrm{L}}$, which also discretize the Wiener increments $dW \longrightarrow \Delta W$ such that $\Delta W \sim \mathcal{N}(0, \Delta\tau_{\mathrm{L}})$. The calculation of observables is now changed from an integral to a sum over measurements $\phi(\tau_\mathrm{L}^i)$ along Langevin time

$$\langle O\rangle \approx \frac{1}{N}\sum_i^N \Delta\tau_\mathrm{L} O(\phi(\tau_\mathrm{L}^i)), \tag{2.22}$$

which, due to discretization and finite simulation time T, is an approximation of the observable. Details on how to solve the Langevin equation numerically will be discussed in detail in chapter 4.

As discussed in this section for a real action, the Langevin equation is a mathematically well-defined method to sample configurations from the distribution $\exp(-S)$, with $S \in \mathbb{R}$. However, in this thesis, we are interested in sampling $\exp(S')$ from a distribution where $S' \in \mathbb{C}$. This is the topic of the following sections.

2.1 Complex Langevin

We now consider the Feynman weight of real-time QFT in eq. (2.1), which amounts to a complex-valued expression, and note the exponent by $iS \in \mathbb{C}$, where the action S can also be complex-valued. The Feynman path integral is now written as

$$\langle O\rangle = \frac{1}{Z}\int D\phi\ O[\phi]e^{iS[\phi]} \tag{2.23}$$

where $Z = \int \mathcal{D}\phi \exp[iS[\phi]]$ denotes the partition function. The distribution e^{iS} is now complex, while the fields ϕ are still real quantities. Parisi [43] and Klauder [44], propose to write the complex Langevin, which is eq. (2.6) applied to a complex action iS, as follows

$$d\phi = i\frac{\delta S[\phi]}{\delta\phi(x)}dt + dW(x) \tag{2.24}$$

where dW is the real Wiener increments from eq. (2.7). The field ϕ obtains an imaginary component during the evolution and is therefore complexified. We can rewrite the evolution equations instead in terms of the real and imaginary part of the field as

$$\phi(x,\tau_\mathrm{L}) = \phi_R(x,\tau_\mathrm{L}) + i\phi_I(x,\tau_\mathrm{L}), \tag{2.25}$$

such that eq. (2.24) turns into two coupled but real-valued equations for the real- and imaginary part of the field

$$\begin{aligned} d\phi_R &= \mathrm{Re}\left[i \frac{\delta S[\phi]}{\delta\phi(x)}\bigg|_{\phi=\phi_R+i\phi_I} \right] d\tau_{\mathrm{L}} + dW(x), \\ d\phi_I &= \mathrm{Im}\left[i \frac{\delta S[\phi]}{\delta\phi(x)}\bigg|_{\phi=\phi_R+i\phi_I} \right] d\tau_{\mathrm{L}}. \end{aligned} \tag{2.26}$$

We have used the standard construction where the noise term $dW(x)$ is real. There is, however, freedom in the Fokker-Planck equation to rotate the noise into the complex plane, but this is not discussed in this thesis[2]; see, e.g., [45] for more details.

After complexifying the fields and separating the real and imaginary parts of the evolution, the distribution we are sampling from is now a real and positive density $P[\phi_R, \phi_I, \tau_{\mathrm{L}}]$. It is not a priori known whether this is equivalent to the complex distribution we expect, and hence we need to show that $\lim_{\tau_{\mathrm{L}}\to\infty} P[\phi_R, \phi_I, \tau_{\mathrm{L}}]$ is equivalent to e^{iS}.

The corresponing Fokker-Planck equation for the complex Langevin equation (eq. (2.26)) evolves a real and positive probability density $P[\phi_R, \phi_I, \tau_{\mathrm{L}}]$ according to

$$\frac{\partial P[\phi_R, \phi_I, \tau_{\mathrm{L}}]}{\partial \tau_{\mathrm{L}}} = L^T P[\phi_R, \phi_I, \tau_{\mathrm{L}}] \tag{2.27}$$

where the Fokker-Planck operator L^T is given by

$$L^T = \frac{\partial}{\partial\phi_R}\left[\frac{\partial}{\partial\phi_R} - \mathrm{Re}\left\{ i\frac{\partial S[\phi]}{\partial\phi}\right\}\right] - \frac{\partial}{\partial\phi_I}\mathrm{Im}\left\{ i\frac{\partial S[\phi]}{\partial\phi}\right\}. \tag{2.28}$$

This can be derived in the same way as in the real-Langevin case by considering the real and imaginary parts of the fields, in the complex Langevin, as two separate dimensions, such that the total number of dimensions is twice that as of the real-Langevin case. However, following the same steps as we did for the real-Langevin when attempting

[2] This freedom is different from the one we will use to insert a kernel in later sections. Inserting a complex noise does not alter the drift term.

to prove the desired stationary distribution, we end up with a Fokker-Planck operator that is neither Hermitian nor positive semi-definite such that the proof of convergence is spoiled. To show the convergence of complex Langevin, we need to compare this Fokker-Planck equation to the complex Fokker-Planck equation evolving the complex distribution $\rho(\phi, \tau_{\rm L})$;

$$\frac{\partial \rho[\phi, \tau_{\rm L}]}{\partial \tau_{\rm L}} = F_{FP}\rho[\phi, \tau_{\rm L}], \quad F_{FP} = \frac{\partial}{\partial \phi}\left[\frac{\partial}{\partial \phi} - i\frac{\partial S[\phi]}{\partial \phi}\right] \tag{2.29}$$

where F_{FP} is called the complex Fokker-Planck operator.

To prove that the stationary distribution of the complex Fokker-Planck equation is indeed e^{iS}, we again perform a similarity transform of both distribution and operator (as in eqs. (2.17) and (2.18)). We used the modified Fokker-Planck operator in the real action scenario to show that the stationary distribution is e^{-S}. However, this is not generally the case in the complex Langevin, as the complex Fokker-Planck operator suffers from being non-Hermitian and, in some cases, even a non-normal operator. Klauder and Peterson [46] noted a shortage of theorems that address unbounded, non-normal operators that can be utilized to support statements of this nature. This statement is still true today.

2.2 Challenges

Now that we understand the complex Langevin framework, I will discuss the challenges facing the method. These challenges can be broadly classified into mathematical foundations and practical issues. The practical aspect encompasses two main problems, namely runaway solutions and convergence to incorrect solutions, which will be examined thoroughly in this thesis. On the other hand, the mathematical foundation requires a formal proof that the Fokker-Planck operators (F_{FP}, L^T) and the Langevin operator (L) generate a unique stationary distribution [47, 48]. This thesis concentrates on resolving the practical challenges mentioned above.

The first challenge is the instability problem, commonly known as runaway solutions, which was observed early in applying the Complex

Langevin method [49, 50]. These instabilities manifest as diverging trajectories from overshooting during large excursions from stable points. In some models, the drift term's streamlined plot reveals paths leading to infinity within a finite time[51, 52]. Although the noise term should prevent such trajectories, the drift term can sometimes dominate, such that $\frac{\partial S}{\partial \phi}\Delta \gg \Delta W$, rendering the theory close to being deterministic. This dominance can lead to numerical instabilities, causing runaway solutions. Initial attempts to mitigate this issue involved reducing the step size; however, this approach was computationally expensive and generally ineffective.

These runaway trajectories should be distinguished from divergent trajectories formed in models where the path integral itself is diverging. Models like this need an inherent regulator in both the path integral and the complex Langevin equation. We can say the stochastic differential equation is ill-defined for such problems without a regulator, as the divergent trajectories do not occur due to numerical instabilities. For well-defined problems, the appearance of runaway trajectories is caused by numerical instabilities. These systems are often called stiff systems.

With the introduction of adaptive step-size schemes [52], the issue of runaway trajectories was overcome for a wide range of problems. This scheme adjusts the step size during the simulation, decreasing it when the drift term is large and increasing it when it is small. Alternatively, an implicit scheme can be employed to address instability issues [38], which will be the primary focus of chapter 4. For this scheme, an adaptive step-size scheme is unnecessary to prevent runaway trajectories, as the scheme is stable for all step sizes. This is because of an inherent undershooting of the true solution, which avoids divergent trajectories. Even though the individual steps of the scheme are more costly, since a non-linear equation must be solved at every step, the implicit scheme can, in cases with highly stiff SDEs, be more efficient than the adaptive step size as it does not require a small step size to avoid numerical instabilities.

The second challenge concerns the convergence to incorrect solutions. This issue was identified early on [53] and led to a decline in interest in complex Langevin during the 1990s and early 2000s. The reasons for convergence to the wrong solution are multifaceted. While it is tempting to attribute its convergence failures to the lack of formal proof for the method, there are examples where the complex Langevin converges correctly despite lacking formal justification [46]. Therefore, we separate the problem of convergence from the mathematical proof and investigate why convergence breaks down. The failure of convergence has been linked to the following factors: the stationary distribution of complex Langevin not being e^{iS} [54], the occurrence of "boundary terms at infinity [39, 55], and ergodicity problems due to poles in the drift term [56]. We will discuss the first two factors in greater detail below.

The failure of the following limit can characterize the first issue:

$$\lim_{\tau_{\mathrm{L}}\to\infty} \rho[\phi, \tau_{\mathrm{L}}] \overset{?}{\propto} e^{iS}. \tag{2.30}$$

To investigate the cause of this failure, we need to revisit the complex Fokker-Planck equation, which evolves the complex-valued distribution ρ in Langevin time. As discussed in section 2.1, the absence of formal proof for the complex Fokker-Planck operator prevents us from determining when this relation generally breaks. However, we can assert that if the Fokker-Planck operator admits a complete set of eigenfunctions, then eq. (2.30) is fulfilled if all eigenvalues, except the zero mode (e^{iS}), have negative real parts, for the same reasons as for real Langevin (see chapter 5).

The other issue is the connection between the complex Fokker-Planck equation evolving the complex distribution $\rho(\phi, \tau_{\mathrm{L}})$ and the Fokker-Planck equation evolving the real distribution $P[\phi_R, \phi_I, \tau_{\mathrm{L}}]$. This can be summarized by needing to prove that

$$\langle O \rangle_\rho = \langle O \rangle_P \tag{2.31}$$

holds for some observable O, as first shown in [39]. This relation can be proven for holomorphic observables as long as the action and drift terms are also holomorphic.

Let us introduce an interpolation function

$$F(t,\tau) = \int \mathcal{D}\phi_R \mathcal{D}\phi_I P[\phi_R, \phi_I, t-\tau] O[\phi_R + i\phi_I, \tau] \tag{2.32}$$

where O is a holomorphic observable, and t and τ are two different Langevin times with $0 \leq \tau \leq t$. This function interpolates between the two observables we want to compare in 2.31. Examining $F(t,0)$, we find

$$F(t,0) = \int \mathcal{D}\phi_R \mathcal{D}\phi_I P[\phi_R, \phi_I, t] O[\phi_R + i\phi_I, 0] = \langle O \rangle_P \tag{2.33}$$

by comparing to eq. (2.8). To derive the left side of eq. (2.31), we consider the following real initial condition for the distribution P:

$$P[\phi_R, \phi_I, 0] = \rho[\phi_R, 0]\delta(\phi_I). \tag{2.34}$$

If the complex Langevin is ergodic, the dependence on the initial distribution $P[\phi_R, \phi_I, 0]$ vanishes in the $t \longrightarrow \infty$ limit [48]. Evaluating the interpolation function at $t = \tau$, we obtain

$$\begin{aligned} F(t,t) &= \int \mathcal{D}\phi_R \mathcal{D}\phi_I P[\phi_R, \phi_I, 0] O[\phi_R + i\phi_I, t] \\ &= \int \mathcal{D}\phi_R \mathcal{D}\phi_I \rho[\phi_R, 0] \delta(\phi_I - \phi_{I,0}) O[\phi_R + i\phi_I, t] \\ &= \int \mathcal{D}\phi_R \rho[\phi_R, 0] O[\phi_R, t] \\ &= \int \mathcal{D}\phi_R \left(e^{tL}\rho[\phi_R, 0]\right) O[\phi_R, 0] = \langle O \rangle_\rho \end{aligned} \tag{2.35}$$

where, in the last line, we used integration by parts and the evolution equation of an observable O (see Ref. [57] for details). As long as the function F is constant in τ, eq. (2.31) is satisfied. Hence, we need to show that

$$\begin{aligned} \frac{\partial}{\partial\tau} F(t,\tau) = \int \mathcal{D}\phi_R \mathcal{D}\phi_I \, \{ & P(\phi_R, \phi_I; t-\tau) L O[\phi_R + i\phi_I, \tau] \\ & - (L^T P(\phi_R, \phi_I; t-\tau)) O[\phi_R + i\phi_I, \tau] \} = 0, \end{aligned} \tag{2.36}$$

which is true in case we can perform integration by parts on one of the terms. This leads to the criterion that for eq. (2.31) to be true, the decay of $P[\phi_R, \phi_I, t-\tau]O[\phi_R+i\phi_I, \tau]$ must be fast enough such that it vanishes at the boundaries [57, 55].

To obtain a testable criterion, we will investigate $\frac{\partial}{\partial\tau}F(t,\tau)$ at $\tau = 0$, which is a weaker condition than $\frac{d}{d\tau}F(t,\tau) = 0$, but as demonstrated in [55], it is sufficient up to some technical conditions. The second term in eq. (2.36) is zero, while the first comprises the so-called *consistency criterion* [57, 55], if integration by parts is allowed, i.e.,

$$\langle LO \rangle = 0 \tag{2.37}$$

where L is the Langevin operator. We denote the simulation to accumulate *boundary terms* if this quantity is non-zero. The integration by parts is spoiled if the simulation accumulates boundary terms. As shown in ref. [55], an equivalent criterion without the weaker assumption of testing only at $\tau = 0$ can be made by testing if the drift term magnitude distribution decays faster than any power law.

In practice, the $\langle LO \rangle$ expectation value is too noisy to yield an accurate result for many models. Instead, we use a cutoff Ω in the real and imaginary directions to limit noise at points with a small amount of statistics. For models on a compact manifold, such as gauge theories on $SU(N)$, only a cutoff in the imaginary direction is necessary since the real remains compact. The boundary terms are now given by

$$B_1(\Omega) = \frac{\partial}{\partial\tau}F(\Omega; t, \tau)\bigg|_{\tau=0} = \langle LO \rangle \Omega \tag{2.38}$$

where we compute multiple values of Ω, and then read off the plateau for large Ω. We will discuss examples of such a computation in section 5.3.2, where the boundary terms calculations of simple systems are carried out.

We can now define the *corectness criterion* [57] as follows; There should be no boundary terms, i.e., eq. (2.38) evaluates to 0 for large cutoff size, or the distribution of the drift term magnitude decays faster than any power law [55]. Secondly, the stationary distribution of the Fokker-Planck must be e^{iS}, which amounts to the complex Fokker-Planck operator having only one unique zero mode, and all the other eigenvalues being negative such that $\lim_{\tau_{\mathrm{L}}\to\infty} P[\phi_R, \phi_I, \tau_{\mathrm{L}}] \propto e^{iS}$.

2.3 Progress on the convergence problem

Several methods have been proposed to tackle the problem of convergence to the wrong solution, but none have solved the problem for all models. Most of these methods have one thing in common; they rely on constraining the complex Langevin evolution into a region close to the original manifold, i.e., the non-complexified manifold. For Gauge theories like $SU(N)$, this will limit the evolution of the complexified fields in the $SL(N,\mathbb{C})$ manifold to be close to $SU(N)$. This is related to the correctness criterion, as this will limit the occurrence of boundary terms at infinity, i.e., localize the distribution of the drift term magnitude.

The first method discussed here attempts to undo the accumulated boundary terms aposteriori [58]. This can be done by first making an ansatz of the form of the interpolation function $F(t,\tau)$ (eq. (2.32)), which relates it to the first and second order boundary term (B_1 and B_2). Then by evaluating the boundary terms, we can subtract the contribution from the observable using

$$\langle O\rangle = \langle O\rangle_{\mathrm{CL}} - \frac{B_2^2}{B_1}. \tag{2.39}$$

The nth order boundary term is given by the nth order τ derivative of $F(t,\tau)$(see eq. (2.38) for B_1). Since this strategy relies on an accurate calculation of the higher order boundary terms, which are inherently costly due to the increasing statistics needed, the numerical cost will be high [58]. It also relies on the ansatz of $F(t,\tau)$ being an appropriate ansatz.

In gauge theories, the Gauge colling method is necessary, but not always sufficint, to obtain correct convergence, even for problems whith a small imaginary contribution in the action. The method was first introduced in [59], then updated with an adaptive version in [60], and the mathematical justification of the method was strengthened in [61]. Gauge cooling has been used successfully in several models [62, 63] and has been necessary for all gauge theory CLE simulations. The idea of the method is to carry out multiple gauge transformations of the field between the Langevin time steps, evolving it toward the compact original manifold. The method minimizes the unitarity norm, defined in $SU(N)$ by a trace of the gauge link product; $\frac{1}{N}\mathrm{Tr}(UU^\dagger - I)$. This reduces the unitarity norm to be small and hence restricts the field d.o.f close to the compact original manifold[3].

For some models, Gauge cooling alone cannot keep the unitary norm small. The method of *Dynamical stabilization* [64] has been developed to make a stronger pull of the evolution towards the original compact manifold. The method adds an additional drift term which points the evolution toward a smaller unitary norm. This will also keep the simulation close to the original (non-complexified) compact manifold. The change in the drift is written as

$$\frac{\partial S[\phi]}{\partial \phi} \rightarrow \frac{\partial S[\phi]}{\partial \phi} + i\alpha_{\mathrm{DS}} M \tag{2.40}$$

where α_{DS} is a tunable parameter of the method and iM points in the direction of the original manifold. A recent application of complex Langevin with dynamical stabilization was carried out for QCD thermodynamics with finite chemical potential [65]. Since the change to the drift is non-holomorphic, we cannot apply the correctness criterion directly; hence a mathematical justification of the method remains open. It can, however, be shown that the additional term vanishes in the continuum limit.

A similar approach relies on introducing a harmonic oscillator trapping potential to the action as a regulator to suppress terms far from the original configuration space [66]. Since the method changes the action and the theory we try to simulate, the additional potential term introduces a bias in the simulation. To regain the original theory, we must extrapolate to the

[3] At the original (non-complexified) manifold, the unitarity norm is zero

original theory by letting the correction term go to zero. A reweighting method using a similar regulator was proposed in [67].

A typical example of regularization of the CLE is the U(1) one-link model with the action $S(x) = i\beta\cos(x)$, which was used in [68]. The regulated action is $\tilde{S}(x) = S(x) + \frac{1}{2}sx^2$, such that for $s > 0$, the Langevin evolution is trapped close to the origin due to the added potential. For larger values of s, it can be shown that the complex Langevin equation of the modified theory ($\tilde{S}$) correctly converges. As discussed above, this does not correspond to the same observable of the U(1) one-link model action S, and hence we need to extrapolate [68] or reweight [67] back to $s = 0$.

The problem with such methods is that it is not a complete solution. It is often impossible to extrapolate back to $s = 0$ as the regulator needs to be very strong to obtain correct convergence. This is especially true in strongly correlated systems.

A kernel-controlled Langevin can be introduced to modify the complex Langevin equation without altering the action or breaking holomorphicity. This exploits freedom in the Fokker-Planck equation, allowing us to change the convergence behavior of the stochastic process and potentially avoid the convergence to the wrong solution. A potential method of obtaining a suitable kernel will be the main topic of chapter 6. We will introduce the kernel-controlled Langevin equation in the next section.

2.4 Kernelled complex Langevin

The last two methods, discussed in the previous section, were based on a change in the drift term or the action. There is, however, a way to change the complex Langevin without changing the action or spoiling the holomorphicity. It can be achieved by exploiting freedom in the Fokker-Planck equation. For real actions, we showed that the stationary distribution of the Fokker-Planck equation was the desired path integral measure of the Euclidean path integral. This Fokker-Planck equation is, however, not unique in having the desired stationary distribution e^{-S}. We can summarize the possible modifications to the original Fokker-Planck equation by inserting a field-dependent kernel $K(x, x', \phi, \tau_{\mathrm{L}})$, which also explicitly depends on the Langevin-time τ_{L}. In this section, I will introduce the necessary concept of a kernel-controlled complex Langevin equation.

The change in the Fokker-Planck operator for a real action (eq. (2.14)) following from a field-dependent kernel, independent of Langevin time[4], inserted as follows

$$\frac{dP[\phi,\tau_{\mathrm{L}}]}{d\tau_{\mathrm{L}}} = \int dx dx' \frac{\delta}{\delta\phi(x)} K(x,x',\phi) \left[\frac{\delta}{\delta\phi(x')} + \frac{\delta S}{\delta\phi(x')}\right] P[\phi,\tau_{\mathrm{L}}]. \tag{2.41}$$

It can be easily verified that a similar derivation of the stationary distribution holds after the insertion of the general kernel $K(x, x', \phi)$ without modification to the arguments. As the kernel resides on the outside of the parenthesis $\left(\frac{\delta}{\delta\phi_i(x)} + \frac{\delta S_E[\phi]}{\delta\phi_i(x)}\right)$, the stationary distribution remains unchanged. In principle, the only criterion we have for the kernel is that it is real-valued and factorizable. This means that we must be able to write K in terms of a matrix H as $K = H^T H$, which guarantees K to be positive semi-definite. In that case, it does not change the negative semi-definiteness of the modified Fokker-operator in eq. (2.18). In practice, the kernel will change the Fokker-Planck operator's eigenvalues. Hence the Langevin equation's convergence properties, which can be exploited to improve the convergence properties of the theory, i.e., the time it takes for the system to thermalize to a stationary distribution and the autocorrelation time of the stochastic process, might be changed.

We will assume the action to be discretized to simplify the following discussion. The integral kernel reduces to a matrix kernel in the following

[4] For a discussion of a kernel controlled Langevin kernel with a Langevin time dependent kernel, see ref. [69]

form[5]

$$\frac{dP[\phi,\tau_{\mathrm{L}}]}{d\tau_{\mathrm{L}}} = \sum_{ij} \frac{\partial}{\partial\phi_i} K_{ij}(\phi) \left[\frac{\partial}{\partial\phi_j} + \frac{\partial S[\phi]}{\partial\phi_j} \right] P[\phi,\tau_{\mathrm{L}}]. \tag{2.42}$$

We can now see the kernel's behavior; it mixes the field at different positions on the specified lattice, which we indicated with the indices i and j. Remember that for the kernel to be factorizable, it also needs to be positive semi-definite, hence, the diagonal elements must be strictly positive.

The corresponding Langevin equation yields

$$d\phi_i = \sum_j \left[-K_{ij}(\phi) \frac{\partial S(\phi)}{\partial\phi_j} + \frac{\partial K_{ij}(\phi)}{\partial\phi_j} \right] dt + \sum_j H_{ij} dW_j \tag{2.43}$$

where H_{ij} is given by the factorization of the kernel $K_{ik} = H_{ji}H_{jk}$. We see that we obtain an additional term in the Langevin equation given by the derivative of the kernel. The term emerges due to the product rule when applying the leftmost derivative in the Fokker-Planck operator to the kernel and the objects inside the brackets in eq. (2.42).

The simplest example of a kernel is a constant diagonal one; $K_{ij}(\phi) = \gamma\delta_{ij}$, where γ is a positive real number. Setting $\gamma = 1$ yields the standard Langevin equation, while a $\gamma \neq 1$ leads to a redefinition of the time increment $dt \rightarrow \gamma dt$. When discretizing the Langevin equation, introducing such a kernel with $\gamma \neq 1$ will change the Langevin step-size of the numerical scheme since the drift term update is proportional to the step-size $\Delta\tau_{\mathrm{L}}$, and the noise coefficient is proportional to $\sqrt{\Delta\tau_{\mathrm{L}}}$.

For the complex Langevin equation, we complexify the fields in the same way as without the kernel, and hence the kernel-controlled complex Langevin equation is given by

$$d\phi_i = \sum_j \left[K_{ij}(\phi)\, i\frac{\partial S(\phi)}{\partial\phi_j} + \frac{\partial K_{ij}(\phi)}{\partial\phi_j} \right] dt + \sum_j H_{ij}(\phi) dW_j \tag{2.44}$$

[5] See chapter 5 for a similar introduction without discretizing the field.

where $\phi_i, K_{ij}(\phi), H_{ij}(\phi) \in \mathbb{C}$ and $dW_i \in \mathbb{R}$. For the real Langevin equation in eq. (2.43) the matrix kernel was chosen to be factorizable, meaning $K = H^T H$ to regain a negative semi-definite modified Fokker-Planck equation. For the case of complex Langevin, we do not have such restriction, and hence we can choose H to be given by the square root of the kernel, i.e., $K = H^2$, not the transposed/hermitian factorization. As we do not know apriori if the complex Fokker-Planck equation has a positive sime-definite spectrum, constraining the choice of the kernel to be positive definite might be too restrictive as a non-positive definite kernel might be required to change the eigenvalues of a non-positive semi-definite operator back into a positive semi-definite one.

As for the complex Langevin equation without a kernel, also in the case with a kernel, we do not have a general proof of convergence for a general action. However, as we will discuss in chapter 5 the correctness criterion can easily be extended to the complex Langevin in the presence of a kernel.

We introduced the kernel-controlled complex Langevin in this section as a possible solution to the problem of convergence to the true solution. The idea is to use the kernel to alter the Fokker-Planck eigenvalues and limit the accumulation of boundary terms. There are two separate points in the correctness criterion, namely that the stationary distribution of $\rho[\phi, \tau_{\mathrm{L}}] = e^{iS}$ and the connection between the complex Fokker-Planck and the real-valued Fokker-Planck, i.e., $\langle O \rangle_\rho = \langle O \rangle_P$, is true. In the early investigation into applying a kernel to the complex Langevin to restore converges by Okamoto et al. in [54, 70], the authors constructed the kernel such that the eigenvalues of the modified Fokker-Planck equation, i.e., after the similarity transform, changed from positive to negative. Another direction would be to limit boundary terms by effectively including a push in a similar direction to the dynamical stabilization and regulator term mentioned in the previous section. This idea will be discussed in chapter 6.

3 Differential programming

Many recent proposals for improved lattice field theory methods rely on some form of optimization, e.g., training a neural network. Examples of this are the integration contour optimization for the Thimbles method [24] to reduce the sign problem and the normalizing flows method to avoid the critical slowing down in Eucledian Monte-Carlo simulations [71]. Similar techniques have not yet been explored and tested out for the complex Langevin method. In this chapter, we will explore some important concepts which can make it possible to optimize a kernel in the complex Langevin equation.

For optimization to be possible, we do, in general, need some kind of derivative. The methods for computing derivatives on a computer can be classified into three different categories

- symbolic differentiation, i.e., computing the derivative symbolically and then evaluating it,
- numerical differentiation, which uses finite difference approximations,
- automatic differentiation (AD), which we will discuss in the next section.

For machine learning applications, the latter one is commonly used, as this is an efficient way to compute the gradient of large models within machine precision. There are many different implementations of AD, and most modern machine learning has this built in. The reason this is normally preferred to symbolic- and numerical differentiation is, in short, a combination of the ability to obtain gradients directly from programming code with high precision.

To obtain the gradient, often referred to in the general context as the sensitivity, of iterative methods, like the numerical solution of the complex Langevin equation, it is necessary with a combination of differentiation techniques mentioned above. These methods are often called sensitivity methods and are usually formulated as solving similar kinds of systems. This would mean solving an additional stochastic differential equation to obtain the gradient for the complex Langevin equation. The most efficient sensitivity framework for systems with a large number of parameters, i.e., on the order of 1000 and larger, is called the adjoint sensitivity method, which will be derived in this chapter.

Although this thesis is directed towards the complex Langevin method, this chapter introduces some topics of the *Differential programming* paradigm in a general context. This chapter will review some aspects of the differential programming paradigm. We will begin by introducing auto-differentiation, for a detailed review covering the concepts of auto-differentiation in detail, see, e.g., ref. [30]. Then develop the necessary ingredients for deriving the adjoint sensitivity method for non-linear systems and (stochastic) differential equations. In the last chapter, we demonstrate the use of the adjoint sensitivity method applied to the optimization of a kernel-controlled complex Langevin equation.

3.1 Automatic differentiation

Before delving into automatic differentiation (AD), it is instructive to examine why AD is the preferred method for large optimization systems. The case against numerical differentiation, i.e., finite difference methods, is relatively straightforward, as it suffers from round-off and truncation errors and scales poorly for a large number of parameters. In many cases, including the optimization schemes we will investigate later in this thesis, the number of parameters is significantly larger than 100. This implies that to evaluate the simple forward finite difference; we need to evaluate the objective function the same number of times as there are parameters.

On the other hand, symbolic differentiation can circumvent numerical differentiation problems. However, it is plagued by the so-called "expression swell" phenomenon, which essentially means that the size of the symbolic expression grows exponentially. Furthermore, it cannot take derivatives of specific programming-oriented operations like conditionals, which is an essential aspect of differential programming.

Automatic differentiation (AD) offers a highly effective approach for computing derivatives and is a combination of the two in many ways. It computes numerical derivatives down to machine precision by using what is called a computational graph of the program, where the nodes are the building blocks of the code, and the lines are the order of the operations. We will touch on this in more detail later on. Still, for now, one can think about these building blocks as mathematical operations with trivial derivatives, like multiplication and addition. AD has two primary modes: forward mode and reverse mode. Forward mode is highly efficient for problems with a small number of input variables and a large number of output variables. In contrast, the reverse mode AD is well-suited for problems with many input variables and a small number of output variables. The forward mode is limited by the number of parameters as it requires one pass through the function that we want the gradient of for every parameter. This is cheap for a small number of parameters, while the adjoint method which only requires one additional pass of the function for all the parameters is more computationally expensive and is memory intensive.

In the section below, I will cover aspects of AD relevant to this thesis, which is necessarily incomplete; for a detailed introduction, see ref. [30].

3.1.1 Forward and Backward auto-differentiation

Forward automatic differentiation, also known as forward mode AD, is an algorithmic approach for evaluating derivatives of a given function. The key concept behind forward AD is the efficient propagation of directional derivatives through a sequence of elementary operations. In forward AD, we compute the derivative with respect to a specific input variable by applying the chain rule in a forward manner, starting from the input variables and moving through the intermediate variables to the output.

Consider a multivariate function $\boldsymbol{f}(\boldsymbol{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^m$, and let $\boldsymbol{x} \in \mathbb{R}^n$ be the input vector. In forward AD, we compute the derivatives of each element of the function $\boldsymbol{f}$ with respect to one of the input variables x_j and then propagate these partial derivatives through the computational graph using the chain rule to obtain the desired gradient vector $\nabla \boldsymbol{f}(\boldsymbol{x})$.

For example, let us assume the function is a scalar function with two inputs x_1 and x_2 and three operations, given by

$$f(x_1, x_2) = \ln(x_1) + x_1 x_2. \tag{3.1}$$

The operations are addition, multiplication between x_1 and x_2, and the natural logarithm of x_1. The expression for computing the logarithm is divided into a Taylor series approximation or a similar approximation, which in turn relies on addition and multiplication operations. In principle, there is no clear definition of how to divide into elementary operations in the computational graph. Still, the important distinction is that we can compute the derivative of $\ln(x)$ much faster using the analytic derivative, $\frac{d}{dx} \ln(x) = 1/x$, than computing the derivative of the Taylor series expansion using AD. It is, therefore, advantageous to supply the algorithm with this derivative in the same way as done for multiplication and addition. We can call these operations the elementary operations of the graph. For this simple example, the computational graph starts with x_1 and x_2, then computes $v_1 = \ln(x_1)$, and in parallel, calculates $v_2 = x_1 x_2$. We combine them into $v_3 = v_1 + v_2$, which yields the output $f(x_1, x_2)$.

We can now find the directional derivative in the x_1 direction by propagating the vector $\boldsymbol{x}^T = [1, 0]$ through a modified graph, where we take the same directional derivative of the operations (nodes): $\dot{v}_1 = 1/x_1$, $\dot{v}_2 = x_2$, and $\dot{v}_3 = \dot{v}_1 + \dot{v}_2$. We denote the directional derivative in the x_1 direction with a dot. We can now read out the derivative $\frac{\partial f(x_1, x_2)}{\partial x_1}$ by the value $\frac{\partial v_3}{\partial x_1} = \dot{v}_3$. These are the operations that are done in forward-mode auto differentiation, although the actual implementation might differ to optimize the algorithm.

Now to compute the full gradient ∇f, we need to evaluate the same graph for the directional derivative along x_2, which is done the same way as above, just by initializing with $\boldsymbol{x}^T = [0, 1]$. Forward AD exhibits excellent performance for problems with a small number of input variables and a large number of output variables, as it requires only a small number of passes through the computational graph. However, its efficiency decreases as the number of input variables grows. Fast implementations of forward-mode AD have been developed in numerous libraries, such as *ForwardDiff.jl* [72], which is used later in the thesis.

The most common implementations utilize dual numbers, which is a truncated Taylor series of the variables $\bar{x} = x + \dot{x}\epsilon$, where ϵ is defined such that $\epsilon^2 = 0$ and $\epsilon \neq 0$[1]. By construction, the numbers follow the symbolic differentiation rules of addition and multiplication,

$$\bar{x}_1 + \bar{x}_2 = (x_1 + x_2) + (\dot{x}_1 + \dot{x}_2)\epsilon, \quad \bar{x}_1\bar{x}_2 = x_1x_2 + (\dot{x}_1x_2 + x_1\dot{x}_2)\epsilon. \tag{3.2}$$

For functions, the chain rule is defined by the Taylor expansion around the dual number

$$f(x + \dot{x}\epsilon) = f(x) + f'(x)\dot{x}\epsilon. \tag{3.3}$$

As an example, we can get the same derivate as in the previous example by propagating the dual numbers $\bar{x}_1 = x_1 + \epsilon$ and $\bar{x}_2 = x_2 + 0\epsilon = x_2$ through the computational graph, using $\ln(\bar{x}_1) = \ln(x_1) + \frac{1}{x_1}\epsilon$, where the output of the function is dual number $\bar{f}$ containing the value and its directional derivative. The implementation of such an algorithm can be highly optimized by defining dual numbers as a type similar to complex numbers. As long as all operations in the code allow dual numbers, the gradient or Jacobian of the expression can be obtained without any changes to the original code.

It is also important to note that, given a function $f : \mathbb{R}^n \longrightarrow \mathbb{R}^n$, where the number of input and output variables are the same, forward mode AD is often the fastest option, as one pass through the graph gives one column of the Jacobian, which makes the total number of forward-passes through the graph n. This is the same number of passes through the graph as we need for the reverse-mode AD, which we will discuss next.

[1] This is a special case of Grassman numbers in a single dimension.

Reverse-mode automatic differentiation [73], also known as reverse AD, is another algorithmic approach for evaluating derivatives of a given function. Unlike forward AD, reverse AD computes the gradients by applying the chain rule backward, starting from the output variables and moving through the intermediate variables to the input. This method is particularly efficient when dealing with functions that have a large number of input variables and a small number of output variables.

The reverse AD process involves two main steps: the forward pass and the backward pass. In the forward pass, we evaluate the function $f(\boldsymbol{x})$ and store the intermediate variables in the computational graph, for simplicity we assume the function to be scalar-valued. In the backward pass, we propagate so-called adjoint variables by applying the chain rule in the reverse direction. The adjoint variables are defined as follows: for each intermediate variable v_i in the computational graph, we define its adjoint variable $\bar{\boldsymbol{v}}_i$ as the partial derivative of the output y with respect to v_i:

$$\bar{v}_i = \frac{\partial y}{\partial v_i}. \tag{3.4}$$

The adjoint variables can be computed efficiently by starting from the output $\bar{y} = 1$ and applying the chain rule in the reverse direction:

$$\bar{v}_i = \sum_j \frac{\partial y}{\partial v_j} \frac{\partial v_j}{\partial v_i} = \sum_j \bar{v}_j \frac{\partial v_j}{\partial v_i}. \tag{3.5}$$

Once all the adjoint variables have been computed, the gradient of the function with respect to the input variables can be obtained as:

$$\frac{\partial y}{\partial x_i} = \bar{x}_i = \sum_j \bar{v}_j \frac{\partial v_j}{\partial x_i}. \tag{3.6}$$

Reverse AD is particularly well-suited for large-scale optimization problems with numerous input variables. It requires only a single pass through the computational graph in the reverse direction to compute the gradient, regardless of the number of input variables. As in the example of the function $f(x_1, x_2) = \ln(x_1) + x_1 x_2$, we can compute the gradient ∇f by first a forward-pass to calculate the values at the nodes in the computational graph, and then one backward pass of the reverse mode AD presented above. The advantages of reverse mode AD come with the price of increased memory requirement, which in many cases is proportional to the number of operations. This is because we need to store all intermediate variables of the computational graph value during the forward pass.

3.1.2 Vector-Jacobian or Jacobian-Vector products

In many scientific applications, we often need to compute vector-Jacobian products (VJP) of the form $\boldsymbol{\lambda}^T J$ or Jacobian-vector products (JVP) of the form $J\boldsymbol{\lambda}$, where $\boldsymbol{\lambda}$ is a vector and J is a Jacobian. Automatic differentiation techniques, like forward and reverse mode AD, are closely connected with the concepts of VJP and JVP. Understanding the differences between VJP and JVP can help us better grasp the computational efficiency and advantages of using AD.

To relate JVP, VJP, forward mode AD, and reverse mode AD can investigate the output of one forward/backward pass of the method. Taking as an example a function $\boldsymbol{g}(x_1, x_2) \in \mathbb{R}^2$ the forward mode AD can be seen as computing

$$\begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} \end{bmatrix} \boldsymbol{e}_i = \boldsymbol{J}_{\boldsymbol{g}(x_1,x_2)} \boldsymbol{e}_i = \begin{bmatrix} \frac{\partial g_1}{\partial x_i} \\ \frac{\partial g_2}{\partial x_i} \end{bmatrix} = \frac{\partial}{\partial x_i} \boldsymbol{g}, \tag{3.7}$$

where $\boldsymbol{e}_i$ the unit vertor in the direction of x_i. This is the same as a JVP, meaning that $\boldsymbol{J}_{\boldsymbol{g}(x_1,x_2)}$ is a Jacobian, and $\boldsymbol{e}_i$ is a vector. On the other hand, reverse mode AD starts at the end, such that

$$\boldsymbol{e}_i^T \boldsymbol{J}_{\boldsymbol{g}(x_1,x_2)} = \nabla g_i(x_1, x_2) = \begin{bmatrix} \frac{\partial g_i}{\partial x_1} \\ \frac{\partial g_i}{\partial x_2} \end{bmatrix} \tag{3.8}$$

is a VJP. Here, $\boldsymbol{e}_i^T$ is the vector that tells us which direction in f we want to get the derivative of. We can see that for scalar functions, i.e., if $\boldsymbol{g}(x_1, x_2) \in \mathbb{R}$, having the AD calculation in the VJP form is more efficient as we obtain the gradient directly from one backward pass.

Now that we have made a connection between the Forward/backward AD and the two different products, we can extend it to a general VJP and JVP. As was eluded to in the AD section, the starting vector of the AD evaluation, both in the forward and backward AD, was connected to the directional derivative we want to calculate, i.e., $\boldsymbol{e}_i$. We could ask the same question using a general vector $\boldsymbol{\lambda}$, such that the meaning of the product becomes a directional derivative in the λ direction of either the function or the input vector depending on the choice of AD. This product can be obtained without needing direct computation of the full Jacobian as we only need to initialize the adjoint variables to be propagated in reverse through the graph. This means that a product $\boldsymbol{\lambda}^T J$, or equivalently $J^T \boldsymbol{\lambda}$, can be computed by one forward and backward pass by the reverse mode AD. Similarly, for forward mode AD, we can compute any product of the form $J\boldsymbol{\lambda}$ in one forward pass using dual variables and initializing the dual variables by the vector $\boldsymbol{\lambda}$.

This fact is important as we will later identify VJP in the adjoint sensitivity methods, which we will be deriving later on. This identification helps speed up the schemes, as every time such a product appears we only require one backward AD pass to obtain the full gradient, in contrast to first computing the Jacobian and then doing the product.

3.1.3 Why Not Use AD for Every Derivative

Automatic differentiation (AD) is a powerful tool for calculating the derivative of any code, provided it is written in a differentiable programming language. Such languages typically construct a computational graph to record all the elementary operations of an algorithm, which can then be utilized in AD calculations. Examples of differentiable programming languages include *Julia* and *Swift*. In these languages, all native code, with some exceptions, is differentiable as the compiler constructs a graph when compiling. The Julia programming language offers multiple AD tools such as *Zygote.jl* [74], *Forward-/ReverseDiff.jl* [72], *Enzyme.jl* [75]. In dynamically typed languages like Python, this functionality can be achieved using external packages like *TensorFlow*, *autograd*, or *JAX*. To build up the computation graph for use in AD in Python, one must write all operations using the built-in functions of the respective library. With the current state of available programming languages and packages, obtaining gradients for custom models has become significantly easier.

However, AD faces challenges when computing the gradient of any iterative method, such as the Newton non-linear solver. AD must propagate through all the iterations to calculate the gradient of the parameters of a non-linear function. This process is computationally expensive, and memory usage accumulates rapidly for reverse-mode AD. Another issue with propagating AD through all iterations is that iterative schemes only approximate the solution to the non-linear system, whereas AD computes the gradient to machine precision. Consequently, an exact gradient of an approximate solution is obtained, which not only provides redundant information but can also lead to differentiation of the error in the approximation rather than the derivative of the approximate solution.

This is a similar argument as the one we used for handling the natural logarithm in our example function $f(x_1, x_2)$ (eq. (3.1)) where we directly used the rule of differentiating $\ln(x)$ instead of adding the inner calculation of the Taylor series expansion to the computational graph. This means that for auto-differentiation to be efficient and stable, we should provide a method for calculating the gradient of expressions like, e.g., iterative solvers, series expansions, and expressions with a straightforward gradient expression.

One approach to address this issue for iterative solvers is to use adjoint methods, which compute the derivative based solely on the final approximate solution. We will discuss this topic in the next section.

3.2 Adjoint sensitivity methods

In this section, the adjoint sensitivity method is derived. These methods have been actively developed since the 1980s for differential equations, with applications to fields such as computational fluid dynamics, in particular areodynamics [76], metherology [77] and geophysocs [78]. The method has been applied to different optimization problems, from shape optimization of airplane wings and cars to constructing more accurate models based on physical measurements. The method has gained significant importance in recent years due to the rise of machine learning. One exciting application of the method is neural ODEs [79], where a neural network is built as a differential equation of the form $\frac{\partial u}{\partial t} = \mathrm{NN}(u, t, \theta)$, this has later been generalized to SDEs in [80, 81]. This efficiently describes time-dependent data, as the network does not have to learn the time dependence. Moreover, these methods allow for integrating pre-existing physics knowledge, bypassing the need to learn this information separately.

In general, we can say that the adjoint sensitivity method is a way of calculating the gradient of a loss function L based on the parameters (θ) or the initial condition of the underlying system $F(u, \dot{u}, t, \theta) = 0$[2]. The structure of the method is similar for various forms of F and requires, in general, solving a similar system to obtain the gradient. An example of systems that we can incorporate into F is a non-linear algebraic system, which we would write $F_{\mathrm{NL}}(u, \theta) = 0$ or a differential equation $F_{\mathrm{DE}}(u, \dot{u}, t, \theta) = 0$. To compute the gradient of these systems, we need to solve a linear- or a differential equation respectively. These equations are called the adjoint equation, which we will derive in a general form below.

Linear systems and non-linear systems

The most elementary adjoint sensitivity equations that can be derived stem from linear and nonlinear equations, specifically, problems of the form $f(u(\theta), \theta) = 0$. In this expression, $u(\theta) \in \mathbb{R}^M$ represents the solution to the nonlinear equations given the parameterization θ of the nonlinear equation. This means that a change in θ will necessitate a change in the solution $u(\theta)$ of the non-linear system f.

The objective is to obtain the gradient $\frac{dL}{d\theta}$, which provides the sensitivity

[2] This is not the most general form of the underlying system, but it incorporates differential equations and non-linear systems (independent of $\dot{u}$).

of the cost function L concerning the parameters θ. For the nonlinear system, this will yield:

$$\frac{dL}{d\theta} = L_\theta + L_u u_\theta = L_\theta - L_u \left[f_u^{-1} f_\theta \right] = L_\theta - \left[L_u f_u^{-1} \right] f_\theta \tag{3.9}$$

Here, the subscripts indicate partial derivatives, e.g. $f_u = \frac{\partial f(u,\theta)}{\partial u}$ and $L_\theta = \frac{\partial L(u,\theta)}{\partial \theta}$. This notation will be used for the rest of this section to simplify derivations. We have used $\frac{d}{d\theta} f = f_u u_\theta + f_\theta = 0$, such that $u_\theta = -f_u^{-1} f_\theta$[3]. We have grouped the $L_u f^{-1}$ terms in the last equality to emphasize the various ways to compute the gradient. The second equality necessitates inverting an $M \times M$ matrix and subsequently multiplying by an $M \times P$ matrix, where M represents the system size, i.e., the number of discrete points in u, and P refers to the number of parameters in the model, i.e., the length of the parameter vector θ. The next step is to rewrite the last bracket ($\lambda^T = L_u f_u^{-1}$) as a linear equation:

$$f_u^T \lambda = L_u^T \tag{3.10}$$

and then use this to calculate the full derivative:

$$\left. \frac{dL}{d\theta} \right|_{f=0} = L\theta - \lambda^T f_\theta \tag{3.11}$$

where we evaluate the derivative at the solution to the linear system $f(u,\theta) = 0$. This reduces the computation of calculating derivatives of the loss function L with respect to its parameters θ to solely computing the Jacobian $f_u = \frac{\partial f(u,\theta)}{\partial u}$ and subsequently solving a linear system. The cost of obtaining the gradient should be equivalent to one iteration of the Newton method.

Lagrangian formulation

Since it will be instructive for the next section on ODEs, we will again derive the adjoint sensitivity equation for the non-linear systems using

[3] Keep in mind that u in this section always refer to the solution of the system $f = 0$.

the Lagrangian formulation. We now formulate the problem as a minimization problem. Given the solution u of $f(u,\theta) = 0$

$$\mathcal{L}(u,\theta) = L(u,\theta) + \lambda^T f(u,\theta) \tag{3.12}$$

where we added a zero, due to $f(u,\theta) = 0$ on the solution. We now take the derivative of this Lagrangian

$$\begin{aligned} \frac{d\mathcal{L}(u,\theta)}{d\theta} &= L_\theta + L_u u_\theta + \lambda^T (f_u u_\theta + f_\theta) && (3.13)\\ &= L_\theta + \lambda^T f_\theta + (L_u + \lambda^T f_u) u_\theta. && (3.14) \end{aligned}$$

In the second line, we have reshuffled the equation such that the terms containing u_θ are grouped. The reason for this grouping is, as we learned in the previous section, because those terms involve large matrix-matrix products and hence will be expensive to calculate. To avoid calculating these quantities, we will instead find λ^T such that $L_u + \lambda^T f_u = 0$, and hence only the two first terms contribute to the final result. Remember that $\mathcal{L} = L$, since $f = 0$. We can now set up a linear system

$$f_u^T \lambda = L_u^T \tag{3.15}$$

which, after obtaining λ, we can use to evaluate the gradient of the loss function

$$\left.\frac{dL}{d\theta}\right|_{f=0} = L_\theta - \lambda^T f_\theta. \tag{3.16}$$

Even though the adjoint method can be applied to non-linear problems with any number of parameters, the method's efficiency compared to the forward method, i.e., computing the gradient for each parameter separately, is related to the number of parameters. The following section will derive an equivalent adjoint method for ordinary differential equations (ODEs).

Ordinary differential equations

We now go on to discuss adjoint sensitivity methods for ordinary differential equations (ODE), preparing the ground for stochastic differential equations in the next section. For a general ODE, we use the form

$$\dot{u} = f(u, t, \theta) \tag{3.17}$$

where again $f(u, t, \theta)$ can be a non-linear equation with u evaluated at the time t with the parameters θ.[4]

For an ODE system, we now need a generic function $g(u, t, \theta)$ that can be evaluated at every point in time t, and from which we define the loss function, via integration as

$$L(\theta) = \int_{T_0}^{T} g(u(t), \theta) dt \tag{3.18}$$

Next, we calculate the derivative of the loss function with respect to the ODE parameters

$$\frac{dL}{d\theta} = \int_{T_0}^{T} g_\theta + g_u u_\theta dt. \tag{3.19}$$

We identify the computationally expensive object u_θ that needs to be evaluated at any step in time during the simulation. It is expensive because, at every point in time, the dependence on θ requires us to evaluate the function u and g at all points $t \in [T_0, T]$, for all the values of θ. To compute the equality we use sensitivity variable $s_i = \frac{\partial u}{\partial \theta_i}$, and then solve the ordinary equation given by taking the partial derivative of the original ODE from the left, such that we obtain a system of ODEs (one for each entry in θ) given by

$$\frac{\partial s_i}{\partial t} = f_u s_i + f_{\theta_i}. \tag{3.20}$$

This is often called a *tangent equation*, or the forward sensitivity equation. We see that this system quickly becomes large for many parameters as we, in practice, need one ODE for each parameter. This sensitivity method is efficient for a small number of parameters, similar to the forward auto differentiation. In fact, we can also identify a Jacobian vector product

[4] We can derive an adjoint sensitivity method for $f(\dot{u}, u, t, \theta) = 0$ by following a similar derivation. Since the SDE we will consider in the following chapters has a similar form as eq. (3.17), we will use this as the starting point of the derivation.

(JVP) in the first term on the right-hand side, which is the same product we could relate to the forward AD in section 3.1.1.

For a large number of parameters, the adjoint sensitivity method is in most cases more efficient. We use the Lagrange formulation to derive the corresponding adjoint equation. Given a solution u for one set of parameters θ, we define the Lagrangian as

$$\mathcal{L} = \int_{T_0}^{T} g(u(t), \theta) dt + \int_{T_0}^{T} \lambda^T \left(\dot{u} - f(u, t, \theta)\right) dt. \tag{3.21}$$

where the second part evaluates to zero for any solution of the ODE (eq. (3.17)). The gradient of this expression reads

$$\begin{aligned}
\frac{d\mathcal{L}}{d\theta} &= \int_{T_0}^{T} g_\theta + g_u u_\theta dt + \int_{T_0}^{T} \lambda^T \left(\dot{u}_\theta - f_\theta - f_u u_\theta\right) dt && (3.22)\\
&= \int_{T_0}^{T} g_\theta - \lambda^T f_\theta dt + \int_{T_0}^{T} \left(\lambda^T \dot{u}_\theta + g_u u_\theta - \lambda^T f_u u_\theta\right) dt && (3.23)\\
&= \int_{T_0}^{T} g_\theta - \lambda^T f_\theta dt + \int_{T_0}^{T} \left(-\dot{\lambda}^T + g_u - \lambda^T f_u\right) u_\theta dt && (3.24)\\
&+ \lambda^T(T_0) u_\theta(T_0) + \lambda^T(T) u_\theta(T) && (3.25)\\
& && (3.26)
\end{aligned}$$

where we in the second line have reshuffled some of the terms, and in line 3, integration by parts is carried out on the $\lambda^T \frac{\partial}{\partial t} u_\theta$ term. The trick now is to force the interior of the second integral's parentheses to zero by tuning λ. In this way, we do not have to calculate u_θ, and we are only left with gradients that can be evaluated without too much added computational cost. We, therefore, need to find the λs, such that $-\dot{\lambda}^T + g_u - \lambda^T f_u = 0$, which is defining the relation of the so-called adjoint ODE equation with the initial value $\lambda^T(T) = 0$. The initial condition arises from the boundary terms obtained by integration by part, as we also want to set $\lambda^T(T) u_\theta(T) = 0$. For this derivation, we assume that the initial condition of the ODE is independent of the parameters θ, i.e., $\frac{\partial u}{\partial \theta}(T_0) = 0$. The adjoint equation thus takes the form

$$\dot{\lambda} = -f_u^T \lambda + g_u^T \tag{3.27}$$

$$\lambda(T) = 0 \tag{3.28}$$

where we have transposed both sides of the equation. To get the full derivative of the loss with respect to the ODE parameters, we need to solve eq. (3.27) and then we insert λ in

$$\frac{dL}{d\theta} = \int_{T_0}^{T} g_\theta - \lambda^T f_\theta dt. \tag{3.29}$$

The advantage of the adjoint sensitivity method is that we only need to evaluate one ODE, no matter how many parameters there are in the ODE. The added cost for evaluating the gradient of the loss function amounts to a backward evaluation of the adjoint ODE eq. (3.27), which in contrast to the forward-sensitivity method, would need to solve the forward-sensitivity ODE once for every parameter. The drawback of the adjoint method is that it requires storing the solution of the forward pass in memory, as it is needed for an accurate solution to the adjoint equation.

Stochastic Differential equations

To generalize this further to be used in a stochastic differential equation (SDE), we need to perform similar derivation for the generic SDE

$$du = a(u, t, \theta)dt + b(u, t, \theta)dW. \tag{3.30}$$

The derivation of the adjoint method will be the same as for the ODE case, where we constructed a Lagrangian by adding zero to the loss function (eq. (3.21)). The constraint is now that the SDE equation is satisfied, i.e., $\dot{u} - s(u, t, \theta) + b(u, t, \theta)\xi = \dot{u} - f(u, t, \theta) = 0$, such that the Lagrangian is given by

$$\mathcal{L} = \int_{T_0}^{T} g(u(t), \theta)dt + \int_{T_0}^{T} \lambda^T \left(\dot{u} - f(u, t, \theta)\right) dt. \tag{3.31}$$

The adjoint method for the SDE now yields

$$d\lambda = (g_u^T - a_u^T \lambda)dt - (b_u^T \lambda)^T \; dW \tag{3.32}$$

where we have rewritten the last term $(b_u dW)^T \lambda = dW \cdot (b_u^T \lambda) = (b_u^T \lambda) \cdot dW = (b_u^T \lambda)^T dW$. At this point, we need to treat the noise term carefully, as this needs to be the same Wiener process as for the forward pass of the SDE. Let's take a step back and look into different ways of implementing this.

The simplest way, which does not require any modifications, is to use a fixed step-size when solving the SDE forward. We save the Wiener process step (ΔW) at every time-step in the forward pass and then use the same step-size in the backward pass in order to deploy the saved Wiener steps ΔW when we solve the adjoint system (eq. (3.32)). This, however, have the drawback of not being applicable for use with adaptive step-size solvers. For the use of adaptive step-size, we have two options. The first one is to treat the Wiener process after the forward pass as deterministic and then interpolate between the points to obtain a deterministic function. This function can now be used to solve the adjoint system using an adaptive step-size, as the interpolation will be defined at every point t. Notice that if we use the interpolation function of ΔW, the adjoint equation reduces to a deterministic equation, i.e., an ordinary differential equation. To our knowledge, this has not been tested before. We will, therefore, demonstrate this as an example later in this chapter for a simple toy model complex Langevin equation[5].

The second way to do this uses the seeded Brownian tree method. One can think of the Wiener process as having fractal-like properties, i.e., if one zooms in on the curve, it will never be a continuous differential function. This method uses this pattern to be able to draw random numbers between the points we have saved in a way that satisfies the same Wiener process, just with more points. This was first implemented in the adjoint sensitivity method of SDEs in [80, 81]. This is the mathematically formal way to

[5] We will also use the method of interpolating the noise in a sensitivity method for chaotic dynamical systems in section 7.4.2, more precisely a stochastic form of the Lorenz system.

calculate the adjoint SDE equation. However, we see that interpolating the forward pass noise helps stabilize the adjoint solve for the system we are interested in. See section 7.4.2, where we tested both methods and were only able to stabilize the system with the interpolation method.

We will continue the discussion by using the method of interpolating the noise. Therefore, the noise ΔW is assumed to be a deterministic function in the backward pass, such that the stochastic integral changes to a deterministic integral. The adjoint system can now be written as, where we use $\eta(t)dt = \Delta W(t)$ as the interpolating function,

$$\dot{\lambda} = g_u^T - (a_u + b_u\eta(t))^T\lambda. \tag{3.33}$$

We can identify the λ term as $a_u + b_u\eta(t) = f_u$ from the ODE adjoint system eq. (3.27). The adjoint system is now solved from T to 0 with the initial value of $\lambda(T) = 0$. We can then put the solution λ into the equation for the sensitivity of the loss function L on varying the parameters θ

$$\frac{dL}{d\theta} = \int_0^T \left(g_\theta - \lambda^T[a_\theta + b_\theta\eta(t)]\right) dt. \tag{3.34}$$

This approach can be combined with the adjoint solution by using the fundamental theorem of calculus to turn the integral into a differential equation. This will save memory as there is no need to store the intermediate values of λ.

It is important to note that we solve eq. (3.34) for every parameter θ at once, just like the reverse mode AD. We can ideentify the the vector-jacobian product (VJP), as the adjoint vector λ times the Jacobian $a_u + b_u\eta(t)$ in eq. (3.33) and $a_\theta + b_\theta\eta(t)$ in eq. (3.34). This is why the adjoint method is often compared to reverse auto-differentiation and the backpropagation algorithm in neural networks, a special case of reverse mode AD.

3.3 Adjoint sensitivity method for complex Langevin

This section will apply the adjoint sensitivity method to the field-independent kernel-controlled complex Langevin equation eq. (6.9). As we derived the adjoint sensitivity method for an SDE in section 3.2, we can directly apply it to the complex Langevin equation. Let us use the form of the complex Langevin with a field-independent kernel as given in chapter 2;

$$d\phi = Ki\frac{\partial S(\phi)}{\partial\phi} + HdW \tag{3.35}$$

where $K = H^T H$. Consider a generic loss function that can be used to optimize the expectation value of an observable $O(\phi)$ to a given value y;

$$L = \left(y - \frac{1}{T}\int_0^T O(\phi)dt\right)^2 = (y - \langle O(\phi)\rangle)^2 . \tag{3.36}$$

Note that this differs from eq. (3.18) as the integral is inside the square. Nevertheless, if we compute the derivative of this loss function we get

$$L_\phi = 2\,(y - \langle O(\phi)\rangle)\,\frac{1}{T}\int_0^T \frac{\partial O(\phi)}{\partial\phi}dt. \tag{3.37}$$

where everything in front of the integral can be summarized into a constant k, such that $L_\phi = \int_0^T kO_\phi dt$. This is equivalent to the gradient of the ODE loss function (eq. (3.19)). Since only the gradient of the loss appears in the adjoint equation, we can apply $g(\phi) = kO(\phi)$ to the method, treating k as a constant.

The adjoint equation for the CLE is given by

$$\frac{d\lambda}{dt} = \frac{\partial g(\phi)}{\partial\phi} - \left[K\frac{\partial^2 S(\phi)}{\partial\phi\partial\phi}\right]^T \lambda \tag{3.38}$$

$$\lambda(T) = 0. \tag{3.39}$$

Note that this is an ordinary differential equation at this point due to the choice of a field-independent kernel. For a field-dependent kernel, an additional noise term will appear. Finally, we can compute the gradient of the loss function eq. (3.36)

$$\frac{dL}{d\theta} = \int_0^T \left[\frac{\partial g}{\partial\theta} - \lambda^T\frac{\partial K}{\partial\theta}\frac{\partial S(\phi)}{\partial\phi}\right]dt - \int \lambda^T\frac{\partial H}{\partial\theta}dW. \tag{3.40}$$

where the second integral is stochastic in the Ito-sense. By assuming dW to be deterministic, i.e., an interpolation of the Wiener process sampled in the initial solution of the complex Langevin, the integral reduces to

$$\frac{dL}{d\theta} = \int_0^T \left[\frac{\partial g}{\partial \theta} - \lambda^T \frac{\partial K}{\partial \theta} \frac{\partial S(\phi)}{\partial \phi} - \lambda^T \frac{\partial H}{\partial \theta} \eta(t) \right] dt. \tag{3.41}$$

where $\eta(t)$ is the interpolation function of the noise. We will use a linear interpolation of the noise. This is the adjoint sensitivity method for stochastic differential equations. In practice, we will combine eq. (3.38) and eq. (3.41) into a system of ODEs.

3.4 Application to toy model

As an example of the adjoint sensitivity method laid out above, we investigate a simple toy model with the action

$$S = \frac{1}{2}\sigma x^2 + \frac{1}{4}\lambda x^4 \tag{3.42}$$

where $\sigma \in \mathbb{C}$ and $\lambda \in \mathbb{R}$, note that this is not the λ used in the adjoint equation, but a parameter of the model. This model have been studied exstensively in [54, 70, 82]. We will also discuss this model in detail in chapter 5. For this section, it is sufficient to know that for the parameters $\sigma = 4i$ and $\lambda = 2$; the complex Langevin simulation converges to the wrong solution. Studies using a field-independent kernel $K = e^{-i\frac{\pi}{3}}$ were shown to solve the problem of convergence for this set of parameters in [54]. In this section, let us try to find such a kernel by optimizing a loss function based on the true solution.

When complexifying the action we change x to a complex variable, such that $x \longrightarrow x_R + ix_I$. The corresponding drift term for such a complexified action is

$$\mathrm{Re}\frac{dS}{dx} = \sigma_R x_R - \sigma_I x_I + \lambda(x_R^3 - 3x_R x_I^2) \tag{3.43}$$

$$\mathrm{Im}\frac{dS}{dx} = \sigma_R x_I + \sigma_I x_R + \lambda(3x_I x_R^2 - x_I^3) \tag{3.44}$$

where we have used $\sigma = \sigma_R + i\sigma_I$. When including a field-dependent (constant) kernel, we get the complex Langevin equation as a system of SDEs containing the real and imaginary part of x

$$\begin{bmatrix} dx^R \\ dx^I \end{bmatrix} = \begin{bmatrix} K_R & -K_I \\ K_I & K_R \end{bmatrix} \begin{bmatrix} \mathrm{Re}\frac{dS}{dx} \\ \mathrm{Im}\frac{dS}{dx} \end{bmatrix} dt + \begin{bmatrix} H_R & -H_I \\ H_I & H_R \end{bmatrix} \begin{bmatrix} dW \\ 0 \end{bmatrix} \tag{3.45}$$

where the two complex kernels, K and H ($K = H^2$), are written as matrices corresponding to the complex number. We follow the same convention with the real and imaginary parts of the kernel noting by $_R$ and $_I$. We use real noise, so the noise vector only contains a non-zero contribution in the first element. In practice, we can compute the matrix-vector product before we apply it to a numerical simulation.

We regain the standard complex Langevin equation for $K = 1$ due to the two kernel matrices reducing to identity matrices. As we know that the standard complex Langevin simulation converges to the wrong solution for the observable $\langle x^2 \rangle$, we need to find a kernel that changes the convergence to regain the correct solution. We, therefore, need to find a suitable value for K_R and K_I. Since we are only after changing the phase, and not the magnitude (overall change in dt) we choose the kernel parameterization to be

$$K = e^{i\theta}, \quad H = e^{i\frac{\theta}{2}}. \tag{3.46}$$

The parameter θ is constrained to $\frac{\pi}{2} \geq \theta \geq -\frac{\pi}{2}$ such that we can threat K as positive definite since the eigenvalues of the 1×1 matrix are the same as the entry. The two values K_R and K_I is then given by $K_R = \cos(\theta)$ and $K_I = \sin(\theta)$, and the noise coefficients $H_R = \cos(\theta/2)$ and $H_I = \sin(\theta/2)$.

We will optimize the kernel parameter θ based on the true solution of an observable O; y_O. The loss function in eq. (3.36) for the $O = x^2$ observable, is given by

$$L = \left(\mathrm{Re} y_{x^2} - \mathrm{Re}\langle x^2 \rangle\right)^2 + \left(\mathrm{Im} y_{x^2} - \mathrm{Im}\langle x^2 \rangle\right)^2, \tag{3.47}$$

where we have split the real and imaginary parts up. The corresponding gradient of the loss function with respect to x follows again the form in eq. (3.37), such that for the real part of x, we get

$$\begin{aligned}\frac{\partial L}{\partial x_R} = &\; 2\left(\mathrm{Re} y_{x^2} - \mathrm{Re}\langle x^2\rangle\right)\frac{1}{T}\int_0^T 2x_R(t)dt \\ &+ 2\left(\mathrm{Im} y_{x^2} - \mathrm{Im}\langle x^2\rangle\right)\frac{1}{T}\int_0^T 2x_I(t)dt \\ =&\frac{1}{T}\int_0^T 2c_R x_R(t) + 2c_I x_I(t)dt = \frac{1}{T}\int_0^T g_{x_R} dt.\end{aligned} \tag{3.48}$$

and the imaginary part

$$\frac{\partial L}{\partial x_I} = \frac{1}{T}\int_0^T -2c_R x_I(t) + 2c_I x_R(t)dt = \frac{1}{T}\int g_{x_I} dt. \tag{3.49}$$

The two constants c_R and c_I are a redefinition of the factor in front of the corresponding integrals in eq. (3.48). Similar rewriting can be done for optimizing multiple observables. We can now read off the expressions inside the integral and use this as $\frac{\partial g}{\partial x}$ in eqs. (3.38) and (3.41). Note that $\frac{\partial g}{\partial \theta} = 0$.

The adjoint equation is a system of ODEs when including the sensitivity of the loss function due to changes in the parameter θ ($\frac{dL}{d\theta} = s_\theta \in \mathbb{R}$), is given by

$$\begin{aligned}\frac{\partial}{\partial t}\begin{bmatrix}\lambda^R(t)\\ \lambda^I(t)\end{bmatrix} &= \begin{bmatrix}g_{x_R}\\ g_{x_I}\end{bmatrix} + \left[K\frac{\partial^2 S(x)}{\partial x \partial x}\right]^T \begin{bmatrix}\lambda_R\\ \lambda_I\end{bmatrix} \\ \frac{\partial}{\partial t}s_\theta(t) &= \begin{bmatrix}\lambda_R & \lambda_I\end{bmatrix}\left\{\left[\frac{\partial K}{\partial \theta}\right]\begin{bmatrix}\mathrm{Re}\frac{dS}{dx}\\ \mathrm{Im}\frac{dS}{dx}\end{bmatrix} + \left[\frac{\partial H}{\partial \theta}\right]\begin{bmatrix}\eta(t)\\ 0\end{bmatrix}\right\}\end{aligned} \tag{3.50}$$

with the inital value $\lambda(T) = \mathbf{0}$ and $s_\theta(T) = 0$. Note that we have used brackets around the objects that are matrices, which are all the factors containing the kernel, i.e., K or H. In the practical implementation of the equation, both a VJP and JVP can be used in combination with an auto-differentiation of choice. We can read the final sensitivity by the value $s_\theta(0) = \frac{dL}{d\theta}$, which is the last point of the solution.

We need to use this sensitivity to optimize the kernel to update the θ parameter. This can be done using an optimization scheme. Since this is a one-dimensional optimization problem, we will apply the standard gradient descent method. The parameter is updated using

$$\theta^{i+1} = \theta^{i} - \gamma \frac{dL}{d\theta} \tag{3.51}$$

where we start the optimization at $\theta^0 = 0$ such that $K^0 = 1$. The parameter γ can be thought of as the step-size of the optimization. The algorithm is then given by;

1. setup up the problem with $\theta^0 = 0$,
2. simulate the complex Langevin equation of the problem (eq. (3.45)),
3. calculate the sensitivity s_θ using adjoint equation eq. (3.50),
4. update θ using eq. (3.51),
5. repeat step 2-4, until the value of the loss eq. (3.47) reach desired tolerance.

For an implementation of the algorithm above, we will use the *DifferentialEquation.jl* [83] library written in *Julia* as this have both SDE and ODE solvers available. We will discuss in detail the implementation of numerical SDE schemes in chapter 4. We will use the standard explicit Euler-Maruyama scheme with an adaptive step-size for this setup. We simulate 10 trajectories up to $T = 500$ in Langevin time and save every 0.01 to memory. Since we will apply the simulation to the adjoint method, we also save the noise ΔW to memory at the same interval. We thermalize the system from $t = -5$ to $t = 0$ in Langevin time before we start sampling.

For the adjoint equation, we use the explicit ODE solver *Tsit5*[84] with an absolute tolerance of 10^{-7}. The simulation is done from $T = 500$ to $t = 0$ by using a negative signed step-size, i.e., $-\Delta t$. Note that we do not include the thermalization phase when calculating the loss sensitivity. In the gradient descent update of the parameter, we use $\gamma = 0.01$ and run the optimization for 100 steps.

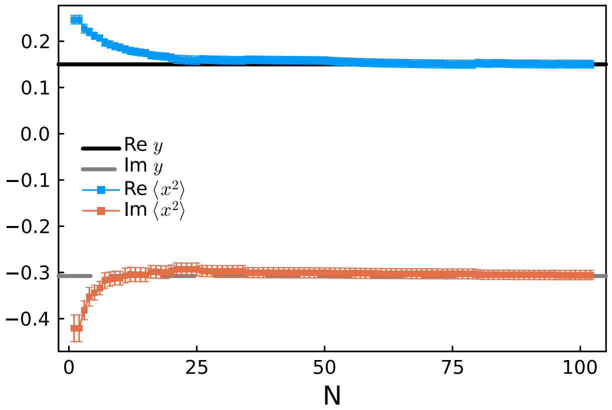

Figure 3.1: Evolution of a kernel optimization using the adjoint sensitivity method on the complex Langevin equation. The kernel is optimized based on the true solution of the observable $\langle x^2 \rangle$ plotted in black and gray lines. The x-axis represents the iteration number of the optimization scheme, and the scatter points are the real- and imaginary parts of the observable obtained from the CLE simulation.

The optimization is shown in fig. 3.1, where we plotted the change in the observable $\langle x^2 \rangle_K$ during the kernel optimization. Every point represents one update of the kernel; as can be seen, the observable quickly get close to the true solution. We find the minimum of the optimization to be $\theta = 0.9$, which is similar to the one found in [54], i.e. $\theta = \pi/3 \approx 1.0$.

This is a simple example where we obtained a suitable kernel for the complex Langevin by optimizing against the known solution. In the context of real-time simulation, this information is available whenever we can simulate the Eucledian correlator. Another loss function we could have used to optimize the kernel is by minimizing the boundary terms $\langle Lx^2 \rangle$ (eq. (5.11)). We could also have obtained this kernel in different ways, like in [54] where the choice of the kernel is based on the imaginary part of the drift term, and as we will see in chapter 5, where we will use the same model and optimize the kernel based on a low-cost loss function that also yields the same kernel. The method does, however, show the efficiency in using adjoint sensitivity methods in combination with the complex Langevin equation, as it can be used to obtain a suitable kernel by an optimization scheme. We will introduce such an optimization scheme for real-time problems in chapter 6.

3.5 Summary and outlook

This chapter gave an introduction to important concepts in differential programming, with emphasis on the adjoint sensitivity methods. The AD framework is a useful tool for many kinds of optimization problems, not only for neural network applications, as it can speed up the implementation process without a large reduction in computational speed. By identifying vector-Jacobian (Jacobian-vector) products, we have seen that AD can be reduced to only a single pass through the function. This product appears in the adjoint sensitivity method, which can speed up the code significantly for a large number of parameters.

The adjoint sensitivity method for a non-linear equation was derived as a simple example before the ODE version. Although it was used as a precursor to the ODE method, it has many uses. An example of this is in Bayesian spectral reconstruction, where we can use the method to efficiently optimize the regulators. An example of such a use case is presented in chapter D. This makes it possible to optimize for a large number of parameters, which without such an adjoint sensitivity scheme, is unattainable.

The adjoint sensitivity method laid out above gives many possibilities for optimizing a kernel for the complex Langevin equation. We can use it to, e.g., optimize kernels to reduce auto-correlation time, optimize the kernel to sample close to the original (non-complexified) manifold in Gauge theories, and in general, be used to avoid both the convergence problem and the stability problem of the CLE. This interesting direction can replace current methods of keeping the sampling of the CLE close to the original manifold by a holomorphic modification to the CLE.

However, we find that in real-time simulations, the gradient is not well-defined after a certain real-time extent due to the complex Langevin becoming a chaotic system. This is the topic of chapter 7. For such kind of problems, a different adjoint sensitivity method is needed. The reason for deriving this method here is that the alternative methods developed for chaotic systems build on similar equations.

4 Stable Solvers for Real-Time Complex Langevin

This chapter is published in *Journal of High Energy Physics 08 (2021) 138* [38].

This study explores the potential of modern implicit solvers for stochastic partial differential equations in the simulation of real-time complex Langevin dynamics. Not only do these methods offer asymptotic stability, rendering the issue of runaway solution moot, but they also allow us to simulate at comparatively large Langevin time steps, leading to lower computational cost. We compare different ways of regularizing the underlying path integral and estimate the errors introduced due to the finite Langevin time steps. Based on that insight, we implement benchmark (non-)thermal simulations of the quantum anharmonic oscillator on the canonical Schwinger-Keldysh contour of short real-time extent.

4.1 Motivation

The sign problem (see ref. [11] for a mini-review) remains one of the central open challenges in modern theoretical physics and hinders progress in various different subfields. It underlies the challenges encountered in the study of transport properties of the quark-gluon-plasma [85, 86, 87, 88, 89, 90, 91, 92], it is the central hurdle in the exploration of the QCD phase diagram at large Baryon density [93, 94, 95, 96, 97, 98] and impacts the study of the thermodynamics of imbalanced Fermi gases [99, 100, 101], to name just three. The term sign problem refers to the fact that many strongly correlated quantum systems of phenomenological relevance can only be expressed through a path integral with complex valued Feynman weight. In turn, Monte-Carlo sampling methods, successful in case that the Feynman weight is purely real, become inapplicable and system-specific strategies must be developed. One of the most technically challenging sign problems occurs in case of quantum systems formulated in Minkowski spacetime, where the Feynman weight amounts to a pure phase.

The sign problem has been shown to be NP-hard [32], which implies that a one-size-fits-all approach is unlikely to exist. Nevertheless, many examples are known in which the sign problem has been successfully overcome or at least tamed. An active research community (for a recent review see ref. [20]) is exploring multiple strategies. Among these are variants of reweighing, extrapolation from complex parameters and the reformulation of the system of interest in new degrees of freedom unaffected by a sign problem.

The present study sets out to contribute to ongoing efforts to beat the sign problem by considering the complexification of system degrees of freedom. This research field has a long history, giving birth to two major promising currents: the Lefshets thimble approach [102] and complex Langevin [103]. In the former, one identifies manifolds in the complex plane, the so-called thimbles, on which the imaginary part of the classical action remains constant and thus ordinary Monte-Carlo sampling may commence. A residual sign problem persists as one has to average over different thimbles. Together with the computationally demanding task of locating the thimbles, these challenges constitute two areas of active research interest.

Complex Langevin on the other hand is based on the concept of stochastic quantization [22, 104]. Quantum and statistical fluctuations of a system are represented by noise in an additional $(d+1)+1$ temporal dimension, which reproduces the correlation functions in $(d+1)$ dimensions. In practice one is required to evolve the field degrees of freedom by a stochastic partial differential equation (SDE) in an additional, so-called Langevin time, generating representations of the quantum system along the way. Expectation values of observables are estimated by taking the mean over these field configurations. Langevin stochastic quantization has proven successful in systems in which naive Monte-Carlo methods are also applicable [105]. It has furthermore been shown to correctly simulate several systems with complex weights and thus complexified field degrees of freedom [103]. Even though straight forward in principle, it has been realized early on by the community that in its standard formulation, the complex Langevin approach suffers from three major shortcomings, which have to be addressed, before the method can serve as a reliable tool in the precision study of strongly correlated quantum systems.

The three main challenges affecting complex Langevin identified in the literature are its *stability*, the *ergodicity in the presence of non-holomorphic actions* and most crucially the *convergence to incorrect results* (for recent insight see e.g. [68, 58]), which is intimately related to the appearance of tail structures in histograms of observables (see Refs.[55, 62]). We believe that it is paramount to disentangle each of these issues, in order to be able to solve them one-by-one. Hence we focus in this study solely on the question of stability, returning to the remaining two in future work. (I.e. in order to remain in the parameter range where the complex Langevin method itself is known to converge to the correct results, we limit ourselves to a short real-time extent in this study.)

The question of stability in complex Langevin is intimately connected to the well-known phenomenon of *runaway solutions*. In general, such divergent behavior can arise from two sources. Either the complex Langevin method itself does not converge to a finite result, or the numerical methods used to implement the discrete Langevin time evolution introduce artifacts, which in turn give rise to unphysical divergencies. In order to make progress on understanding the former, we must disentangle numerical artifacts from methods artifacts.

Observed early on [50], runaways are now commonly treated by deploying adaptive step-size prescriptions [52] in the solution of the stochastic Langevin dynamics. One motivation for our work is the fact that even though adaptive step-size has proven to alleviate the problem of runaways in many systems in practice, it does not prevent their occurrence in principle. I.e. runaway solutions may appear even if adaptive step-size is deployed (see e.g. [59]), which in gauge theories has led to the introduction to the additional gauge-cooling approach [106].

In this paper we set out to explore solvers that can prevent the occurrence of runaway solutions by construction. To this end, we consider the stability of complex Langevin from the point of view of the stiffness of the underlying SDEs. While no precise definition of stiffness exists, we take the pragmatic view that it refers to systems in which naive explicit time-stepping prescriptions fail to recover the correct solution. Surveying the landscape of complex Langevin implementations, we find that a majority of studies rely on the simple forward Euler discretization. Early on, improvements in the spirit of deterministic Runge-Kutta methods have been proposed [107], but to our knowledge only two studies [108, 109] has embraced a higher-order method that takes into account the stochastic character of the Langevin evolution equation.

Our study aims at bringing to the table some of the progress made in the solution of SDEs in other fields. In particular, we propose to deploy implicit solvers, which are designed with stiff problems in mind. Besides the simple Euler-Maruyama scheme, which we use extensively in this paper, we will discuss what ingredients are needed in order to set up higher-order schemes for SDEs, compared to the case of purely deterministic equations.

Once stable solvers are available, we can proceed to investigate the stability and accuracy of the complex Langevin method itself. Our goal lies in simulating real-time physics, which in the continuum requires a form of regularization. After exploring different ways how a regularization may be incorporated in the Langevin evolution, we implement high accuracy simulations of the (0+1) dimensional anharmonic oscillator in thermal equilibrium and as a genuine initial value problem from a Gaussian density matrix.

The paper is organized in the following way: we start in section 4.2 with an introduction of the equations underlying the complex Langevin approach and discuss some explicit and implicit SDE solvers for their solution. In the following, we prepare the grounds for numerical simulations by introducing and discretizing the anharmonic oscillator model in section 4.3.1. We discuss different ways to regularize its path integral section 4.3.2 and will learn how to describe the errors made by a finite step size in the

Langevin evolution. Armed with this insight, we carry out benchmark complex Langevin simulations of the quantum anharmonic oscillator on the canonical Schwinger-Keldysh contour at short real-times both in thermal equilibrium and in a non-equilibrium setting in section 4.4. We close with a summary and outlook in section 4.5.

4.2 Complex Langevin and SDE solvers

The task at hand is to compute quantum statistical expectation values of an observable O. Conventionally such expectation values are formulated in terms of a Feynman path integral

$$\langle O\rangle = \frac{1}{Z}\int \mathcal{D}\phi\; O[\phi]e^{iS[\phi]}, \quad S[\phi] = \int d^dxL[\phi]. \tag{4.1}$$

where $Z = \int \mathcal{D}\phi \exp[iS[\phi]]$ denotes the partition function.

In *Stochastic Quantization*, we obtain the expectation values from the evolution of the system in an artificial Langevin time $\tau_{\rm L}$ (for an in-depth review of the approach, see Ref.[53]). The Langevin-like evolution equation for the field $\phi(x,\tau_{\rm L})$ in its simplest form consists of a drift term, derived from its classical action $S[\phi]$, as well as a Gaussian noise term $\eta(x,\tau_{\rm L})$

$$\begin{aligned}&\frac{d\phi}{d\tau_{\rm L}} = i\frac{\delta S[\phi]}{\delta\phi(x)} + \eta(x,\tau_{\rm L}) \quad \text{with}\\ &\langle\eta(x,\tau_{\rm L})\rangle = 0, \quad \langle\eta(x,\tau_{\rm L})\eta(x',\tau'_{\rm L})\rangle = 2\delta(x-x')\delta(\tau_{\rm L}-\tau'_{\rm L}).\end{aligned} \tag{4.2}$$

The quantity x may e.g. refer to a four-vector $x = (x_0, \mathbf{x})$ with Minkowski (real-time) x_0 and the 3 spatial dimensions $\mathbf{x}$. It is important to keep in mind that the physical time (x_0) and the fictitious Langevin time $\tau_{\rm L}$ are not related to each other. Note that the delta function in the correlator of the noise encompasses all dimensions of x. This prescription places an independent stochastic process at each space-time point.

Due to the complex drift term, the field degrees complexify and we can rewrite the evolution equations instead in terms of the real and imaginary part of the field as

$$\phi(x, \tau_{\rm L}) = \phi_R(x, \tau_{\rm L}) + i\phi_I(x, \tau_{\rm L}), \tag{4.3}$$

such that eq. (4.2) turns into two coupled but real-valued equations for the real- and imaginary part of the field degrees of freedom

$$\begin{aligned} \frac{d\phi_R}{d\tau_{\rm L}} &= \mathrm{Re}\left[i \frac{\delta S[\phi]}{\delta\phi(x)}\bigg|_{\phi=\phi_R+i\phi_I} \right] + \eta(x, \tau_{\rm L}), \\ \frac{d\phi_I}{d\tau_{\rm L}} &= \mathrm{Im}\left[i \frac{\delta S[\phi]}{\delta\phi(x)}\bigg|_{\phi=\phi_R+i\phi_I} \right]. \end{aligned} \tag{4.4}$$

We have here used the standard construction in which the noise term $\eta(x, \tau_{\rm L})$ is real. These are the stochastic partial differential evolution equations we will solve in the subsequent sections. The first central task is to find appropriate numerical solvers to accommodate these equations.

4.2.1 Numerical schemes

Stochastic partial differential equations (SDE) are a central modern tool in the modeling of various phenomena in science, technology and in particular finance. Most of the equations arising in these research fields do not lend themselves to an analytic treatment and thus require numerical solvers. To this end, the past two decades have seen vigorous research activity in the development of accurate and efficient algorithms. One major impulse towards these developments can be found in the by now classic book by Klöden and Platen [110], which not only contains a comprehensive survey of both explicit and implicit SDE solvers but also provides a pedagogic introduction into the underlying Ito and Stratonovic calculus. One central message of the book states that the series expansions, commonly used to set up deterministic discretization schemes, need to be amended by additional terms in the stochastic case due to the different scaling properties of stochastic variables. In particular, it is shown that deterministic algorithms may yield much lower convergence rates in an SDE setting than for the PDEs they were originally designed for.

The goal of this section is to introduce some of these numerical schemes and to explicitly match the complex Langevin equations to the mathematical notation used in the literature, preparing us for a straightforward implementation through standard libraries, such as the SDE solver package found in the Julia language.

Let us formulate a general stochastic differential equation for N stochastic variables ϕ^i in Langevin time τ_{L}, enumerated by the superscript j.

$$d\phi^j(\tau_{\mathrm{L}}) = a^j(\phi,\tau_{\mathrm{L}})d\tau_{\mathrm{L}} + \sum_j b^{jk}(\phi,\tau_{\mathrm{L}})dW^k. \tag{4.5}$$

Its evolution is governed by a diffusion term conventionally denoted by $a^j(\phi,\tau_{\mathrm{L}})$, which may depend on all other stochastic variables, as well as the Langevin time explicitly. We incorporate N independent Wiener processes dW^j which obey the standard relations

$$\left\langle \int_0^{\tau_{\mathrm{L}}} dW^j \right\rangle = 0 \quad \text{and} \quad \left\langle \int_0^{\tau_{\mathrm{L}}} dW^j \int_0^{\tau_{\mathrm{L}}} dW^k \right\rangle = \delta^{jk} \int_0^{\tau_{\mathrm{L}}} d\tau'_{\mathrm{L}}. \tag{4.6}$$

They affect the dynamical degrees of freedom ϕ^j via the mixing matrix $b(\phi,\tau_{\mathrm{L}})$. This general non-constant noise coefficient matrix may have a non-trivial dependence on Langevin time and the stochastic variables.

In the concrete case of stochastic quantization, we replace the discrete parameter j with the combination of an discrete index for field degrees per spacetime point and the continuous parameter x. Hence the sum over j turns into a combined sum and integral $\sum_j \rightarrow \sum_j \int dx$. Kronecker deltas remain for discrete indices, while we have to introduce Delta functions for spacetime $\delta_{jk} \rightarrow \delta_{jk}\delta(x-x')$. This leads us to the expression

$$d\phi^j(x,\tau_{\mathrm{L}}) = a^j(\phi,x,\tau_{\mathrm{L}})d\tau_{\mathrm{L}} + \int dx' \sum_k b^{jk}(\phi,x,x',\tau_{\mathrm{L}})dW^k(x',\tau_{\mathrm{L}}), \tag{4.7}$$

which can be matched to our complex Langevin eq. (4.2) using

$$a^j(\phi,x,\tau_{\mathrm{L}}) = i\frac{\delta S[\phi]}{\delta\phi^j(x)}, \quad b^{jk}(\phi,x,\tau_{\mathrm{L}}) = \sqrt{2}\delta^{jk}\delta(x-x'). \tag{4.8}$$

Note that in its standard form, the CL noise coefficients are constant in ϕ and τ_{L}.

As a first step, we need to take care of the drift term, which originates from a discretized action. To this end, one discretizes physical spacetime on which the field degrees live, turning continuous x into a discrete set of coordinates x_m. The integral in the classical action may be approximated using a Newton-Cotes formula with the weights $\omega_m = \omega(x_m)$, which yields the following drift and noise terms for the continuous Langevin time SDE

$$a^{j,m}(\phi, \tau_{\text{L}}) = \frac{i}{\omega_m} \frac{\partial S[\phi]}{\partial \phi^j(x_m)}, \quad b^{jk,ml}(\phi, \tau_{\text{L}}) = \sqrt{\frac{2}{\omega_m}} \delta_{jk} \delta_{ml}. \tag{4.9}$$

This expression for the diffusion term a and the noise coefficient matrix b can be used straightforwardly in numerical schemes for stochastic differential equations. In the remainder of this section, we discuss some of these schemes in more detail.

The most common implementation of the CL dynamics deploys the simple, i.e. lowest order, Euler-Maruyama (EM) scheme. Discretizing Langevin time in equidistant steps of size $\Delta\tau_{\text{L}}$ we introduce the discrete Wiener process increment $\Delta W^j_\lambda = W^j_{\lambda+1} - W^j_\lambda$. The subscript λ denotes at which Langevin time step the random variables are evaluated. The update step is then given by the following expression, where summation over repeated indices is implied

$$\begin{aligned} &\phi^{j,n}_{\lambda+1} = \phi^{j,n}_\lambda + \Delta\tau_{\text{L}} \left[\theta a^{j,n}(\phi_{\lambda+1}) + (1-\theta) a^{j,n}(\phi_\lambda)\right] + b^{jk,nm}(\phi_\lambda)\, \Delta W^{k,m}_\lambda \\ &\text{with} \quad \langle \Delta W^{j,n}_\lambda \rangle = 0, \quad \langle \Delta W^{j,n}_\lambda \Delta W^{k,m}_\lambda \rangle = \Delta\tau_{\text{L}} \delta_{jk} \delta_{nm}. \end{aligned} \tag{4.10}$$

Here we actually refer to a whole class of EM schemes, which differ by the choice of a single real-valued parameter θ. It controls the level of implicitness. For $\theta = 0$ one recovers the fully explicit forward EM scheme, while for $\theta = 1$ the implicit variant ensues. The choice of $\theta = 1/2$ is special, as it refers to a semi-implicit Crank-Nicholson-like implementation of the EM scheme.

In contrast to the deterministic Euler schemes, the different variants of the EM scheme for a non-trivial noise term share a numerical accuracy of strong order $O(\sqrt{\Delta\tau_{\mathrm{L}}})$. I.e. in general they perform worse than in the deterministic case, which is a common ailment afflicting the direct application of deterministic schemes to SDEs. In the case of simple CL dynamics we are fortunate however, in that the noise term remains trivial and thus the simple EM scheme can be shown to be of strong order $O(\Delta\tau_{\mathrm{L}})$.

While the same numerical accuracy is shared among the different members of the EM family of schemes, their numerical stability varies significantly. It is well known that the forward EM scheme is at best conditionally stable, while the fully implicit scheme $\theta = 1$ is robust against instabilities, being what is called in the literature L-stable. Similarly, it can be shown (c.f. Crank-Nicolson) that the semi-implicit scheme $\theta = 1/2$ is also unconditionally asymptotically stable [111].

We stress that stability and accuracy are two separate qualities of a scheme, where stability only refers to the ability of the numerical solver to follow the true solution within the limitations placed by the accuracy of the scheme. Unconditional stability however also guarantees that as long as the true solution remains bounded, the scheme will not produce divergent runaway solutions. This property is what leads us to propose the deployment of (semi-)implicit solvers for complex Langevin, as it allows us to disentangle possible breakdown of the stochastic quantization prescription from a breakdown of the numerical solver.

In general, an implicit scheme is more costly than its explicit cousin at each individual update step. We need to solve a non-linear system of equations arising from the drift term $a^{j,n}(\phi_{\lambda+1})$ in eq. (4.10), which is commonly implemented by a variant of Newton's method. As we will see, the favorable stability properties may however allow one to choose larger step sizes in Langevin time, leading to an overall reduction in computation cost.

Similarly as for deterministic differential equations, one may improve on the simple Euler schemes by developing *Runge-Kutta* solvers for SDEs, usually referred to as *SRK* schemes. They offer a straightforward way to increase the accuracy of the solver by combining approximations of the stochastic variables at intermediate steps within one update interval. The simplest of these is the *Runge-Kutta Milstein* scheme [110, 112] of strong order $\mathcal{O}(\Delta\tau_{\rm L})$ for diagonal noise

$$\begin{aligned}\phi^{j,n}_{\lambda+1} =&\phi^{j,n}_{\lambda} + \Delta\tau_{\rm L}\ \left[\theta a^{j,n}\left(\phi_{\lambda+1}\right) + (1-\theta)a^{j,n}\left(\phi_{\lambda}\right)\right] + b^{j,n}\left(\phi_{\lambda}\right)\ \Delta W^{j,n}_{\lambda}\\ &+\frac{1}{2\sqrt{\Delta\tau_{\rm L}}}\left(b^{j,n}(\Upsilon_\lambda) - b^{j,n}(\phi_\lambda)\right)\left\{(\Delta W^{j,n}_{\lambda})^2 - \Delta\tau_{\rm L}\right\}\quad \text{with}\\ \Upsilon^{j,n}_{\lambda} =&\phi^{j,n}_{\lambda} + a^{j,n}\left(\phi_{\lambda}\right)\Delta\tau_{\rm L} + b^{j,n}\left(\phi_{\lambda}\right)\sqrt{\Delta\tau_{\rm L}}.\end{aligned} \tag{4.11}$$

Again we have indicated a whole family of schemes, whose implicitness is governed by the θ parameter. Note that these schemes differ from a naive application of the deterministic second-order Runge-Kutta (RK2) prescription through the presence of a term quadratic in the Wiener process in the second line of eq. (4.11). In case of simple CL eq. (4.2) with constant real noise coefficients, the Milstein scheme reduces to the EM scheme, reaffirming that for trivial noise the simplest algorithm already offers order 1.0 accuracy.

Let us also touch on higher-order schemes. The next order one can reach is 1.5 [110], at which the SRK prescription for constant additive noise reads

$$\begin{aligned}\phi_{\lambda+1} =&\phi_{\lambda} + \Delta\tau_{\rm L}\ \frac{1}{2}\left[a\left(\phi_{\lambda+1}\right) + a\left(\phi_{\lambda}\right)\right] \qquad (4.12)\\ &+ b\ \Delta W_{\lambda} + \frac{1}{2\sqrt{\Delta\tau_{\rm L}}}\left\{a(\Upsilon^{\lambda}_{+}) - a(\Upsilon^{\lambda}_{-})\right\}\left\{\Delta Z_{\lambda} - \frac{1}{2}\Delta W_{\lambda} d\tau_{\rm L}\right\}\\ \text{with}\quad &\Upsilon^{\lambda}_{\pm} = \phi_{\lambda} + a\left(\phi_{\lambda}\right)\Delta\tau_{\rm L} \pm b\sqrt{\Delta\tau_{\rm L}}.\end{aligned}$$

Here we use vector notation and we have explicitly chosen $\theta = 1/2$ for simplicity of the presentation. In this equation, we find that a genuinely new contribution arises even for trivial noise. It consists of a combination of the drift term together with Wiener processes ΔW and ΔZ. The latter one refers to additional independent processes with the same mean and variance as the ΔW as well as $\langle \Delta W^j \Delta Z^j \rangle = 0$. By incorporating the proper contributions arising from Ito's lemma in the series expansions underlying these Runge-Kutta schemes one may thus construct consecutive improvements to the naive EM scheme.

In our study, we will draw upon the implementation of the abovementioned schemes through the SDE module in the *DifferentialEquations.jl* [113, 114, 115] library provided in the Julia language. The concrete implementations of eqs. (4.10) to (4.12) differ slightly due to performance improvements outlined in the literature (see the documentation of [113]), which however has no effect on their stability and accuracy properties.

All the methods listed above can be implemented with an adaptive stepsize prescription. This offers two concrete benefits. On the one hand, the stability properties of a simulation can be improved, as the step size is adapted to fulfill the Courant–Friedrichs–Lewy stability condition at each update. For stiff problems and explicit solvers, this approach is limited in practice by the step size becoming so small that the number of steps along Langevin time grows beyond available computational power. On the other hand adaptive step size also allows us to increase the step size at intermediate times to reduce the computational burden while staying within a predefined accuracy tolerance for the update step. One drawback of adaptive step size is the fact that an analytic investigation of the properties of the solver becomes more involved. We will thus deploy adaptive step size in all simulations except those where we study the finite time discretization artifacts and look at the corrections from the discretized Fokker-Planck equation.

Many different adaptive step prescriptions are deployed in the literature. One of the more sophisticated approaches implemented e.g. in the Julia library compares updates of solvers of different order and takes the difference as an error estimate [114]. The step size then is chosen to keep this error estimate below a pre-defined threshold. More simply we may monitor the size of the drift term in the Langevin equation and adjust the time step such that the change induced by the drift term remains below a certain threshold. We have found that some adaptive step-size algorithms implemented in the literature do not contain a limit on the maximum step size, which may spoil the accuracy of the outcome and required us to implement such an upper limit by hand.

4.2.2 On the issue of large excursions

Having reviewed different explicit and implicit prescriptions for the solution of the complex Langevin SDE, we may now explore how these methods fare in addressing the issue of stability. A common challenge that plagues complex Langevin simulations is the occurrence of large excursions. While the overwhelming majority of trajectories contributing to the final expectation value are located in a well-contained area around the origin, some paths are found to venture significantly further out into the complex plane. A simple example of this behavior can be found in the system of a single degree of freedom, evolving in the potential $V(\phi) = i\phi^4$. On average it leads to paths that stay within around 2 dimensionless units from the origin, however excursions up to $|\phi| \sim 10$ sporadically occur.

These excursions can be understood by inspecting the flow field $-4i\phi^3$, as shown in fig. 4.1 along the line where $\phi_R = \phi_I$. As $\phi_I \gtrsim \phi_R$ the flow lines tilt upward, while for $\phi_I \lesssim \phi_R$ they tilt downwards. As one moves exactly on top of the line, the field lines will keep going straight out towards infinity. This is a property of the continuum theory and not an artifact of the numerical solution.

In principle, this is not a problem, since the noise term makes sure that one never stays on this line indefinitely. However, as the size of ϕ increases, the size of the flow $4i\phi^3$ also increases significantly compared to the noise term. In turn, after the noise kicks the system away from the diverging path, it now follows a path dominated by the drift term with only a small contribution of the noise term. Such a path tends to go out to even larger values of $|\phi|$ until it eventually returns to the dominating region. A representative example is shown in fig. 4.1 as the green solid line.

It has been understood that such excursions constitute one of the rea-

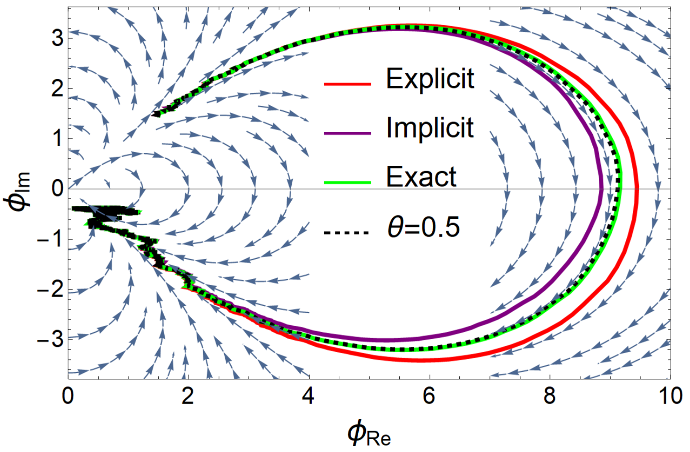

Figure 4.1: Example of complex Langevin paths for a single degree of freedom in the potential $V(\phi) = i\phi^4$ based on different solvers: (red) explicit EM, (violet) implicit EM, (dashed) unitary $\theta = 1/2$ EM and the exact solution given as green line. Each path is initiated at $(1.5, 1.47)$ close to the divergent flow line. The Langevin time step size is kept constant at $\Delta\tau_{\mathrm{L}} = 10^{-4}$. Note the characteristic over and undershooting of the explicit and implicit method respectively.

sons for the stability issues of numerical implementations of complex Langevin. Let us have a look at how well the true path is recovered by the simple EM schemes for different settings of implicitness for a fixed step size. In general, the explicit method is prone to overshooting the correct trajectory (red solid line), while the fully implicit method also fails to stay close to it but does so by undershooting the correct result (violet solid line). The overshooting of the explicit method easily leads to divergent behavior as the errors accumulate. For the implicit method on the other hand the simulation remains stable. Note however that its accuracy still suffers due to the deviation from the true trajectory. The semi-implicit EM scheme on the other hand combines the best of both worlds, as it offers the unconditional stability of the implicit scheme and limits the undershooting to a minimum, as can be seen in the dashed black line in fig. 4.1. Note that due to the large flow the actual Langevin time spend in one of these excursions is very small compared to the total length of the trajectories needed to accumulate reasonable statistical uncertainties.

We can understand the behavior seen in fig. 4.1 already from an inspection of the discretization prescription in the free theory. There the drift term is linear ($a(\phi) = iM\phi$) and we can rewrite the EM scheme of eq. (4.10) as

$$\phi^{\lambda+1} = (1 - i\Delta\tau_{\mathrm{L}}\theta M)^{-1}\left\{(1 + i\Delta\tau_{\mathrm{L}}(1-\theta)M)\phi^{\lambda} + \sqrt{\Delta\tau_{\mathrm{L}}}\eta^{\lambda}\right\}. \quad (4.13)$$

Taking the expectation value of the field at $\tau_{\mathrm{L}} = (\lambda + 1)\Delta\tau_{\mathrm{L}}$ we get

$$|\langle\phi^{\lambda+1}\rangle| = \left|\frac{1 + i\Delta\tau_{\mathrm{L}}(1-\theta)M}{1 - i\Delta\tau_{\mathrm{L}}\theta M}\right| |\langle\phi^{\lambda}\rangle|. \quad (4.14)$$

For $\theta = 0$ the fraction simplifies to $|1 + i\Delta\tau_{\mathrm{L}}M| > 1$, which induces an increase in the magnitude of the field value for each step. On the other hand for $\theta = 1$ one finds $|1 - i\Delta\tau_{\mathrm{L}}M|^{-1} < 1$, which represents shrinkage of the magnitude. For the special case of ($\theta = \frac{1}{2}$) we obtain a semi-implicit scheme with $|\langle\phi^{\lambda+1}\rangle| = |(1 - i\Delta\tau_{\mathrm{L}}M/2)/(1 + i\Delta\tau_{\mathrm{L}}M/2)|\,|\langle\phi^{\lambda}\rangle| = |\langle\phi^{\lambda}\rangle|$, which in the free case exactly preserves the magnitude of the expectation value of the field.

We chose the above example to illustrate a key qualitative difference between solvers of varying degrees of implicitness. It should be mentioned that for the specific scenario shown here, the differences are only sizeable since we do not use an adaptive step size. A $\Delta\tau_L = 10^{-4}$ allows for stable dynamics close to the origin where the drift is small. However at $|\phi| \sim 10$ we have a relatively large drift term of around $\Delta\tau_L 4\phi^3 = 0.4$. In practice, using adaptive step size one would reduce $\Delta\tau_{\mathrm{L}}$ along the excursion, keeping the deviation from the exact solution small. The conclusion that explicit schemes accumulate errors according to overshooting of the true trajectory and implicit schemes according to undershooting it remains unchanged.

4.3 Towards stable real-time simulations of the quantum anharmonic oscillator

4.3.1 Formulating and discretizing the model

Our main goal in this study is to implement stable simulations of the early real-time dynamics of a strongly coupled quantum anharmonic oscillator. This system amounts to a (0+1)d field theory prototype and has been studied in the literature in detail [106, 116], establishing itself as a benchmark for the success of different real-time approaches. Real-time expectation values for an observable O arising in a system that evolves from a mixed initial state ρ can be described via the Schwinger-Keldysh closed time-path formalism. In its canonical implementation, it deals with field degrees of freedom placed on a time contour, with both a forward-facing branch along the time axis (housing ϕ_+) and a backward branch (housing ϕ_-), both of which are attached at the initial time to the density matrix $\rho(\phi_1, \phi_2)$

$$\langle O(\phi)\rangle = \frac{1}{Z}\int d\phi_1 \int d\phi_2 \; \rho(\phi_1,\phi_2) \int_{\phi_2}^{\phi_1} D\phi^+ D\phi^- \; O(\phi) \; e^{iS[\phi_+]-iS[\phi_-]}. \tag{4.15}$$

Let us consider the case of thermal equilibrium with inverse temperature $\beta = 1/T$. The density matrix takes on the standard Boltzmann form $\rho \propto \exp[-\beta H]$ and we may conveniently absorb the sampling over initial conditions into a path integral along the imaginary time axis, compactified to a length of β

$$\langle O(\phi)\rangle = \frac{1}{Z}\int D\phi_E e^{-S_E[\phi_E]} \int_{\phi_E(\beta)}^{\phi_E(0)} D\phi^+ D\phi^- \; O(\phi) \; e^{iS[\phi_+]-iS[\phi_-]}. \tag{4.16}$$

The corresponding Schwinger Keldysh contour now contains three parts: the forward and backward real-time branch, as well as the Euclidean contour all of which contribute with their own action and which we will summarize as $iS[\phi_+] - iS[\phi_-] - S_E \rightarrow iS[\phi]$.

The real-time action for the anharmonic oscillator explicitly reads

$$S = \int dx_0 \left\{ \frac{1}{2} \left(\frac{\partial \phi}{\partial x_0} \right)^2 - V(\phi) \right\}, \quad V(\phi) = \frac{1}{2} m\phi^2 + \frac{\lambda}{4!}\phi^4, \tag{4.17}$$

where λ refers to the coupling constant. We give all of our results in units of m, which is the same as setting $m = 1$. The first step to take is to discretize the time coordinate $x_0 \in \mathbb{C}$ along the Schwinger-Keldysh contour. To this end, we introduce a contour parameter $\xi \in \mathbb{R}$ which on the forward and backward branch refers to a real-valued time, while on the Euclidean branch refers to a negative imaginary time. The time integral in the action becomes a line integral over the contour parameter, discretized into N_C steps a_j. The fields, evaluated at the discrete real-time steps are denoted by $\phi(\sum_j a_j) = \phi_j$. We deploy the trapezoidal rule for the action integral, which amounts to averaging over the left and right Riemann sums. Consistently we approximate the derivative by finite differences choosing forward difference at the point j and backward differences at the point $j + 1$, leading us to the standard expression

$$S = \frac{1}{2} \sum_j \left\{ \frac{\left(\phi_{j+1} - \phi_j\right)^2}{a_j} - a_j \left[V(\phi_{j+1}) + V(\phi_j) \right] \right\}. \tag{4.18}$$

To implement the complex Langevin equations of motion, we need to calculate the drift term $i\frac{\delta S}{\delta \phi_j}$. Using the discretization scheme above we obtain

$$\begin{aligned} i\frac{\delta S[\phi]}{\delta \phi_j} = \frac{i}{\frac{1}{2}\left(|a_j| + |a_{j-1}|\right)} \bigg\{ & \frac{\phi_j - \phi_{j-1}}{a_{j-1}} - \frac{\phi_{j+1} - \phi_j}{a_j} \\ & - \frac{1}{2}\left[a_{j-1} + a_j\right] \frac{\partial V(\phi_j)}{\partial \phi_j} \bigg\}. \end{aligned} \tag{4.19}$$

The factor $\frac{1}{\frac{1}{2}\left(|a_j|+|a_{j-1}|\right)}$, which we have included here explicitly in the drift term must then also be consistently included in the noise term $b^{jk} = \sqrt{\frac{2}{\frac{1}{2}(|a_k|+|a_{k-1}|)}}\delta_{jk}$. Note however that via a rescaling of the drift term, one may drop the factors of a if one consistently drops them also from all the Kronecker deltas associate with functional derivatives. At this stage we have only discretized the physical coordinates, leaving us with a continuous Langevin time prescription to stochastically quantize the anharmonic oscillator.

The discretization of the action already introduces numerical artifacts in the solution of the continuous-time complex Langevin equation. In order to make sure that we do not misinterpret such errors as arising from the finite Langevin time discretization in an actual simulation, we take a closer look at them here.

Analogous to constructing the transfer matrix operator we can go backward from the discretized path integral to the corresponding operator expressions while leaving the real-time step size a_j finite. In that case, we have to deal with the fact that the Campbell-Baker-Hausdorff formula gives non-trivial contributions when decomposing the action integral into individual exponentials. Let us define the exponentiated Hamiltonian of the system via the following matrix elements

$$\langle\phi_{j+1}|\exp(ia_jH)|\phi_j\rangle = \exp\left[\frac{1}{2}\frac{\left(\phi_{j+1}-\phi_j\right)^2}{a_j} - \frac{1}{2}a_jV(\phi_j) - \frac{1}{2}a_jV(\phi_j)\right]. \tag{4.20}$$

According to our eq. (4.18) the RHS contains the potential evaluated at neighboring values of the field ϕ_{j+1} and ϕ_j, which requires that the potential operator acts on both the left and right state. Since only one complete set of momentum eigenstates is involved in transforming the kinetic term back to its operator form we end up with the expression

$$\exp(ia_j\hat{H}_{\mathrm{s}}) = \exp\left[\frac{a_j}{2}V(\hat{\phi})\right]\exp\left[\frac{a_j\hat{\pi}^2}{2}\right]\exp\left[\frac{a_j}{2}V(\hat{\phi})\right] + O(a^2). \tag{4.21}$$

Had we considered just one single potential term in eq. (4.18) the corresponding operator expressions would have turned out to be

$$\exp(ia_j\hat{H}_\mathrm{r}) = \exp\left[\frac{a_j\hat{\pi}^2}{2}\right]\exp\left[a_jV(\hat{\phi})\right] + O(a) \quad \text{or} \tag{4.22}$$

$$\exp(ia_j\hat{H}_\mathrm{l}) = \exp\left[a_jV(\hat{\phi})\right]\exp\left[\frac{a_j\hat{\pi}^2}{2}\right] + O(a). \tag{4.23}$$

Using as parameters $\lambda = 24$ and $m = 1$, similar to what we will deploy in the actual complex Langevin simulations in the following sections, we can now study the effects of the finite real-time spacing explicitly. To this end we compute the forward correlator $\langle\phi(x_0)\phi(0)\rangle$ using matrix mechanics in the truncated Hilbert space spanned by the 32 lowest-lying energy eigenstates of the harmonic oscillator, according to the different effective Hamilton operators $\hat{H}$ defined above.

As can be seen in fig. 4.2, we find a characteristic artifact introduced by the finite real-time steps. Its main manifestation is the appearance of a non-zero value of the imaginary part of the correlator at the origin. Neighboring values of the imaginary part are correspondingly also shifted away from their true values. When using the $O(a)$ discretization of eq. (4.22) this effect is of the order of 12% for $a = 0.08$ (with respect to the maximum value the imaginary part of the correlator takes on.) The strength of the effect scales as expected linearly with a so that we obtain 6% deviation for $a = 0.04$ and 4% deviation for $a = 0.08/3$. Switching from eq. (4.22) to (4.23) changes the sign in the shift of the imaginary part, indicating that the discretization amounts to a complex phase factor. Combining the opposing phase factors in the symmetric formulation of eq. (4.21) cancels out the effect to a significant extent, reducing the deviation from the continuum results to 0.2% already for $a = 0.08$. We will make sure to keep these discretization artifacts below the percent level in the Complex Langevin simulations in the following.

4.3.2 Regularizing the model

To make the continuum theory well-defined on the canonical Schwinger-Keldysh real-time contour, we need to introduce an infinitesimal damping

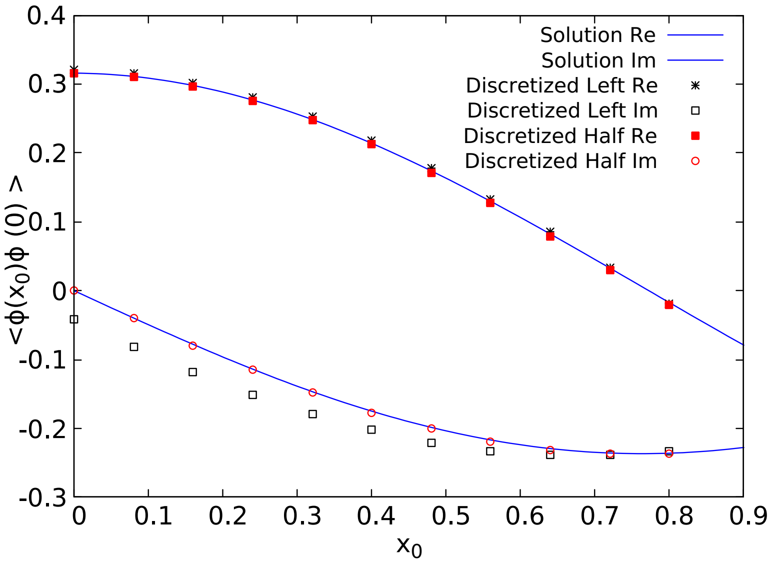

Figure 4.2: Visualization of real-time discretization artifacts in the unequal-time correlation function $\langle\phi(x_0)\phi(0)\rangle$. True values (solid lines) are obtained from matrix mechanics in the truncated Hilbert space spanned by 32 energy eigenstates of the harmonic oscillator. Note the shift in the imaginary part when using the first order discretization of eq. (4.22) (open squares) for a lattice spacing of $am = 0.08$. The symmetric discretization of eq. (4.21) significantly improves the agreement with the continuum results as seen in the open circles.

term into the otherwise purely oscillatory behavior of the Feynman path integral weight. Otherwise, the continuous Langevin-time evolution will not be able to converge to a finite result. In an analytic setting, this is conventionally achieved by introducing by hand an additional term $R > 0$ in the action such that the new effective action reads $\bar{S} = S + iR(\phi, \epsilon)$. A simple example of such a regulator is $R = \frac{1}{2}\epsilon\phi^2$ is e.g. discussed in [117].

It is long known that such a regulator also controls the rate of convergence in a complex Langevin simulation [106, 118]. Similar to an analytic computation where the correct result is obtained by setting $\epsilon \longrightarrow 0$ only a posteriori, a simulation operates at finite ϵ, which, depending on its chosen value may significantly distort the computed expectation values. Only after an extrapolation over several different simulations do we recover the true solution. It goes without saying that the closer we can simulate to the correct solution, the less troubled the extrapolation procedure will be. Therefore we will attempt to deploy as small a regulator as possible.

In this section, we will briefly discuss the classic approach to introduce a tilt in the Schwinger-Keldysh contour [106] and report on our observation that the implicit scheme itself offers a regularization by construction.

Tilted Schwinger-Keldysh contour

One way to regularize our model is to introduce an imaginary tilt in the Schwinger-Keldysh contour, as deployed e.g. in ref.[106] and shown in the leftmost panel of fig. 4.3. Since in the thermal setting we are interested in correlators on the forward contour (the values of the mixed correlators are related via the KMS relation) one tries to keep the tilt on the forward branch small. The backward branch on the other hand may tilt downward more steeply, as long as it reaches the negative imaginary axis before $-i\beta$. To be more concrete, the tilted Schwinger-Keldysh contour (fig. 4.3 a) has two distinct parts. Part one (C_1) is tilted under an angle α from 0 to $t_{max} - \sin(\alpha)\beta i$. Part two ($C_2$) is tilted such that it arrives at the imaginary axis at the point $-i\beta$, which due to periodic boundary conditions coincides with the starting point of the first part of the contour.

The information about the shape of the real-time contour is fully contained in the choice of the, in general complex, time step a_j. It denotes the distance between point j and $j+1$ on the contour and appears explicitly in the complex Langevin drift term in eq. (4.19). A tilt of the real-time contour manifests itself as a non-zero imaginary part in a_j such that

$$\begin{aligned}
\bar{S} =&\frac{1}{2}\sum_j \left\{ \frac{\left(\phi_{j+1}-\phi_j\right)^2}{a_j^R + ia_j^I} - (a_j^R + ia_j^I)\left[V(\phi_{j+1}) + V(\phi_j)\right] \right\} \\
=&\frac{1}{2}\sum_j \left\{ \frac{\left(\phi_{j+1}-\phi_j\right)^2}{|a_j|}(a_j^R - ia_j^I) - (a_j^R + ia_j^I)\left[V(\phi_{j+1}) + V(\phi_j)\right] \right\} \\
=&S + \frac{1}{2}\sum_j \left\{ \frac{\left(\phi_{j+1}-\phi_j\right)^2}{|a_j|}(-ia_j^I) - (ia_j^I)\left[V(\phi_{j+1}) + V(\phi_j)\right] \right\} \\
=&S + \frac{1}{2}\sum_j \left\{ \frac{\left(\phi_{j+1}-\phi_j\right)^2}{|a_j|} + \left[V(\phi_{j+1}) + V(\phi_j)\right] \right\}(-ia_j^I) \\
=&S + i\sum_j R(\phi, a_j^I).
\end{aligned} \tag{4.24}$$

We see that if a_i has a negative imaginary part, the overall prefactor becomes $(-ia_j^I) = +i|\mathrm{Im}(a_i)|$, turning the corresponding $R > 0$ into a positive quantity. Coming to the conclusion that such a positive term R successfully acts as a regulator however is not as straightforward as it appears at first sight. In the case of a complex Feynman weight, the field themselves becomes complexified and R exhibits both a real- and imaginary part. In the free theory ref.[117] has shown that a positive $R = \frac{1}{2}\epsilon\phi^2$ allows us to take a well defined late Langevin-time limit of the complex Langevin dynamics with a regularization of the two-point function that amounts to $\langle\phi(k)\phi(-k)\rangle = i/(k^2 - m^2 + i\epsilon)$ supporting the downward tilt of the Schwinger-Keldysh contour as an appropriate regulator.

Reducing the tilt of C_1 reduces the strength of the regularization. It is well known and we have reconfirmed in our numerical experiments that concurrently the stochastic dynamics become more and more stiff. I.e. when deploying an explicit scheme the probability to encounter runaway solutions increases significantly as the tilt is reduced. In the classic work of [106], an explicit solver was combined with a 0.01β tilt of the contour. While this tilt is small at the early times considered in previous and also

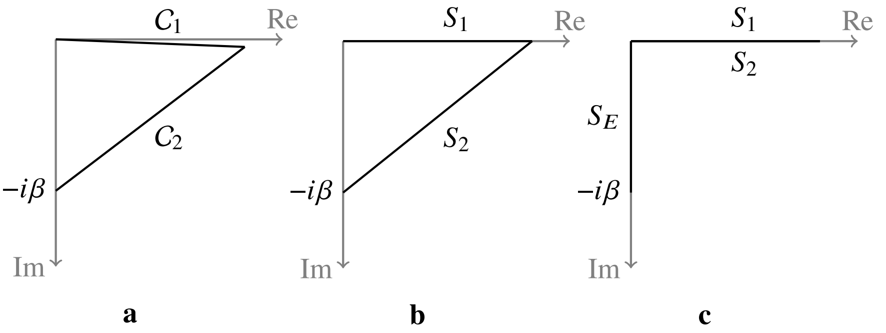

Figure 4.3: Three different realizations of the thermal Schwinger-Keldysh contour for a system with temperature $T = \frac{1}{\beta}$. The leftmost setting (a) corresponds to the contour adopted e.g. in ref.[106]. Our first goal is to be able to remove the tilt on the forward contour (b) and ultimately in preparation for the non-equilibrium setting to move both the forward and backward branch very close to the real-time axis (c). We find that the inherent regularization of the implicit solver allows us to realize scenarios (b) and (c) in practice.

this study, it already introduces a deviation from the true solution in the unequal-time correlation functions, which goes beyond the statistical errorbars of the simulation. With the goal of extending CL simulations to later real-times in the future, we will be urged to reduce the tilt even further.

As an example let us carry out a simulation using a similar setup as in [106], with a tilt of 0.01β in the anharmonic oscillator action (eq. (4.18)). In order to remain in the region where complex Langevin converges to the correct result, we select as maximum real-time extent $x_0^{\max} = 0.5$. As a solver the general Euler-Maruyama scheme (eq. (4.10)) with adaptive step-size[1] is chosen. The θ value in 4.10 is set to $\theta = \frac{1}{2}$ corresponding to a semi-implicit scheme, which, as we have seen in section 4.2.2, preserves the magnitude of the expectation value throughout the simulation well. We average over a total of 500 trajectories, each of which reaches a total

[1] All simulations based on the implicit scheme can be carried out without adaptive step-size. The numerical cost in that case will simply be higher, as an overall smaller step-size is needed to reach the same accuracy. Nevertheless, compared to simulating with an explicit scheme, we can deploy a much larger step size. The implicit scheme already works well with $\Delta\tau_{\mathrm{L}} = 10^{-3}$, while the explicit scheme requires us to go to $\Delta\tau_{\mathrm{L}} = 10^{-5}$.

Langevin time of $\tau_{\mathrm{L}}m = 100$. Observables are read out every $\delta\tau_{\mathrm{L}}m = 0.1$ in Langevin time.

We plot the real- and imaginary part of the unequal time correlation function $G_{++}(x_0) = \langle\phi(0)\phi(x_0)\rangle - \langle\phi(0)\rangle\langle\phi(x_0)\rangle$ on the forward branch vs. the contour parameter ξ in the top panel of fig. 4.4. While a small effect, we can already distinguish between the analytic solution on the real-time axis (black solid) and the solution on the tilted contour (green solid) within the precision of our simulation. The analytic solution here is obtained again using matrix mechanics in the truncated Hilbert space spanned by the 32 energy eigenstates of the harmonic oscillator.

As shown in the magnified insets, close to $x_0 = 0.5$ the tilt leads to a visible deviation from the true solution. I.e. such a tilt does affect the solution at early times and will become sizeable once the simulation can be extended to a phenomenologically relevant real-time extent.

In the lower panel of fig. 4.4 the field expectation value $\langle\phi\rangle$ and the equal time correlation function $\langle\phi^2\rangle$ are plotted vs. the contour parameter along both branches of the contour. We find that they agree with the constant value predicted by the true solution. I.e. as is known, these quantities are less susceptible to the tilt as the unequal-time correlation function.

Regularization via an implicit scheme

Previous studies and the preceding subsection have shown that introducing a large enough tilt in the Schwinger-Keldysh contour, as in section 4.3.2, allows us to regularize the oscillatory behavior of the path integral. Depending on the size of the tilt it does so effectively enough for even an explicit solver to capture the ensuing complex Langevin dynamics. The price to pay is a systematic deviation of the correlation function from the result on the real-time axis which grows with the maximum extent of the forward contour.

In this study, our goal is to explore the potential of implicit solvers for complex Langevin. We have already seen how their simplest formulation, in form of the EM scheme, avoids the occurrence of runaway solutions in

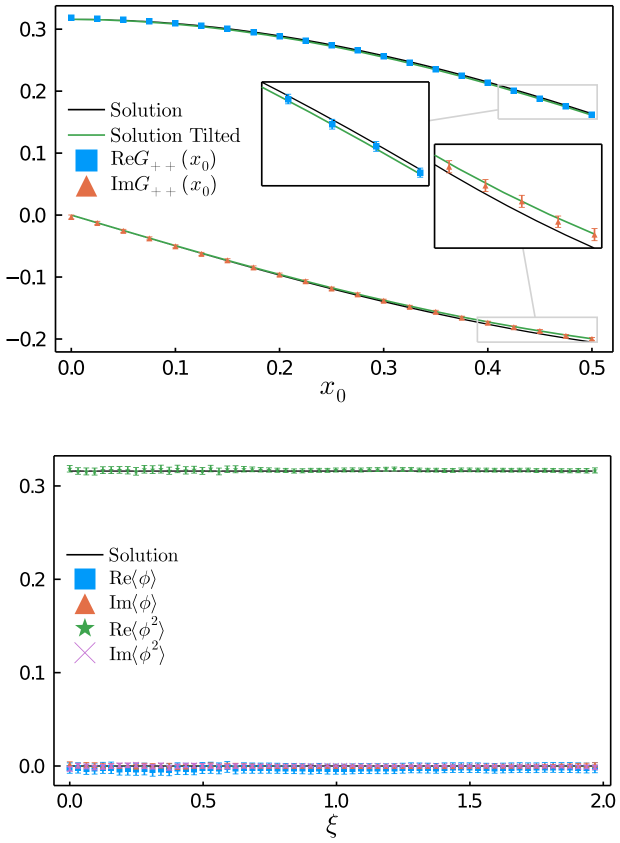

Figure 4.4: Simulation results for the anharmonic oscillator based on the semi-implicit Euler-Maruyama scheme on a 0.01β tilted contour: (top) unequal-time correlation function $G_{++}(x_0) = \langle\phi(0)\phi(x_0)\rangle - \langle\phi(0)\rangle\langle\phi(x_0)\rangle$ plotted for real-time values along the forward branch of the tilted Schwinger-Keldysh contour together with the true solution on the real-time axis (black solid) and along the tilted contour (green solid). Close to $x_0 = 0.5$ our simulation can already distinguish between the two. (bottom) The field expectation value $\langle\phi\rangle$ and the equal-time correlation function $\langle\phi^2\rangle$ evaluated on both branches. Agreement with the true solution is observed within errors within uncertainty (errors appear larger here only since the y-axis scale is reduced compared to the top panel).

section 4.2.2. We will return to the evolution equations and show that the formulation of the general EM scheme harbors additional terms, which play a role in the regularization of the path integral itself.

Let us focus on the simple but relevant case of the free theory here with $V(\phi) = \frac{1}{2}m^2\phi^2$. The update step of the general EM scheme reads

$$\phi_j^{\lambda+1} = \phi_j^\lambda + i\epsilon_j \left[\theta \frac{\partial S^{\lambda+1}}{\partial \phi_j} + (1-\theta)\frac{\partial S^\lambda}{\partial \phi_j}\right] + \sqrt{\epsilon_j}\eta_j^\lambda, \tag{4.25}$$

where $\epsilon_j = \frac{\Delta\tau_L}{|\omega_j|}$. To simplify the derivation below, we assume without loss of generality that the step size $\epsilon_i = \epsilon$ is constant along the contour. In the free theory we may in addition write the action in a simple matrix form as $\frac{\partial S^\lambda}{\partial \phi_j} = M\phi_j^\lambda$. Substituting this into eq. (4.25) yields

$$(I - i\epsilon\theta M)\,\phi^{\lambda+1} = \left\{(I + i\epsilon(1-\theta)M)\phi^\lambda + \sqrt{\epsilon}\eta^\lambda\right\}. \tag{4.26}$$

The explicit entries in M_{ij} are obtained via eq. (4.19) as

$$M_{jk} = \begin{cases} \frac{1}{a_{j-1}} + \frac{1}{a_j} - \frac{1}{2}\left[a_{j-1} + a_j\right] m^2, & j = k \\ -\frac{1}{a_j}, & j = k-1 \\ -\frac{1}{a_{j-1}}, & j = k+1. \end{cases} \tag{4.27}$$

In order to proceed, we bring the implicit part of the update over to the RHS and assume that ϵ is sufficiently small to expand the inverse matrix. The relevant quantitative criterion here is that the magnitude of the eigenvalues of $\epsilon\theta M$ are smaller than 1, i.e., the max $[|\lambda_1|, |\lambda_2|, ...] < 1$. In turn, we obtain

$$\phi^{\lambda+1} = (I - i\epsilon\theta M)^{-1}\left\{(I + i\epsilon(1-\theta)M)\phi^\lambda + \sqrt{\epsilon}\eta^\lambda\right\} \tag{4.28}$$

$$= \sum_{k=0}^{\infty} (i\epsilon\theta M)^k \left\{(I + i\epsilon(1-\theta)M)\phi^\lambda + \sqrt{\epsilon}\eta^\lambda\right\}. \tag{4.29}$$

Let us truncate the expansion at second order in ϵ and focus on the contributions to the drift term

$$\phi^{\lambda+1} = \left\{\left(1 + i\epsilon M - \epsilon^2\theta M^2\right)\phi^\lambda + \sqrt{\epsilon}\eta^\lambda\right\} + O(\epsilon^{3/2}). \tag{4.30}$$

The correction to the drift term of second order in ϵ may be absorbed into an effective action for the general EM scheme

$$S_\theta = \frac{1}{2}\phi\Big(M + i\epsilon\theta M^2\Big)\phi = S_{\text{explicit}} + \frac{i\epsilon}{2}\theta\sum_j S_j^2. \tag{4.31}$$

The expression for S_θ tells us that the difference between the EM scheme with finite θ and the fully explicit one lies in the presence of one additional term. It is proportional to the complex unit i and both depend on the time step and the implicitness parameter. Similar to the regulator term from a tilted contour in eq. (4.24) it is positive and thus leads to a damping of the oscillations of the path integral. eq. (4.31) thus constitutes a new means of regularization unavailable to explicit solvers.

The above argument is further supported by numerical tests, which show that the regularization becomes weaker as the Langevin time step is reduced. Intuitively it also agrees with the behavior of the numerical solvers we discussed in the context of large excursions in section 4.2.2. The term proportional to ϵ^2 in eq. (4.30) features a minus sign, which leads to the stable undershooting of the true solution shown in fig. 4.1. In turn, it is this correction that will prevent the Langevin dynamics from diverging in the late Langevin time limit, realizing the role of a regularizer in the underlying path integral.

4.3.3 Finite Langevin time step errors

We have argued in the preceding section that implicit solvers provide a novel intrinsic regularization of the underlying complex path integral. This regularization depends on the implicitness parameter but more importantly depends on the finite Langevin time step $\Delta\tau_\mathrm{L}$. It tells us that for finite step size our system remains well defined but that when moving towards continuous Langevin time, the dynamics will become more difficult to tame, as we concurrently remove our regularization. This conundrum can be avoided if it is possible to analytically correct for the finite Langevin step size corrections in our observables. Then we may choose a small but not too small value of $\Delta\tau_\mathrm{L}$ (depending on the parameters of the system) and carry out the simulation in a well-defined manner, accounting for the difference to the continuous Langevin solution a posteriori. In this section, we set out to derive such correction terms.

Our strategy is as follows: As a first step, we follow [107] and show that the effects of an implicit solver scheme at finite Langevin step size $\Delta\tau_{\rm L}$ can be cast in the language of an effective action for the Fokker-Planck equation. In order to exploit the well-established methods underlying the derivation of the Fokker-Planck equation from Langevin dynamics, we restrict ourselves to a scenario with purely real Feynman weights, i.e. the imaginary time one. We continue in a second step to guess how the effective action obtained in the real case generalizes to the complex case. This heuristic step is supported by numerical evidence, which confirms that it allows us to correct numerical artifacts introduced by finite $\Delta\tau_{\rm L}$ in practice.

Let us again focus on the simple but relevant free theory with $V(\phi) = \frac{1}{2}m^2\phi^2$. The update step of the general EM scheme, now for the Euclidean action reads

$$\phi_j^{\lambda+1} = \phi_j^{\lambda} - \epsilon_j \left[\theta\frac{\partial S^{\lambda+1}}{\partial\phi_j} + (1-\theta)\frac{\partial S^{\lambda}}{\partial\phi_j}\right] + \sqrt{\epsilon_j}\eta_j^{\lambda}, \tag{4.32}$$

where $\epsilon_j = \frac{\Delta\tau_{\rm L}}{|\omega_j|}$ and the negative sign in the drift term arises from the Wick rotation into imaginary time. For $\theta = 0$ these dynamics have been investigated in ref.[107].

To simplify the derivation below, we assume without loss of generality that the step size $\epsilon_i = \epsilon$ is constant along the contour. Remember that in the free theory can write the action in a simple matrix form as $\frac{\partial S^\lambda}{\partial\phi_i} = M\phi^\lambda$. Substituting all into eq. (4.32) yields

$$\left(I + \epsilon\theta M\right)\phi^{\lambda+1} = \left\{(I - \epsilon(1-\theta)M)\phi^\lambda + \sqrt{\epsilon}\eta^\lambda\right\}. \tag{4.33}$$

Let us bring the implicit part of the update over to the RHS and take ϵ is small enough to expand the first term in parentheses

$$\phi^{\lambda+1} = (I + \epsilon\theta M)^{-1} \left\{ (I - \epsilon(1-\theta)M)\phi^{\lambda} + \sqrt{\epsilon}\eta^{\lambda} \right\} \tag{4.34}$$

$$= \sum_{k=0}^{\infty} (-\epsilon\theta M)^{k} \left\{ (I - \epsilon(1-\theta)M)\phi^{\lambda} + \sqrt{\epsilon}\eta^{\lambda} \right\}. \tag{4.35}$$

The above expression, up to order $\epsilon^{5/2}$, can be written in index notation, using $M_{jk}\phi_k^{\lambda} = \frac{\partial S^{\lambda}}{\partial \phi_j} = S_j^{\lambda}$ as

$$\begin{aligned} \phi_j^{\lambda+1} &= \phi_j^{\lambda} - \epsilon S_j^{\lambda} + \epsilon^2 \theta M_{jk} S_k^{\lambda} + \left(\sqrt{\epsilon}\delta_{jk} - \epsilon^{3/2}\theta M_{jk} \right) \eta_k^{\lambda} + O(\epsilon^{5/2}) \\ &= \phi_j^{\lambda} - f_j^{\lambda}[\phi]. \end{aligned} \tag{4.36}$$

We will now derive the corresponding Fokker-Plank equation and the effective action based on the above update prescription. The standard approach (see e.g. [119]) is to rewrite the probability distribution for ϕ, denoted as $\mathcal{P}[\phi]$ at discrete Langevin time step $\lambda + 1$ in terms of its values at step λ using a delta-distribution. The argument of the delta distribution contains the Langevin update step from λ to $\lambda + 1$ and is averaged over the ensemble

$$\mathcal{P}^{\lambda+1}[\phi] = \int [d\phi'] \left\langle \prod_j \delta\left(\phi_j - \phi'_j + f_j[\phi'] \right) \right\rangle \mathcal{P}^{\lambda}[\phi']. \tag{4.37}$$

After expanding the delta function in powers of f_j and integrating over ϕ' one arrives at the Kramers-Moyal expansion for the discretized stochastic process,

$$\mathcal{P}^{\lambda+1}[\phi] = \mathcal{P}^{\lambda}[\phi] + \sum_{n=1}^{\infty} \frac{1}{n!} \nabla_{j_1} ... \nabla_{j_n} \left(\langle f_{j_1} ... f_{j_n} \rangle \mathcal{P}^{\lambda}[\phi] \right). \tag{4.38}$$

A Fokker-Planck equation may be obtained by considering terms up to the order ϵ^2, which are encoded in the correlation functions of the update term f. To make these explicit we use the following properties of the noise $\langle \eta_j \rangle = 0$, $\langle \eta_j \eta_k \rangle = 2\delta_{jk}$, $\langle \eta_j \eta_k \eta_l \rangle = 0$ and $\langle \eta_j \eta_k \eta_l \eta_m \rangle =$

$4\left(\delta_{jk}\delta_{lm}+\delta_{jl}\delta_{km}+\delta_{jm}\delta_{kl}\right)$, which leads to the following four expressions:

$$\begin{aligned}\langle f_j\rangle =&\epsilon S_j-\epsilon^2\theta M_{jk}S_k+O(\epsilon^3),\\ \langle f_jf_k\rangle =&\epsilon^2S_jS_k+2\epsilon\delta_{jk}-2\epsilon^2\theta(M_{kl}\delta_{jl}+M_{jl}\delta_{kl})+O(\epsilon^{5/2}),\\ \langle f_jf_kf_l\rangle =&2\epsilon^2\left(S_j\delta_{kl}+S_k\delta_{jl}+S_l\delta_{jk}\right),\\ \langle f_jf_kf_lf_m\rangle =&4\epsilon^2\left(\delta_{jk}\delta_{lm}+\delta_{jl}\delta_{km}+\delta_{jm}\delta_{kl}\right).\end{aligned} \tag{4.39}$$

To the lowest order in ϵ we obtain the following Fokker-Planck equation

$$\frac{\partial}{\partial\tau_{\mathrm{L}}}\mathcal{P}=\nabla_j\left[\left(S_j+\nabla_j\right)\mathcal{P}\right]+O(\epsilon^{3/2}), \tag{4.40}$$

which by a change of variable, $\mathcal{P}=e^{-S/2}\Psi$, can be shown to converge to the correct equilibrium distribution in the case of a real-valued action. Now the Kramer-Moyal expansion up to corrections of order $O(\epsilon^3)$ on the other hand contributes additional terms to the Langevin time evolution of the probability distribution

$$\begin{aligned}\partial_t\mathcal{P}=\nabla_j\left(S_j+\nabla_j\right)\mathcal{P}+\epsilon\Bigg\{&-\theta M_{jk}\nabla_j(S_k\mathcal{P})+\frac{1}{2}\nabla_j\nabla_k(S_jS_k\mathcal{P})\\ -\theta(M_{kj}+M_{jk})\nabla_j\nabla_k\mathcal{P}+\nabla_j\nabla^2(S_j\mathcal{P})&+\frac{1}{2}\nabla^2\nabla^2\mathcal{P}\Bigg\}+O(\epsilon^{5/2}).\end{aligned} \tag{4.41}$$

Since it is the equilibrium distribution, which is of main interest to us, let us set the LHS to zero. To be more concise we will rewrite $M_{jk}=-\nabla_jS_k$ and use eq. (4.40) to make the replacement $\nabla_j\mathcal{P}=-S_j\mathcal{P}+O(\epsilon)$ within the curly brackets of eq. (4.41), consistent with the order of the approximation.

The new terms at this order then give

$$\nabla_j \left\{ \theta M_{jk} S_k + \frac{1}{2}(S_{jk} S_k + S_j S_{kk} - S_j S_k^2) - \theta(M_{kj} + M_{jk}) S_k \right. \tag{4.42}$$

$$\left. + \nabla_k (S_{jk} - S_j S_k) + \frac{1}{2} \nabla_j (-S_{kk} + S_k^2) \right\} \mathcal{P} \tag{4.43}$$

$$= \nabla_j \left\{ -\theta M_{kj} S_k + \frac{1}{2}(S_{kj} S_k + S_j S_{kk} - S_j S_k^2) \right. \tag{4.44}$$

$$+ (S_{jkk} - S_{jk} S_k - S_{jk} S_k - S_j S_{kk} + S_j S_k^2) \tag{4.45}$$

$$\left. + \frac{1}{2}(-S_{kkj} + S_{kk} S_j + 2 S_{kj} S_k - S_k^2 S_j) \right\} \mathcal{P} \tag{4.46}$$

$$= \nabla_j \left\{ \theta S_{jk} S_k - \frac{1}{2} S_{jk} S_k + \frac{1}{2} S_{jkk} \right\} \mathcal{P} \tag{4.47}$$

$$= \nabla_i \left\{ -\left(\frac{1}{2} - \theta \right) S_{jk} S_k + \frac{1}{2} S_{jkk} \right\} \mathcal{P} \tag{4.48}$$

$$= \nabla_j \left\{ \frac{1}{2} \nabla_j S_{kk} - \frac{1}{2} \left(\frac{1}{2} - \theta \right) \nabla_j S_k^2 \right\} \mathcal{P}. \tag{4.49}$$

Reexpressed as a modified action we arrive at the intermediate result

$$0 = \nabla_j \left[(\bar{S}_j + \nabla_j) \mathcal{P} \right], \quad \bar{S} = S + \frac{\epsilon}{2} \sum_k \left\{ S_{kk} - \left(\frac{1}{2} - \theta \right) S_k^2 \right\}. \tag{4.50}$$

In the above expression, we see that the θ parameter governs the size and sign of a real-valued addition to the action. For $\theta > \frac{1}{2}$ the contribution is positive and for $\theta < \frac{1}{2}$ it is negative, distinguishing clearly between the implicit regime and the explicit regime. Note that similar to our discussion of the large excursions, the semi-implicit case of $\theta = \frac{1}{2}$ is special, as it cancels all corrections associated with the S_k^2 term.

The derivation outlined above cannot be translated one-to-one into the complex case. We would need to instead express the complex Langevin evolution in terms of the real-valued joint probability distribution of the real- and imaginary part of the complexified fields. Doing so, we were unable to derive a similarly closed-form as eq. (4.49). The structure of the correction terms obtained in the real case however invites a heuristic generalization to the complex domain using the replacement $-S \to iS$, which leads to

$$0 = \nabla_j \left[\left(-i\bar{S}_j + \nabla_j\right) \mathcal{P} \right], \quad \bar{S} = S + \frac{\epsilon}{2} \sum_k \left\{ S_{kk} + i \left(\frac{1}{2} - \theta \right) S_k^2 \right\}. \tag{4.51}$$

Let us find out whether this expression describes the dynamics of the complex Langevin simulation in practice. Similar to the discussion for the real-valued case in [107], we can attempt to counteract the effects introduced by a finite Langevin step size ϵ in the action by a redefinition of the fields. In our case the leading order change in fields according to eq. (4.51) amounts to

$$\tilde{\phi}_j = \phi_j - \frac{i\epsilon}{2} \left(\frac{1}{2} - \theta \right) S_j, \tag{4.52}$$

where S_j is nothing but the drift term. Note that S_{jj} in the free theory is just a constant, so that acting with one more derivative on it makes that term vanish. Thus the redefinition of the fields changes the action to first order in ϵ such that it cancels the S_j^2 contribution in eq. (4.51)

$$\bar{S}[\tilde{\phi}] \sim \bar{S}[\phi] - \frac{i\epsilon}{2} \left(\frac{1}{2} - \theta \right) \sum_k S_k^2. \tag{4.53}$$

I.e. if eq. (4.51) is the correct generalization then observables evaluated in terms of $\tilde{\phi}_i$ instead of ϕ should show reduced deviations from the continuous Langevin time result. For the equal time two-point function we e.g. obtain the following corrected expression

$$\left\langle \tilde{\phi}_j^2 \right\rangle = \left\langle \phi_j^2 \right\rangle \underbrace{- \epsilon \left(\frac{1}{2} - \theta \right) \left\langle \phi_j (iS_j) \right\rangle}_{\Sigma} - \frac{\epsilon^2}{4} \left(\frac{1}{2} - \theta \right)^2 \left\langle S_j^2 \right\rangle. \tag{4.54}$$

For later reference, we denote the correction term linear in ϵ as Σ. In order to assess the validity of the above arguments, let us simulate the harmonic oscillator on the real-time contour c) of fig. 4.3, i.e. on a contour without tilt in the real-time branch, up to a maximum extent of $x_0^{\max} = 0.5$. Deploying an equidistant real-time spacing on the forward and backward branch of the contour and a Langevin time step of $\Delta\tau_\mathrm{L} = 10^{-2}$, we compute the difference between the numerical result and the analytic solution $\Delta\phi^2 = \langle\phi^2\rangle_\mathrm{CL} - \langle\phi^2\rangle_\mathrm{QM}$ as the colored boxes in the top panel of fig. 4.5 vs. the contour parameter ξ. We find characteristic features in both the real- and imaginary part of this quantity. The artifacts introduced by the implicit solver at finite Langevin time show opposite sign in the imaginary part and same sign in the real-part comparing the forward and backward branch. On the Euclidean time interval, only the real-part receives significant modifications.

Interestingly, the correction term $(\phi - \tilde{\phi})^2$ taken from eq. (4.54), when plotted as the colored triangles in fig. 4.3 already follows the behavior of the deviations in a qualitative fashion. At the same time, we observe that it appears to consistently over-predict those artifacts. Limiting ourselves to the corrections linear in Langevin step size ϵ, shown in gray, we find that they capture the artifacts even more accurately. This difference between the linear and quadratic terms in ϵ to us hints at the need to include higher-order corrections in the expansion of eq. (4.51) to arrive at a reliable correction term beyond leading order. To conclude, we find that the linear correction terms derived from a heuristic generalization of the robust result in eq. (4.50) capture the discrete dynamics of our complex Langevin simulation in a qualitative fashion, lending numerical support to eq. (4.51).

The correction terms obtained in eq. (4.51) contain the first and second derivative of the action S_k and S_{kk}. We can gain additional insight into what role they play from the following considerations based on the translation invariance of the integrals over the fields. Let us start by stating the fact that

$$\begin{aligned}\langle(\phi_j + \phi_0)^n\rangle &= \int D\phi(\phi_j + \phi_0)^n \exp(iS(\phi_j)) \\ &= \int D\phi(\phi_j)^n \exp(iS(\phi_j - \phi_0)) \qquad (4.55)\end{aligned}$$

where we have shifted the integral in the second line, such that the constant ϕ_0 has been moved into the exponent. While the distribution obtained

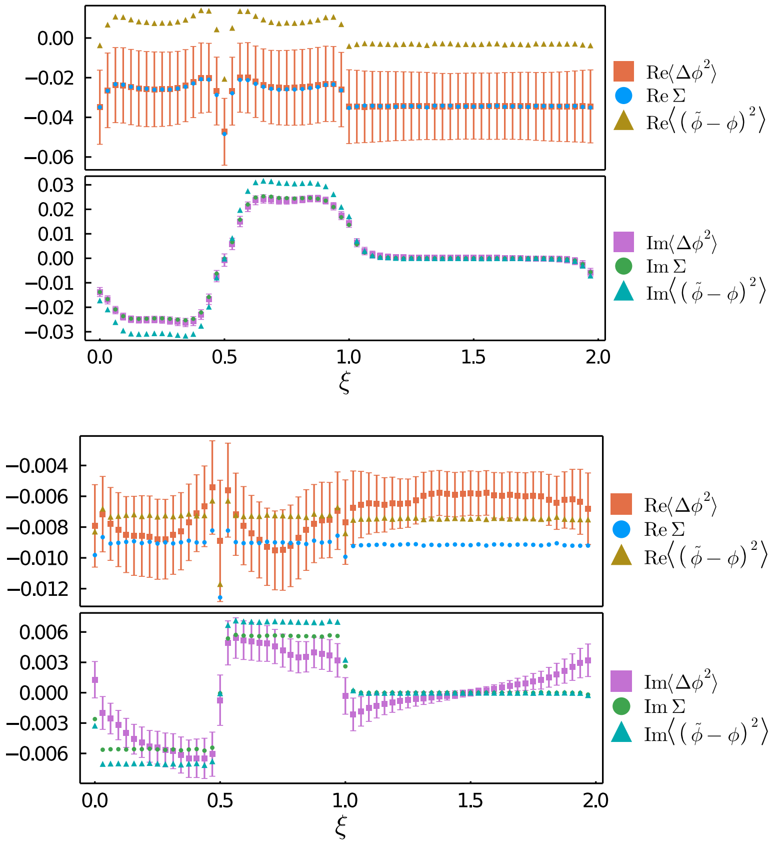

Figure 4.5: Comparison of the finite Langevin step-size artifacts in the equal-time correlator $\Delta\phi^2 = \langle\phi^2\rangle_{\mathrm{CL}} - \langle\phi^2\rangle_{\mathrm{QM}}$ to the estimates of that error based on eq. (4.54). The filled triangles denote the full estimate including the terms proportional to ϵ^2, which provide the correct qualitative behavior but systematically overestimate $\Delta\phi^2$. On the other hand the leading order expression Σ, proportional to ϵ, shown as filled circles captures the error even quantitatively within the statistical uncertainty. The position along the contour is parametrized by ξ, which for $\xi < 1$ points to real-time values and for $1 < \xi < 2$ refers to imaginary times. (top) Estimation of the errors in the free theory (harmonic oscillator) using $\Delta\tau_{\mathrm{L}} = 10^{-2}$, as well as (bottom) for the anharmonic oscillator at $\lambda = 24$ with $\Delta\tau_{\mathrm{L}} = 10^{-3}$. In both cases, the data is based on 1000 separate trajectories each of total length $\tau_{\mathrm{L}} = 200$.

from CL is different, the expectation values of the shifted system will still be the same. We can now exploit the presence of ϕ_0 in the integrand and the weight to derive relations between different n-point correlation functions. Using the first and second derivative with respect to ϕ_0 we have

$$\nabla_{\phi_0}\langle(\phi_j+\phi_0)^n\rangle|_{\phi_0=0} = n\langle(\phi_j)^{n-1}\rangle = \langle(\phi_j)^n(-iS_j)\rangle \tag{4.56}$$

$$\nabla^2_{\phi_0}\langle(\phi_j+\phi_0)^n\rangle|_{\phi_0=0} = n(n-1)\langle(\phi_j)^{n-2}\rangle = \langle(\phi_j)^n(-iS_{jj}-(S_i)^2)\rangle. \tag{4.57}$$

Using the first derivative with $n = 1$ and the second derivative with $n = 0$ we obtain the following two expressions respectively

$$\langle 1\rangle = \langle\phi_j(-iS_j)\rangle, \qquad 0 = \langle -iS_{jj}-(S_i)^2\rangle. \tag{4.58}$$

For the free case, S_{jj} is a constant. The first term is particularly interesting as it tells us that for continuous Langevin time, the term $\langle\phi_j(-iS_j)\rangle$ should be constant and correspond to the normalization of the system. In the presence of discrete Langevin time steps, we found that it is a term proportional to $\langle\phi_j(-iS_j)\rangle$, which describes the corrections and which are not constant as shown in fig. 4.5. We thus interpret the corrections in eq. (4.54) as counteracting in part the deviations from the correct normalization of the continuum theory.

When we derived the modifications to the Fokker-Planck equation in the free theory, we were able to express them in terms of the quantity S_j. We may ask whether this expression also holds in the interacting theory. To this end, we carry out simulations of the anharmonic oscillator at short real-times on the untilted Schwinger-Keldysh contour up to $x_0^{\max} = 0.5$, where complex Langevin is known to converge to the correct solution (for more details see section 4.4.1). The same solver as for the harmonic oscillator is deployed and we choose a Langevin step-size of $\Delta\tau_{\mathrm{L}} = 10^{-3}$. In the lower panel of fig. 4.5 we plot the resulting deviations from the analytic solution (filled boxes), compared to the naive application of eq. (4.54) to the interacting theory (filled triangles). Again we observe that the expression up to second order in ϵ slightly overestimates the artifacts but that restricting us to the linear term in ϵ allows us to capture the discretization errors within uncertainties.

With an expression at hand that allows us to correct the finite Langevin step size corrections for small values of ϵ, we are able to exploit the regularization properties of the implicit solvers in practice. What remains for each explicit system is to choose a step-size and θ parameter, keeping in mind the trade-off between regularization artifacts and numerical cost. Having too small of a step size in the implicit scheme will reduce the effect of the regulator, which in turn will lead to the appearance of large excursions. Even though these excursions do not represent a problem in principle (the approach is inherently stable) they may lead to high computational cost if a fixed accuracy goal is prescribed. Using an intermediate step sizes $\sim 10^{-3}$ appears to give the best trade-off for the interacting systems considered in this study. The finite step size provides an effective regulator to the path integral and the finite step-size artifacts can be remedied by the correct procedure discussed above.

We emphasize that the implicit EM scheme provides enough of an intrinsic regularization that we may forego a tilting of the Schwinger-Keldysh contour all together. I.e. we gain access to the fields very close to the actual forward and backward real-time branch of the canonical Schwinger-Keldysh contour, which is particularly useful in the study of non-equilibrium field theory, in which the forward and backward correlators are not related via the KMS relation.

Armed with the insight laid out in the previous sections we are now ready to carry out stable simulations of the quantum anharmonic oscillator at short real times.

4.4 Stable CL simulations at short real-times

In this section, we present numerical results of simulating real-time complex Langevin on the canonical Schwinger-Keldysh contour with short time extent of $x_0^{\max} = 0.5$ using the implicit EM scheme. We will start out with a system in thermal equilibrium which is formulated on contour c) of fig. 4.3. As a second example, we take a look at a system with Gaussian initial conditions, where only the forward and backward real-time branch of the contour remains and a Euclidean branch is absent.

4.4.1 Dynamics in thermal equilibrium

Our simulation uses the same parameters as adopted in the classic work of ref.[106], i.e. $\lambda = 24$ and $m = 1$. To discretize the real-time contour c) in fig. 4.3 for a temperature $T = 1/\beta = 1$ and real-time extent of $x_0^{\max} = 0.5$, we use 16 points for the forward and backward branch each and an additional 32 points along the negative imaginary time axis. This choice of an equidistant $|a| = 0.031$ guarantees that the finite time spacing artifacts to the correlation functions remain at the permille level.

To regularize the path integral we deploy the general EM scheme with its implicitness parameter set to $\theta = 0.6$. We use the adaptive step size prescription of the Julia stochastic processes library with a maximum step size of $\Delta\tau_{\rm L} = 0.005$. This choice provides an efficient enough regularization to avoid costly excursions, while at the same time the deviation from the continuous Langevin result remains smaller than our statistical uncertainty. For the computation of the correlation functions of interest, field configurations are collected based on 500 different trajectories. We read out observables on each of them in intervals of $\delta\tau_{\rm L} = 0.1$ up to a total Langevin time of $\tau_{\rm L} = 100$.

In the top panel of fig. 4.6 we show the unequal time correlation function $G = \langle\phi(0)\phi(\xi)\rangle - \langle\phi(0)\rangle\langle\phi(\xi)\rangle$, which for $\xi < 1/2$ amounts to $G_{++}(x_0 = \xi) = \langle\phi(0)\phi(\xi)\rangle - \langle\phi(0)\rangle\langle\phi(\xi)\rangle$ and for $\xi > 1/2$ to $G_{+-}(x_0 = 1 - \xi) = \langle\phi(0)\phi(\xi)\rangle - \langle\phi(0)\rangle\langle\phi(\xi)\rangle$. The corresponding continuum Langevin time solution is plotted as solid black curve, which is obtained from a matrix mechanics computation based on the truncated Hilbert space spanned by the lowest 32 energy eigenstates of the harmonic oscillator. The magnified insets confirm that our solution accurately reproduces the continuum solution on the forward and backward branch of the canonical Schwinger-Keldysch contour up to these early real-times.

In the lower panel of fig. 4.6 we plot the Euclidean correlator $G_E(x_0 =$

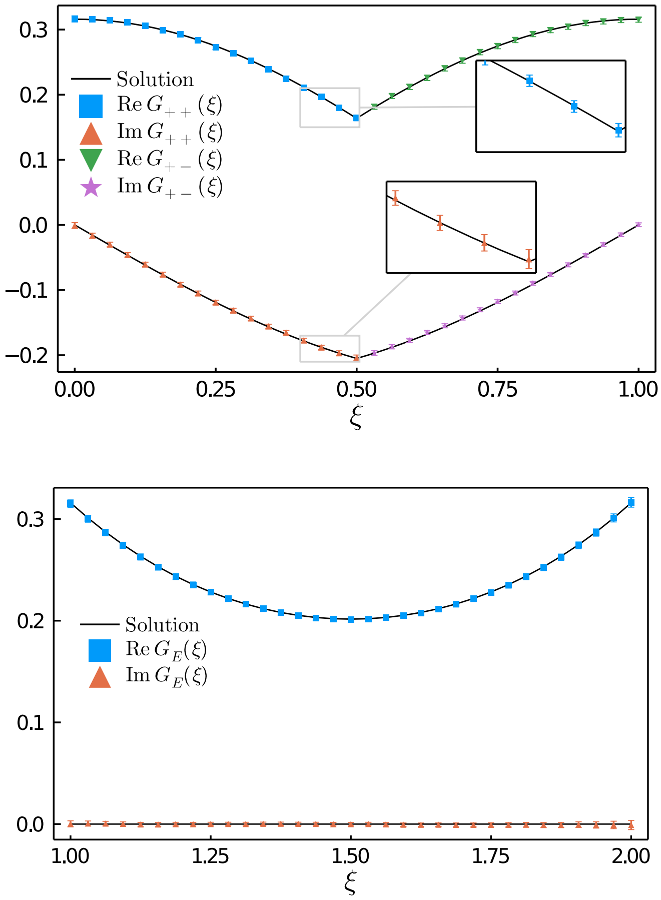

Figure 4.6: (top) Two unequal time correlation functions along the real-time branches of the canonical Schwinger-Keldysh contour. For $\xi < 1/2$ the datapoints represent $G_{++}(x_0 = \xi) = \langle\phi(0)\phi(\xi)\rangle - \langle\phi(0)\rangle\langle\phi(\xi)\rangle$ and for $\xi > 1/2$ we have $G_{+-}(x_0 = 1-\xi) = \langle\phi(0)\phi(\xi)\rangle - \langle\phi(0)\rangle\langle\phi(\xi)\rangle$. Note the excellent accuracy in reproducing the continuous Langevin time result given as black solid line. (bottom) The Euclidean correlator $G_E(x_0 = -i(\xi - 1)) = \langle\phi(0)\phi(\xi)\rangle$ evaluated on the imaginary time branch of the canonical Schwinger-Keldysh contour together with the continuous Langevin time solution (black solid)

$-i(\xi - 1)) = \langle \phi(0)\phi(\xi) \rangle$. It features a vanishing imaginary part and a real part which correctly exhibits a symmetry around $x_0 = -i\beta/2$, corresponding to the contour parameter $\xi = 1.5$ here. Again the continuum solution from matrix mechanics is given as solid black line and we find excellent agreement.

Let us take a look at another set of observables, which have been discussed in the literature. The top panel of fig. 4.7 we show the field expectation value $\langle \phi \rangle$ (filled box and triangle) and the equal time correlation function $\langle \phi^2 \rangle$ (filled star and cross) along the whole extent of the simulation contour parametrized by ξ. Within the statistical uncertainties of our simulation, we find full agreement with the continuous-time Langevin solution. Should one be interested in higher precision results, one will eventually find minute differences from the continuum result, similar to those shown in the lower panel of fig. 4.5.

We foresee that the new insight obtained in eq. (4.54) will help us in future studies to distinguish artifacts arising from finite Langevin-time discretization from those connected to a convergence to the wrong result. One concrete example is the equal-time correlation function, whose deviation from a constant value has previously been taken as an indication for the arrival at an unphysical solution. If the errors from a finite $\Delta\tau_\mathrm{L}$ are accounted for, can the remaining deviation be unambiguously associated with wrong convergence.

The last result in this section we present in the lower panel of fig. 4.7. In preparation for the simulation of genuine field theory in higher dimensions and for the simulation out-of-equilibrium in the next section, we compute the forward $G_{+-}(x_0) = G^{>}(x_0)$ and backward correlator $G_{-+}(x_0) = G^{<}(x_0)$ together with the analytic solution as solid lines. In thermal equilibrium, the information of these two quantities is redundant due to the KMS relation and because we are in a quantum mechanical setting their interrelation is actually trivial. Their real parts agree, while their imaginary parts are the negative of each other. They nevertheless take on a central role in field theory, as their difference $\rho = G^{>} - G^{<}$ encodes the spectral function of the system, which harbors a wealth of phenomeno-

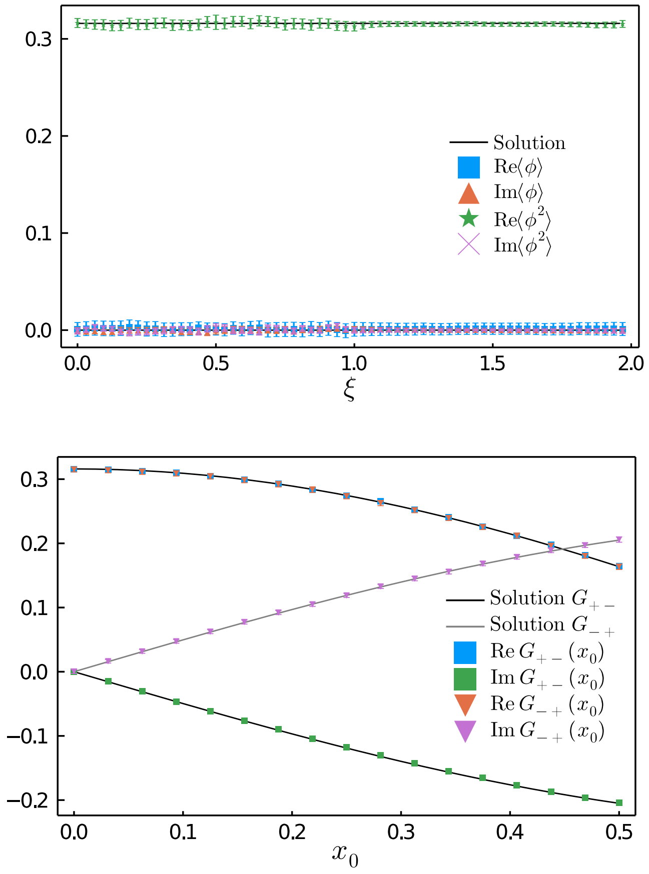

Figure 4.7: (top) The field expectation value $\langle\phi\rangle$ (box and triangle), as well as the equal-time correlation function $\langle\phi^2\rangle$ (filled star and cross) evaluated in thermal equilibrium along the whole simulation contour, parametrized by ξ. The continuous Langevin time solution from matrix mechanics is given as solid line. (bottom) The phenomenologically relevant forward $G_{+-} = G^>$ and backward $G_{-+} = G^<$ correlation functions. Note that only in the thermal setting the information they contain is redundant with that of the G_{++} correlator.

logically relevant pieces of information. Thus an accurate reproduction of these correlation functions between fields on different branches of the contour is an important benchmark for the complex Langevin procedure. In addition, only when we simulate close enough to the real-time axis, do we have access to these quantities in an undistorted fashion and in turn compute the spectral function of the system.

4.4.2 Non-equilibrium dynamics

Having confirmed the efficacy of the implicit solver for the simulation of the early real-time dynamics of the anharmonic oscillator in thermal equilibrium the next step is to move to an out-of-equilibrium setting.

We follow ref.[106] and choose a Gaussian initial density matrix. Its form allows us to incorporate the information about initial conditions into a modification of the action of the system on the first and last point on the Schwinger-Keldysh contour. Note that here the contour consists only of a forward and backward real-time branch, which are not connected via periodic boundary conditions. The most general form of the Gaussian density matrix [34] leads to the following expression for the system action

$$\begin{aligned} S_{\mathrm{G}}[\phi_+,\phi_-] =& S[\phi_+] - S[\phi_-] - iS_0(\phi_+[t=0],\phi_-[t=0]) \quad \text{with} \\ S_0[\phi_+,\phi_-] =& i\dot{\phi}_0(\phi_+ - \phi_-) - \frac{\sigma^2+1}{8\zeta^2}\left((\phi_+ - \phi_0)^2 + (\phi_- - \phi_0)^2\right) \\ &+ \frac{i\eta}{2\zeta}\left((\phi_+ - \phi_0)^2 - (\phi_- - \phi_0)^2\right) \\ &+ \frac{\sigma^2-1}{4\zeta^2}(\phi_+ - \phi_0)(\phi_- - \phi_0)\,. \end{aligned} \tag{4.59}$$

The five independent parameters, which specify the Gaussian initial state, represent the initial values of the field expectation value, the two-point correlation function and their derivatives

$$\begin{aligned} \phi_0 =&\langle \phi(t=0)\rangle, \quad \dot{\phi}_0 = \langle \dot{\phi}(t=0)\rangle, \\ \zeta^2 =&\langle \phi(t=0)\phi(t=0)\rangle_c, \\ \eta\zeta =&\frac{1}{2}\langle \dot{\phi}(t=0)\phi(t=0) + \phi(t=0)\dot{\phi}(t=0)\rangle_c, \\ \eta^2 + \frac{\sigma^2}{4\zeta^2} =&\langle \dot{\phi}(t=0)\dot{\phi}(t=0)\rangle_c. \end{aligned} \tag{4.60}$$

The subscript c refers to the connected correlator, in which the expectation value of the field and its derivatives are subtracted.

The drift term of the discretized complex Langevin dynamics (eq. (4.19)) is affected by the Gaussian initial density matrix only at the boundaries of the contour. Consistent with the trapezoidal rule underlying the discretization of the action integral, we choose forward derivatives at the starting point and backward derivatives when considering the endpoint of the contour. No changes are needed at intermediate contour steps. Similar to [106] we set $\eta = 0$ and $\dot{\phi}_0 = 0$, which leads to the following two explicit terms to implement at the boundary

$$\begin{aligned} \frac{\delta S_{\mathrm{G}}}{\delta \phi_0} =&\frac{1}{|a_0|}\Bigg\{-\frac{\phi_1-\phi_0}{a_0} - \frac{1}{2}a_0\frac{\partial V(\phi_0)}{\partial \phi_0} \\ &+ \frac{1}{2}i\left[\frac{\sigma^2+1}{4\zeta^2}(\phi_0-\bar{\phi}) - \frac{\sigma^2-1}{4\zeta^2}(\phi_{N_C}-\bar{\phi})\right]\Bigg\}, \\ \frac{\delta S_{\mathrm{G}}}{\delta \phi_{N_C}} =&\frac{1}{|a_{N_C-1}|}\Bigg\{\frac{\phi_{N_C}-\phi_{N_C-1}}{a_{N_C-1}} - \frac{1}{2}\frac{\partial V(\phi_{N_C})}{\partial \phi_{N_C}} \\ &+\frac{1}{2}i\left[\frac{\sigma^2+1}{4\zeta^2}(\phi_{N_C}-\bar{\phi}) - \frac{\sigma^2-1}{4\zeta^2}(\phi_0-\bar{\phi})\right]\Bigg\}. \end{aligned} \tag{4.61}$$

The Langevin equation remains in its standard form for the field degrees on the forward and backward branch respectively

$$\partial_{\tau_{\mathrm{L}}}\phi_{\pm}(x) = i\frac{\delta S_{\mathrm{G}}[\phi_+,\phi_-]}{\delta \phi_{\pm}(x)} + \eta(x,t), \tag{4.62}$$

with no changes to the noise term.

As in previous studies in the literature, we deploy here $m = 1$ and a relatively small coupling of $\lambda = 1$. This choice leaves us safely in the regime where CLE converges to the right solution. The field starts out at a finite expectation value $\phi_0 = \langle\phi(t = 0)\rangle = 1$ at rest $\dot{\phi}_0 = 0$. The spread in the values of the initial field, encoded in the correlation function is set symmetrically $\sigma = 1$ to a value of $\zeta = 1$. Mixing terms between field and derivatives vanish via $\eta = 0$. We distribute 32 points along each of the two real-time branches to cover the maximum time extent of $x_0^{\max} = 0.5$.

Similar to the thermal case we deploy the EM solver with $\theta = 0.6$ implicitness parameter using the Julia adaptive step size prescription with a maximum Langevin step size of $\Delta\tau_{\mathrm{L}} = 0.005$. Statistics are collected on 500 different trajectories of length $\tau_{\mathrm{L}} = 100$, reading out observables on intervals $\delta\tau_{\mathrm{L}} = 0.1$. Comparisons to matrix mechanics are also available in this scenario, however, the energy eigenfunctions of the harmonic oscillator are not well suited for truncating this particular Hilbert space. Instead, we discretize the Hamiltonian in the coordinate basis using 1024 points in the distance range $\langle x\rangle \in [-10, 10]$, the result of which will be shown as solid lines in the subsequent plots.

Our out-of-equilibrium simulation results are collected in fig. 4.8. Now with time translational invariance gone, we can follow the non-trivial behavior of the field expectation value $\langle\phi\rangle$, plotted as solid squares and triangles in the upper panel along the contour, parameterized by ξ. The initial conditions of $\phi_0 = 1$ as well as unit variance manifest themselves in the value $\langle\phi^2\rangle(\xi = 0) = 2$. Our results agree within statistical uncertainties with the analytic solution on the real-time axis.

The most interesting unequal-time correlation functions in the out-of-equilibrium scenario are the forward and backward quantities $G_{+-} = G^{>}$ and $G_{-+} = G^{<}$, which provide access to the spectral function of the system. Plotted in the lower panel of fig. 4.8, we find that here the statistical error remains larger than in the thermal case at the same collected statistics, indicating the presence of larger excursions in the Langevin dynamics as in the more strongly coupled thermal case. Compared to the

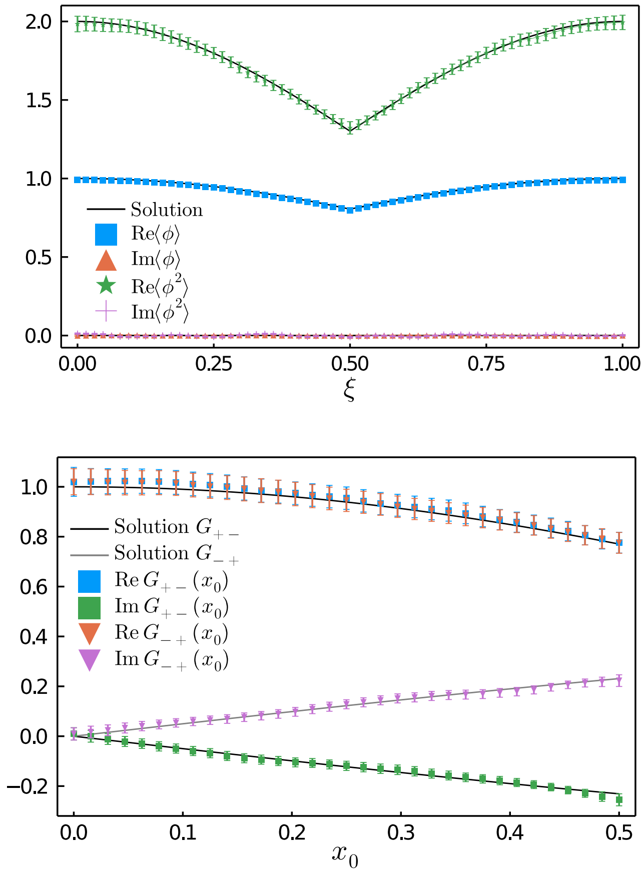

Figure 4.8: (top) The field expectation value $\langle\phi\rangle$ (box and triangle), as well as the equal-time correlation function $\langle\phi^2\rangle$ (filled star and cross) evaluated out-of equilibrium for a Gaussian density matrix, along the whole simulation contour, parametrized by ξ. The continuous Langevin time solution from matrix mechanics is given as solid line. (bottom) The phenomenologically relevant non-equilibrium forward $G_{+-} = G^{>}$ and backward $G_{-+} = G^{<}$ correlation functions.

continuous Langevin time solution from matrix mechanics, the numerical solution again shows excellent agreement.

4.5 Summary and Outlook

In this study, we have explored and showcased the potential of implicit solvers in real-time complex Langevin simulations. With the intention to disentangle the issue of numerical artifacts, such as runaway trajectories, from foundational issues, such as the convergence to wrong results, our focus in this paper remained restricted solely to early real-times.

Two central benefits of the implicit solvers were laid out in detail. On the one hand, the implicit solvers can be shown to be unconditionally asymptotically stable, preventing the occurrence of runaway trajectories, as long as the underlying complex Langevin dynamics remain finite. Using the Langevin dynamics of the free theory as a simple but relevant example, we showed in section 4.2 that the difference between implicit and explicit methods lies in the accumulation of errors that either undershoot or overshoot the true trajectory. While the undershoot in the implicit case also leads to a reduction in accuracy of the solution, it manages to prevent the occurrence of runaways.

In section 4.3 we carried out a comparison of the update prescription for the explicit and implicit EM scheme, which revealed that the effect of the latter can be captured in one additional term in an effective action. That term takes the form of a regulator $+iR$ and depends on the implicitness parameter θ, as well as Langevin step size $\Delta\tau_{\mathrm{L}}$. Since $R > 0$, it indeed dampens the oscillations in the underlying path integral. We conclude that this additional term provides an intrinsic regularization of the path integral unavailable to the explicit solvers.

Subsequently, we analyzed the finite Langevin time discretization artifacts in terms of an effective action in the Fokker-Planck equation for the case of a purely real path integral. We then heuristically generalized the result to the complex case and provided numerical support that our educated guess indeed captures the numerical artifacts introduced due to finite

Langevin time steps in the free theory and even the strongly coupled interacting case. This correction formula allows us to exploit the inherent regularization properties of the implicit solvers in practice, as we may now simulate the system at a small but finite Langevin step size $\Delta\tau_{\rm L}$ in a well-defined manner and correct for the effect of the regulator a posteriori.

The first three sections have provided us with insight into the regularization properties of different numerical schemes, insight into the effects of finite real-time discretization and we have derived the form of finite Langevin time steps artifacts that allow us to compensate for the effect of the regulator. We thus proceeded in section 4.4 to carry out benchmark numerical simulations of the anharmonic oscillator in $(0+1)d$ on the canonical Schwinger-Keldysh contour without tilt and maximum real-time extent of $x_0^{\rm max} = 0.5$ (Source code for this simulation is written in Julia and is available at [120]). Both in the thermal case and in a scenario with Gaussian non-thermal initial conditions, we find excellent agreement between the complex Langevin simulation and the analytic solution from matrix mechanics. The fact that the implicit solver gives access to the backward path on the real-time axis allows us for the first time to compute the actual forward and backward correlators $G_{+-} = G^{>}$ and $G_{-+} = G^{<}$, whose difference encodes the phenomenologically relevant spectral function of the system.

We believe that the availability of implicit solvers and an improved understanding of discretization artifacts will help to improve the reliability of the complex Langevin approach and provides new momentum to attack the pressing open challenges associated with it. The stability and regularization properties of the implicit schemes offer benefits in other applications of complex Langevin beyond real-time simulations, such as the treatment of strongly interacting systems at finite chemical potential (for a recent review on CL and the QCD phase diagram see e.g. [98]).

Many different paths forward exist. One aspect we are following up on is the role of regularization in the path integral for the success of complex Langevin convergence. When we introduce a tilt in the Schwinger-

Keldysh contour it led us in eq. (4.24) to a regulator term that incorporates all terms of the action. We may instead ask how the system reacts to introducing a regulator on individual terms in the action

$$S \;=\; i\sum_j [\frac{(\phi_j-\phi_{j-1})^2}{a_j} - a_j\frac{\sigma}{2}\phi_j^2 - a_j\frac{\lambda}{24}\phi_j^4], \qquad (4.63)$$

by modifying in either the kinetic, mass or self-interaction term the lattice spacing from $a_j \to a_j - i\kappa$ with $\kappa > 0$. In the following we will thus work with the contour b) of fig. 4.3, where the forward branch is located on the real-time axis and only the backward branch tilts downwards to intersect with the imaginary time axis at β.

We have seen that for $x_0^{\max} = 0.5$ the complex Langevin approach in the strongly coupled thermal scenario with $\lambda = 24$ converges to the correct solution given by matrix mechanics. Extending the contour to later real-times, we encounter significant deviations already at $x_0^{\max} = 0.8$. A prominent characteristic of the incorrect solution is an artificial downward shift in the real-part of the unequal-time correlation function, as shown by the red data points in fig. 4.9. In addition, the curvature of the imaginary part of the correlator, given as open squares also deviates from the true solution beyond statistical uncertainty.

In the regime $0.75 < x_0^{\max} \lesssim 1$ we observe that this incorrect convergence can be overcome by a choice of regularization on the forward branch. Interestingly, when introducing an imaginary part in the a_j's associated with the interaction term (blue filled square and open circle) the simulation outcome remains unchanged. On the other hand, modifying the kinetic term with a small imaginary part $a_j - i \times 10^{-3}$ leads to a significant improvement as indicated by the brown-filled circle and open triangles, which agree with the analytic solution from matrix mechanics. On the other hand, such a regularization based strategy fails to achieve its purpose, once the real-time extent of the Schwinger-Keldysh contour goes beyond unity in units of the mass. Our goal in future work is to gain a systematic understanding of how the regularization achieves to recover the correct results, possibly by studying the associated Fokker-Planck

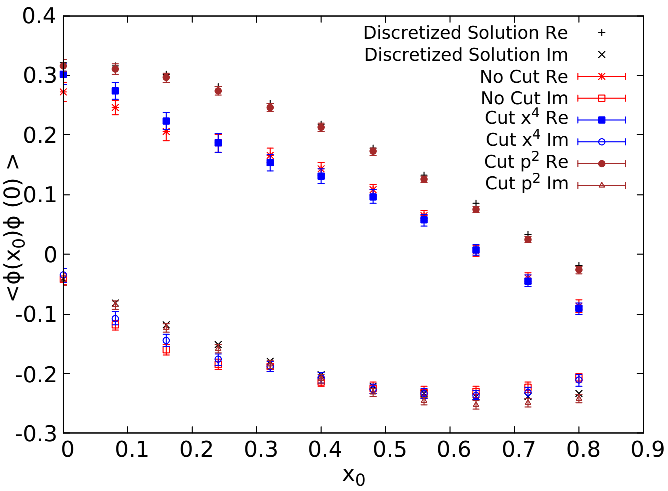

Figure 4.9: Comparison of the unequal-time correlation function G_{++} in thermal equilibrium from matrix mechanics (black crosses) with complex Langevin simulations carried out on a Schwinger-Keldysh contour of intermediate real-time extent $x_0^{\max} = 0.8$. Here the forward branch of the contour resides on the real-time axis and the backward contour tilts down to intersect with the imaginary axis at β (c.f. b) in fig. 4.3). The direct simulation based on the explicit EM scheme converges to an incorrect result given by the red data points. No improvement is observed for regularizing the ϕ^4 term (blue). The correct solution is recovered when regularizing the momentum term (brown).

equation in low-dimensional models.

Furthermore, the availability of implicit and in particular higher-order solvers benefits the systematic exploration of kernels for the Langevin dynamics (for a modern perspective on CL kernels see e.g. [121]). In the real-valued case, kernels can be used to improve the convergence properties of the stochastic quantization procedure. In complex Langevin, they have been studied with mixed success as means to remedy the convergence to wrong solutions. Robust numerical SDE solvers (c.f. the Runge-Kutta Milstein scheme of eq. (4.11)), which can accommodate non-trivial kernels with Langevin-time and field dependencies will allow us to explore a much broader class of kernels than before in future studies.

5 Kernels, Thimbles and the Correctness Criterion

This chapter is an extension of the appendix of the paper [31] and the main content of proceeding [40]. We start by deriving the correctness criterion for a kernel-controlled complex Langevin equation (KCLE). The derivation is based on the work done in [39], and is extended to KCLE by showing the necessary steps that change due to a kernel. This allows us to compute boundary terms, as the first part of the correctness criterion test.

We then investigate the use of a kernel in simple models where we compare the sampled configurations to the corresponding thimble structure of the model. We see that the KCLE samples differently, relative to the thimbles for a set of optimal kernels. These kernels were selected as they showed to have no boundary terms. For these toy models, it is also possible to calculate the Fokker-Planck operator eigenvalues such that we can see when the complex Fokker-Planck equation has the correct stationary distribution. This gives us insight when the KCLE correctly converges, as this information cannot be seen directly from the boundary terms calculation alone.

5.1 Correctness criterion in the presence of a kernel

In this section, we discuss the correctness criterion in the presence of a kernel in the CL evolution. As mentioned in section 2.2 there are two parts to the correctness criterion that need to be fulfilled in order for complex Langevin to converge to the correct solution. We must avoid boundary terms for the real-valued distribution $\Phi(x^R, x^I)$ and the complex Fokker-Planck eq. (2.29) must have the correct equilibrium distribution. If both conditions are fulfilled, the correctness criterion is satisfied, hence

$$\lim_{\tau_\mathrm{L}\to\infty} \frac{1}{\tau_\mathrm{L}} \int_0^{\tau_\mathrm{L}} d\tau_\mathrm{L}' O[\phi_R + i\phi_I] \overset{?}{=} \frac{1}{Z} \int \mathcal{D}\phi \; O[\phi] e^{iS_M[\phi]} \tag{5.1}$$

holds.

To check if the equilibrium distribution of $\rho(x,\tau_L)$ is $\exp[\mathrm{iS_M}]$ we need to either solve the Fokker-Planck equation explicitly or inspect the eigenvalue spectrum of the Fokker-Planck equation [122]. To make an inference about correct convergence based on the eigenspectrum, the eigenvectors of the Fokker-Planck operator must form a complete set, as otherwise there exist non-orthogonal zero modes competing with the e^{iS_M} stationary distribution. For a non-self-adjoint operator, this is not always the case.

To show the connection between the eigenvalues of the Fokker-Planck equation and the equilibrium distribution we use a similarity transform to define the operator G from the Fokker-Planck operator L including the kernel

$$\begin{aligned} G(x) =& UL(x)U^{-1} = e^{-\frac{1}{2}iS_M(x)}L(x)e^{\frac{1}{2}iS_M(x)} \\ =& \left(\frac{\partial}{\partial x}+\frac{1}{2}i\frac{\partial S_M}{\partial x}\right)K[x]\left(\frac{\partial}{\partial x}-\frac{1}{2}i\frac{\partial S_M}{\partial x}\right), \end{aligned} \tag{5.2}$$

which by definition has the same eigenvalues as L. The transformation is carried out here to follow closely the conventional way of proving the correct convergence for a real action S. I.e., when S is real, G becomes a self-adjoint and hence negative semi-definite operator. For complex actions, iS_M, this transformation is not necessary for the following arguments. It is however useful in practice as a pre-conditioner for calculating the eigenvalues of the Fokker Planck operator. The complex distribution $\rho(x,\tau_L)$ is also transformed based on the same transformation, such that

$$\tilde{\rho}(x,\tau_L) = e^{-i\frac{1}{2}S_M}\rho(x,\tau_L), \quad \text{where} \quad \dot{\tilde{\rho}}(x,\tau_L) = G(x)\tilde{\rho}(x,\tau_L) \tag{5.3}$$

is the Fokker-Planck equation for the transformed operator. Since we are interested in the stationary distribution, we construct the eigenvalue equation

$$G(x)\psi_n(x) = \lambda_n\psi_n(x). \tag{5.4}$$

Due to the form of the operator, we know that it must have at least one zero eigenvalue, $\lambda = 0$, associated with the eigenvector $e^{i\frac{1}{2}S_M}$.

The formal solution of the Fokker-Planck equation after the similarity transform of eq. (5.3) is given by

$$\tilde{\rho}(x;\tau_L) = e^{\tau_L G(x)}\tilde{\rho}(x,0) \tag{5.5}$$

and by expanding $\tilde{\rho}(x;0)$ in the eigenbasis ψ_n, and using $\psi_0 e^{\lambda_0 t} = e^{\frac{1}{2}iS_M}$ we get

$$\begin{aligned}\rho(x;\tau_L) =& e^{\frac{1}{2}iS_M(x)}\tilde{\rho}(x;\tau_L) &(5.6)\\ =& e^{\frac{1}{2}iS_M(x)}\sum_{n=0}^{\infty} a_n\psi_n(x)e^{\lambda_n\tau_L} = ce^{iS_M(x)} + \sum_{n=1}^{\infty} a_n\psi_n(x)e^{\lambda_n\tau_L} &(5.7)\end{aligned}$$

such that when $\tau_L \to \infty$ only the first term is left, namely the equilibrium distribution $\exp[\mathrm{iS_M}]$. This is however only true if Re $\lambda_n \leq 0$, in which case the spectrum of G provides information of the equilibrium distribution of the Fokker-Planck equation.

The second condition, which needs to be satisfied is that the sampling of CL gives the same distribution as the complex Fokker-Planck equation. To establish that it does, we follow the correctness criterion of ref. [57]. Let us show that the criterion also holds in the presence of a kernel by revisiting some central steps of the original proof. We start with the Fokker-Planck equation for complex Langevin, which operates on a real distribution $\Phi(x^R, x^I; t)$ for the complexified degrees of freedom x^R and x^I. Let us take a look at the Fokker-Planck equation, which evolves the distribution of an observable O

$$\begin{aligned}\partial_{\tau_L} O(x^R, x^I) =& \Big[(H_R\partial_{x^R} + H_I\partial_{x^I})^2\\ &+ \mathrm{Re}\left\{iK[x^R+ix^I]\nabla S_M + \frac{\partial K[x^R+ix^I]}{\partial x^R}\right\}\partial_{x^R}\\ &+\mathrm{Im}\left\{iK[x^R+ix^I]\nabla S_M + \frac{\partial K[x^R+ix^I]}{\partial x^R}\right\}\partial_{x^I}\Big]\, O(x^R,x^I)\\ =& L_K^T O(x^R, x^I),\end{aligned} \tag{5.8}$$

where we can identify the operator L_K to be the bilinear adjoint of the Fokker-Planck operator L_K^T[57]. If we assume that O is holomorphic, we know that $\partial_{x^I} O = i\partial_{x^R} O \longrightarrow i\partial_z O$, where for the last equality we have used the following relation between derivatives $\partial_{x^R} O(x^R + ix^I) \longrightarrow \partial_z f(z)$ with $z = x^R + ix^I$. Replacing derivatives yields the following Langevin equation for the holomorphic observable O expressed in the complex variable z

$$\partial_t O = \left[K[z]\partial_z^2 + iK[z]\nabla S_M \partial_z + \frac{\partial K[z]}{\partial z}\partial_z \right] f \tag{5.9}$$

$$= \left[\partial_z + i\nabla S_M\right] K[z]\partial_z O = \tilde{L}_K^T O, \tag{5.10}$$

where in the last equality we have used that $K[z]\partial_z^2 + (\partial_z K[z])\partial_z = \partial_z K[z]\partial_z$ based on *integration by parts*. We have now shown that $(\tilde{L}_K^T - L_K^T)O = 0$ for a Fokker-Planck equation with a field-dependent kernel. In turn, we conclude that the correctness criterion also holds for a kernelled complex Langevin equation.

For the above derivation to hold there may not arise any boundary terms given by[68]

$$B_n = \int dx^R dx^I \Phi(x^R, x^I) \left(\tilde{L}_K^T\right)^n O(x^R + ix^I) \tag{5.11}$$

where $\tilde{L}_K^T$ is the Langevin operator given by

$$\tilde{L}_K^T = (\partial_z + i\nabla S_M) K[z]\partial_z. \tag{5.12}$$

The formal criterion is then that the observable $\langle \tilde{L}_K^T O\rangle$ should be zero. This expression for B includes contributions from the full range of values of the d.o.f. between $-\infty$ to ∞. Including all of these will introduce significant amounts of noise in the expectation value. This can be avoided by introducing a cut-off Ω for the values for x^R and x^I in the calculation of the observable. The boundary terms of eq. (5.11) are thus calculated using

$$\begin{aligned} B_n^{\Omega} &= \left\langle \left(\tilde{L}_K^T\right)^n O(x^R + ix^I) \right\rangle_{\Omega} \\ &= \left\langle \begin{cases} \left(\tilde{L}_K^T\right)^n O(x^R + ix^I), & \text{if } x^R \le \Omega_{x^R} \text{ and } x^I \le \Omega_{x^I} \\ 0, & \text{otherwise} \end{cases} \right\rangle \end{aligned} \tag{5.13}$$

where Ω_{x^R} and Ω_{x^I} denote the individual cutoffs for the real- and imaginary part respectively. In the case of scalar fields (which in contrast to gauge fields do not feature a compact dimension), we need to cut off in both x^R and x^I direction. We will in this paper stick to considering the cut-off to be a square. For all the values outside the square we set the contributions to the expectation value to zero.

Since the observable of interest in the simple models is z^2 (i.e. it is the most difficult to capture accurately), we find the boundary terms observable from eq. (5.13) to be

$$\begin{aligned}\tilde{L}_K^T\, z^2 =&(\nabla_z + i\nabla S_M)K(z)\nabla_z z^2 = (\nabla_z + i\nabla S_M)K(z)2z\\ =&2((\nabla_z K(z))z + K(z) + i\nabla S_M K(z)z)\\ =&2K(z)(1 + i\nabla S_M z) + 2(\nabla_z K(z))z,\end{aligned} \tag{5.14}$$

which for a field-independent kernel reduces to $\langle \tilde{L}_K^T\; z^2\rangle_\Omega = \langle 2K + iz\nabla S_M\rangle_\Omega$.

We have discussed both ingredients necessary to establish correct convergence of our simulation in the presence of a kernel, i.e. the behavior of the Fokker-Planck spectrum and boundary terms. The boundary terms can be calculated in practice without problems, while the eigenvalues of the Fokker-Planck operator of eq. (5.4) so far remain out of reach for realistic systems, due to computational cost.

5.2 Lefschetz thimbles

Now we briefly introduce the Lefschetz thimble method, which we will use in the next sections. For a detailed review of the method, see ref. [24]. We will define the thimbles for a complex action $S(x)$, and as for the complex Langevin method, we extend the field variable x into the complex plane, i.e., $x \to z = x + iy$. The Lefschetz thimbles [23, 102, 24] are defined as the paths of the steepest descent of the real-part of the action. They can be obtained by evolving the gradient descent/flow equation

$$\frac{\partial z}{\partial \tau} = -\overline{\frac{\partial S(z)}{\partial z}}, \tag{5.15}$$

which evolves the complexified x towards the thimble. The endpoints of the thimbles are located at the critical points, this is the fixed point of the flow (which are the stationary solutions of the drift term in the complex Langevin equation) given by the solution to

$$\frac{\partial S(z)}{\partial z} = 0. \tag{5.16}$$

Along the thimble, the imaginary part of the field variable z is constant. This can be seen from the change in the action due to the flow equation eq. (5.15), which is given by

$$\frac{\partial S}{\partial z} = \frac{\partial S}{\partial z}\frac{\partial z}{\partial \tau} = \frac{\partial S}{\partial z}\overline{\frac{\partial S}{\partial z}} = \left|\frac{\partial S}{\partial z}\right|^2, \tag{5.17}$$

this is a real quantity; hence, the imaginary part of the action does not change. This is important because we can sample along these thimbles with constant imaginary parts, avoiding the sign problem. We will not get into more detail about the method as the next section focuses on how the complex Langevin method samples relative to the Lefschetz thimbles.

5.3 Constant kernels and correct convergence in simple models

In this section, we investigate concrete examples of one-degree-of-freedom models and how we can tune kernels to get correct convergence. In the literature (see e.g. [54, 70]) kernels for these models have been constructed by hand. The motivation behind this section is to understand how the kernels affect the behavior of the complex Langevin simulation. To this end, we connect kernel-controlled complex Langevin to the Lefschetz thimbles and the correctness criterion [57].

We investigate the one-degree-of-freedom model with the action

$$S = \frac{1}{2}\sigma x^2 + \frac{\lambda}{4}x^4, \tag{5.18}$$

which leads to the following partition function

$$Z = \int d\phi e^{-S}, \tag{5.19}$$

i.e., we use the same convention as in the literature [54, 70, 82]. This model is interesting as it exhibits similar properties as the interacting real-time model: the convergence problem appears, breaking both the boundary term condition and the equilibrium distribution of the Fokker-Planck equation for various parameters.

We will therefore take a closer look at two specific sets of parameters. The first one is $\sigma = 4i$ and $\lambda = 2$ where we can find an optimal kernel, and as the second parameter, we choose $\sigma = -1 + 4i$ with the same $\lambda = 2$, where for correct convergence we have to go beyond a constant, field-independent kernel.

In section 6.3.1, we will look at a variant of this model corresponding to $\sigma = i$ and $\lambda = 0$ in eq. (5.18). The optimal field independent kernel $K = -i$ transforms the complex Langevin equation to sample exactly on the Lefschetz thimble. In contrast, the models considered here are non-linear and therefore have more than one critical point. The relation to the Lefschetz thimbles is, therefore, not as simple. The critical points for eq. (5.18), can be found via

$$\frac{\partial S(x)}{\partial x} = 0. \tag{5.20}$$

which are located at $x = 0, \pm\sqrt{\sigma/\lambda}$ [57]. We see that the smaller the real-part of the σ parameter becomes, the further out into the complex plane the two critical points away from the origin are located.

5.3.1 Non-uniqueness of the optimization

In [54], they found that the optimal kernel for the one-degree-of-freedom models rotates the drift term such that the direction is towards the origin. They first show this by using the simple model with parameters $\sigma = i$ and $\lambda = 0$ and show that the optimal kernel $K = -i$ yields $-x$ as the drift term.

In the optimization of kernels, we will use this fact such that the drift term is as close as possible to $-x$. For non-linear systems, it is not as trivial as for the linear system, since for a field-independent kernel, there does not exist a kernel that converts the drift term into $-x$. What we can do instead is to find a kernel that makes the drift term as close as possible to be $-\phi$. This will be discussed in detail in section 6.4.2, where we use this argument to construct a loss function to optimize a kernel for a real-time system. We will define the optimization function here as

$$L_D = \left\langle \left| D(x) \cdot (-x) - ||D(x)||\ ||x|| \right|^{\xi} \right\rangle \tag{5.21}$$

with $D = K\delta S/\delta x$. L_D was constructed with the idea in mind that to remove boundary terms we wish to penalize drift away from the origin. In this section, we discuss the fact that there exist multiple critical points to L_D, which may or may not correspond to a kernel that restores correct convergence.

Let us start with the parameter set $\sigma = 4i$ and $\lambda = 2$ in eq. (5.18). For this choice, ref. [54] showed that a constant kernel can be constructed that restores correct convergence.

In a one-degree-of-freedom model, where the constant kernel is nothing but a complex number, we can optimize by brute force. Using the parametrization

$$K = e^{i\theta}, \quad H = \sqrt{K} = e^{i\frac{\theta}{2}} \tag{5.22}$$

we only have to consider a single compact parameter: $\theta \in [0, 2\pi)$. A scan of the θ values reveals two minima of the L_D loss function. One at $\theta_1 = \frac{\pi}{3}$ and one at $\theta_2 = \frac{2\pi}{3}$, where the first one corresponds to the kernel found manually in ref [54]. When deriving the optimal kernel, the authors also obtained two solutions, which correspond to these two kernels. They selected the correct one by requiring the kernel to belong to the first Riemann sheet when taking a square root. In our case, we too need to select the correct one and in this simple model can use the correctness criteria directly to do so.

To proceed in this direction, let us take a look at the complex Langevin distribution according to the two kernels found in the optimization process and compare them to the Lefschetz thimble structure of the model. The thimble here consist of three different parts as shown by red lines in fig. 5.1, together with the critical points (green points). Note that the thimbles always cross through the critical points. The distribution of the complex Langevin evolution is shown as a point cloud. The three different distributions shown in each panel correspond to the case of (top left) $K_0 = 1$, (top right) $K_1 = \exp[-i\pi/3]$ and (bottom) $K_2 = \exp[-i2\pi/3]$. One can clearly see that for the trivial kernel complex Langevin tries to sample parallel to the real axis.

In the top right and bottom of fig. 5.1, we have plotted the complex Langevin distribution obtained after introducing one of the two kernels that minimize L_D. Again we find that the angle of the noise term decides where CL samples. We see that the highest density of the CL distribution lies along the direction in which the thimble passes through the critical point at the origin. Further out from the origin, the distribution follows closely the angle of the noise term, which is $H_1 = \sqrt{e^{-i\pi/3}} = e^{-i\pi/6}$ for the first kernel (top left) and $H_2 = \sqrt{e^{-i2\pi/3}} = e^{-i\pi/3}$ for the second (bottom). I.e. we can distinguish that sampling with the first kernel leads to samples slightly closer to the thimbles going out along the real-axis, compared to the other kernel which favors sampling more closely along the parts of the thimble that eventually run off to infinity. We will give a formal explanation for this behavior in the next paragraphs.

As shown in section 5.1 the correctness criterion consist of two parts, the first one states that no boundary terms may appear and the second requires that the eigenvalues of the complex Fokker-Planck equation need to have a negative real part. For this simple model we can compute both of these criteria, which is illustrated in fig. 5.2.

The left plot contains the boundary terms for the real-part of the observable $\langle x^2 \rangle$. Each of the curves corresponds to one of the three kernels K_i. They are computed using the boundary term expectation value of eq. (5.13). We see that both of the kernels lead to very small values of the

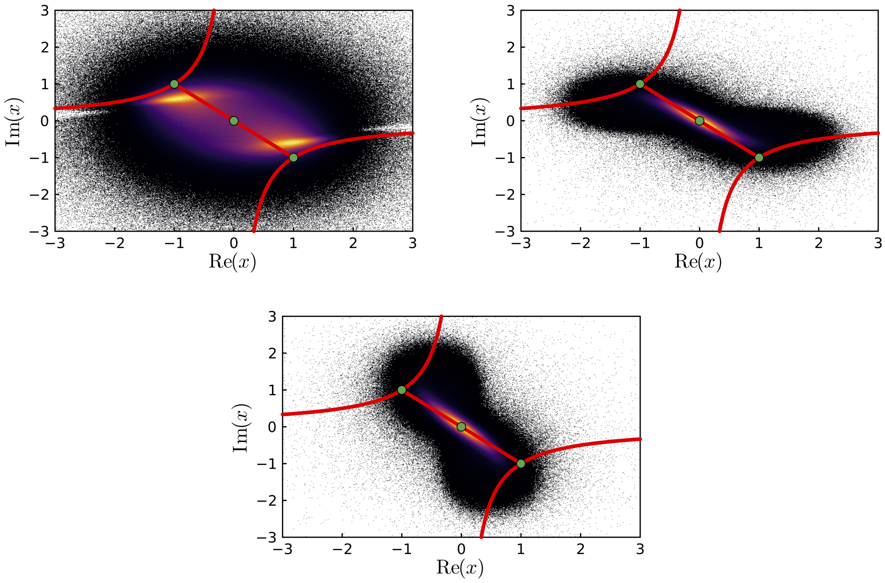

Figure 5.1: Distribution of the complex Langevin simulation and the Lefschetz thimble (red line) for the model of eq. (5.18) with $\sigma = 4i$ and $\lambda = 2$, using different kernels; $K_0 = 1$ (top left), $K_1 = \exp[-i\pi/3]$ (top right) and$K_2 = \exp[-i2\pi/3]$ (bottom). The green points denote the critical points given by the solution to eq. (5.20). The color in the distribution heat map corresponds to the number of samples at the corresponding position (a lighter color refers to a higher value).

boundary terms for this observable, while the complex Langevin process without kernel exhibits a clear boundary term. However, at this point we cannot yet say which of the two kernels produces the correct solution, if any.

In order to see which of them is correct, we need to look at the right plot in fig. 5.2 where the five eigenvalues of the Fokker-Planck operator are plotted, which have the largest real-part (blue lines). They are plotted against different kernel parameters θ and the red lines indicate the position of the two kernels that optimize L_D. The eigenvalue calculation is carried out using a restarted Arnoldi method solver, which internally uses a Krylov-Schur method. We see that there is a region of θ where the

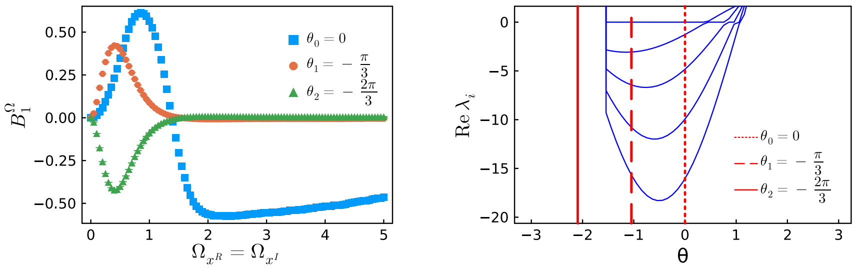

Figure 5.2: (left) Boundary term according to the x^2 observable for the model of eq. (5.18) with $\sigma = 4i$ and $\lambda = 2$, evaluated for the three different kernels K_i discussed in the main text. (right) the five eigenvalues of the Fokker-Planck operator with the largest real part (blue lines) plotted against the kernel parameter θ. The position of the two kernels that optimize L_D are indicated by red lines.

eigenvalues are all satisfying $\mathrm{Re}(\lambda) \leq 0$, which includes the kernel $\theta_1 = -\frac{\pi}{3}$. It is exactly this kernel, which, when incorporated into the complex Langevin evolution gives the right solution for the model. For smaller θs, the eigenvalues will eventually cross the zero. This is the region where one finds the second kernel $\theta_2 = -\frac{2\pi}{3}$. We can therefore attribute the failure to restore correct convergence with the second kernel to a violation of the correctness criterion pertaining to the spectrum of the complex Fokker-Planck equation.

The interesting point here is that the boundary terms do not seem to distinguish between the two kernels as both lead to quickly diminishing distributions.

5.3.2 Limitation of constant kernels and boundary terms

Let us now go to the set of parameters $\sigma = -1 + 4i$ and $\lambda = 2$, for which there does not exist a constant kernel, which restores correct convergence. It is however possible to construct a field-dependent kernel that solves the problem [70].

We can understand this behavior, as the constant kernel that is optimal in

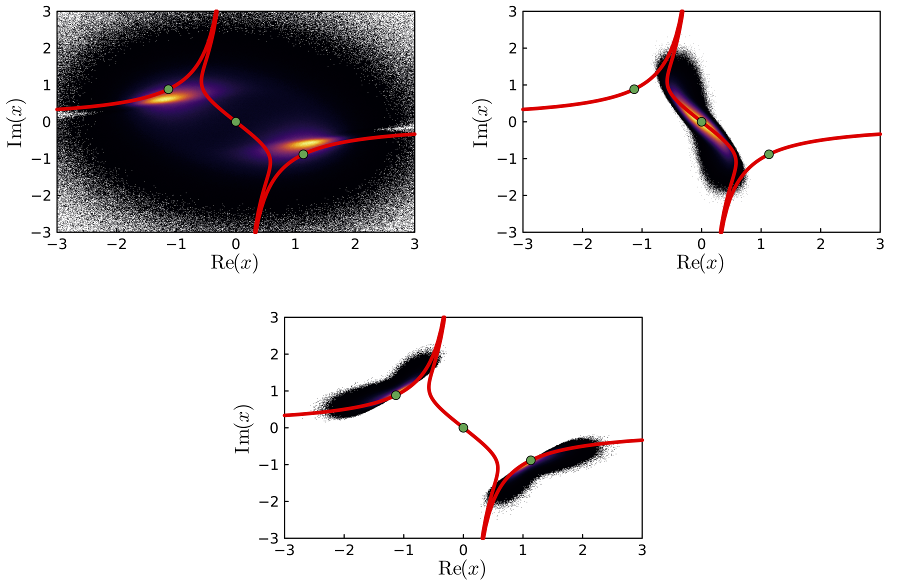

Figure 5.3: Distribution of the complex Langevin simulation and the Lefschetz thimbles (red line) for the model of eq. (5.18) with $\sigma = -1 + 4i$ and $\lambda = 2$, using different kernels; $K_0 = 1$ (top left), $K_3 = e^{-i\frac{3\pi}{4}}$ (top right) and $K_4 = e^{i\frac{\pi}{2}}$ (bottom). The green points are the critical points given by the solution to eq. (5.20).

the sense of removing boundary terms, does not achieve correct convergence of the complex Fokker-Planck equation to the correct e^{-S}.

This can be seen again by plotting the CL distribution for some of the local minima of the L_D loss function. For this parameter space there are more than two, but we have picked out two of the solutions which have the interesting property that they both have no boundary terms, and still does not converge to the true solution. The kernels that are picked have the parameters $\theta_3 = -\frac{3\pi}{4}$ and $\theta_4 = \frac{\pi}{2}$

The CL distribution together with the thimble is plotted in fig. 5.3 for the three different kernels $K_0 = 1$ (top left), $K_3 = \exp(i\theta_3)$ (top right) and $K_4 = \exp(i\theta_4)$ (bottom). We see that the thimbles show three distinct structures, connecting at infinity. To obtain the thimbles we evolve the

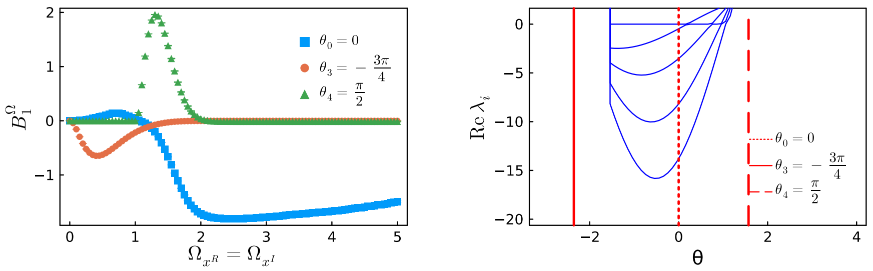

Figure 5.4: (left) Boundary term according to the x^2 observable for the model of eq. (5.18) with $\sigma = -1 + 4i$ and $\lambda = 2$, evaluated for the three different kernels K_i discussed in the main text. (right) the five eigenvalues of the Fokker-Planck operator with the largest real part (blue lines) plotted against the kernel parameter θ. The position of the two kernels that optimize L_D are indicated by red lines.

gradient flow equation starting from a small offset from the critical points (which all are saddle points) and then combine the six part of the thimbles. The CL distribution without a kernel (top left plot in fig. 5.3) again favors sampling parallel to the real-axis, while the two other kernels sample completely different parts of the thimbles. The distribution for K_3 is located along the thimble crossing the origin. The other kernel (K_4), follows the other two thimbles crossing the critical points away from the origin. We can explain this behavior with the angle of the noise coefficient. For K_3 we have an angle of $-\frac{3\pi}{8}$ against the real axis and for K_4 we have an angle of $\frac{\pi}{4}$ against the real axis.

In fig. 5.4 (left) the boundary terms for this set of model parameters is calculated for the observable Re $\langle x^2 \rangle$ and plotted for increasing square box cutoff. We see that without a kernel, there are boundary terms present, as the blue datapoints do not go to zero for large cutoff. This can also been seen directly from the distribution in fig. 5.3 which exhibits a large spread and hence the falloff of the distribution is not fast enough. For the two kernels, K_3 and K_4, that correspond to a local minimum in L_D, the system does not show any boundary terms. This is an important point as even though we have avoided boundary terms, the CL dynamics under the kernels K_3 and K_4 still does not converge to the correct solution. In turn it appears that it is in general not enough to remove the boundary terms to achieve correct convergence. In fact one also need to be sure that the complex Fokker-Planck equation converges to the desired equilibrium distribution.

In fig. 5.4 (right) we show the five eigenvalues of the Fokker-Planck equation with the largest real-part plotted against the parameter θ which determines the kernel $K = e^{i\theta}$. For the parameters chosen here, we find that both kernels lie outside of the admissible region[1], where Re $\lambda \leq 0$. Interestingly at $\theta = 0$ the eigenvalues actually all lie in the lower half complex plane but there the boundary criterion is not fulfilled. But as one increases the imaginary part of σ, e.g. at $\sigma = -1 + 5i$, one finds that the eigenvalues for the identity kernel $K = 1$ already take on positive real-parts.

5.4 Summary

In this chapter, we have derived the correctness criterion for a kernel-controlled complex Langevin equation. We ended up with the formula to calculate boundary terms, similar to the standard boundary term calculation but now including a kernel.

We proceed by choosing optimal kernels for a toy model, for two different parameter sets, based on minimizing the drift not pointing toward the origin. The optimal kernels for this loss function also corresponded to zero boundary terms, however, only one of them corresponded to a correct convergence of the CLE.

This means that we cannot rely solely on the boundary terms to check if the kernel-controlled complex Langevin simulation converged to the correct solution, or not. We also need to check the Fokker-Planck eigenvalues,

[1] An interesting observation was made in [70], that combining kernels, which sample different parts of the thimble into a field-dependent kernel seems to work well. The motivation was to find a kernel that would reduce the drift term to $-x$ when either the x^2 or the x^4 term in the action dominates. A similar argument for constructing a field-dependent kernel can now be made via the minima of the L_D loss function, which favor sampling different parts of the thimbles.

which is equivalent to stating that we need to also check whether the stationary distribution is $\exp[-S]$. Checking the second criterion for larger problems is however not feasible with today's methods, as it requires us to calculate the Fokker-Planck eigenvalues. The size of the Fokker-Planck operator scales as N^d, where N is the number of points along every dimension d. For a 4-dimensional model with 10 lattice points in every dimension ($d = 10^4$) using $N = 100$ points to express these points, this means a matrix of the size 10^{20^4}, which is too large to be able to find eigenvalues of.

We have also investigated the connection between the application of a kernelled Langevin equation and Lefschetz thimbles. This showed that applying a kernel will change the CLE distribution based on the match between the kernel's angle and the thimble's angle at the critical points.

6 Towards learning optimized kernels for complex Langevin

This chapter is published in *Journal of High energy Physics 04. (2023) 057* [31].

We present a novel strategy aimed at restoring correct convergence in complex Langevin simulations. The central idea is to incorporate system-specific prior knowledge into the simulations, in order to circumvent the NP-hard sign problem. In order to do so, we modify complex Langevin using kernels and propose the use of modern auto-differentiation methods to learn optimal kernel values. The optimization process is guided by functionals encoding relevant prior information, such as symmetries or Euclidean correlator data. Our approach recovers correct convergence in the non-interacting theory on the Schwinger-Keldysh contour for any real-time extent. For the strongly coupled quantum anharmonic oscillator we achieve correct convergence up to three-times the real-time extent of the previous benchmark study.

6.1 Motivation

Strongly correlated quantum systems underlie some of the most pressing open questions in modern theoretical physics. Whether it is the transport of highly energetic partons through a liquid of deconfined quarks and gluons [123], created in heavy-ion collisions [5] or the transport of non-relativistic fermions [6], captured in the iconic Hubbard model [21] at low energies. When formulated in Minkowski time, quantum field theories so far have defied a treatment by conventional Monte-Carlo simulation techniques, due to the presence of the notorious sign problem [11, 12]. And while progress has been made in extracting real-time dynamics from Euclidean time simulations using e.g. Bayesian inference [124], the sign problem prevails by rendering the extraction ill-posed and equally exponentially hard.

The sign problem has been proven to be NP-hard [32], which entails that no generic solution method is likely to exist. In turn, if we wish to make inroads towards overcoming the sign problem, system-specific solutions are called for.

Over the past decade, several approaches to tackle the sign problem have been put forward [11, 20]. They can be divided into system-specific and system-agnostic approaches. The reformulation strategies discussed e.g. in Refs. [125, 126, 127] are an example of the former class, where the partition function of the original system is re-expressed in terms of new degrees of freedom, for which no sign problem exists. While highly successful in the systems for which a reformulation has been discovered, no systematic prescription exists to transfer the approach to other systems. The other approaches, among them reweighting, extrapolation from sign-problem free parameter ranges [128, 100, 129, 130], density of states [131, 132, 133], tensor networks [25, 26], Lefschetz thimbles [23, 102, 24] and complex Langevin (CL) [104, 53] all propose a generic recipe to estimate observables in systems with a sign problem. As the NP-hard sign problem however requires system-specific strategies, all of these methods are destined to fail in some form or the other. Be it that their costs scale excessively when deployed to realistic systems (e.g. reweighting, Lefschetz thimbles, tensor networks) or that they simply fail to converge to the correct solution (complex Langevin).

Both the Lefschetz Thimbles and complex Langevin belong to the class of complexification strategies [20]. They attempt to circumvent the sign problem by moving the integration of the Feynman integral into the complex plane. After complexifying the degrees of freedom, the former proposes to integrate over a specific subspace on which the imaginary part of the Feynman weight remains constant (thimble), while the latter proposes to carry out a diffusion process of the coupled real- and imaginary part of the complexified degrees of freedom.

In this paper our focus lies on the complex Langevin approach, as it has been shown to reproduce correctly the physics of several strongly correlated model systems, albeit in limited parameter ranges [47]. Most importantly in its naive implementation it scales only with the volume of the system, similar to conventional Monte-Carlo simulations. In the past, complex Langevin had suffered from two major drawbacks: the occurrence of unstable trajectories, called runaways and the convergence to incorrect solutions. In a previous publication [38] we have shown how to avoid runaways by deploying inherently stable implicit solvers (c.f. the use of adaptive step size [52]). In this study we propose a novel strategy to restore correct convergence in the complex Langevin approach.

One crucial step towards establishing complex Langevin as reliable tool to attack the sign problem is to identify when it converges to incorrect solutions. The authors of ref. [57] and later [55] discovered that in order for CL to reproduce the correct expectation values of the underlying theory, the histograms of the sampled degrees of freedom must fall off rapidly in the imaginary direction. Otherwise boundary terms spoil the proof of correct convergence. The absence of boundary terms has been established as necessary criterion and efforts are underway [134] to compensate for their presence to restore correct convergence.

With QCD at the center of attention, the gauge cooling strategy [59, 135], based on exploiting gauge freedom, has been proposed. It has recently been amended by the dynamic stabilization approach [64, 136], which modifies the CL stochastic dynamics with an additional drift term. Both are based on the idea that by pulling the complexified degrees of freedom closer to the real axis, boundary terms can be avoided. Their combination has led to impressive improvements in the correct convergence of complex Langevin in the context of QCD thermodynamics with finite Baryo-chemical potential [137] and is currently explored in the simulation of real-time gauge theory [138].

We focus here on scalar systems formulated in real-time on the Schwinger-Keldysh contour (for a Lefschetz thimble perspective see [116, 139]). For scalars, gauge freedom does not offer a rescue from the convergence problems. The fact that dynamical stabilization introduces a non-holomorphic modification of the drift term means that the original proof of convergence is not applicable, which is why we refrain from deploying it here. Furthermore the boundary term correction requires that the eigenvalues of the Fokker-Planck equation associated with the original system lie in the lower half of the complex plane, which is not necessarily the case in the scalar systems that we investigate.

The convergence problem in real-time complex Langevin is intimately connected with the extent of the real-time contour [106]. In a previous publication [38] we showed that for a common benchmark system, the strongly correlated quantum anharmonic oscillator, real-time simulations directly on the SK contour are feasible for times up to $mt_{\max} = 0.5$. Convergence quickly breaks down when extending the contour beyond this point.

Within the complex Langevin community, coordinate transformations and redefinitions of the degrees of freedom have been used in the past to weaken the sign problem in a system specific manner (see e.g. discussion in [121]). All of these reformulations can be captured mathematically by introducing a so called kernel for complex Langevin. It amounts to a simultaneous modification of the drift and noise contribution to the CL stochastic dynamics. In the past it has been used to improve the autocorrelation time in real-valued Langevin simulations [53] and has been explored in simple model systems to restore the convergence of complex Langevin (see e.g. [54]). The construction of the kernels, as discussed in the literature applies to a specific system only and so far no systematic strategy exists to make kernels work in more realistic theories.

Our study takes inspiration from both conceptual and technical developments in the machine learning community. In machine learning, an optimization functional, based on prior knowledge and data is used to train an algorithm to perform a specific task. The algorithm depends on a set of parameters, e.g. the weights of a neural network, which need to be tuned to minimize the prescribed optimization functional. Highly efficient automatic differentiation programming techniques [30] have been developed to compute the dependence of the outcome of complex algorithms on their underlying parameters. Here we utilize them to put forward a systematic strategy to incorporate prior knowledge about the system into the CL evolution by learning optimal kernels.

In section 6.2 we review the concept of kernelled Langevin, first in the context of Euclidean time simulations and subsequently for use in complex Langevin. In section 6.3 we show how the concept of a kernel emerges in a simple model system and how it relates to the Lefschetz thimbles of the model. Subsequently we discuss that a constant kernel can be used to restore convergence of real-time complex Langevin for the quantum harmonic oscillator. The kernel found in this fashion will help us to improve the convergence of the interacting theory too. Section 6.4 introduces the central concept of our study: a systematic strategy to learn optimal kernels for complex Langevin, based on system-specific prior information. Numerical results from deploying a constant kernel to the quantum anharmonic oscillator are presented in section 6.4.3 (Source code for the kernel optimization and simulation is written in Julia and available at [140]), leading to a significant extension of correct convergence. In the appendices, we discuss some of the limitations of constant kernels and show in the context of simple models that correct convergence requires not only the vanishing of boundary terms but in addition requires the spectrum of the associated Fokker-Plank equation to remain negative.

6.2 Neutral and non-neutral modifications of Langevin dynamics

Stochastic quantization, the framework underlying Langevin simulations, sets out to construct a stochastic process for fields in an artificial additional time direction $\tau_{\rm L}$ with a noise structure, which correctly reproduces the quantum statistical fluctuations in the original theory. In the context of conventional Monte-Carlo simulations in Euclidean time, where expectation values of observables are given by the path integral

$$\langle O\rangle = \frac{1}{Z}\int \mathcal{D}\phi\, O[\phi]e^{-S_E[\phi]}, \quad S_E[\phi] = \int d^dx L_E[\phi], \qquad (6.1)$$

with Euclidean action S_E, the goal thus is to guarantee at late Langevin times a distribution of fields $\Phi[\phi] \propto \exp\big(-S_E[\phi]\big)$. The chain of configurations $\phi(\tau_L)$ underlying the distribution $\Phi[\phi]$, can then be used to evaluate the expectation values of observables O from the mean of samples $\langle O\rangle = \lim_{\tau_{\rm L}\to\infty}\frac{1}{\tau_{\rm L}}\int_0^{\tau_{\rm L}} d\tau'_{\rm L} O[\phi(\tau'_{\rm L})]$. The simplest stochastic process, which realizes this goal and which is therefore commonly deployed is

$$\begin{aligned}&\frac{d\phi}{d\tau_{\rm L}} = -\frac{\delta S_E[\phi]}{\delta\phi(x)} + \eta(x,\tau_{\rm L}) \quad \text{with}\\ &\langle\eta(x,\tau_{\rm L})\rangle = 0, \quad \langle\eta(x,\tau_{\rm L})\eta(x',\tau'_{\rm L})\rangle = 2\delta(x-x')\delta(\tau_{\rm L}-\tau'_{\rm L}).\end{aligned} \tag{6.2}$$

Its drift term is given by the derivative of the action S_E and the noise terms η are Gaussian. The associated Fokker-Planck equation reads

$$F_{\rm FP} = \int d^dx \, \frac{\partial}{\partial\phi(x)}\left(\frac{\partial}{\partial\phi(x)} + \frac{\delta S_E[\phi]}{\delta\phi(x)}\right), \quad \frac{\partial\Phi(\phi,\tau_{\rm L})}{\partial\tau_{\rm L}} = F_{\rm FP}\Phi(\phi,\tau_{\rm L}). \tag{6.3}$$

For an in-depth review of the approach see e.g. ref. [53].

In the following we will discuss the fact that there exists the freedom to introduce a so called *kernel* into eq. (6.3), which as a *purely real* quantity allows us to modify the above Fokker-Planck equation without spoiling the convergence to the correct stationary solution $\Phi[\phi] = \lim_{\tau_L\to\infty}\Phi[\phi,\tau_L] \propto \exp\big(-S_E[\phi]\big)$. One may use this freedom to improve autocorrelation times of the simulation and for other problem-specific optimizations as has been explored in the literature.

Subsequently we will turn our attention to the case of *complex Langevin*, where the simplest stochastic process proposed by stochastic quantization is not guaranteed to converge to the correct solution. In that case we will explore how a reparametrization of the associated Fokker Planck equations through in general *complex kernels* can be used to not only change the convergence speed but actually to change the stationary distribution itself, allowing us to recover correct convergence where the naive process fails.

6.2.1 Kernelled Real Langevin

As alluded to above, there exists a freedom to reparametrize the Fokker-Planck eq. (6.3) by introducing a real-valued kernel function $K_{ij}(x, x', \phi; \tau_{\rm L})$

$$F_{\rm FP} = \sum_{i,j} \int d^d x \int d^d x' \, \frac{\partial}{\partial \phi_i(x)} K_{ij}(x, x', \phi; \tau_{\rm L}) \left(\frac{\partial}{\partial \phi_j(x')} + \frac{\delta S_E[\phi]}{\delta \phi_j(x')} \right). \tag{6.4}$$

Written in its most generic form, it may couple the different degrees of freedom of the system (according to the ij indices), it may couple different space-time points (according to its x and x' dependence) and may depend explicitly both on the Langevin time $\tau_{\rm L}$, as well as the field degrees of freedom ϕ. The corresponding Langevin equation reads

$$\begin{aligned} \frac{d\phi_i(x,\tau_{\rm L})}{d\tau_{\rm L}} =& \sum_j \Bigg\{ - \int d^d x' K_{ij}(x, x'; \phi) \frac{\delta S_E[\phi]}{\delta \phi_j(x', \tau_{\rm L})} \\ &+ \int d^d x' \frac{\delta K_{ij}(x, x'; \phi)}{\delta \phi_j(x', \tau_{\rm L})} \\ &+ \int d^d x' H_{ij}(x, x'; \phi) \eta(x', \tau_{\rm L}) \Bigg\} \quad \text{with} \\ K(x, x'; \phi) =& \sum_k \int d^d x'' H_{ik}(x, x''; \phi) H_{jk}(x', x''; \phi), \end{aligned} \tag{6.5}$$

where in the last equation we assume that K is factorizable. In practice we will either choose kernels, which can be factorized using the square root of their eigenvalues or will start directly by constructing the function H that can be combined into an admissible K.

Let us gain a bit of intuition about the role of the kernel when considering it in its simplest form, a constant scalar kernel, which multiplies each d.o.f. with a real number γ. Inspecting eq. (6.5) we find that, as it appears in front of the drift term and as square root in front of the noise term, γ simply leads to a redefinition of the Langevin time coordinate $\tau'_{\rm L} = \gamma \tau_{\rm L}$. While the stationary solution is left unchanged, the convergence time has been modified.

Even for more general kernels, the fact that K appears in the generalized Fokker-Planck eq. (6.4) on the outside of the parenthesis $\left(\frac{\delta}{\delta\phi_i(x)} + \frac{\delta S_E[\phi]}{\delta\phi_i(x)}\right)$ tells us that the stationary distribution remains unchanged. It goes without saying that choosing $K_{ij}(x,x';\phi) = \delta_{ij}\delta(x'-x)$ we regain the standard Langevin eq. (6.2).

6.2.2 Kernelled Complex Langevin

Let us now consider the application of stochastic quantization to complex-valued path integrals, in particular to those describing real-time physics in Minkowski time. Here the observables are given by Feynman's path integral

$$\langle O\rangle = \frac{1}{Z}\int \mathcal{D}\phi\, O[\phi]e^{iS_M[\phi]}, \quad S_M[\phi] = \int d^dx L_M[\phi], \tag{6.6}$$

which houses the Minkowski time action of the theory S_M. Stochastic quantization in this case *proposes* to modify the real-valued stochastic process of eq. (6.2) via the substitution $-S_E \longrightarrow iS_M$ such that

$$\begin{aligned}&\frac{d\phi}{d\tau_\mathrm{L}} = i\frac{\delta S_M[\phi]}{\delta\phi(x)} + \eta(x,\tau_\mathrm{L}) \quad \text{with}\\ &\langle\eta(x,\tau_\mathrm{L})\rangle = 0, \quad \langle\eta(x,\tau_\mathrm{L})\eta(x',\tau_\mathrm{L}')\rangle = 2\delta(x-x')\delta(\tau_\mathrm{L}-\tau_\mathrm{L}').\end{aligned} \tag{6.7}$$

It is obvious that even if one starts out with purely real degrees of freedom at $\tau_\mathrm{L} = 0$, the presence of the complex drift term necessitates the complexification $\phi = \phi_\mathrm{R} + i\phi_\mathrm{I}$, each of which will obey a coupled stochastic evolution.

In the complexified scenario, the question of correct convergence is not as simple to answer as in the purely real case. The most stringent criterion refers to whether complex Langevin reproduces the correct expectation values

$$\lim_{\tau_\mathrm{L}\to\infty}\frac{1}{\tau_\mathrm{L}}\int_0^{\tau_\mathrm{L}} d\tau_\mathrm{L}' O[\phi_R + i\phi_I] \overset{?}{=} \frac{1}{Z}\int \mathcal{D}\phi\, O[\phi]e^{iS_M[\phi]} \tag{6.8}$$

of the theory, defined on the right. And indeed it has been found that the dynamics of eq. (6.7) may violate the equal sign of eq. (6.8). I.e. complex

Langevin converges, but it does not converge to the correct solution. In this study we set out to recover correct convergence by introducing kernels into the complex Langevin dynamics.

To this end we consider a not-necessarily real kernel function $K(x,x';\phi)$ which enters the complexified dynamics as

$$\begin{aligned}\frac{d\phi}{d\tau_{\rm L}} &= \int d^dx' \big\{ iK(x,x';\phi)\frac{\delta S_M[\phi]}{\delta\phi(x',\tau_{\rm L})} + \frac{\partial K(x,x';\phi)}{\partial\phi(x',\tau_{\rm L})} \\ &\qquad + H(x,x';\phi)\eta(x,\tau_{\rm L})\big\} \\ \text{with} \quad & \langle\eta(x,\tau_{\rm L})\rangle = 0, \quad \langle\eta(x,\tau_{\rm L})\eta(x',\tau'_{\rm L})\rangle = 2\delta(x-x')\delta(\tau_{\rm L}-\tau'_{\rm L}) \\ \text{and} \quad & K(x,x';\phi) = \int d^dx'' H(x,x'';\phi)H(x',x'';\phi).\end{aligned} \tag{6.9}$$

Expressed as two separate but coupled stochastic processes for the real- and imaginary part of the complexified field we obtain

$$\begin{aligned}\frac{d\phi_R}{d\tau_{\rm L}} &= \int d^dx' \left\{ \text{Re}\left[K[\phi] i \frac{\delta S_M[\phi]}{\delta\phi} + \frac{\delta K[\phi]}{\delta\phi} \right] + \text{Re}\left[H[\phi]\right]\eta \right\}\Bigg|_{\phi=\phi_R+i\phi_I}, \\ \frac{d\phi_I}{d\tau_{\rm L}} &= \int d^dx' \left\{ \text{Im}\left[K[\phi] i \frac{\delta S_M[\phi]}{\delta\phi} + \frac{\partial K[\phi]}{\partial\phi} \right] + \text{Im}\left[H[\phi]\right]\eta \right\}\Bigg|_{\phi=\phi_R+i\phi_I}.\end{aligned} \tag{6.10}$$

Note that at this point we are dealing with two different concepts of Fokker-Planck equations. One describes how the probability distribution $\Phi[\phi_R,\phi_I]$ of the real- and imaginary part ϕ_R, ϕ_I of the complexified field evolve under eq. (6.9)

$$\begin{aligned}\frac{\partial\Phi}{\partial\tau_{\rm L}} = \Bigg[&\left(\frac{\partial}{\partial\phi^R}H_R + \frac{\partial}{\partial\phi^I}H_I \right)^2 - \frac{\partial}{\partial\phi^R}\text{Re}\left\{ iK\frac{\partial S_M}{\partial\phi} + \frac{\partial K}{\partial\phi} \right\} \\ &- \frac{\partial}{\partial\phi^I}\text{Im}\left\{ iK\frac{\partial S_M}{\partial\phi} + \frac{\partial K}{\partial\phi} \right\} \Bigg] \Phi = L_K\Phi.\end{aligned} \tag{6.11}$$

We define the operator for the real Fokker Planck equation, which has been separated into real and imaginary part, as L_K, not to be confused with the original now complex Fokker Planck equation F_{FP}, which was only defined on the real part of ϕ. For the term quadratic in derivatives, we have split the kernel K into the product of H functions, as shown in eq. (6.9), such that each derivative acts on either the real or the imaginary part of H respectively. Since it is the noise term of the Langevin eq. (6.9) that translates into a term quadratic in derivatives in the Fokker-Planck language, it is there that H appears in eq. (6.11).

It is important to recognize that the correct late Langevin-time distribution of this Fokker-Planck equation is purely real and therefore is not related in a trivial manner to the Feynman weight $\exp[\mathrm{iS_M}]$ of the original path integral, as has been established in simple models in the literature as discussed e.g. in refs. [122, 141, 142, 103, 143].

The other Fokker-Plank equation is not a genuine Fokker-Planck equation, in the statistical sense, as it does not describe the evolution of a real-valued probability density $P[\phi, \tau_L]$ but instead that of a complex-valued distribution $\rho(\phi, \tau_L)$

$$\frac{\partial}{\partial \tau_L}\rho(\phi, t) = F_{FP}\rho(\phi, \tau_L), \tag{6.12}$$
$$F_{FP} = \sum_{i,j}\int d^dx \int d^dx' \; \frac{\partial}{\partial\phi_i(x)}K_{ij}(x, x', \phi; \tau_\mathrm{L})\left(\frac{\partial}{\partial\phi_j(x')} - i\frac{\delta S_M[\phi]}{\delta\phi_j(x')}\right).$$

It is this equation whose late time limit we expect to reproduce the Feynman weight $\lim_{\tau_L\to\infty}\rho(\phi, \tau_L) = \exp[\mathrm{iS_M}]$ and we will refer to in the following as the complex Fokker-Planck equation.

Significant progress in the understanding of the convergence properties of complex Langevin had been made starting with ref. [57] in the form of so-called correctness criteria.

The criteria most often discussed in the literature are boundary terms (for a detailed exposition see Refs. [57, 55]). They tell us if the expectation value calculated from the real distribution $\Phi(\phi^R, \phi^I; \tau_\mathrm{L})$ (eq. (6.11)), which we can sample using the CL, is the same as the expectation value obtained from the complex distribution $\rho(\phi; \tau_\mathrm{L})$. The latter one can only be obtained from solving the complex Fokker-Planck equation, eq. (6.12). The two expectation values, $\langle O\rangle_\Phi(\tau_\mathrm{L}) = \langle O\rangle_\rho(\tau_\mathrm{L})$ only agree if $\Phi(\phi^R, \phi^I; \tau_\mathrm{L})$

falls off exponentially fast. If it does not fall of sufficiently fast, it will produce boundary terms and the equal sign in eq. (6.8) is not valid. This criterion is however not sufficient as it does not guarantee the equilibrium distribution of the complex Fokker-Planck equation to be $\exp[i\mathrm{S}_{\mathrm{M}}]$. These two criteria combined are however sufficient to claim convergence of the CL to the true solution. For a proof that the correctness criterion still holds after introducing a kernel into the CL, we revisit the proof in section 5.1.

How can a kernel help to restore the correct convergence? Not only do we need to make sure that no boundary terms arise in sampling eq. (6.10) but also that the complex Fokker-Planck equation has a unique and correct complex stationary distribution. I.e. we need in general a non-neutral modification of the complex Langevin dynamics.

If we were to introduce a real-valued kernel, similarly to the case of conventional real-valued Langevin, we will be able to change the speed of convergence but not the stationary solution. On the other hand, since the drift term is complex, there is no reason not to consider also complex valued kernels, which will act differently on the stochastic process for ϕ_R and ϕ_I, representing a genuine non-neutral modification in the corresponding Fokker-Planck eq. (6.11). Similarly in the complex Fokker-Planck eq. (6.12) the presence of a complex K_{ij} can change the stationary distribution through a reshuffling of the associated eigenvalues, as is discussed in more detail in section 5.1. For a comprehensive discussion of different modifications to complex Langevin, including kernels, see also ref. [121].

In the following sections we will start off with constructing an explicit example of a field-independent kernel that improves convergence in the free theory and find that it can restore correct convergence in the interacting theory to some degree. We will then continue to present our novel strategy to learn optimal kernels for the restoration of correct convergence and showcase their efficiency in a benchmark model system. Subsequently we discuss the limitations of field-independent kernels and shed light on how kernels connect to the correctness criteria.

6.3 A field independent kernel for real-time complex Langevin

In this section, we will manually construct one specific field-independent kernel and demonstrate its use to improve convergence in real-time simulations of the quantum anharmonic oscillator. The form of the kernel is motivated by insight gained in a simple one d.o.f. model and reveals an interesting connection between kernelled Langevin and the thimble approach. Since in the following only low dimensional model systems are considered, we will refer to the dynamical degrees of freedom from now on as x.

6.3.1 A kernel for the simplest real-time model

Following ref. [54] let us investigate the simplest model of real-time physics, the integrals

$$\langle x^n \rangle = \frac{1}{Z} \int dx\, x^n \exp[-\frac{1}{2} i x^2], \quad Z = \int dx\, \exp[-\frac{1}{2} i x^2]. \tag{6.13}$$

Attempting to solve this expression using the complex Langevin approach for $x(\tau_L)$, leads to a stochastic process

$$\frac{dx}{d\tau_L} = -ix + \eta, \tag{6.14}$$

with Gaussian noise η. Equation (6.14) fails at reproducing correct values of $\langle x^n \rangle$.

We can understand this failure by recognizing that without regularization the original integral in eq. (6.13) is not well defined and this lack of regularization is inherited by the Langevin eq. (6.14).

One way to proceed is to explicitly modify the action by introducing a regulator term, such as ϵx^2. The integral becomes well-defined and its value is obtained when we let $\epsilon \to 0$ at the end of the computation. In a numerical setting this would require to explicitly include the regulator term, carry out the corresponding simulation for different values of ϵ and extrapolate $\epsilon \to 0$. There are two drawbacks to this strategy: first it requires several evaluations of the simulation, which can be expensive for large systems. The second reason is that the relaxation time for the simulation grows, the smaller ϵ becomes, and hence in practice we cannot make ϵ arbitrarily small.

Let's consider an alternative strategy of solving the integral of eq. (6.13), which relies on contour deformations, the so-called Lefschetz thimble method. We will carry out a change of variables in the integral which moves the integration path into the complex plane and which in turn will weaken the oscillatory nature of the integrand. This method is based on a continuous change of variables according to the following gradient descent equation

$$\frac{d\tilde{x}}{d\tau} = \overline{\frac{dS_E[\tilde{x}]}{d\tilde{x}}}, \tag{6.15}$$

which complexifies the degree of freedom, $\tilde{x} = a + ib$. Equation (6.15) evolves the formerly real-valued x towards the so called Lefschetz thimble which is the optimal contour deformation where the imaginary part of the action stays constant.

Following the steps outlined in [144], we solve the flow of eq. (6.15) analytically which gives $\tilde{x}(x,\tau) = x(\cosh(\tau) - i\sinh(\tau))$. For large values of τ it leads to

$$\tilde{x}(x,\tau) \overset{\tau\gg 1}{\approx} x(1-i)\frac{1}{2e^{-2\tau}} = \frac{x}{2e^{-2\tau}}e^{-i\frac{\pi}{4}}. \tag{6.16}$$

The above equation tells us that the optimal thimble in this system lies on the downward 45° diagonal in the complex plane $z(x) = xe^{-i\frac{\pi}{4}}$. On this contour the integrand of the original integral eq. (6.14) reduces to a real Gaussian e^{-x^2} for which no regularization is required.

If we flow for just a very small $\tau = \epsilon$, we obtain on the other hand $\cosh(\tau) - i\sinh(\tau) \approx 1 - i\epsilon$ and

$$\begin{aligned} \int dx \exp[-\frac{1}{2}ix^2] &= \int dx \frac{\partial\tilde{x}}{\partial x}\exp[-\frac{1}{2}i\tilde{x}^2] \qquad (6.17)\\ = (1-i\epsilon)&\int dx\, \exp[-\frac{1}{2}ix^2(1-i\epsilon)^2]\\ \approx (1-i\epsilon)&\int dx\, \exp[-\frac{1}{2}ix^2 - \epsilon x^2]. \end{aligned}$$

We see that the term ϵ here takes on the role of a regulator in the action but due to its presence also in the Jacobian, the value of the integral is not changed. This is different from introducing the regulator only in the action itself.

Hence the obvious benefit of the deformation method is that we can introduce a regulator to tame oscillations without the need to extrapolate that regulator in the end. The closer we approach the optimal thimble, the easier the integral will be to solve numerically.

How can such a coordinate transformation be implemented in complex Langevin? Intuitively the action in the integral is what influences the drift in complex Langevin and the measure is related to the noise structure. The above tells us that the change we introduced will therefore affect the drift quadratically, while it occurs in the noise linearly. Thinking back to eq. (6.9), we see that this is just how a field-independent kernel modifies the complex Langevin equations.

For the optimal thimble with $z(x) = xe^{-i\frac{\pi}{4}}$ the modification in the drift therefore becomes $K = e^{-i\frac{\pi}{2}} = \frac{1}{i}$ and for the noise $H = \sqrt{K} = \sqrt{-i}$. This leads to the following stochastic process

$$\frac{dx}{d\tau_L} = -x + \sqrt{-i}\eta, \tag{6.18}$$

which had been identified as optimal already in ref. [54]. This stochastic process converges to the correct solution of the integral eq. (6.13). Interestingly the imaginary unit has disappeared from the drift term since the kernel K exactly canceled it there and instead moved it over into the noise term.

As the last step, let us show explicitly that the choice of kernel above indeed amounts to a coordinate transform. Following [121] we have

$$\frac{dx}{d\tau_L} = -\, HH^T \frac{\partial S_E(x)}{\partial x} + H\eta \tag{6.19}$$

$$\Rightarrow H^{-1}\frac{dx}{d\tau_L} = -H^T \frac{\partial S_E(x)}{\partial x} + \eta \tag{6.20}$$

$$\Rightarrow \frac{du}{d\tau_L} = -H^T (H^T)^{-1} \frac{\partial T(u)}{\partial u} + \eta = -\frac{\partial T(u)}{\partial u} + \eta. \tag{6.21}$$

Here $x = Hu$ and $T(u) = S_E(Hu) = S_E(x)$. We find that introducing a kernel $K = HH^T$ in the evolution equation for x has the same effect as carrying out a coordinate transformation to $u = H^{-1}x$.

As an example of the complex Langevin dynamics in the absence (left panel) and presence (right panel) of the kernel discussed above, we show the corresponding scatter plots in fig. 6.1. The kernel has indeed rotated the noise into the direction of the thimble, along which the system now samples. Note that while the naive CL dynamics have been implemented using the semi-implicit Euler-Maruyama scheme to avoid runaways, we are able to carry out the kernelled dynamics with a fully explicit solver without adaptive step size. The reason is that on the deformed contour the integral has already been regularized.

This result shows that in the simple model discussed here we can find a kernel that both restores correct convergence of the complex Langevin dynamics and at the same time removes the need for a regulator. Both are related to the fact that the kernel effectively instituted a coordinate transform that amounts to a deformation of the integration contour into the complex plane. In the next section, we will consider a similar kernel for the harmonic oscillator.

We investigated the relationship between the Lefschetz thimbles and kernel-controlled complex Langevin in section 5.3.2, where a similar analysis was performed in a system where a $\frac{\lambda}{4}x^4$ term has been added to the action. As such a one-degree-of-freedom model has a non-linear drift term, a constant kernel, in that case, will not suffice to remove the imaginary unit from the drift term, and hence the complex Langevin

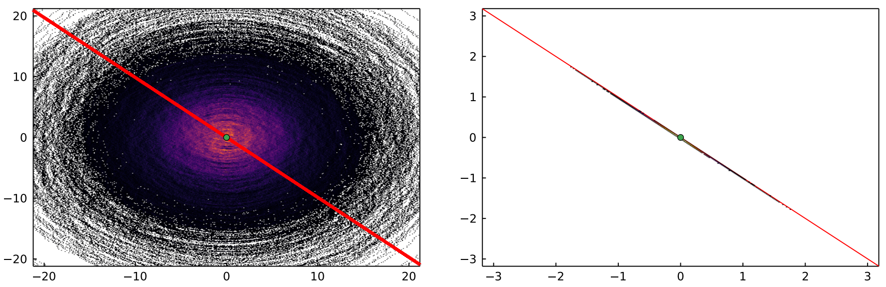

Figure 6.1: Distribution of a complex Langevin simulation (scatter points) and the Lefschetz thimble (red line) for the model eq. (6.13). (left) simulation according to the naive CL eq. (6.14) and (right) simulation after introducing the kernel in eq. (6.18). The color of the scatter points refers to the number of measurements recorded at the corresponding position, a lighter color indicates a larger number. Note that the optimal kernel has moved the sampling onto the single thimble present in this simple system.

will not sample directly on the thimble as was the case for the example above.

6.3.2 A kernel for the harmonic oscillator

When constructing a kernel for the harmonic oscillator, we encounter similar difficulties related to stability and convergence of complex Langevin process as in the previous section. In order to see how an optimal kernel can be chosen we revisit the discussion originally found in refs. [54, 53].

The continuum action of this one-dimensional system is given by

$$\begin{aligned} S_M &= \int dt \left\{ \frac{1}{2} \left(\frac{\partial x(t)}{\partial t} \right)^2 - \frac{1}{2} m^2 x^2(t) \right\} \\ &= \int dt \left\{ \frac{1}{2} x(t) \Big(-\partial_t^2 - \frac{1}{2} m^2 \Big) x(t) \right\}, \end{aligned} \tag{6.22}$$

In quantum mechanics the fields ϕ are the position x and the coordinates, previously called x, are the time t. The corresponding complex Langevin equation reads

$$\frac{dx(t,\tau_L)}{d\tau_L} = -i\left(\partial_t^2 + m^2\right)x(t,\tau_L) + \eta(t,\tau_L). \tag{6.23}$$

In the absence of a regularization, this stochastic process is unstable and does not show convergence to the correct result. In analogy with the results for the simple model system in the previous section we will argue analytically that correct convergence can be achieved in this system via a kernel with the property $-\left(\partial_t^2+m^2\right)K(t-t') = i\delta(t-t')$. This kernel will render the drift term trivial, proportional to x itself and move all complex structure into the noise.

Following [54, 53] we solve eq. (6.23) analytically and obtain for the two-point correlator in Fourier space

$$\begin{aligned}&\langle x(\omega,\tau_L)x(\omega',\tau_L')\rangle\\ &\quad = \delta(\omega+\omega')\frac{i}{\omega^2-m^2}\left(e^{i(\omega^2-m^2)|\tau_L-\tau_L'|} - e^{i(\omega^2-m^2)(\tau_L+\tau_L')}\right).\end{aligned} \tag{6.24}$$

Obviously this expression does not have a well defined value in the late Langevin-time limit. Introducing an explicit regulator of the form $i\epsilon x(t)^2$ yields

$$S_M = \int dx\frac{1}{2}\left\{\partial_0\phi(x)\partial_0\phi(x) - (m^2 - i\epsilon)\phi^2(x)\right\} \tag{6.25}$$

and improves the situation, as now the stochastic process correctly converges to

$$\lim_{\tau_L\to\infty}\langle x(\omega,\tau_L)x(\omega',\tau_L)\rangle = \delta(\omega+\omega')\frac{i}{\omega^2-m^2+i\epsilon}. \tag{6.26}$$

A careful analysis of the associated Fokker-Planck equation in ref. [117] however reveals that the relaxation time towards the correct solution scales with $1/\epsilon$. I.e. carrying out a CL simulation based on a small regulator ϵ will lead to slow convergence. In addition one also needs to take the limit $\epsilon \to 0$ as well as $\Delta\tau_\mathrm{L} \to 0$, which may not commute [42].

Since the action of the harmonic oscillator in Fourier space decouples into a collection of non-interacting modes we may deploy a similar strategy for each mode as we considered in the simple model of the preceding section. I.e. we introduce a kernel, which moves the integration onto the single thimble for each mode.

$$\frac{\partial}{\partial \tau_{\rm L}} x(\omega,\tau_L) = i\tilde{K}(\omega)\,\frac{\delta S_M[x]}{\delta x(\omega)} + \sqrt{\tilde{K}(\omega)}\xi(\omega,\tau_L), \tag{6.27}$$

$$\langle \xi(\omega,\tau_L)\rangle = 0, \quad \langle \xi(\omega,\tau_L)\xi(\omega',\tau_{\rm L}')\rangle = 2\delta(\omega+\omega')\delta(\tau_L-\tau_L'). \tag{6.28}$$

This train of thought leads us to choose the following field-independent kernel, which had been explored in ref. [54] before

$$\tilde{K}(\omega) = \frac{iA(\omega)}{\omega^2-m^2+i\epsilon}, \quad K(t) = \int \frac{d\omega}{(2\pi)}\tilde{K}(\omega)e^{-i\omega t}, \tag{6.29}$$

where $A(\omega)$ is a real, positive and even function of ω. Thus for a constant $A(\omega)$, $\tilde{K}(\omega)$ is nothing but the propagator of the free theory in momentum space.

The corresponding correlation function is found to read

$$\begin{aligned}\lim_{\tau_{\rm L}\to\infty} \langle \phi(\omega,\tau_{\rm L})\phi(\omega',\tau_{\rm L})\rangle &= \delta(\omega+\omega')\frac{\tilde{K}(\omega)}{A(\omega)} \\ &= \delta(\omega+\omega')\frac{i}{\omega^2-m^2+i\epsilon}\end{aligned} \tag{6.30}$$

which is the correct result. The most important difference to simply introducing a regulator in the action however lies in the fact that now the relaxation time for each mode is proportional to $1/A(\omega)$ and not proportional to $1/\epsilon$ and no extrapolation in ϵ needs to be carried out. For completeness let us note the corresponding coordinate space complex Langevin process

$$\frac{\partial}{\partial \tau_{\rm L}} x(t,\tau_L) = i\int dt'\, K(t-t')\,\frac{\delta S_M[x]}{\delta x(t')} + \chi(t,\tau_L), \tag{6.31}$$

$$\chi(t,\tau_L) = \int d\omega e^{i\omega t}\sqrt{\tilde{K}(\omega)}\xi(\omega,\tau_L). \tag{6.32}$$

6.3.3 A kernel for real-time Langevin on the thermal SK contour

The analytic study of the one d.o.f. model and the harmonic oscillator have provided us with insight into how a kernel can be used to both satisfy the need for regularization of the path integral and achieve convergence to the correct solution of the associated complex Langevin equation in practice.

We will now construct the corresponding kernel for the harmonic oscillator at finite temperature, discretized on the Schwinger-Keldysh contour. Numerical simulations will confirm the effectiveness of the kernel in the non-interacting theory.

The Schwinger-Keldysh contour for a quantum system at finite temperature encompasses three branches. The forward branch along the conventional time axis reaches up to a real-time $t_{\rm max}$ and the degrees of freedom associated with it are labeled $x^+(t)$. The backward branch with $x^-(t)$ returns to the initial time t_0 in reverse and the Euclidean branch which houses $x^E(-i\tau)$ and extends along the negative imaginary time axis. The physical length of the imaginary time branch dictates the inverse temperature of the system. A sketch of our contour setup is shown in the left panel of fig. 6.2.

In the action of the system, the integration over time is rewritten into an integration over a common contour parameter γ. The d.o.f. on the different branches are then distinguished by the values of the contour parameter $x(\gamma)$ and we will drop the superscript in the remainder of the text.

As sketched in the right panel of fig. 6.2, we will refer to the equal- and unequal-time two-point correlation functions along the SK contour in the following, plotted against the contour parameter. The reader can identify the values along the forward and backward branch as being mirrored, connecting to the values on the Euclidean branch that show the expected periodicity of a thermal theory.

When discretizing the action for use in a numerical simulation the direc-

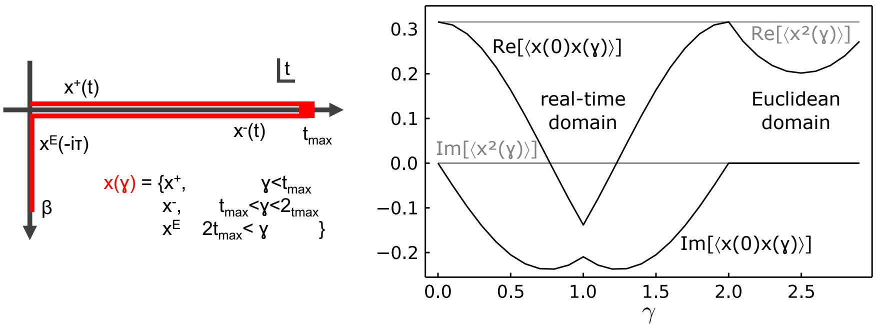

Figure 6.2: (left) Sketch of the Schwinger-Keldysh contour deployed in our study with forward $x^+(t)$ and backward $x^-(t)$ branches on the real-time axis, connected to an imaginary time branch $x^E(-i\tau)$. The contour parameter γ is used to address all branches in a unified manner. (right) Sketch of the visualization of our observables along the contour parameter γ for the example of $mt_{\text{max}} = 1$. The analytic solution of the real- and imaginary part of the equal time correlator $\langle x^2(\gamma)\rangle$ and the unequal time correlator $\langle x(0)x(\gamma)\rangle$ will be plotted in the real-time $\gamma < 2mt_{\text{max}}$ and subsequently Euclidean domain $\gamma > 2mt_{\text{max}}$.

tion of each branch of the SK contour is encoded in a contour spacing $a_i \in \mathbb{C}$. Computing the drift term for an arbitrary contour yields

$$i\frac{\partial S_M[x]}{\partial x_j} = \frac{i}{\frac{1}{2}\left(|a_j| + |a_{j-1}|\right)}\left\{\frac{x_j - x_{j-1}}{a_{j-1}} - \frac{x_{j+1} - x_j}{a_j} \right. \\ \left. - \frac{1}{2}\left[a_{j-1} + a_j\right]\frac{\partial V(x_j)}{\partial x_j}\right\}. \tag{6.33}$$

This expression simplifies if we use a constant magnitude step-size $|a_i| = |a|$, such that the prefactor in the above equation can be reduced to $\frac{i}{|a|}$. In that case we can go over to a convenient matrix-vector notation

$$i\nabla_x S_M[\boldsymbol{x}] = \frac{1}{|a|}\, iM\boldsymbol{x}, \tag{6.34}$$

where

$$M_{jk} = \begin{cases} \frac{1}{a_{j-1}} + \frac{1}{a_j} - \frac{1}{2}\left[a_{j-1} + a_j\right] m^2, & j = k \\ -\frac{1}{a_j}, & j = k - 1 \\ -\frac{1}{a_{j-1}}, & j = k + 1. \end{cases} \tag{6.35}$$

Based on the findings in the previous sections the form of the optimal discrete free theory kernel in coordinate space will be the inverse propagator

$$K = H\, H^T = iM^{-1}, \tag{6.36}$$

where H is the factorized kernel used in the noise term. The form of this kernel relies on the matrix M to be invertible, and M^{-1} to be factorizable, both of which holds. We obtain $H = \sqrt{iM^{-1}}$ by using the square root of the eigenvalues. Written in differential form with Wiener processes $d\boldsymbol{W}$, the corresponding Langevin equation reads

$$d\boldsymbol{x} = \frac{1}{|a|}\left(\frac{i}{M} iM\boldsymbol{x}\right) d\tau_L + \sqrt{2\frac{i}{M}} d\boldsymbol{W} = -\frac{1}{|a|}\boldsymbol{x}\, d\tau_L + \sqrt{2\frac{i}{M}} d\boldsymbol{W}, \tag{6.37}$$

which leaves us with a complex non-diagonal noise coefficient $\sqrt{\frac{2i}{M}}$ and a drift term pointing in the direction of $-\boldsymbol{x}$.

Let us demonstrate the effect of this kernel by carrying out a simulation for the following parameters. We discretize the canonical SK contour with $N_t = 50$ points on the forward and backward branch each and $N_\tau = 5$ points on the imaginary branch. Note that we do not introduce any tilt here. Choosing a mass parameter $m = 1$, the imaginary branch extends up to $m\tau_{\max} = 1$. As real-time extent, we choose $mt_{\max} = 10$. The value chosen here is arbitrary as the kernelled dynamics of the free theory are stable and converge for any real-time extent. The results of the simulation without a kernel are given in the top panel of fig. 6.3 and rely on the implicit Euler-Maruyama scheme to avoid the occurrence of runaway solutions. The results with our choice of kernel are shown in the bottom panel and were obtained using a simple forward-stepping Euler scheme at $\Delta\tau_{\mathrm{L}} = 10^{-3}$ without adaptive step size. In each case we generate

100 different trajectories, saving configurations at every $m\Delta_{\tau_L} = 0.1$ in Langevin time up to a total of $m\tau_L = 100$. Each panel in fig. 6.3 showcases four quantities plotted against the contour parameter γ. Their values for $0 < m\gamma < 10$ are obtained on the forward branch, those for $10 < m\gamma < 20$ on the backward branch and the small piece $20 < m\gamma < 21$ denotes the Euclidean time results. The real- and imaginary part of the equal time expectation value $\langle x^2(\gamma)\rangle$ are plotted as green and pink data points respectively. The real- and imaginary-part of the unequal time correlator $\langle x(0)x(\gamma)\rangle$ on the other hand are plotted as orange and blue data points. The analytically known values from solving the Schrödinger equation are underlaid as black solid lines.

The results without a kernel show both deviations from the correct result and exhibit relatively large uncertainties. The reason lies in the slow relaxation rate to the correct result due to the presence of an explicit regulator. Here the regulator is provided by our use of an implicit numerical scheme (Euler-Maruyama with implicitness parameter $\theta = 1.0$ and an adaptive step-size with a maximum step size of 10^{-3}), but could equally well be introduced by adding a small term $i\epsilon x^2$ to the system action. Using a stronger regulator, e.g., tilting the contour, would yield a shorter relaxation time, but any such explicit regulator distorts the results away from the actual $\epsilon \to 0$ physical solution. It is interesting to note that it is the equal time observable $\langle x^2\rangle$ that is performing the worst. We will see later on that this is the hardest observable to accurately reproduce.

For the bottom plot we use the free theory propagator kernel, eq. (6.36). This simulation now aligns excellently with the true solution for all the observables.

After applying the kernel, the problem is regularized and thus less stiff, and we can revert to using a fixed step-size explicit Euler-Maruyama scheme. The step-size here is $d\tau_{\mathrm{L}} = 10^{-3}$. The fact that we do not need to impose an explicit regulator term is important as in this case we only need to take the limit $\Delta\tau_{\mathrm{L}} \to 0$ to obtain a physical result, and do not need to extrapolate the regulator term to zero ($\epsilon \to 0$). This might be important considering the recent work in ref. [145], which shows that one

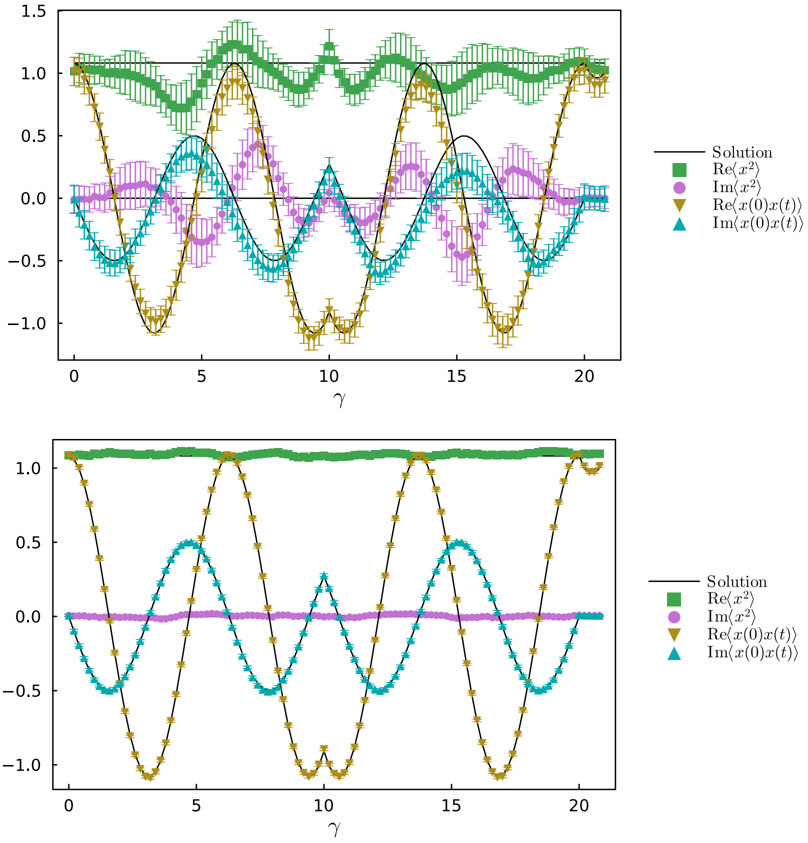

Figure 6.3: The result of a complex Langevin simulation of the real-time harmonic oscillator without a kernel (top) and with a kernel (bottom) for the observables $\langle x^2\rangle$, and the correlator $\langle x(0)x(t)\rangle$. Simulations are carried out with 100 trajectories and up to $m\tau_L = 100$ in Langevin time, saving at every $m\Delta_{\tau_L} = 0.1$ using the implicit scheme Euler-Maruyama with $\theta = 1.0$ with adaptive step-size with tolerance 10^{-3} (top) and the explicit scheme Euler-Maruyama with $\theta = 0$ with fixed step-size $\Delta\tau_\mathrm{L} = 10^{-3}$ (bottom). Values of the correlators from the solution of the Schrödinger equation are given as solid black lines.

encounters subtleties in taking the limit of the regulator $\epsilon \longrightarrow 0$.

6.3.4 A kernel for the quantum anharmonic oscillator

Not only did the kernel in the free theory change the convergence behavior of the complex Langevin simulation, it also removed the need of a regulator in the action. The obvious next step is to explore the interacting theory where the problem of convergence to the wrong solution is more severe. The potential term in the action is now given by

$$V(x) = \frac{1}{2}mx(t)^2 + \frac{\lambda}{4!}x(t)^4, \tag{6.38}$$

where we use $m = 1$ and $\lambda = 24$. This choice of parameters has been deployed in the past as benchmark for strongly-interacting real-time complex Langevin in refs. [135, 38]. As ref. [135] formulated the real-time dynamics on a tilted contour they found correct convergence up to $t^{\max} = 0.8$, while ref. [38] worked with an untilted contour and observed onset of incorrect convergence already above $t^{\max} = 0.5$. In the following we will remain with an untilted contour.

We find that using the free kernel of eq. (6.36) the convergence to the correct solution can be extended slightly to around $t^{\max} = 0.75$. If in addition we modify the free theory kernel by rescaling the contributions from the kinetic term with a common prefactor g and modify the mass term away from the free theory value m

$$M_{jk}(g, m_g) = \begin{cases} \frac{g}{a_{j-1}} + \frac{g}{a_j} - \frac{1}{2}\left[a_{j-1} + a_j\right] m_g^2, & j = k \\ -\frac{g}{a_j}, & j = k-1 \\ -\frac{g}{a_{j-1}}, & j = k+1. \end{cases} \tag{6.39}$$

convergence can be pushed up to $t_{\max} = 1.0$ by using the heuristically determined parameter values $g = 0.8$ and $m_g = 1.8$. The CL equation we simulate is given by

$$d\boldsymbol{x} = \frac{1}{|a|}\left[\frac{i}{M(g, m_g)} i \left(M(1, m)\boldsymbol{x} + \frac{\lambda}{3}x^2\boldsymbol{x}\right)\right] d\tau_L + \sqrt{\frac{2i}{M(g, m_g)}} d\boldsymbol{W}. \tag{6.40}$$

We carry out simulations, assigning $N_t = 10$ points to the forward and backward branches each and $N_\tau = 10$ points to the imaginary branch of the contour. Here we use the implicit Euler-Maruyama scheme with implicitness parameter $\theta = 0.5$. Even though we do not need a regulator in the presence of the kernel, the system retains some of its stiffness in contrast to the free theory. The use of an explicit scheme with e.g. adaptive step size is possible, however we find it more efficient to rely on an implicit scheme, as it allows the use of much larger Langevin step sizes.

The results of two simulations with a maximum real-time extent $t_{\max} = 1.0$ are shown in fig. 6.4. One is carried out without a kernel and using an implicit scheme (top) and the other in the presence of a kernel based on the parameters $g = 0.8$ and $m_g = 1.8$ (bottom). The graphs show the real and imaginary part of the equal time $\langle x(t)x(t)\rangle$ (green and pink data points) and unequal time correlator $\langle x(0)x(t)\rangle$ (orange and blue datapoints) plotted against the contour parameter γ. For $0 < m\gamma < 1$ and $1 < m\gamma < 2$ it refers to the forward and backward branch of the contour and for $2 < m\gamma < 2.9$ denotes the imaginary time branch.

The top panel indicates that naive complex Langevin fails to converge to the correct solution at this real-time extent of $mt_{\max} = 1$. It is interesting to point out the failure of CL at two specific points along the contour, the first one is the starting point at $\gamma = 0$, which is connected by periodic boundary condition to the Euclidean path. Then at the turning point of the contour at maximum real-time extent, corresponding to $m\gamma = 1$, the real part of the $\langle x^2\rangle$ observable lies significantly away from the true solution. This points seems to be most affected by the convergence problem of the CLE.

In the lower panel, the simulation in the presence of the modified free theory kernel is presented. The outcome of the kernelled complex Langevin evolution is very close to the correct solution and shows only small statistical uncertainties. Note however that especially the observable $\langle x^2\rangle$ still shows some deviation from the true result beyond the statistical error bars

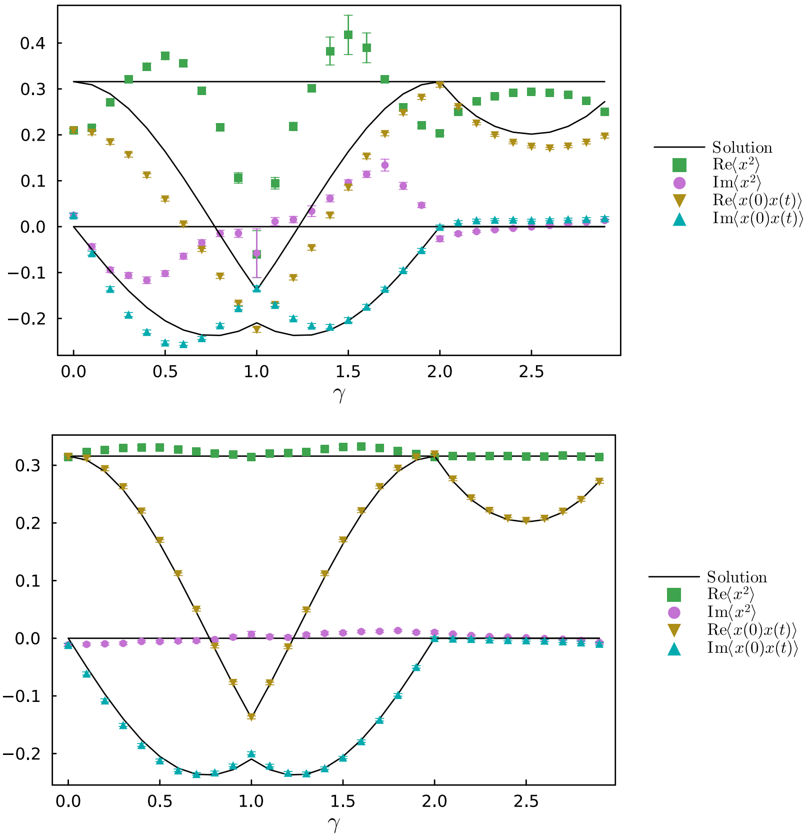

Figure 6.4: Anharmonic oscillator at $m = 1$ and $\lambda = 24$ up to $t_{\max} = 1$ in real-time without (top) and in the presence (bottom) of the heuristic free theory kernel with $g = 0.8$ and $m_g = 1.8$ from eq. (6.40). Both simulations are carried out generating 100 trajectories simulated up to $m\tau_L = 100$ in Langevin time, saving configurations at every $m\Delta\tau_L = 0.01$. We deploy the Euler-Maruyama solver with $\theta = 1.0$ without kernel (top), and $\theta = 0.5$ with kernel (bottom). Values of the correlators from the solution of the Schrödinger equation are given as solid black lines.

indicating that exact correct convergence has not yet been achieved[1]

The above results are promising, as they indicate that in principle the convergence problem of real-time complex Langevin can be attacked by use of a kernel. At the same time explicitly constructed kernels, such as the modified free theory kernel are limited in the range of real-time extent in which they are effective. The question at hand is how to systematically construct kernels that will restore convergence at even larger real-time extent.

6.4 Learning optimal kernels

In this section, we introduce our novel strategy to systematically construct kernels to improve the convergence of real-time complex Langevin. Our goal is to overcome the limitations of explicitly parametrized kernels, such as the one of eq. (6.39). While optimal parameter values g and m_g were found for this kernel, they only achieved correct convergence for a limited $mt_{\max} \leq 1$. Most importantly it is not clear how to systematically modify that kernel for realizing convergence at larger real-time extent.

Instead we set out to use a generic parametrization of the kernel. We propose to use an expansion in a set of complete basis functions of the dynamical d.o.f. In this study, as a proof of principle, we will restrict ourselves to a field-independent kernel, which can be understood as the first term in an expansion in powers of the field. This field-independent kernel for the quantum anharmonic oscillator on the Schwinger-Keldysh contour will take the form of a τ_L independent matrix K with $(2N_t + N_\tau)^2$ entries, multiplying the $2N_t$ d.o.f. on the forward and backward contour and the N_τ ones on the imaginary time branch. It is the values of these matrix entries that we set out to tune in order to achieve optimal convergence.

And even though simple model systems indicate that a field-dependent

[1] This behavior may be understood in terms of boundary terms. The kernel manages to significantly reduce the magnitude of boundary terms for $\langle x^2 \rangle$ where it differs from the true solution the boundary terms, while small, are not exactly zero.

kernel is needed to achieve correct convergence in case of strong complex drift terms, we find that an optimal field-independent kernel can already extend the range of convergence of the anharmonic oscillator out to $mt_{\text{max}} = 1.5$, three times larger than the previous record set for CL in ref. [106].

In order to obtain kernel values that restore correct convergence, we formulate an optimization problem based on a cost functional, which incorporates prior knowledge about the system of interest. Taking advantage of modern programming techniques that allow us to compute the dependence of a full complex-Langevin simulation on the entries of the kernel we propose to iteratively learn the optimal kernel. The fact that we incorporate prior information into the simulation opens a novel path to beat the notorious sign problem, i.e. for the first time complex Langevin can be amended by system-specific information in order to restore correct convergence.

6.4.1 The optimization functional

In order to guarantee that a complex Langevin simulation converges to the true solution we must fulfill the correctness criteria of [57]. First we must ensure the absence of boundary terms and second that the late-time distribution of the complex Fokker-Planck equation is indeed $\exp[\mathrm{iS_M}]$. Constructing a loss function for both criteria however is only feasible for very low dimensional models, as it entails calculating the eigenvalues and eigenvectors of the complex Fokker-Planck operator, which is prohibitively expensive already for the anharmonic oscillator discussed here.

Instead we will retain only the first ingredient of the correctness criteria, the absence of boundary terms and use other prior information in order to guide the kernelled complex FP equation to the correct stationary solution. The boundary terms can be calculated via the expectation value $\langle L_c O \rangle_Y$ where L_c is the Langevin operator and O refers to any observable (for a detailed discussion see e.g. [68]). In section 5.1 we demonstrate that the correctness criterion still holds with a kernel and how to calculate these boundary terms.

Besides the boundary terms, we often possess additional relevant prior information about the system at hand. We can e.g. compute correlation functions in Euclidean time using conventional Monte-Carlo methods. In addition, we know that in thermal equilibrium the correlation functions on the forward and backward branch are related due to the KMS relation. In order to exploit this prior information it is vital for the CL equations to be formulated on the canonical SK contour, whose real-time branches lie parallel to each other and connect to the Euclidean branch at the origin. In a tilted contour setup, access to the Euclidean branch is limited and the comparison of the values on the forward and backward branch is much more involved. In addition, symmetries provide powerful constraints to the simulation, as e.g. time-translation invariance in a thermal system renders local observables such as $\langle x^n(\gamma)\rangle$ constant along the full contour.

We quantify the distance of the simulated result from the behavior dictated by prior knowledge via a loss function L^{prior}. The comparison is carried out on the level of expectation values of observables, where apriori known values from conventional Euclidean simulations are referred to as $\langle O\rangle_{\text{MC}}$ and those from the complex Langevin simulation in the presence of a kernel by $\langle O\rangle_K$.

In principle one can distinguish between four categories of prior knowledge:

- Euclidean correlators (L^{eucl}), which are accessible via conventional Monte-Carlo simulations:

  $$L^{\text{eucl}} = \sum_O \int d\tau \Big| \langle O(\tau)\rangle_K - \langle O(\tau)\rangle_{\text{MC}} \Big|^2 / \sigma^2_{\langle O(\tau)\rangle_K}$$

  $\tau \in$ imaginary time

- Model symmetries (L^{sym}), which exploit that the expectation values of observables O must remain invariant under a symmetry transformation T_ξ governed by a continuous (or discrete) parameter ξ:

$$L^{\text{sym}} = \sum_O \int d\xi |\langle T_\xi O\rangle_K - \langle O\rangle_K|^2/\sigma^2_{\langle O(\tau)\rangle_K}$$

- Contour symmetries (L^{rt}), which arise predominantly in systems in thermal equilibrium:

$$L^C = \sum_O \int d\gamma \Big|\langle O^{C^+}(\gamma)\rangle_K - F[\langle O^{C^-}(\gamma)\rangle_K]\Big|^2/\sigma^2_{\langle O(\tau)\rangle_K}$$

  with F analytically known

- Boundary terms (L^{BT}), which can be explicitly computed from the outcome of the kernelled Langevin simulation:

$$L^{\text{BT}} = \sum_O \Big|\langle L_c(K)O\rangle_K\Big|^2/\sigma^2_{\langle O(\tau)\rangle_K}$$

In practice one wishes to combine as many of these different contributions as possible. To this end they must be added as dimensionless quantities. This is why each of the terms above is normalized by the variance of the complex Langevin simulation. In order for the combined functional to provide a meaningful distinction of the success of convergence (also in the case of e.g. the free theory as shown in fig. 6.3) we propose to introduce an overall normalization for the combined prior functional

$$L^{\text{prior}} = \mathcal{N}_{\text{tot}}\Big(L^{\text{eucl}} + L^{\text{sym}} + L^C + L^{\text{BT}}\Big). \tag{6.41}$$

There is an element of arbitrariness in what overall normalization to choose, and we find that the best distinction between wrong and correct convergence is achieved if one uses the relative error of the most difficult observable to reproduce. In case of the systems studied here this amounts to the relative error of the equal time correlator obtained in the complex Langevin simulation with respect to the correct known value from Euclidean simulations $\mathcal{N}_{\text{tot}} = \max_\gamma\{\sigma_{\langle x^2\rangle_K}(\gamma)/\langle x^2\rangle_{\text{MC}}(\gamma)\}$.

We carry the subscript K in the expectation values above, in order to emphasize that the loss functional depends implicitly on the choice of kernel used in the underlying complex Langevin simulation. The number of observables O contained in the cost functional is not specified here and depends on the problem at hand. In practice, we find that often including the apriori known Euclidean one- and two-point functions already allow us to reliably distinguish between correct and incorrect convergence.

In the next section, we will discuss both fully general numerical strategies to locate the minimum of the optimization functional, as well as an approximate low-cost approach, which we have deployed in the present study.

6.4.2 Optimization strategies

General approach

The task at hand is to find the critical point of a cost functional that is comprised of a subset of the contributions listed in the previous section, i.e. of $L^{\rm eucl}$, $L^{\rm sym}$, L^{C} or $L^{\rm BT}$. Generically each contribution can be written as the expectation value of a known function G, depending on the dynamical degrees of freedom x and the kernel K, i.e. $L^{\rm prior}[K] = |\langle G[x,K]\rangle_K|^2$. In order to make the dependence of the expectation value on the kernel explicit we consider the d.o.f. within the simulation to explicitly depend on K as $x(K)$. This allows us to remove the subscript K from the expectation value so that $L^{\rm prior}[K] = |\langle G[x(K),K]\rangle|^2$.

Let us characterize the kernel via a set of variables κ. We emphasize that this does not limit the general nature of the approach, as κ may refer to the prefactors of a general expansion of the kernel in a complete set of basis functions.

To efficiently locate its critical point we deploy standard numerical optimization algorithms, which utilize the information of the gradient of the functional with respect to the parameters of the kernel. The computational challenge lies in determining the gradient robustly. In the continuum, the gradient of the loss reads

$$\nabla_K L^{\text{prior}}[K] = 2\frac{\langle G[x(K),K]\rangle}{|\langle G[x(K),K]\rangle|}\langle\nabla_K G[x(K),K]\rangle \tag{6.42}$$

$$= 2\frac{\langle G[x(K),K]\rangle}{|\langle G[x(K),K]\rangle|}\{\langle\nabla_x G[x(K),K]\cdot\nabla_K \mathbf{x}\rangle + \langle\nabla_K G[x,K]\rangle\} \tag{6.43}$$

In order to evaluate eq. (6.43) we need to compute the change in the field $x(K)$, which depends on the kernel. This requires taking the gradient of the CL simulation itself. While a demanding task, dedicated methods to evaluate such gradients have been developed, which underpin the recent progress in the machine learning community. They are known as differential programming techniques (for an in-depth review see e.g. ref. [30]).

As a first option, we considered using direct auto-differentiation[2] on the full loss function, as we are dealing with the standard setting of estimating the gradient of a highly dimensional functional whose output is a single number. For small systems with a number of degrees of freedom $\mathcal{O}(10)$, forward-auto-differentiation is feasible as it requires multiple runs of the full CL simulation. As the number of independent d.o.f. grows, backward-auto-differentiation offers us to reduce the number of necessary simulation runs, trading computational cost for increased memory demands to store intermediate results of the chain rule it computes internally. We find that already for the quantum anharmonic oscillator this direct computation of the gradient is too costly and thus not practical.

A more advanced approach, which promises to avoid the cost and memory limitations of direct auto-differentiation are so-called sensitivity analysis methods, such as e.g. adjoint methods for stochastic differential equations. A detailed discussion of these methods is beyond the scope of this paper

[2] Auto-differentiation is a method to compute derivatives to machine precision on digital computers based on an efficient use of the chain rule, exploiting elementary arithmetic operations in the form of dual variables (see e.g.[146]). We have used the Julia library *Zygote.jl*[74] and *ForwardDiff.jl* for computing gradients.

and the interested reader is referred to refs. [147, 148, 83] for further details.

We find that for the specific case of real-time complex Langevin, these methods in their standard implementation, as provided e.g. in [83] are challenged in estimating the gradient robustly. We believe that the difficulty here lies in the stiffness of the underlying stochastic differential equation. One possible way out is to deploy sensitivity analysis methods specifically developed for chaotic systems, such as *Least Square Shadowing algorithms*, discussed e.g. in Refs. [149, 150]. While these methods at this point are still too slow to be deployed in CL simulations, the rapid development in this field over the past years is promising.

Our survey of differential programming techniques indicates that while possible in principle, the optimization of the loss functional $L(K)$ is currently plagued by issues of computational efficiency. We believe that implementing by hand the adjoint method for the real-time complex Langevin systems considered here will offer a significant improvement in speed and robustness compared to the generic implementations on the market. This line of work goes beyond the scope of this manuscript and will be considered in an upcoming study.

To make headway in spite of these methods limitations we in the following propose an approach to compute an approximate low-cost gradient, which in practice allows us to significantly reduce the values of the optimization functional.

A low cost update from an heuristic gradient

Our goal is to compute a gradient, which allows us to approximately minimize the cost functional $L^{\text{prior}}[K]$ without the need to take derivatives of the CL simulation. The approach we propose here relies on using a different optimization functional, whose form is motivated by the need to avoid boundary terms. While updating the values of the kernel according to a heuristic gradient obtained from this alternative functional, we will monitor the values of the true optimization functional, selecting the kernel which achieves the lowest value of $L^{\text{prior}}[K]$.

We saw that the optimal kernel for the free theory reduces the drift term to a term in the direction of $-x$. This drift term points towards the origin. In this spirit we construct a functional that penalizes drift away from the origin.

The starting point is the following expression, where we define $D = -iK\partial S_M/\partial x$ as the drift term modified by the kernel

$$D(x,K)\cdot(-x) = ||D(x,K)||||x||\cos\theta. \tag{6.44}$$

Here cosθ denotes the angle between the drift and the optimal direction. As we wish to align the drift and $-x$, our optimization problem becomes finding a kernel K such that

$$\min_K \{D(x,K)\cdot(-x) - ||D(x,K)||\ ||x||\}. \tag{6.45}$$

We can write down different loss functionals which encode this minimum

$$\begin{aligned} L_D &= \left\langle \left| D(x)\cdot(-x) - ||D(x)||\ ||x|| \right|^{\xi} \right\rangle \\ &= \frac{1}{T}\int \left| D(x(\tau_{\text{L}}))\cdot(-x(\tau_{\text{L}})) - ||D(x)||\ ||x|| \right|^{\xi} \end{aligned} \tag{6.46}$$

The choice of ξ determines how steep the gradients on the functional are and we find that in practice a value between $1 < \xi < 2$ leads to most efficient minimization, when L_D is used to construct the heuristic gradient we describe below.

Note that turning the drift towards the origin differs from the strategy employed by dynamic stabilization. The scalar counterpart to minimizing the unitarity norm is driving the values of the complexified x towards the real axis. In addition, in dynamical stabilization a non-holomorphic term is added to the action. Here the CL equation is modified only by a kernel, which still leads to a holomorphic complex Langevin equation that leaves the correctness criteria intact.

The exact gradient of the functional L_D of eq. (6.46) also contains the costly derivatives over the whole CL simulation. However we find that in practice for values $1 \leq \xi \leq 2$ in L_D these contributions can be neglected. We believe the reason to lie in the fact that L_D consists of the difference between two terms that contain the same powers of x. I.e. we find that carrying out the optimization using only the explicit dependence of L_D on the kernel K, which is computed using standard auto-differentiation. The approximate gradient allows us to locate kernel values, which significantly reduce the values of the true optimization functional $L^{\text{prior}}[K]$. The kernels identified in this way in turn achieves correct convergence on contours with larger real-time extent than previously possible.[3].

The full optimization scheme can be summarized as follows:

1. Initialize the kernel parameters yielding the initial kernel K_1
2. Carry out the CL simulation with K_1 and save the configurations $\{x_j\}_1$, where the subscript indicates that this is the first iteration
3. Compute the values of the loss functions L_D and $L^{\text{prior}}[K_1]$
4. Compute the gradients of the loss function $L_D(\{x_j\}_1, K_1)$ with respect to the kernel parameters using auto-differentiation
5. Update the kernel parameters using one step of the ADAM optimization scheme
6. Rerun the CL simulation with the new kernel K_{i+1} and save a new set of configurations $\{x_j\}_{i+1}$
7. Loop over step 3 - 6 for N steps, or until L_D have reached a minimum and then select the kernel parameters with the smallest $L^{\text{prior}}[K_i]$

We will demonstrate the efficiency of the proposed optimization based on the heuristic gradient in the next section, where we learn optimal kernels for the quantum harmonic and anharmonic oscillator on the thermal Schwinger-Keldysh real-time contour.

[3] Note that disregarding the costly terms in the gradient of the true cost functional $L^{\text{prior}}[K]$ introduced in section 6.4.1 did not lead to a viable minimization of its values.

6.4.3 Learning optimal kernels for the thermal harmonic oscillator

To put the strategy laid out in the previous section to a test we set out here to learn a field-independent kernel for the quantum harmonic oscillator on the canonical Schwinger-Keldysh contour at finite temperature. In section 6.3.3 we had identified one kernel by hand, which actually minimizes the low-cost functional eq. (6.46). We will compare it to the learned kernel at the end of this section.

We simulate on the canonical Schwinger-Keldysh contour with real-time extent $mt_{\max} = 10$ and an imaginary time branch of length $m\beta = 1$. The contour will be discretized with steps of equal magnitude $|a_i| = |a|$ such that $N_t = 25$ points are assigned to the forward and backward branch each and $N_\tau = 5$ to the imaginary time axis.

The field-independent kernel therefore is a complex 55×55 matrix, which we parametrize via two real matrices A and B such that $K = e^{A+iB}$. This choice is arbitrary and is based on the observation that the minimization procedure is more robust for the exponentiated matrices than when using $A + iB$ directly. The kernel is initialized to unity before the start of the optimization by setting all elements of A and B to zero. The optimization itself, as discussed in the previous section, is carried out using the approximate gradient following from L_D with a choice of $\xi = 2$.

One needs to choose the actual cost functional L^{prior} based on prior knowledge through which to monitor the optimization success. We decide to include the known values of the Euclidean two-point correlator $\langle x(0)x(-i\tau)\rangle$ and exploit the knowledge about the symmetries of the system, which require that $\langle x\rangle = \langle x^3\rangle = 0$, as well as $\langle x^2(\gamma)\rangle = \langle x^2(0)\rangle$.

This leads us to the following functional

$$\begin{aligned}
&L^{\text{prior}} = \mathcal{N}_{\text{tot}}\Big(\qquad (6.47)\\
&\sum_{i\in SK} \Big\{ |\langle x(\gamma_i)\rangle_K|^2/\sigma^2_{x_{\text{K}}} + |\langle x^3(\gamma_i)\rangle_K|^2/\sigma^2_{x^3_{\text{K}}}\\
+ \quad & |\langle x^2(0)\rangle_{\text{MC}} - \langle x^2(\gamma_i)\rangle_K|^2/\sigma^2_{x^2_{\text{K}}} \Big\}\\
+ &\sum_{i\in\text{Eucl.}} |\langle x(0)x(\tau_i)\rangle_{\text{MC}} - \langle x(0)x(\tau_i)\rangle_K|^2/\sigma^2_{xx_{\text{K}}} \Big)
\end{aligned}$$

where the first sum runs over all points of the discretized Schwinger-Keldysh contour, while the second sum only contains the correlator on the Euclidean branch. As discussed before, the overall normalization is based on the uncertainty of the equal-time x^2 correlator.

Since we start from a trivial kernel, we must make sure that our simulation algorithm provides a regularization and remains stable even for stiff dynamics. Therefore we solve the complex Langevin stochastic differential equation using the *Implicit Euler-Maruyama scheme* with implicitness parameter $\theta = 1.0$ and adaptive step-size. For every update of the CL configurations we simulate 30 different trajectories up to a Langevin time of $m\tau_L = 30$, with a thermalization regime of $m\tau_L = 5$ in Langevin time before we start collecting the configurations at every $m\Delta_{\tau_L} = 0.05$ in Langevin time. To calculate the expectation values, we compute sub-averages from the saved configurations in each trajectory separately. The final mean and variance are then estimated from the results of the different trajectories.

The iterative optimization of the kernel values, based on the low-cost functional and its approximate gradient, is performed using the ADAM (Adaptive Moment) optimizer with a learning rate of 0.001. This is an improved *gradient descent* optimizer, which combines gradient descent momentum and an adaptive learning rate.

Since we know that the complex Langevin simulation will be the slowest part of the optimization scheme we will only run the full CL simulation for every five optimization steps. For this simple model it would not be a computation time problem to update the expectation values in L_D after every kernel update, but for realistic models in higher dimensions this might be too expensive. As the distributions of the observables should be similar for a small change in the kernel we indeed find that not updating the CL configurations at every update steps still allows us to obtain a good estimate of the heuristic gradient.

Starting with the unit kernel, the functionals $L_D = 6.58 \times 10^{11}$ and $L^{\mathrm{prior}} = 107$ show appreciable deviation from zero. After 32 steps of the ADAM optimizer we manage to find values of K which reduce the value of $L^{\mathrm{prior}} = 26.5$ indicating that the apriori known information has been well recovered.

The results for the simulation with the optimal learned kernel are plotted in fig. 6.5, based on 100 trajectories (top) and 400 trajectories (bottom) each of which progresses up to $m\tau_L = 100$ in Langevin time . The x-axis refers to the contour parameter γ, such that at $m\gamma = 10$ we are at the turning point of the real-time branch of the Schwinger-Keldysh contour and at $m\gamma = 20$ the contour has returned to the origin, before extending along the imaginary axis to $mt = -i$. We plot the real- and imaginary part of the unequal time correlator $\langle x(0)x(\gamma)\rangle$ as orange and blue data points, while the real- and imaginary part of the equal time expectation value $\langle x^2(\gamma)\rangle$ are given in green and pink respectively. The analytically known values from solving the Schrödinger equation are underlaid as black solid lines.

How has the learned kernel improved the outcome? When comparing to a simulation without kernel in the top panel of fig. 6.3 we see that using the same amount of numerical resources (i.e. 100 trajectories at $m\tau_L = 100$) the learned kernel has reduced the resulting errorbars significantly. On the other hand in the top panel of fig. 6.5 residual oscillations in $\langle x^2\rangle$ seem to persist. One may ask whether these indicate incorrect convergence, which is why we provide in the lower panel the result after including 400 trajectories at the same Langevin time extent. One can see that not only the errorbars further reduce but also that the oscillatory artifacts have diminished. The improvement amounts to another factor of two in terms

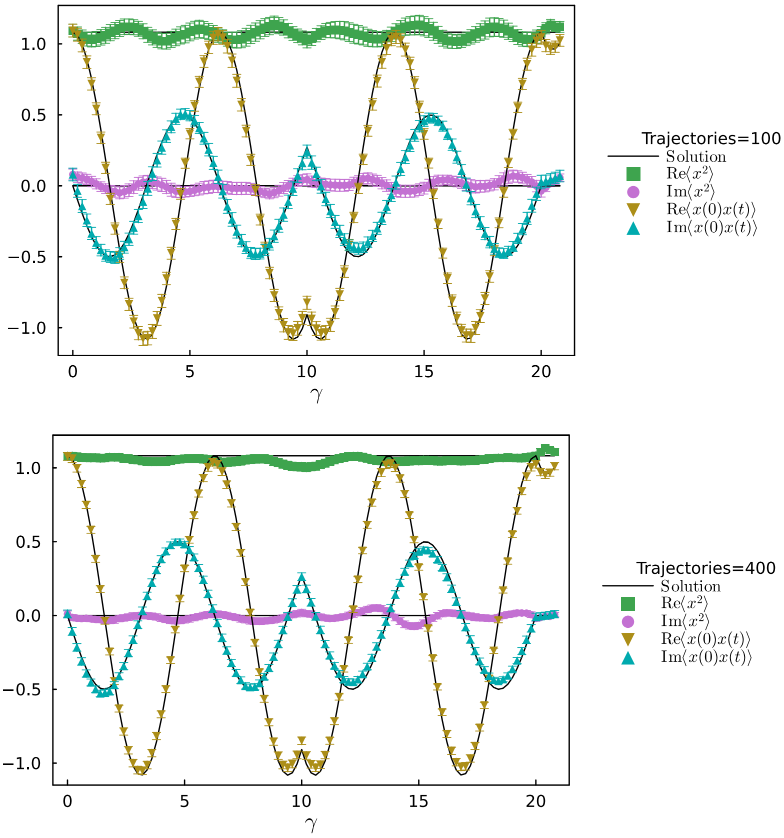

Figure 6.5: Harmonic oscillator in the presence of our learned optimal kernel, based on the heuristic gradient from the loss function in eq. (6.46). We simulate on the SK contour with $mt_{\mathrm{max}} = 10$ in real-time and choose $m = 1$. Simulating up to $m\tau_L = 100$ in Langevin time, we combine samples from 100 (top) and 400 (bottom) different trajectories. Note that improved statistics diminishes the residual oscillatory artifacts in $\langle x^2 \rangle$. Values of the correlators from the solution of the Schrödinger equation are given as solid black lines.

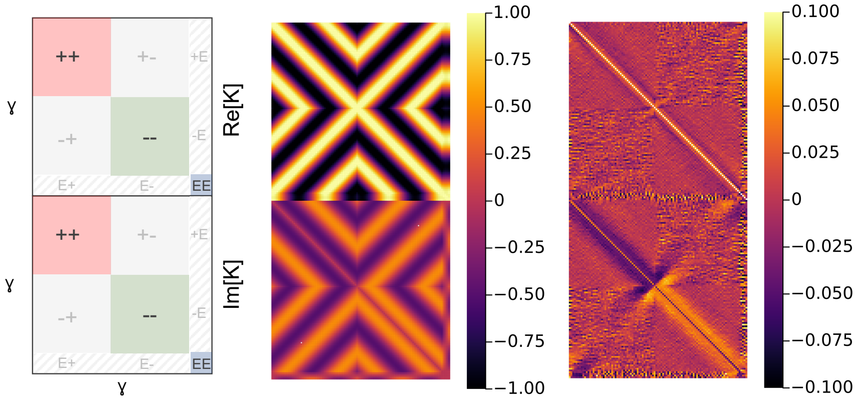

Figure 6.6: (left) An explanatory sketch of our visualization of the complex kernel used in the simulations. The top major panel denotes the values of the real part and the lower major panel those of the imaginary part of the kernel. Inside each panel the values of the kernel are ordered along the contour parameter, indicating which parts of the kernel couple which range on the contour. (center) The free theory kernel in eq. (6.36), constructed explicitly in the previous section for $mt_{\max} = 10$. The repeating pattern indicates an oscillatory behavior in coordinate space arising from the fact that this kernel is just the propagator of the free theory. (right) In the optimal learned kernel based on the low-cost update we have subtracted the unit matrix from the real-part to avoid it dominating the other structures. We find that the learned kernel exhibits some of the structure of the manually constructed kernel but in general has a more simple form, which nevertheless manages to achieve correct convergence of the complex Langevin dynamics.

of L^{prior} from the value $L^{\text{prior}} = 26.5$ in the top panel to $L^{\text{prior}} = 13.4$ in the lower panel. We emphasize that we *did not* use the analytically known solution of the system for the optimization procedure.

Let us inspect the learned kernel and compare it to the free theory propagator kernel of eq. (6.36). In fig. 6.6 we visualize the structures of the kernel by plotting a heat-map of the matrix entries of the complex matrix kernel. The right sketch shows how the matrix is structured, where the top panel refers to the real part and the lower panel to the imaginary part. The entries of the matrices are laid out corresponding to the contour parameter γ. The smaller regions inside the two panels indicate how the kernel mixes points along the time contour. The ++ corresponds to the mixing of the forward branch of the contour, while +− mixes the forward and backward branch time points. There exists also a small strip involving the Euclidean points, mixing with the real-time points ($E+$ and $E-$), as well as a small corner (EE) mixing within the Euclidean points.

The different regions shown in the sketch can easily be recognized in the two kernel structure plots. Note that we have subtracted the unit matrix from the real-part of the optimized kernel to more clearly expose off-diagonal structures, if present. The manually constructed free theory propagator kernel (middle), as expected from being the inverse free propagator, exhibits an oscillatory pattern. It leads to a significant coupling between the forward and backward time points, due to an anti-diagonal structure in the real and imaginary parts, forming an oscillatory cross pattern. This anti-diagonal behavior is much less pronounced in the optimized kernel (left). In its real-part it mainly exhibits a diagonal which is not as wide as in the manually constructed kernel. There is however a small negative structure present, an off-diagonal band, similar to the black part in the middle panel.

For the imaginary part, the patterns close to the diagonal are similar between the manually constructed kernel and the optimal learned kernel. Both possess a diagonal close to zero and a broad sub/super-diagonal that switches sign at the turning points between the ++ and −− part of the time contour. We also see that the anti-diagonal structure is similar for a very short part in the +− and −+ quadrants in the imaginary panel. The rest of the +− and −+ quadrant seems to contain noise.

While some similarities exist between the explicit kernel and the optimal learned kernel, it appears that correct convergence requires some non-trivial structure in the imaginary part of K. The learned kernel achieves correct convergence with much less structure than the manually constructed one.

6.4.4 Learning optimal kernels for the strongly coupled anharmonic oscillator

After successfully testing the learning strategy for a field-independent kernel in the free theory in the previous section, we are now ready to attack the central task of this study: learning an optimal kernel for a strongly coupled quantum system in order to extend the correct convergence of the corresponding real-time complex Langevin simulation.

We deploy the same parameter set as before with $m = 1$ and $\lambda = 24$. In section 6.3.4 we showed that for a real-time extent of $mt_{\text{max}} = 1$ an explicit kernel based on insight from the free theory can be constructed, which allows us to restore correct convergence within statistical uncertainties (see fig. 6.4).

Here we set out to learn an optimal kernel based only on the combination of our low-cost functional and prior knowledge of the Euclidean two-point functions and time-translation invariance of the thermal system. Since we restrict ourselves to a field-independent kernel we expect that our approach will be able to improve on the manually constructed kernel but will itself be limited in the maximum real-time extent up to which correct convergence can be achieved.

As testing ground we selected three different real-time extents, $mt_{\text{max}} = 1$, $mt_{\text{max}} = 1.5$ and $mt_{\text{max}} = 2$, all of which show convergence to the wrong solution when performing naive complex Langevin evolution.

We discretize the real-time contour with a common magnitude of the lattice spacing $|a_i| = |a|$. I.e. depending on the maximum real-time extent the number of grid points changes. E.g. in case of $mt^{\text{max}} = 2$ we use $N_t = 20$ on the forward and backward part of the real-time contour each, and $N_\tau = 10$ for the imaginary part of the contour. Due to the stiffness of the complex Langevin equations in the interacting case, all CL simulations are performed with the *Euler-Maruyama* scheme with $\theta = 0.6$ and adaptive step-size. We simulate 40 different trajectories up to $m\tau_L = 40$ in Langevin time, computing observables at every $m\Delta\tau_L = 0.02$ step.

The setup for learning the optimal kernel is very similar to that in the previous section. The kernel parametrization is given by $K = e^{A+iB}$, where A and B are real matrices. We search for the critical point of the true loss function L^{prior} (see eq. (6.47)) via the heuristic gradient obtained from the loss function L_D of eq. (6.46). We find that minimization proceeds efficiently, when choosing the parameter $\xi = 1$ in L_D. The optimal kernel is chosen according to the lowest values observed in L^{prior}.

We find that for a trivial unit kernel where complex Langevin fails, the cost functional L^{prior} based on prior information indicates values of $L^{\text{prior}}_{mt_{\max}=1} = 942$, $L^{\text{prior}}_{mt_{\max}=1.5} = 597320$ and $L^{\text{prior}}_{mt_{\max}=2} = 12923$. The left column of fig. 6.7 shows from top to bottom the rows correspond to results of the naive CL simulation for $mt_{\max} = 1$, $mt_{\max} = 1.5$ and $mt_{\max} = 2$ respectively. As in previous comparison plots the real- and imaginary part of the unequal time correlation function $\langle x(0)x(\gamma)\rangle$ is given by orange and blue data points, while the real- and imaginary part of the equal time expectation value $\langle x^2(\gamma)\rangle$ is represented by the green and pink symbols respectively. The analytically known values from solving the Schrödinger equation are underlaid as black solid lines.

The results of real-time CL in the presence of the optimal learned kernel for the anharmonic oscillator are shown in the right column of fig. 6.7. For $mt_{\max} = 1$ we achieve to lower the value of $L^{\text{prior}}_{mt_{\max}=1} = 14.3$. At this low value all the correlation functions plotted, agree with the true solution within uncertainties. Note that we manage to restore correct convergence for the unequal time correlation function on the real-time axis, even though no prior information about these points was provided in L^{prior} nor L_D. In contrast to the use of the modified free theory kernel, we see here that $\langle x^2\rangle$ does not show a systematic shift on the real-time branches anymore.

We continue to the second row, where, via an optimal learned kernel, we achieve extending the correctness of CL into a region inaccessible to the modified free theory kernel at $mt^{\max} = 1.5$. The value of the functional encoding our prior knowledge has reduced to $L^{\text{prior}}_{mt_{\max}=1.5} = 48.1$. We find that the unequal time correlation function values are

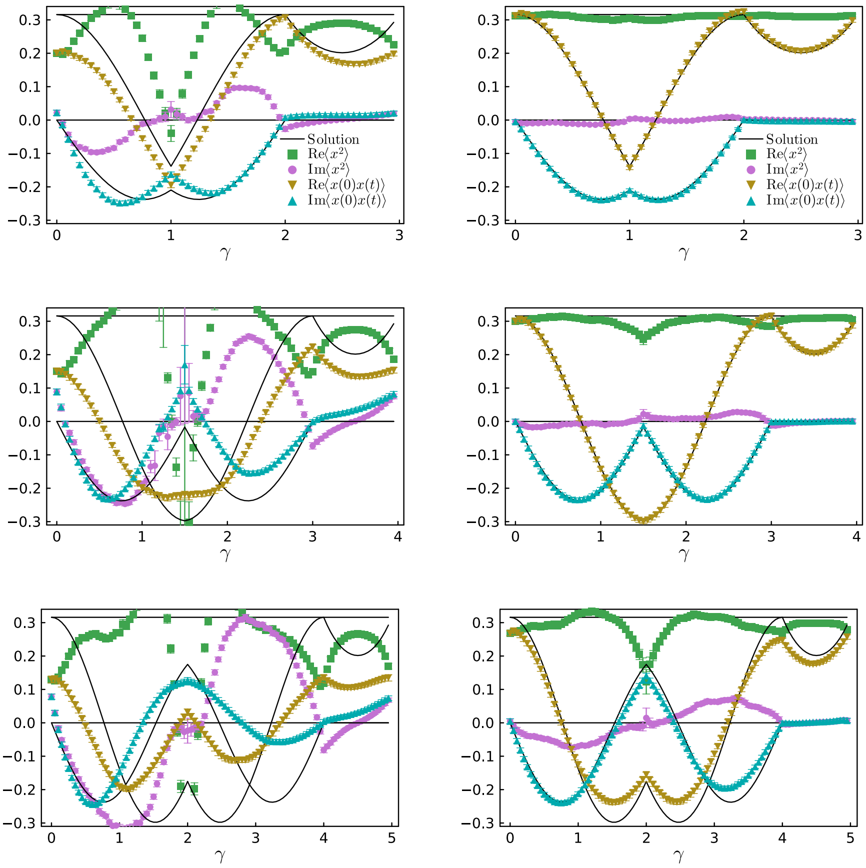

Figure 6.7: Complex Langevin simulation of the strongly coupled anharmonic oscillator on the thermal Schwinger-Keldysh contour in the absence (left) of a kernel and in the presence of the optimal learned field-independent kernel (right). The top row corresponds to results from a contour with real-time extent $mt_{\text{max}} = 1$, while the center row shows results for $mt_{\text{max}} = 1.5$ and the bottom row for $mt_{\text{max}} = 2$. Values of the correlators from the solution of the Schrödinger equation are given as solid black lines.

reproduced excellently, while the real- and imaginary part of $\langle x^2 \rangle$ show residual deviations from the correct solution around those points along the SK contour, where the path exhibits sharp turns, i.e. at the end point $\gamma = t_{\max}$ and the point where the real-time and Euclidean branch meet $\gamma = 2t_{\max}$.

The results shown in the third row clearly spell out the limitation of the field-independent kernel we deploy in this study. At $mt^{\max} = 2$ we do not manage to reduce the value of the cost functional below $L^{\text{prior}}_{mt_{\max}=2} = 759$. Correspondingly in the bottom row of fig. 6.7 it is clear that CL even in the presence of the field-independent kernel fails to converge to the correct solution. Interestingly the imaginary part of the unequal-time two-point correlator still agrees very well with the true solution on the forward branch while its real part already shows significant deviations from the correct solution. This deviation of the unequal time correlation function affects also the values of the equal-time correlation function which is far from constant and thus leads to a penalty in L^{prior}, correctly indicating failure of correct convergence.

There are two possible reasons behind the failure of convergence at $mt^{\max} = 2$. One is that the low-cost gradient obtained from L_D is unable to bring the kernel close to those values required for restoring correct convergence. The other is that the field-independent kernel is not expressive enough to encode the change in CL dynamics needed to restore correct convergence. In simple models it is e.g. known from ref. [70] that field-independent kernels may fail to restore correct convergence for large imaginary drift. We believe that, as a next step, the investigation of field dependent kernels is most promising.

The unequal time correlation function is most relevant phenomenologically, as it encodes the particle content and occupation numbers in the system. We thus compare in fig. 6.8 the values of $\langle x(0)x(t) \rangle$ along the forward real-time extent of the contour for $mt^{\max} = 1.0, 1.5$ and 2.0 to the correct solution given as black solid line. Here we can see in more detail that for a real-time extent of 1 and 1.5 CL with the optimal learned kernel converges to the true solution within uncertainties. At 2 the real part of

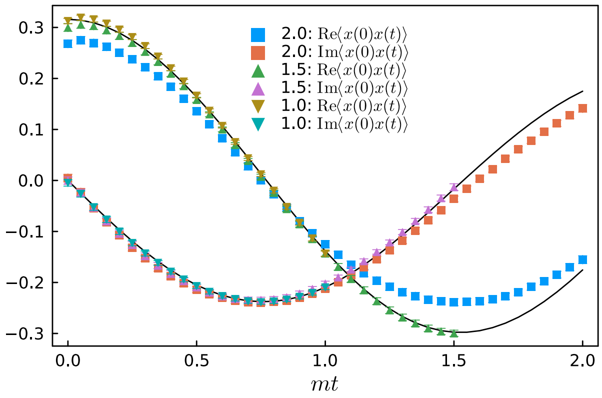

Figure 6.8: A detailed comparison of the unequal time correlation functions $\langle x(0)x(t)\rangle$ from fig. 6.7 evaluated in the presence of the optimal learned field-independent kernel on contours with $mt_{\max} = 1, 1.5$ and 2 respectively.The different colored circles correspond to the real-part while the squares to the imaginary part of the correlator. Values of the correlators from the solution of the Schrödinger equation are given as solid black lines.

the correlator begins to deviate from the correct solution. Note that the most difficult points to achieve convergence at are $t = 0$ and at $t = t_{\max}$. Similarly we find that these points are also the ones, where the equal time correlator deviates the most from the correct solution, an important fact as this allows this deviation to contribute to the penalty in L^{prior}.

In fig. 6.9 we plot a heat map of the values of the kernels with $mt_{\max} = 1.5$ (center) and $mt_{\max} = 2$ (right) compared to the free theory propagator kernel from eq. (6.36) (left) for $mt_{\max} = 1.5$. (for a sketch of the structure of the heat map see the left panel of fig. 6.6). We have subtracted the unit matrix from the real-part of the two optimized kernels. They both exhibit a diagonal band in the real part, which is thinner than the one in the free theory kernel. It is interesting to see that both show non-trivial structures passing through the $t^{\max}$ point and when connecting to the Euclidean branch. In the imaginary part the structures have more similarity with the free theory propagator kernel, where a sign change occurs as one moves away from the diagonal. The difference in the optimal kernels between

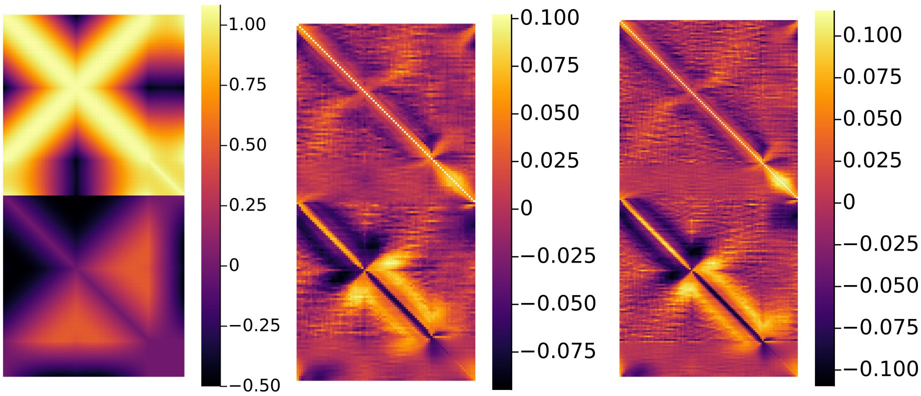

Figure 6.9: (left) Free theory kernel for the SK contour with $mt_{\text{max}} = 1.5$. (center) The optimal learned kernel in the interacting theory for $mt_{\text{max}} = 1.5$, which achieves correct convergence of CL. The diagonal entries with values close to unity are subtracted from the kernel. (right) The kernel obtained as a result of the optimization procedure in the case of $mt_{\text{max}} = 2.0$, which does not achieve correct convergence. At the turning point at t_{max} and when connecting to the Euclidean domain the kernel for the interacting theory shows nontrivial structure not present in the free theory.

$mt_{\text{max}} = 1.5$ and $mt_{\text{max}} = 2$ is small overall.

6.5 Summary and Conclusion

In this paper we proposed a novel strategy to recover correct convergence of real-time complex Langevin simulations by incorporating prior information into the simulation via a learned kernel. The effectiveness of the strategy was demonstrated for the strongly coupled anharmonic oscillator on the Schwinger Keldysh contour by extending correct convergence in this benchmark system up to $mt_{\text{max}} = 1.5$, three times the previously accessible range of $mt_{\text{max}} = 0.5$.

After discussing the concept of neutral and non-neutral modifications of Langevin dynamics by use of real and complex kernels, we demonstrated that an explicitly constructed complex kernel can be used to improve the convergence behavior of real-time complex Langevin on the Schwinger-Keldysh contour. Taking insight from a single d.o.f. model and the harmonic oscillator, approximately correct convergence in the strongly coupled anharmonic oscillator was achieved up to $mt_{\max} = 1$. As no systematic extension to the explicit construction of that kernel exists, we instead proposed to learn optimal kernels using prior information.

The ingredients to learning an optimal kernel are prior information and an efficient prescription for computing gradients. Prior information comes in the form of apriori known Euclidean correlation functions, known symmetries of the theory and the Schwinger-Keldysh contour, as well as information on the boundary terms. Here we included only the first two types of information, which sufficed to achieve improvements in convergence. We surveyed different modern differential programming techniques that in principle allow a direct optimization of the kernel based on the full prior information, but found that in their standard implementations they are of limited use in practice due to runtime or memory limitations. Instead we constructed an approximate gradient based on an alternative optimization functional, inspired by the need to avoid the presence of boundary terms. This optimization functional possesses a gradient, which can be approximated with much lower cost than that of the original optimization functional. The low-cost gradient in practice is computed using standard auto-differentiation. By minimizing with this gradient and monitoring success via the full prior information cost functional we proposed, we were able to locate optimal kernels.

Our strategy was successfully applied first to the harmonic oscillator on the thermal SK contour. We managed to restore correct convergence with an optimal learned field-independent kernel that shows a simpler structure compared to the manually constructed kernel. This result bodes well for future studies, where we will investigate in detail the structure of the optimal learned kernel to draw conclusions about the optimal analytic structure for extending the approach to a field-dependent kernel.

The central result of our study is the restoration of correct convergence in the strongly correlated anharmonic oscillator on the thermal SK contour up to a real-time extent of $mt_{\max} = 1.5$, which is beyond the reach of any manually constructed kernel proposed so far. We find some remnant deviations of the equal-time correlation function $\langle x^2 \rangle$ from the true solution at the turning points of the SK contour. The phenomenologically relevant unequal-time correlation function $\langle x(0)x(t) \rangle$ on the real-time branch on the other hand reproduces the correct solution within statistical uncertainty.

While our strategy based on a field-independent kernel is successful in a range three times the previous state-of-the-art, we find that the restricted choice of kernel limits its success at larger real-time extent.

We conclude that our study provides a proof-of-principle for the restoration of correct convergence in complex Langevin based on the inclusion of prior information via kernels. Future work will focus on extending the approach to field-dependent kernels, carefully reassess the discretization prescription of the SK at the turning points and improve the efficiency of the differential programming techniques necessary to carry out a minimization directly on the full prior knowledge cost functional.

7 Towards a reliable gradient for the complex Langevin equation

The previous chapter introduced a novel optimization algorithm that uses prior knowledge to optimize a kernel suitable to make the kernel-controlled complex Langevin equation correctly converge. When trying to optimize a kernel based on eq. (6.47), we see that we are not able to obtain a reliable gradient for the optimization scheme, i.e., the value of the loss function is not reducing. Therefore, an alternative gradient formulation is needed for optimizing the kernel-controlled complex Langevin equation based on eq. (6.47). This chapter will present a possible direction for achieving this goal by using sensitivity methods used in fields with chaotic systems. We will uncover the underlying problem of why the standard method to obtain gradient is ill-defined in the real-time complex Langevin system and build towards a reliable gradient by introducing a different adjoint sensitivity method.

We have already seen that adjoint methods are effective tools for the optimization of iterative schemes, including stochastic differential equations, which we showed an example of in section 3.3[1]. It is a way to obtain gradients by only having to solve a similar system once, namely the corresponding adjoint equation. This can be used for many different systems, like non-linear equations, differential equations, and stochastic differential equations. The first two have been successfully implemented in various systems, with an increasing interest in the last decade. It can help solve inverse problems as the spectral reconstruction example in chapter D, estimate simulation errors, and be used in control processes and systems [148] to name a few applications. The stochastic differential equation version has only recently been introduced formally in [80, 81].

Sensitivity methods tend to fail when the dynamical system is chaotic, and we seek to calculate long-time averaged quantities. In these cases,

[1] An example of a non-linear system is shown in chapter D, where we optimize a regulator for the Bayesian reconstruct of spectral functions.

the computed gradient is larger than expected and will exponentially increase for a longer simulation time. This failure is caused by the so-called "butterfly effect", which appears in systems that are overly sensitive to initial conditions and changes in system parameters. The reason for such failure to obtain reliable gradients in chaotic systems was first explained in [151], where they identified the problem in the Lorenz system and showed that it is due to positive *Lyapunov exponents*. Much research has been carried out to overcome this failure, resulting in different methods that have made it possible to obtain such sensitivities in fields like computational fluid dynamics [152, 153] and climate modelling [154, 155] where chaotic systems often arise.

Several methods of avoiding the problem of unstable gradients have been proposed [156, 157, 154, 155, 158, 159], but in this section, we will focus on the class of algorithms that are based on the *shadowing lemma* [153, 149, 152]. In short, these algorithms do not compute the gradients based on a perturbation in the parameter of interest but rather a perturbation in the trajectory based on the parameter of interest, i.e., a *shadowing trajectory*.

The outline of this chapter is as follows; we start in section 7.1 by introducing what a chaotic system is in terms of the Lyapunov exponents of a dynamical system. In section 7.2 we apply the formalism to the real-time complex Langevin equation and show that it is chaotic. Then, in section 7.3, we use the Lyapunov exponents to show why standard methods of obtaining gradients fail in the real-time complex Langevin before we go on to introduce the shadowing sensitivity methods in section 7.4 and its stochastic version. Finally, in section 7.4.2, we apply the stochastic NILSAS scheme to obtain gradients based on the parameters of the stochastic Lorenz system.

7.1 Chaotic systems and Lyapunov exponents

Chaos theory is a branch of mathematics that studies dynamical systems that exhibit complex and unpredictable behavior, even in the presence of deterministic rules. These systems are characterized by their sensitive dependence on initial conditions, which means that small changes in the starting points can lead to vastly different outcomes over time. This phenomenon is also known as the "butterfly effect". Chaos theory has found applications in various fields, such as physics, meteorology, biology, engineering, and economics.

One of the crucial concepts in chaos theory is the Lyapunov exponent [160], which helps quantify the rate of divergence or convergence of nearby trajectories in a dynamical system. The Lyapunov exponent measures the system's stability and predicts chaotic behavior; a positive Lyapunov exponent indicates that the system is chaotic, while a negative exponent signifies stability. In ordinary differential equations, this can be visualized by investigating the trajectories around a single stable (critical) point, where positive exponents would circle around the stable point end eventually end at the point. A positive Lyapunov exponent would start diverging away from the point, and an exponent of 0 would circle around the point in a stable orbit.

As a general concept, let's investigate a dynamic system of the form

$$\frac{dx(t)}{dt} = f(x, \theta, t) \tag{7.1}$$

where x is the state variable and θ the system parameters. We are interested in the evolution of the separation between two trajectories $x(t)$ and $x'(t)$, noted by $z(t) = x(t) - x'(t)$. The initial condition of the two trajectories is an infinitesimal distance apart such that $z(0) \longrightarrow 0$. The stability of the ODE can be assessed by investigating the $z(t)$; for a stable system, we have $z(t) \longrightarrow 0$. The evolution of the separation is given by

$$\frac{\partial z(t)}{\partial t} = \frac{\partial f(x, \theta)}{\partial x} z(t) = J(x, t) z(t) \tag{7.2}$$

where we have noted the Jacobian of the system as $J(x, t)$. The general solution to this system is given by

$$z(t) = H[x, t] z(0) = e^{\int_0^t dt' J(x,t)} z(0). \tag{7.3}$$

Note that the evolution operator $H[x,t]$ depends on all the previous values of x. The amplitude of the separation is given by

$$|z(t)|^2 = |H[x,t]z(0)|^2 = z^T(0)H^THz(0). \tag{7.4}$$

We can see that if H is an orthogonal matrix, i.e., $H^TH = I$, then we get $|z(1)|^2 = |z(0)|^2 \rightarrow 0$, and hence the separation of the two trajectories remains infinitesimal. To quantify the separation, we look at the infinite time limit of the operator and linearize the system. We define

$$\lim_{t\rightarrow\infty} \left[H^TH\right]^{1/2t} = P \tag{7.5}$$

which will reach a stationary value for a finite t if the system is ergodic. The eigenvalues μ_i of P now yield the Lyapunov exponents by

$$\lambda_k = \log \mu_k. \tag{7.6}$$

The Lyapunov exponents λ_i now describe the magnitude of the separation change, i.e.,

$$|z(t)| = e^{\lambda t}|z(0)|, \tag{7.7}$$

while the eigenvectors of P describe the direction of the change. The most important of the Lyapunov exponents is the one with the largest real part, denoted as λ_1 and often called maximum Lyapunov exponents (MLE). This will dominate the change in separation distance $z(t)$. If the MLE is negative, then $e^{\lambda_1 t}$ tends to 1 in the infinite limit, such that $H^TH = I$. The system can be said to be stable for $\lambda_1 < 0$.

If the MLE is positive ($\lambda_1 > 0$) the separation distance will exponentially increase. In these cases, the system is not considered stable since an infinitesimal change in the system yields an exponentially different trajectory. To compute the Lyapunov exponents, we need to compute $H(x,t)$. This can be done by rewriting the time evolution operator into an evolution equation given by

$$\frac{d}{dt}H(x,t) = J(x,t)H(x,t). \tag{7.8}$$

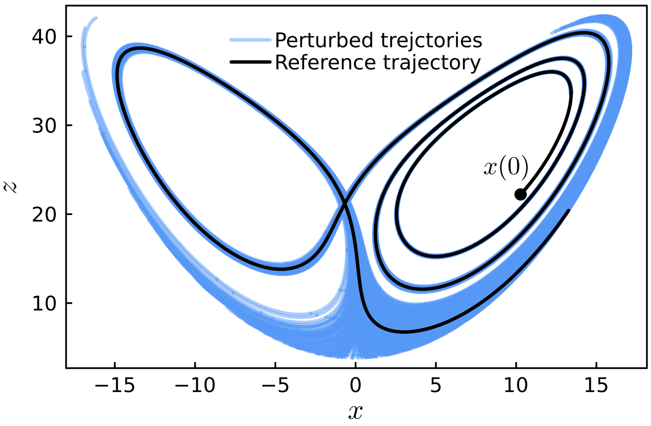

Figure 7.1: This shows several realizations of the Lorenz system, where all trajectories are initialized around $x(0)$ with a Gaussian distribution with variance 0.001. The black line is the reference trajectory, initialized as $x(0)$ without any perturbation. We see that the cloud of trajectories stays together in the beginning but quickly deviates.

with the initial condition $H(0) = I$. We can evolve the equation up to a final time T, then compute the eigenvalues of $P = H^T H$; then we can read out the MLE by sorting the eigenvalues by the real part.

A convenient way to think about it is if we propagate a cloud of initial vectors $x(0) + \delta x(0)$, where $\delta x(0)$ is distributed according to a Gaussian distribution, forward in time. The distribution of vectors will expand and contract in different directions, the magnitude of this behavior is what we refer to as the Lyapunov exponents (LEs). It is important to mention that the LEs are not local objects and hence correspond to a mode in the long-time behavior of the system. The direction of the deformation of the distribution is given by the covariant Lyapunov vectors (CLV)[2] $\psi^1(x), \psi^2(x), \ldots$, which satisfy the equation

$$\frac{d}{dt}\psi^i(x(t)) = J(x,t)\psi^i(x(t)) - \lambda_i \psi^i(x(t)) \tag{7.9}$$

[2] This term is misleading, as it does not refer to the eigenvector of the corresponding LEs, but rather the corresponding eigenvectors of the local LEs, which is often called the characteristic Lyapunov exponents.

where λ_i is the corresponding LE. These vectors indicate the direction of the deviation given by the corresponding LE. For a given LE, with the magnitude λ_i, the corresponding difference between the two trajectories (z) increases in the direction of ψ^i. Let's try to visualize this using the Lorenz attractor, which we will introduce in section 7.4.2. Figure 7.1 show the trajectory of $x(t)$, without a perturbation added (reference trajectory) in black. Then the cloud of points initialized around $x(0)$ with a Gaussian distribution with variance 0.001, is shown in blue. Initially, all the trajectories will follow the same path; each time it gets close to zero on the x-axis, the perturbed trajectories start to deviate from the reference trajectory, i.e., the variance of the distribution of the cloud increases. However, it stays close to the reference trajectory until it crosses zero on the x-axis the second time. At this point, the variance of the cloud increases fast, as some of the trajectories follow the left side and the rest on the right-hand side. The system has so-called unstable CLV directions corresponding to the positive Lyapunov exponents.

It has been observed that the computation of the flow matrix H fails for large times t [160]. This can be avoided with the help of a suitable re-ortonormalization scheme. One of these methods is the *Benettin algorithm* [161], which computes the n'th largest Lyapunov exponents (LE). We will concentrate on the exponent with the largest real part, which is often called MLE (λ_1), so we do not need to compute more than one LE. This scheme re-orthonormalize the solution at checkpoints during the evolution of the so-called tangent equation[3], given by

$$\dot{u}(t) = Ju(t) \tag{7.10}$$

where $u(t)$ is a vector, which is initialized with a random value u_0 and then orthonormalized, such that $\bar{u}_0 = \frac{u(t)}{||\bar{u}(t)||}$. This flow's late time limit is independent of the initial condition [161]. We evolve the equation over a time ΔT until the first checkpoint at t_1. We then save $u(t_1)$ to memory and orthonormalize the vector before continuing the evolution of $\bar{u}(t_1) = \frac{u(t_1)}{||\bar{u}(t_1)||}$. We then repeat this process up to a final time T, with

[3] This is often referred to as a flow equation in lattice field theory

a total of $M = \frac{T}{\Delta T}$ intermediate checkpoints. The MLE is then given by [161]

$$\lambda_1 = \frac{1}{M} \sum_{i}^{M} \ln ||u(t_i)||. \tag{7.11}$$

Note that we only use the values at the checkpoints before normalizing them ($\bar{u}$) and use this as the initial condition for the evolution ΔT to the next checkpoint.

For simplicity, we will define a chaotic system to be a system with positive Lyapunov exponents. Although this is not a formal definition, it is sufficient for describing the problem of the gradient of the complex Langevin. Next, we will show that the real-time complex Langevin evolution of the anharmonic oscillator is, in fact, chaotic by calculating the Lyapunov exponents. It is important to note that the Lyapunov exponents calculated here only approximate the system's behavior, as the complex Langevin equation is inherently stochastic.

7.2 Lyapunov exponent of Real-time CLE

Before we continue discussing the properties of a chaotic system, we will calculate the Lyapunov exponents of the real-time systems discussed in chapter 6. This measure can tell us whether or not the system is chaotic and the dynamical system's stability. The complex Langevin is a dynamical system in the fictitious Langevin time, which is different from most dynamical systems considered in the context of chaotic systems.

We assume the Lyapunov method for the abovementioned dynamical systems can be generalizable to stochastic dynamical systems. This is because the complex Langevin, with a constant (field-independent) kernel, has a constant noise coefficient. In the linearization of the system, as was done to obtain eq. (7.8), the noise terms drop out, as the Jacobian does not contain the noise term. There are, however, some subtleties to this when it comes to field-dependent noise terms, which we will not go into here, see [162] for a detailed overview.

7.2.1 Lyapunov exponent of free theory

We will first discuss the free theory, as this is a linear SDE and, therefore, trivial to obtain the Lyapunov exponent. The SDE for the free theory is given, as described in section 6.3.3, by

$$d\phi = iM\phi dt + dW \tag{7.12}$$

where $iM\phi$ is a linear rewriting of the drift term. We saw in chapter 6 that this action supplied to the Schwinger-Keldysh contour has a large relaxation time, and by using the free theory propagator kernel ($K = (iM)^{-1}$), the relaxation time was reduced significantly.

Computing the Lyapunov exponents (LEs) of this system is trivial as we have already linearized the SDE, which yields a constant Jacobian of $J_{\text{free}} = iM$. The LEs are the real-part of the eigenvalues of the Jacobian. Supplied with a free theory propagator kernel, we get a completely degenerate set of Lyapunov exponents at $\lambda_i = -1$, since the Jacobian is given by $J_{\text{free,K}} = -I$.

For a real-time of $t^{\text{max}} = 10$, we get the Lyapunov exponent, computed by the largest eigenvalues of M, to be $\lambda_1^{K=I} = -10^{-5}$, while with the kernel, we get $\lambda_1^{K=-iM^{-1}} = -1$. This indicates, in the same way as was shown in section 6.3.3, that by applying the free theory kernel, we speed up the convergence rate of the simulation. The Lyapunov exponent without the free-propagator kernel applied gets closer and closer to zero for increasing real-time, and already at $t^{\text{max}} = 30$, we have a Lyapunov exponent of the order $\lambda_1^{K=I} = -10^{-8}$. Hence, the stability measure indicates how far it is feasible to simulate in real-time, which for the free theory without a kernel, quickly becomes expensive.

The MLEs for the free theory with and without a kernel are plotted in the upper panel of fig. 7.2, named "K^I free" for the free theory without a kernel, and "K^M free" with the free theory propagator kernel. As we saw, the MLEs for the kernel are constant equal to -1, while without a kernel quickly increase towards 0.

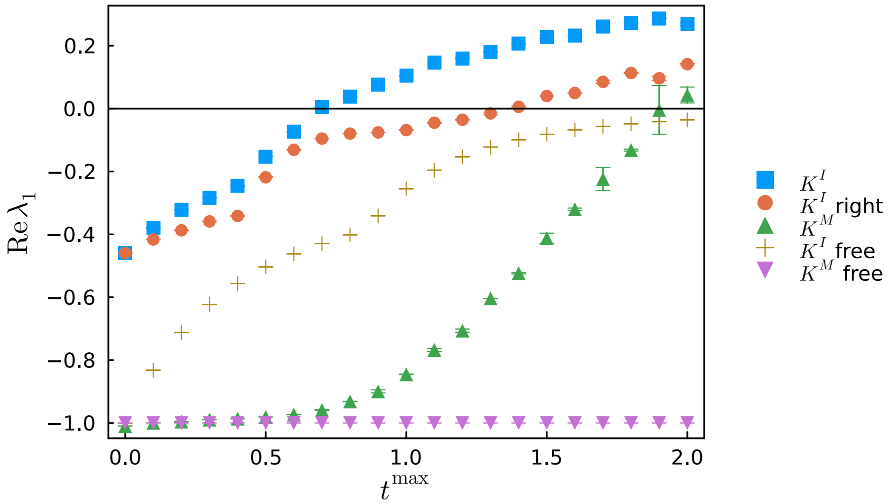

Figure 7.2: Calculation of maximum Lyapunov exponents (MLE) for increasing real-time extent using different contour, system, and kernel setups. The blue squares (K^I) show the anharmonic oscillator (AHO) without a kernel on the canonical Schwinger-Keldysh (SK) contour. The orange circles are the same model on the right SK contour. The greed triangles (K^M) are the same model on the canonical SK using the free theory propagator kernel. The free theory is shown for the yellow crosses and purple triangles without (K^I free) and with (K^M free) a kernel. We treat the CLE systems with MLE above 0 to be chaotic systems.

7.2.2 Interacting theory

We can now investigate the Lyapunov exponent for the real-time system where the complex Langevin converges to the wrong solution at a small real-time extent. For the free theory, a small Lyapunov exponent indicates a slow convergence to the expected solution. It is, therefore, interesting to investigate if similar behavior is present in the interacting theory where we know there is a failure in convergence to the wrong solution.

Since the system is non-linear, we cannot base the Lyapunov exponent's (LE) calculation on the drift coefficient's eigenvalues. We will therefore use the method described in the previous section, where we need to evolve the tangent equation for MLE $u(t)$. For the real-time complex Langevin system, the tangent equation is written by

$$\frac{d}{dt}u(t) = K\, i\frac{\partial^2 S[\phi]}{\partial \phi^2}u(t), \tag{7.13}$$

which is initialized with $u(0)$ as a random vector normalized to one. We then read off the MLE by eq. (7.11).

We plot the result for different real-times in the upper panel of fig. 7.2 for different setups. The blue squares (K^I) are the anharmonic oscillator with $m = 1$, $\lambda = 24$, and time discretization of $a = 0.1$ on the canonical Schwinger-Keldysh contour (panel **c** in fig. 4.3). The orange circles have the same setup, except for the contour is now panel **b** in fig. 4.3, which follows the real-time axis to $mt^{\max}$, and tilted directly down to $-i\beta$. We use the name Schwinger-Keldysh (SK) right for this contour, adopted from [135].

We see that the canonical SK contour leads to less stability, as expected. It also crosses the zero line already at $mt^{\max} = 0.7$, while the SK right contour crosses the zero line at $mt^{\max} = 1.3$. This means that after this point, the system is considered to be chaotic. We will come back to why this is important in the next section. It is also interesting to compare this to when the complex Langevin converges to the wrong solution. This is at $mt^{\max} \approx 0.7$ for the canonical SK contour, while $mt^{\max} \approx 1.1$ for the SK right contour. Although this is not a criterion for when the CLE converges incorrectly, it does indicate the effect of a new important contribution to the dynamical system at this transition point. Since we know the system to have more than one critical point (classical solution), it indicates that the critical points not located at zeros get a stronger attraction force for larger real-time. This can contribute to the system becoming chaotic, as the CLE can transition between the different attractors (critical points) in a chaotic way.

We have also included the anharmonic oscillator simulation with the free theory propagator for comparison, which is denoted K^M. This converges to the wrong solution at around $mt^{\max} \approx 1.0$, while the MLE is still negative. This shows that the link between the convergence to the wrong solution and the Lyapunov exponent is unrelated. It is also worth noting that we see a steep increase in the MLE when using this kernel. This indicates that after $mt^{\max} \approx 1.0$, the difference between the anharmonic oscillator (AHO) and the free theory (harmonic oscillator) is increasing. In practice, this means that the in the AHO, the ϕ^4 term is increasingly dominant for a larger real-time.

These findings have several impacts on the numerical simulation of the system. First, this indicates that much care is needed when selecting numerical schemes for simulation to a large real-time extent. Secondly, the problem of obtaining a gradient of the simulation, e.g., using the adjoint sensitivity method, is due to the Lyapunov exponent becoming positive. We will come back to the latter one in the next section.

We know that the value of the Lyapunov exponents dictates the stability of the simulation. And we already showed that there is a big difference in stability for the free theory with and without the free propagator kernel. However, the difference is larger when transitioning between a negative MLE and a positive one, which means a transition from a stable attractor to an unstable attractor. This indicates that we need a high-accuracy solver to get reliable results. In fact, it might be necessary to modify the complex Langevin equation with, e.g., a kernel to make the computation accurate enough.

We can reduce the MLE down to being negative again using a suitable kernel obtained by the use of an optimization scheme. In fig. 7.3, we performed a kernel optimization using the L_D low-cost loss function and calculated the MLE after each optimization step for several real-time extents. In the upper panel, we see the decrease in MLE after each step, where $mt^{\max} = 0.5, 1.0, 1.5$ continue decreasing, while the $mt^{\max} = 2.0, 3.0$ stays constant early on.

The low-cost gradient is able to obtain a reliable gradient of the system, which at this point might be counterintuitive. However, we can think of

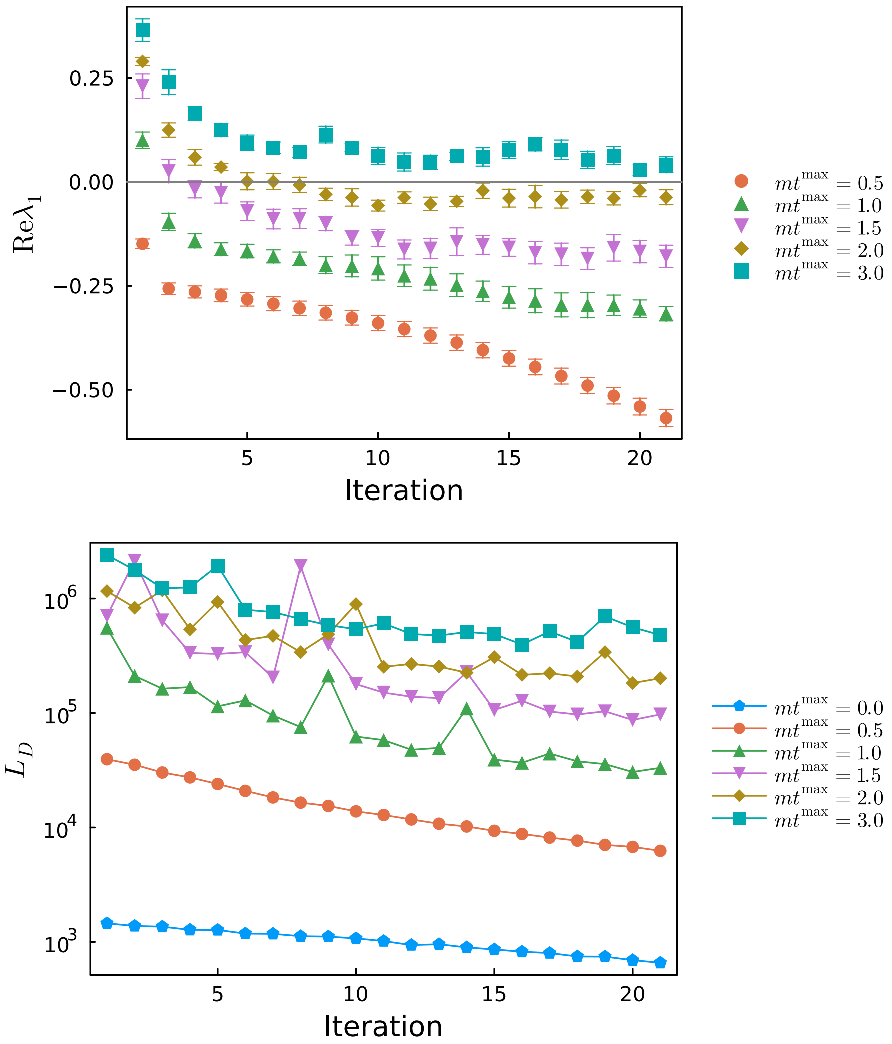

Figure 7.3: The iterations from a kernel optimization using the low-cost gradient, where we have plotted the MLE (top panel) and the loss function L_D (bottom panel). We see a constantly decreasing MLE and L_D for small $mt^{\max}$, while for larger $mt^{\max}$, both the MLE and L_D flatten out after just a few iterations.

this loss function to be on the level of the configurations, i.e., approximate each configuration to be independent of the simulation. In this approximation, the gradient is not dependent on the dynamical system in Langevin time. Hence, it is not affected by the problems of an ill-defined gradient, as we will discuss in the next section. Keep in mind that the gradient of L_D is an approximation.

Even though the gradient of L_D is not susceptible to the problems with chaotic systems, the change in the kernel parameters is. This means that since we need a finite change to the kernel, this is considered a change in the parameters of the system. In the next section, we will see that this change is similar to the change in the initial condition of a chaotic system. Hence, we do expect that any update to the kernel, even with a correct gradient, does not guarantee a reduction in the loss. This is due to the unpredictability of the system. We can see that for a longer real-time extent in fig. 7.3, the update in the MLE is oscillating more and more. On average, L_D is decreasing when using the low-cost gradient, which means that for this specific loss function, we are able to optimize a kernel.

In the lower panel fig. 7.3, we show the improvements in L_D. We see that the loss is decreasing for all the real-time extents, but for the longer real-times, it oscillates. This behavior is common when optimizing a loss function, but we see a difference between the $mt^{\max} = 0.5$ and the longer real-time extent. Although there is an overall improvement for the $mt^{\max} > 0.5$, changes in the kernel are unpredictable due to the positive or small negative Lyapunov exponent.

Let us speculate on a possible connection between the Fokker-Planck operator eigenvalues and the Lyapunov exponents. While they are not the same mathematical construct, we know that both quantities express information about the stability and convergence properties of the stochastic dynamical system. This could mean linking the two concepts, e.g., the MLE yielding an upper bound for the eigenvalue with the largest eigenvalue. If this eigenvalue is positive, we know that the complex Langevin is not converging to the true solution [46] (see chapter 5), i.e., the stationary distribution of the Fokker-Planck equation is not e^{iS}. Although this is speculation, such a connection would be important as we can safely say that the boundary terms break the convergence of the complex Langevin.

Now that we have shown that the real-time complex Langevin equation is chaotic at the real-time extent we are interested in, in the next section, we will use this to show why the gradient of such a system is ill-defined when formulating the derivative in the standard way.

7.3 When a gradient is ill-defined

For some systems, the adjoint sensitivity method laid out in section 3.2 fails to give a reliable gradient; in fact, for some cases, it even diverges for increasing simulation time. Although the gradient of the complex Langevin equation does not diverge, it does yield incorrect results similar to other chaotic systems. The category of system where the gradient of the simulation is ill-defined can be categorized under chaotic systems and, more precisely, systems with positive Lyapunov exponents.

We can, in principle, compute these gradients with finite differences, automatic differentiation, and tangent/adjoint sensitivity methods. For chaotic systems, all of them yield incorrect gradients due to the positive Lyapunov exponent. The fundamental reason for the breakdown is that the infinite time limit and the infinitesimal perturbation limit do not commute for time-average quantities, such that [151]

$$\begin{aligned}&\lim_{\delta\theta\to 0}\lim_{T\to\infty}\frac{1}{T}\int_0^T\frac{g(x(t,\theta+\delta\theta),\theta+\delta\theta)-g(x(t,\theta),\theta)}{\delta\theta}dt\\ \neq&\lim_{T\to\infty}\frac{1}{T}\int_0^T\lim_{\delta\theta\to 0}\frac{g(x(t,\theta+\delta\theta),\theta+\delta\theta)-g(x(t),\theta)}{\delta\theta}dt\end{aligned} \tag{7.14}$$

does not hold. The first equation states that we first integrate up to the $T \to \infty$ and then take the limit $\delta\theta \to 0$. While the other equation states that we first take the derivative of the integrand with respect to the parameter θ and then let $T \to \infty$. As we cannot compute the integral and then take the limit $\delta\theta \to 0$ on a computer, we need to compute it via the right-hand side, which is incorrect in this case. In fact, it is not even defined for chaotic systems, as the integrand itself tends to infinity in the $T \longrightarrow \infty$ limit.

To see this we use the chain rule on the integrand of the right-hand side

$$\lim_{\delta\theta \to 0} \frac{g(x(t;\theta+\delta\theta),\theta+\delta\theta) - g(x(t;\theta),\theta)}{\delta\theta} = \frac{\partial g}{\partial x}\frac{\partial x}{\partial \theta}. \tag{7.15}$$

For a chaotic system $\frac{\partial g}{\partial \theta}$ increase exponentially, and hence grows to ∞ for $t \longrightarrow \infty$. We can see this by investigating the tangent equation (which we derived in eq. (3.20))

$$\frac{d}{d\theta}\frac{du(t)}{dt} = \frac{dv(t)}{dt} = \frac{\partial f}{\partial u}v(t) + \frac{\partial f}{\partial \theta} = Jv(t) + \frac{\partial f}{\partial \theta} \tag{7.16}$$

where $J = \frac{\partial f}{\partial u}$ is the Jacobian. The solution to the tangent equation will have a solution of the form

$$v(t) = e^{\int_0^t J(\tau)d\tau} v(0) + C(u), \tag{7.17}$$

where $C(u)$ is due to the second term on the right-hand side of the tangent equation. We can now identify $H = e^{\int_0^t J(\tau)d\tau}$ as the same object we used to calculate the Lyapunov exponent, and hence we see that $v(t)$ will, in fact, increase exponentially in the same way as an infinitesimal change in the initial condition.

This means that attempts to obtain the gradient using the adjoint sensitivity method of a loss function, including a chaotic system, will have an ill-defined gradient. This practically means, as we mentioned in section 6.4, that we cannot directly optimize the loss function eq. (6.43) using the adjoint sensitivity method for real-time extents where CLE converges to the wrong solution. We will now demonstrate this difficulty using the real-time complex Langevin equation.

7.3.1 Testing the adjoint sensitivity method on real-time complex Langevin

To demonstrate the difficulties with adjoint methods applied to the real-time complex Langevin, we must compare it to real-time extents with positive Lyapunov exponents. In the context of the optimization scheme laid out in section 6.4, this is not easy, as the complex Langevin already converges correctly in the region of negative LEs. This means that the loss function tends toward zero for an increasing simulation time. If we instead optimize a kernel based on a fixed Langevin time extent, i.e., we use the same initial condition and random seed up to a finite time T for every update step. We are able to see improvement as the kernel tries to correct the statistical error in the observable estimation.

For example, we can optimize the x observable, which should be zero at all points along the contour. For a finite simulation time, the estimated mean of the observable, $\langle x \rangle_T$, is only equal to zero within statistical and systematic error. For this optimization, however, we disregard the statistical error, such that we are only interested in optimizing the estimated mean to be zero.

We use a simulation time of $T = 5$, with 5 different trajectories, with the canonical Schwinger-Keldysh contour, the lattice spacing of $a = 0.1$, and the implicit Euler-Maruyama scheme. The loss function is setup to minimize the $\langle x \rangle_{T=5}$ observable, and is written as

$$L_x(K) = \left[\mathrm{Re}\langle x \rangle_{T=5}\right]^2 + \left[\mathrm{Im}\langle x \rangle_{T=5}\right]^2 . \tag{7.18}$$

The optimization uses the *ADAM* optimizer, with a learning rate of 0.002. We can think of the optimization as a simplified version of an optimization scheme to reduce the burn-in time, i.e., the time it takes for the simulation to start sampling from the stationary distribution of the Fokker-Planck equation. A proper investigation into such a method can be interesting as it can potentially speed up computations of more complicated models.

In fig. 7.4, we have plotted the loss function for the first 15 iterations of the optimization scheme for different real-time extents. We have normalized the loss function based on the value of the first iteration for readability. As expected from the Lyapunov exponents of the same real-time extents, we see that only $mt^{\max} = 0, 0.2, 0.3$, and 0.5 obtains reliable gradients. In all of these real-time extents, the loss improvement is smooth for the first 5 iterations, but after this, the $mt^{\max} = 0.5$ starts increasing again.

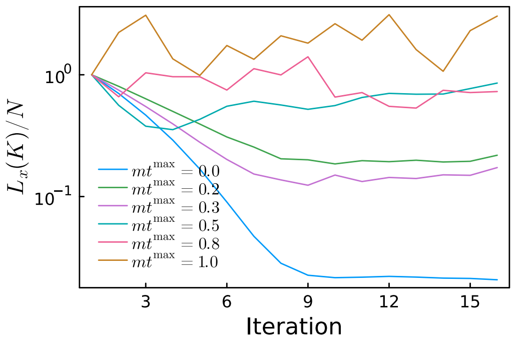

Figure 7.4: Adjoint sensitivty method optimizing of simple loss funciton eq. (7.18) for different real-time extents. We see that for the $mt^{\max}$ that is not chaotic, the loss decreases exponentially until it reaches a plateau. The exception is $mt^{\max} = 0.5$, which initially decreases, but pushes the MLE up such that the CLE becomes chaotic. We see no improvement in the loss for higher $mt^{\max}$.

This is because the kernel pushes the Lyapunov exponent to be positive, making the gradient ill-defined. The optimization does not improve the loss for the two last ones ($mt^{\max} = 0.8, 1.0$), which is expected since they both exhibit positive MLE.

7.4 Shadowing sensitivity algorithms

We have seen that the adjoint sensitivity method fails for the real-time complex Langevin equation. I have investigated possible methods to obtain a reliable gradient of the real-time complex Langevin, and in this section, I will review a potential solution to the problem. The method is adopted to work with stochastic differential equations and use a stochastic Lorenz attract as a test model.

I will discuss the group of sensitivity methods that rely on the *shadowing lemma*, which is the most prominent method to obtain gradients of chaotic dynamical systems [153]. Although many versions of the method have been proposed based on the same lemma, we will introduce the idea using the initial method presented in [153]. In the following section, I will apply a different version, called the NILSAS (non-intrusive least square adjoint shadowing) method, to the stochastic Lorenz system. Here I will only present the changes done to the algorithm and refer to [152] for more information on the derivation. The reason for the difference in method is that I found the intuition of how the method works easier to understand for the initial case. At the same time, the NILSAS is a more efficient method, which is constructed such that previous adjoint sensitivity implementations can easily be converted.

The *least square shadowing* (LSS) sensitivity algorithm is based on finding a so-called "shadow trajectory", which is a perturbed trajectory that stays close to the unperturbed trajectories at all times. The existence of such a trajectory follows from the shadowing lemma [163], which states that;

Consider a solution u to $\dot{u} = f(u;\theta)$, if this system has a hyperbolic strange attractor and if some system parameter θ is slightly perturbed by some $\delta\theta$, then for any $\delta > 0$ there exist $\epsilon > 0$, such that for every u that satisfies $||du/dt - f(u;\theta+\delta\theta)|| < \epsilon$, there exists a true solution $\bar{u}$ and a time transformation $\tau(t)$, such that $||\bar{u}(\tau(t)) - u(t)|| < \delta$, $|1 - d\tau/dt| < \delta$ and $\frac{d\bar{u}(\tau)}{d\tau} - f(\bar{u};\theta+\delta\theta) = 0$.

In words, this means that it is possible to find a trajectory $\bar{u}(\tau(t))$ that closely follows the original solution $u(t)$. The shadowing solution satisfy the original system $\dot{u} = f(u;\theta)$ up to a round off error ϵ, and by introducing a time transformation $\tau(t)$ the system $\frac{d\bar{u}(\tau)}{d\tau} - f(u(\tau);\theta+\delta\theta) = 0$ holds.

For the shadowing lemma to work, the attractor associated with the system must be hyperbolic. This means that the tangent space can be divided into stable, neutrally stable, and unstable components throughout the attractor, corresponding to negative, zero, and positive Lyapunov exponents. Although not all attractors can be broken down in this way, the *Chaotic Hypothesis* [157] suggests that many high-dimensional chaotic systems behave as if they were hyperbolic. For instance, the Lorenz attractor [164], which we will introduce in section 7.4.2, has only one non-hyperbolic point, which is the unstable fixed point at the origin. Since most phase space trajectories don't go through this point, the shadowing lemma applies to the majority of trajectories on the Lorenz attractor[4].

One way to find the shadowing trajectory is to perform a least square minimization of the shadowing trajectory. The minimization problem can be written as

$$\begin{aligned} &\min_{\bar{u}} \frac{1}{2} \int_0^T ||\bar{u}(\tau, \theta + \delta\theta) - u(t, \theta)||^2 dt, \quad \text{such that} \\ &\frac{d\bar{u}(\tau, \theta + \delta\theta)}{d\tau} = f(\bar{u}(\tau, \theta + \delta\theta), \theta + \delta\theta) \end{aligned} \tag{7.19}$$

which minimzes the difference between the shadowing trajectory $\bar{u}(\tau)$ and the original solution $u(t)$. This is in the literature called the *Least Squares Shadowing* (LSS) problem. The LSS problem is said to be well-conditioned, compared to the initial chaotic problem, which means that the condition number stays constant close to zero for the LSS, while it increases exponentially for the original problem [149].

We can now use this minimization problem to obtain a gradient of a chaotic dynamical system, as we now have a method of obtaining perturbation to a trajectory without it diverging from the original one. We are interested in the gradient of a loss function $L^\infty(\theta) = \lim_{T\to\infty} \frac{1}{T} \int_0^T g(u(t), \theta) dt$, as we discussed in section 3.2. We insert the shadowing trajectory in the loss function

$$L^\infty(\theta) \approx L^T_{lss}(\theta) = \frac{1}{\tau(T) - \tau(0)} \int_{\tau(0)}^{\tau(T)} g(\bar{u}(t), \theta) dt. \tag{7.20}$$

Because of the shadowing trajectory $\bar{u}$ is well-conditioned[153], the gradient of the loss function L_{lss} is also well defined at a finite time T. In contrast to the standard definition in eq. (7.14), the limit of $T \longrightarrow 0$ and the derivative $\delta\theta \longrightarrow 0$ now commmute, as the difference between $L^T_{lss}(\theta)$ at θ and $\theta + \delta\theta$ uniformly converges to the derivative as $\delta\theta$ vanishes [153].

[4] It should be noted that for systems that include non-hyperbolic points, the magnitude bounds in the shadowing lemma ϵ and δ needs to be chosen small.

We can compute the derivative $\frac{dL^T_{lss}(\theta)}{d\theta}$ by a linearization of the constrained minimization problem (LSS problem, eq. (7.19)). This can be obtained by Taylor expanding the shadowing solution $\bar{u}(\tau, \theta + \delta\theta)$, around $\delta\theta$, and then dividing by $\delta\theta$ in the LSS minimization problem (eq. (7.19)). We can write

$$\begin{aligned} &\bar{u}(\tau(t, \theta + \delta\theta), \theta + \delta\tau) = u(t) + v(t)\delta\theta, \\ &\text{where} \quad v(t) = \frac{d}{d\theta}\bar{u}(\tau(t,\theta), \theta) \end{aligned} \tag{7.21}$$

and

$$\tau(t, \theta + \delta\theta) = \int_0^t (1 + \eta(t')\delta\theta)dt', \quad \text{where} \quad \eta(t) = \frac{d}{d\theta}\frac{d\tau(t)}{dt} \tag{7.22}$$

where we have defined the time-dialation $\eta(t)$, which is introduced due to the time transformation $\tau(t)$. The new minimization problem can now be written in terms of $v(t)$, which is the quantity of interest in most sensitivity methods. Since we have changed the minimization problem to a problem in v, the construction needs to be similarly converted. This can be done by operating with $\frac{d}{d\theta}$ on the constraint from the LSS problem. Hence we obtain the constrained minimization problem in v [153]

$$\min_v \frac{1}{2}\int_0^T ||v||^2 dt, \quad \text{with} \quad \frac{dv}{dt} = \frac{\partial f}{\partial u}v + \frac{\partial f}{\partial \theta} + \eta f. \tag{7.23}$$

To obtain the derivative of the loss function, we can use the result from this problem directly in [153]

$$\frac{dL^T_{lss}}{d\theta} = \int_0^T \frac{\partial g}{\partial u}v(t) + \frac{\partial g}{\partial \theta} + \eta(t)(g(u(t)) - L(\theta))dt \tag{7.24}$$

We can understand the terms inside the integral as follows; the first term describes the contribution by the perpendicular distance between the shadowing trajectory and the original trajectory, and the second term is the changes in the loss function due to a change in the parameter (this also appears in the standard adjoint sensitivity method (eq. (3.29))). The last term is a correction term due to the time transformation $\tau(t)$. For a derivation of the equation above, we refer to the appendix of [153].

Now we have all the ingredients to construct a routine to solve the problem. When deriving the adjoint equation, we did a lengthy discussion on how to solve a similar constrained problem, where we have an optimization problem constrained by a differential equation. Therefore we set up a new Lagrangian of the following form [149]

$$\Gamma = \int_0^T \left(v^T v + \alpha^2 \eta^2 + 2\omega^T \left[\frac{dv}{dt} - \frac{\partial f}{\partial u} v - \frac{\partial f}{\partial \theta} - \eta f \right] \right), \tag{7.25}$$

where we introduced an adjoint variable ω[5]. Following the steps introduced in section 3.2, we arrive at the following set of equations [153]

$$\frac{dv}{dt} - \frac{\partial f}{\partial u} v - \frac{\partial f}{\partial \theta} - \eta f = 0 \tag{7.26}$$

$$\frac{dw}{dt} + \frac{\partial f}{\partial u}^T w - v = 0 \tag{7.27}$$

$$w(0) = w(T) = 0 \tag{7.28}$$

$$\alpha^2 \eta - w^T f = 0 \tag{7.29}$$

that we need to solve in order to obtain the sensitivity $\frac{dL^T_{lss}}{d\theta}$ by inserting the resulting values of $v(t)$.

The abovementioned algorithm is called the *Least Square Shadowing* sensitivity method [149]. It has, however, been much progress in this field in the last 10 years, both in terms of more efficient algorithms and in the reliability of the method. One of these improvements is the NILSAS (Non-Intrusive Least Squares Adjoint Shadowing) algorithm [152]. This also uses the shadowing lemma and transforms the linearized LSS problem (eq. (7.23)) into a discrete minimization problem subjected to a set of adjoint equations. These equations are the same as the ones we already have derived for the adjoint sensitivity method (eq. (3.27)). In practical applications, the NILSAS algorithm is implemented in a non-intrusive manner, meaning that it does not require any modifications to the underlying numerical solvers or the original dynamical system. This

[5] This have the same purpose as the adjoint variable λ for the standard adjoint sensitivity method.

is an important feature, as it enables the application of the algorithm to a wide range of existing simulation codes and models with minimal effort. In the next chapter, we will introduce the method and modify the parts needed to make it applicable to a stochastic differential equation.

7.4.1 Stochastic NILSAS

In the previous section, we introduced the simplest version of a least square shadowing sensitivity method, where the basic principle was to solve a constrained optimization problem to obtain the sensitivity. For many applications of sensitivity schemes, the adjoint equation is already accessible or can be implemented without many changes to the code. To this end, a more efficient strategy using the adjoint sensitivity equations has been constructed. This is called the NILSAS (non-intrusive least square adjoint shadowing) sensitivity method [152]. The central difference between these two methods is that the LSS uses the shadowing direction, while the NILSAS uses the adjoint shadowing direction.

The adjoint shadowing direction is similar to the shadowing direction, which is $v(t)$ for the LSS algorithm, i.e., the direction from the unperturbed trajectory $u(t)$ to the perturbed trajectory $u'(\tau)$. Instead of satisfying the tangent equation (eq. (7.26)), it satisfies the inhomogeneous adjoint [152] equation

$$\frac{d\overline{v}(t)}{dt} + \left(\frac{\partial f}{\partial u}\right)^T \overline{v}(t) = -\frac{\partial g(u,t)}{\partial u}, \quad \overline{v}(T) = 0 \tag{7.30}$$

where an important difference is that it contains $g(u,t)$, which is part of the loss function. The initial condition is defined at the last point T in the time span of the simulation of the original system. Since the equation is only defined in the reverse direction, the corresponding Lyapunov exponents and their corresponding covariance Lyapunov vectors are also the adjoint versions. This is nothing more than the same formalism, as derived in section 7.1, just in the reverse time direction. Note that this equation, as we discussed in chapter 3, is of an adjoint equation formulation as it contains the transpose of the Jacobian $\left(\frac{\partial f}{\partial u}\right)$ times the vector (VJP product).

The minimization problem used in this method is

$$\min_{a\in\mathbb{R}^M} \frac{1}{2}\int_0^T ||\overline{v}+\overline{W}a||^2, \quad \text{subjected to} \quad \int_0^T \langle \overline{v}+\overline{W}a, f(u,t)\rangle = 0 \tag{7.31}$$

where $\langle\cdot,\cdot\rangle$ is the inner product in Euclidean space, and $\bar{W}$ is a matrix containing a subspace of adjoint CLVs, which is set to $M > m_{us}$ where m_{us} is the number of unstable CLVs in the system. The matrix M spans the entire non-stable CLV subspace [152]. We can find $\bar{W}$ by the so-called homogenous equation [152]

$$\frac{d\overline{W}(t)}{dt} + \left(\frac{\partial f}{\partial u}\right)^T \overline{W}(t) = 0, \quad \overline{W}(T) = Q \tag{7.32}$$

where Q is any random unit vector. After solving the minimization problem eq. (7.31), we can obtain the sensitivity by applying $\overline{v}(t)$ to

$$\frac{dL}{d\theta} = \int_0^T \overline{v}(t)\frac{\partial f}{\partial\theta} + \frac{\partial g(u(t))}{\partial\theta}dt. \tag{7.33}$$

The way we calculate the NILSAS algorithm has many similarities to the Benettin algorithm [161] for the Lyapunov exponent calculation we discussed in section 7.1. We solve the homogenous- (eq. (7.32)) and the inhomogenous (eq. (7.30)) equation with a checkpointing scheme and re-orthonormalize the vector $\overline{v}(t)$ and $\overline{v}(t)$ at intermediate checkpoints. The full algorithm can be found in section 7.A.

The NILSAS algorithm is developed for deterministic systems, i.e., differential equations and not stochastic differential equations. The conversion into a stochastic differential equation can be done by changing the homogenous (eq. (7.32) and in-homogenous adjoint (eq. (7.30)) equation into the adjoint equation for an SDE eq. (3.32). As far as the author knows, no shadowing sensitivity method has been tested before for a chaotic stochastic differential equation. We will therefore apply the method to a known chaotic stochastic different equation, namely the stochastic Lorenz system [164] in the next section.

Adopting the algorithm for stochastic differential equations (SDEs) will follow the exact version of the deterministic case laid out in the previous section, except for the adjoint (homogenous and in-homogenous) equations. For an SDE of the form

$$dx = a(x, t, \theta)dt + b(x, t, \theta)dW, \tag{7.34}$$

with Wiener increments dW, we again use the interpolated noise version, such that $dW \longrightarrow \eta(t)dt$. We can now rewrite the equations as

$$\frac{d\overline{W}(t)}{dt} + \left(\frac{\partial a}{\partial u} + \frac{\partial b}{\partial u}\eta(t)\right)^T \overline{W}(t) = 0, \quad \overline{W}(T) = Q \tag{7.35}$$

for the homogenous equation and

$$\frac{d\overline{v}(t)}{dt} + \left(\frac{\partial a}{\partial u} + \frac{\partial b}{\partial u}\eta(t)\right)^T \overline{v}(t) = -\frac{\partial g(u,t)}{\partial u}, \quad \overline{v}(T) = 0 \tag{7.36}$$

for the inhomogeneous equation. We have seen that the adjoint sensitivity method for SDEs with the interpolating noise successfully yields gradients in the complex Langevin equation when the Lyapunov exponent is negative. A thorough investigation into when the approximation of interpolating noise should be carried out, including a derivation of the shadowing sensitivity method based on the shadowing lemma for a stochastic differential equation to see whether we need to correct for the noise in the cases where we e.g. have field dependence in the noise coefficients. This is, however, out of the scope of this thesis.

We now have an implementation of a shadowing sensitivity algorithm that can be applied to stochastic differential equations, such as the complex Langevin. In the next section, we will put it to the test on the stochastic Lorenz system.

7.4.2 Sensitivity of stochastic Lorenz system

As an illustrative example of implementing the shadowing sensitivity algorithm, we will utilize the stochastic version of the standard Lorenz system. The deterministic Lorenz system [164] is a popular toy model frequently employed for testing algorithms in chaotic systems, which originates back to Lorenz's observation of unusual behavior (now recognized as chaotic behavior) in a more extensive system of atmospheric convection. Lorenz and his colleagues sought to replicate this behavior in a smaller and simpler model, creating the Lorenz system, which exhibits the same chaotic patterns. The system of equations constituting the Lorenz system is given by

$$\begin{aligned} \dot{x} &= \sigma(y - x), \\ \dot{y} &= x(\rho - z) - y, \\ \dot{z} &= xy - \beta z, \end{aligned} \tag{7.37}$$

where x, y, and z represent the state variables, and σ, ρ, and β are positive parameters corresponding to the Prandtl number, Rayleigh number, and a geometric factor, respectively.

Typically, the values of the σ and β parameters are fixed at 10 and $\frac{8}{3}$, respectively, while the ρ parameter is varied. The system exhibits chaotic behavior when $\rho > 24.74$. For $\rho < 24.74$, the trajectory converges to one of the model's fixed points; however, in the chaotic regime, the trajectory experiences a repulsive force from each critical point, causing it to oscillate back and forth.

The deterministic Lorenz system is well-known for its sensitivity to initial conditions, leading to chaotic behavior often visualized as the iconic Lorenz attractor. This sensitivity presents a challenge for calculating gradients, as they increase exponentially with time. Consequently, the longer the system is simulated, the more significant the divergence between infinitesimal perturbations in initial conditions or model parameters regarding the final position.

Since the complex Langevin equation is a stochastic differential equation (SDE), we will evaluate the NILSAS method on a stochastic version of the Lorenz system by introducing an additive noise to each equation in the system. The modified system of equations is as follows:

$$
\begin{aligned}
dx &= (\sigma(y-x))dt + \alpha dW_1, \\
dy &= (x(\rho - z) - y)dt + \alpha dW_2, \\
dz &= (xy - \beta z)dt + \alpha dW_3,
\end{aligned}
\tag{7.38}
$$

where α is a positive constant, and $dW_1(t)$, $dW_2(t)$, and $dW_3(t)$ are independent Wiener processes. The stochastic terms account for random disturbances in the system.

This same model has been considered in [154, 159], where both obtained reliable sensitivities (derivatives) of the parameters ρ and σ. The methods considered in these papers differ from the shadowing methods, as they do not seek to find a shadowing trajectory but by inserting a spring term in the definition of the derivative that can force the trajectories to avoid diverging from each other.

In the presence of noise, the system can display various noise-induced phenomena, such as stochastic resonance, noise-induced transitions, and noise-enhanced stability. These phenomena stem from the interplay between deterministic and stochastic forces, resulting in unexpected system behaviors. We will assign the noise coefficient a value of $\alpha = 10$ and compute the gradient of the expectation value of the z-component, i.e.,

$$
\langle z \rangle = \frac{1}{T} \int_0^T z dt \tag{7.39}
$$

with respect to the parameters ρ and α.

We compare the results of the expectation value gradient to a fitting of a tangent line. To do this we compute expectation values around the desired parameters ρ and α and then fit a line through the points to measure the gradient, which can be seen in fig. 7.5. The change of $\langle z \rangle$ based on ρ follows a linear trend, while the changing α is not. To obtain the gradient at $\alpha = 10$, we fit a tangent line to the points close to $\alpha = 10$. The computed gradient is only an approximation of the true gradient. Still, we do not need higher accuracy to demonstrate the algorithm for an SDE optimization scheme. The fitted lines give the

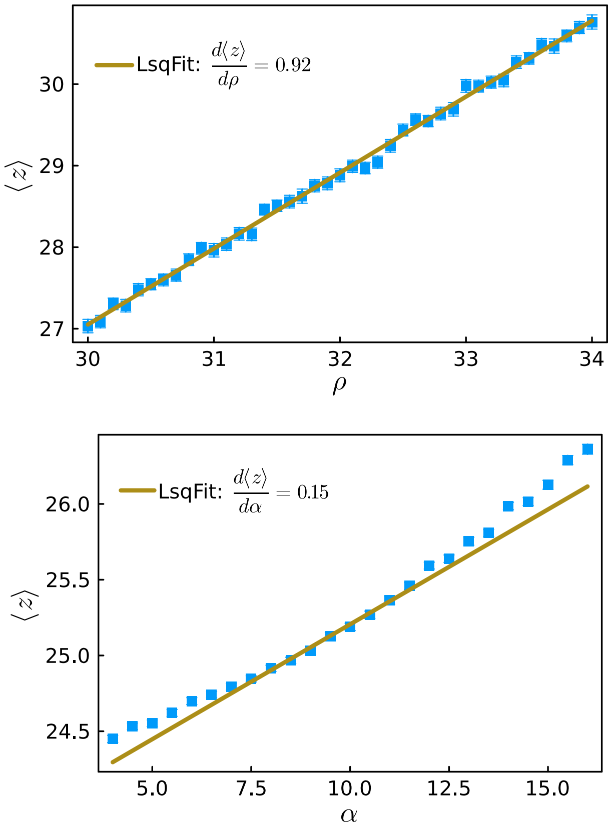

Figure 7.5: We compute the expectation value of $\langle z\rangle$ for different values of ρ (top) and α (bottom) around the point where we try to calculate the derivative to approximate the derivative. The fit a tangent model, for the fit to the change in α values (bottom), only the points close to α = 10 are used in the fit.

approximated gradient $\frac{d\langle z\rangle}{d\rho}\Big|_{\rho=32} \approx 0.92$ and $\frac{d\langle z\rangle}{d\alpha}\Big|_{\alpha=10} \approx 0.15$. The points are calculated by simulating 100 different trajectories from 0 to 100 in time, sampling at every 0.001. We compute the expectation values first for the individual trajectories, then take the mean of the trajectories and compute the standard error. We use the implicit Euler-Maruyama scheme (eq. (4.10) with $\theta = 1.0$).

We set up the NILSAS algorithm, see section 7.A for the full algorithm, with simulation time between the checkpoints by $\Delta T = 0.005$ and the setting $M = 3$ which correspond to the system having 2 unstable CLVs. The adjoint solution is obtained by solving an ODE containing all the equations in one, i.e., the two adjoint equations eqs. (7.35) and (7.36) and all the integrations in eq. (7.42). Since the system size is not big, we use the built-in linear solver in Julia to solve the linear system eq. (7.50).

To see the improvement when using the NILSAS method, we will compare the result of the NILSAS method with the forward auto-differentiation method section 3.1 and the adjoint sensitivity method. The auto differentiation (AD) version is computed by propagating the dual number through the full SDE simulation. The adjoint sensitivity solution uses the same method as discussed in section 3.2. We have also tested computing the gradients using the method in [80, 81], which applies a Brownian tree method for the noise process, both for the adjoint sensitivity method and in the NILSAS. Both of them have adjoint equations that diverge, such that no points were obtained to show here.

The results are shown in fig. 7.6 where we see $\frac{d\langle z\rangle}{d\rho}$ in the upper plot and $\frac{d\langle z\rangle}{d\alpha}$ in the lower plot. We find that the NILSAS method reproduces the gradient found in fig. 7.5. We calculate the gradient for 30 different trajectories and then compute the mean and standard error from these. Note that the AD and the adjoint result are shown as absolute values in the logarithm plot in the upper panel of the plots; we also remove the error bars when they cross below 0.

Our results show that the NILSAS algorithm yields accurate gradients. At the same time, the forward auto-differentiation and the adjoint method

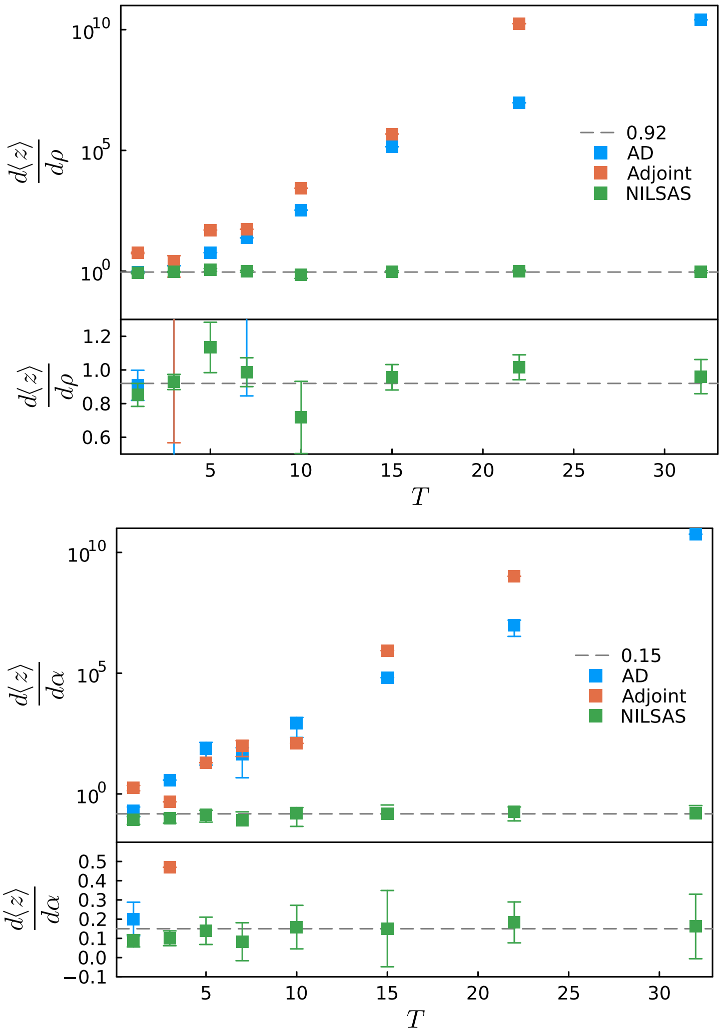

Figure 7.6: Comparison of the NILSAS algorithm, standard adjoint sensitivity method, and forward-AD to compute the change in the z-component due to change in (Top) ρ and (Bottom) α of the Stochastic Lorenz system. The lower panel in the two plots is a zoomed-in version to show the NILSAS result compared to the solution obtained by the tangent line in fig. 7.5, which is the dashed gray line. To be able to use the log axis in the upper panels, we have taken the absolute value of the AD result for the AD and adjoint result; we have also removed the errorbar of the values where the bar would extend from top to bottom in the plot to improve readability.

fail to produce reliable results, as expected, due to the chaotic nature of the stochastic Lorenz system. Furthermore, we observe that the error in the gradients computed using forward auto-differentiation increases exponentially with longer simulation times, highlighting the advantage of the NILSAS algorithm for sensitivity analysis in chaotic systems.

This is the first time that any shadowing sensitivity method has been applied to a stochastic differential equation. The method is stable for a long simulation time, although we see that it quickly reaches the long time average for both observables. This is in contrast to the adjoint sensitivity method, which tends towards the solution in the first two points but does not reach it before it exponentially increases.

7.5 Summary and concluding remarks

This chapter delves into the complexities facing the gradient related to the real-time complex Langevin simulations. We started by investigating the reason why the gradient of a loss function that encompasses a real-time Complex Langevin equation is ill-defined. To do this we introduced the Lyapunov exponents, which measure stability in a system. We applied it to both the free theory and the interacting theory of the real-time CLE and saw that for an increasing real-time, the Lyapunov exponent with the largest real-part (MLE) increased in both cases. For the free theory, the MLE tended towards 0, while in the interacting theory, we had a similar increase but now obtained a positive MLE. This examination hints at a transition in the dynamical system, where the complex Langevin converges to the wrong solution.

A positive MLE indicates a chaotic system, which we linked the failure of obtaining a reliable gradient to. The breakdown of the gradient happens because the limit of the simulation time T and the derivative limit do not commute. We showed an example using the adjoint sensitivity method to optimize the $\langle x \rangle$ observable for increasing real-time. We saw that its gradient optimization failed at the real-time extent where the Lyapunov exponent becomes positive.

To avoid this problem we reviewed the method of shadowing sensitivity analysis, which is a method to obtain gradients of chaotic dynamical systems. We then adopted the NILSAS scheme for stochastic differential equations, which we successfully applied to the stochastic Lorenz attractor. This shows the method's applicability to optimizing kernels in the real-time complex Langevin equation, where the standard sensitivity method fails.

We conclude that our study provides a framework to explain the stability and chaotic behavior of the real-time complex Langevin in terms of Lyapunov exponents. Future work will focus on building a tailored sensitivity method to the complex Langevin equation that can handle the chaoticity appearing in real-time systems. Further investigation into the shadowing sensitivity method or an alternative method is needed. If this succeeds, we can optimize kernels in the complex Langevin equation, extending the real-time contour extent.

7.A Appendix: NILSAS algorithm

To set up the NILSAS algorithm one needs to provide it with the number of unstable CLVs, such that $M \geq m_{us} + 1$, where m_{us} is the number of unstable CLVs, the number of segments K and the segment length ΔT. The total simulation time is $T = K\Delta T$. The algorithm is summarized below in a similar form as stated in [152].

- Integrate the primal system for sufficiently long time before $t = 0$ so that $u(t = 0)$ is on the attractor.
- Compute the trajectory $u(t)$, $t \in [0, T]$, by integrating the primal system.
- Generate terminal conditions for $\overline{W}i$ and $\overline{v}_i$ on the last segment $i = K - 1$:
  - (a) Randomly generate a $m \times M$ full rank matrix, Q'. Perform QR factorization: $Q_K R_K = Q'$.
  - (b) Set $p_K = 0$.

- Compute $\overline{W}_i$ and $\overline{v}_i$ on all segments. For $i = K - 1$ to $i = 0$ do:

  (a) To get $\overline{W}_i(t)$, whose columns are homogeneous adjoint solutions on segment i, solve:

  $$\frac{d\overline{W}_i}{dt} + \left(\frac{\partial a}{\partial u} + \frac{\partial b}{\partial u}\eta(t)\right)^T \overline{W}_i = 0, \quad \overline{W}_i(t_{i+1}) = Q \tag{7.40}$$

  To get $\overline{v}_i(t)$, solve the inhomogeneous adjoint equation:

  $$\frac{d\overline{v}_i}{dt} + \left(\frac{\partial a}{\partial u} + \frac{\partial b}{\partial u}\eta(t)\right)^T \overline{v}_i = -\frac{\partial g(u,t)}{\partial u}, \quad \overline{v}_i(t_{i+1}) = p_{i+1} \tag{7.41}$$

  (b) Compute the following integrations;

  $$C_i = \int_{t_i}^{t_{i+1}} \overline{W}_i \overline{W}_i^T dt, \quad d_i^{wv} = \int_{t_i}^{t_{i+1}} \overline{W}_i^T \overline{v} dt, \tag{7.42}$$

  $$d_i^{wf} = \int_{t_i}^{t_{i+1}} \overline{W}_i^T \left[a + b\eta(t)\right] dt, \tag{7.43}$$

  $$d_i^{vf} = \int_{t_i}^{t_{i+1}} \overline{v}^T \left[a + b\eta(t)\right] dt, \tag{7.44}$$

  $$d_i^{wf_s} = \int_{t_i}^{t_{i+1}} \overline{W}_i^T \left[\frac{\partial a}{\partial \theta} + \frac{\partial b}{\partial \theta}\eta(t)\right] dt, \tag{7.45}$$

  $$d_i^{vf_s} = \int_{t_i}^{t_{i+1}} \overline{v}^T \left[\frac{\partial a}{\partial \theta} + \frac{\partial b}{\partial \theta}\eta(t)\right] dt, \tag{7.46}$$

  $$d_i^{g\theta} = \int_{t_i}^{t_{i+1}} \frac{\partial g}{\partial \theta} dt, \tag{7.47}$$

  where $d_i^{wv}, d_i^{wf}, d_i^{wf_\theta} \in \mathbb{R}^M$; $d^{vf}, d^{vf_\theta}, d^{g\theta} \in \mathbb{R}$; $C \in \mathbb{R}^{M \times M}$ is the covariant matrix.

  (c) Orthonormalize homogeneous adjoint solutions via QR factorization:

  $$Q_i R_i = Wi(t_i) \tag{7.48}$$

  (d) Rescale the inhomogeneous adjoint solution using Q_i:

  $$p_i = \overline{v}_i(t_i) - Q_i b_i, \quad \text{where} \quad b_i = Q_i^T \overline{v}_i(t_i). \tag{7.49}$$

- Compute the adjoint shadowing direction $\{\overline{v}_i\}_{i=0}^{K-1}$.

    (a) Solve the NILSAS problem on multiple segments:

$$\min_{a_0,\ldots,a_{K-1}\in\mathbb{R}^M} \sum_{i=0}^{K-1}\left(\frac{1}{2}a_i^T C_i a_i + (dw_i^{wv})^T a_i\right), \quad \text{s.t.}$$

$$a) \quad a_{i-1} = R_i a_i + b_i, \quad i = 1, \ldots, K-1, \tag{7.50}$$

$$b) \quad \sum_{i=0}^{K-1}(d_i^{wf})^T a_i + \sum_{i=0}^{K-1} d_i^{vf} = 0.$$

    This is a least squares problem in $\{a_i\}_{i=0}^{K-1} \subset \mathbb{R}^M$. In appendix A of [152], a suggestion on how to solve this problem is laid forward using the Lagrange multipliers optimization method.

- Compute the derivative by:

$$\frac{dJ_{\text{avg}}}{ds} \approx \frac{1}{T}\sum_{i=0}^{K-1}\int_{t_i}^{t_{i+1}}\left(\bar{v}_i^T\left[\frac{\partial a}{\partial\theta} + \frac{\partial b}{\partial\theta}\eta(t)\right] + \frac{\partial g}{\partial\theta}\right)dt \tag{7.51}$$

$$= \frac{1}{T}\sum_{i=0}^{K-1}\left(d_i^{vf\theta} + a_i^T d_i^{wf\theta} + d_i^{g\theta}\right) \tag{7.52}$$

8 Summary and Outlook

In this thesis, I have put new ideas forward to improve the complex Langevin simulation of real-time dynamics of quantum systems on the Schwinger-Keldysh contour. I have mainly focused on improving the numerical stability (removing runaway trajectories) of the real-time complex Langevin using an implicit scheme and increasing the range of the real-time extent accessible with the complex Langevin equation using a novel optimization scheme to find a suitable kernel.

I explored the potential of implicit solvers for the real-time complex Langevin simulation in chapter 4. We showed that by using an unconditionally stable scheme, like the implicit Euler-Maruyama scheme, the real-time Langevin equation remained stable, and no runaway trajectories occurred, even on the canonical Schwinger-Keldysh contour. The implicit scheme made such a simulation possible for two reasons; first, it is a scheme that can handle stiff equations, and second, because the method undershoots the true trajectories, it adds an effective regulator to the action. We showed this analytically for the free theory case.

This allowed access to the backward path of the Schwinger-Keldysh contour, which is important to be able to simulate non-equilibrium physics on the Schwinger-Keldysh contour. Since the effective regulator is proportional to the Langevin step-size, we only need to obtain the limit of $dt \to 0$, and not both $dt \to 0$ and the regulator limit $\epsilon \to 0$ if a regulator is inserted by hand in the action or as a tilt of the contour.

Future work regarding implicit solvers is implementing an efficient version for gauge theories. I have not discussed the underlying construction in the main part of the thesis. Still, from this preliminary work carried out during this thesis project, I conclude that it can be achieved by the so-called stochastic RKMK [165, 166], a geometric version of the Runge-Kutta schemes for stochastic differential equations. In appendix A, I have introduced the main ingredients. The main idea in these schemes is to use the Magnus solution of the SDE, which is of the form [167]

$$U_k(t) = e^{\Omega_k(t)} U_k(0) \tag{8.1}$$

where $\Omega_k(t)$ is given by solving a Stochastic differential equation in the *Lie Alebra*. This method reduces to the geometric Euler-Maruyama scheme, used in most complex Langevin gauge theory simulations, e.g., [59, 65], if the SDE in the exponent is discretized with one step of the explicit Euler-Maruyama scheme. However, applying one step of an implicit scheme when discretizing $\Omega_k(t)$ corresponds to an implicit scheme when updating the links $U_k(t)$.

We have seen that a suitable kernel for the real-time complex Langevin can regulate the action. This means that using such a kernel does not need an implicit scheme or adaptive step-size to avoid runaway trajectories. This is because the kernel can be considered a coordinate transformation of the integration contour. It introduces a similar regularization as in the case of contour deformation, which also is a coordinate transform of the integration contour. This means that by introducing a suitable kernel we not only add an effective regulator to the action but also change the integration measure. This leaves the path integral measure intact. This is not the case when regulating the action by, e.g., tilting the contour or with the implicit scheme since this adds a regulator to the action without changing the integral measure. Stabilizing the CLE using a kernel in gauge theory is a promising direction and was tested successfully for an anisotropic kernel in [63]. Thus research into other kernels that can both regularize and stabilize the CLE is a promising way forward for real-time dynamics using complex Langevin.

We introduced in chapter 6 a novel optimization scheme, based on the direction of the drift term, that we showed was able to both introduce a regulator and correct convergence of the complex Langevin for intermediate real-time extents. The procedure optimizes a constant kernel by a gradient of the drift direction loss function (L_D) which does not need to evaluate the full stochastic differential equation, only the configurations sampled from the simulation. This allowed us to extend the real-time simulation up to $mt^{\max} = 1.5$ for the anharmonic oscillator on the canonical

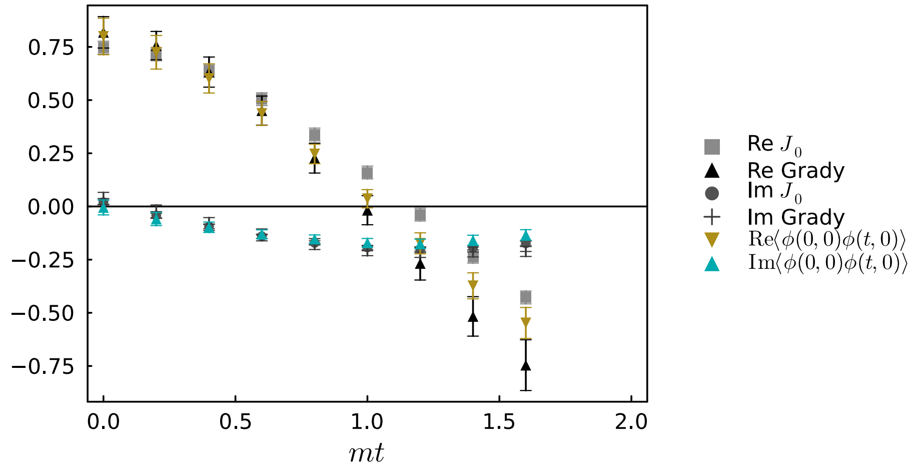

Figure 8.1: Simulation of the (1+1)D ϕ^4 field theory using $m = 1$, $\lambda = 1$, $\beta = 0.4$ and $mt^{\max} = 1.6$. The lattice size is $a_s = a_t = 0.2$ with 8 points in the spatial direction. We use the canonical Schwinger-Keldysh contour with 8 points forward and backward paths along the real-time axis and 2 in the Euclidean part. We compare the results obtained using the thimble method in [139], which are the black and gray points. Grady and J_0 are two different methods of obtaining the Jacobian. We plot the correlator at $p = 0$, such that $\phi(t, p = 0) = \frac{1}{N_s}\sum_n^{N_s}\phi_{t,n}$. The plot shows that the kernel-controlled CLE can obtain the same result as the current state-of-the-art direct real-time simulation method.

Schwinger-Keldysh contour. This is 3 times the extent possible without a kernel. We also showed that a kernel for the free theory could be obtained, which in the same way as the free theory propagator kernel, regularizes the action and reduces the simulation's relaxation time. For example, we optimized a kernel for $mt^{\max} = 10$, where the final kernel regulated the theory and allowed for convergence to the correct solution.

Future work to optimize kernels is to apply the scheme to (1+1)D and (2+1)D ϕ^4 field theory. The computational complexity of the complex Langevin scales proportionally to the volume of the lattice; while applied with a kernel this changes to volume squared. Although this will affect the computation time, small system sizes in (2+1)D scalar theory is still within reach with a medium-sized desktop machine. I show a preliminary result in fig. 8.1, where we compare the solution to the result obtained using the Lefschetz-Thimble method in [139]. The thimble result is the gray and black points, where we show both the thimble results from [139] using two different methods to obtain the Jacobian. We compute the real-time correlator in (1+1)D for $p = 0$, with $m = 1$, $\lambda = 1$, $\beta = 0.4$ and $mt^{\max} = 1.6$, we refer to [139] for details on the model setup. The result was obtained using the same optimization method discussed in section 6.4.

The kernel optimization scheme can potentially also be applied to gauge theories. A suitable loss function for theories can be to minimize the unitarity norm, as this has shown to be the dominant criterion to avoid convergence to the wrong solution. In principle had the potential of replacing both the gauge cooling [59] and dynamical stabilization [136]; kernel optimization is, however, more computationally expensive than those two methods as it would require a field-dependent kernel.

However, in the case of real-time simulation, we do not know if optimizing a kernel such that the field values sampled by the complex Langevin are localized close to the un-complexified manifold is optimal. For problems with a complex-valued Eucledian action, these have turned out, so far, to be optimal. At the same time, in real-time simulations, it is unclear whether this remains the important sampling region, i.e., it is unclear whether a small unitarity norm is still a valid criterion for successful large real-time extent.

An important part of further investigation into kernel optimization should be to find a better functional parameterization. By this, I mean to find a kernel parameterization that contains fewer parameters and, more importantly, can be scaled between real-time lengths. This would allow the optimization to start at a small real-time extent and then optimize the pre-trained kernel further in a longer real-time extent. For very long real-time extents, this is necessary. This would also allow for more flexible kernels which do not connect all space-time points but only the most important ones. Finding such a parameterization is hard, as we do not have any prior information on what form such a kernel should take.

Another improvement can be made by allowing for field-dependent kernels, which require an extra term in the complex Langevin equation (eq. (6.9)). This will allow the kernel to change depending on the field configuration, allowing for a much more specialized kernel. This has already been tested for a simple toy model in [70], using a field-dependent kernel that changes between two constant kernels based on the magnitude of x^2. This can, in principle, be extended to larger systems, like real-time simulation, and would be an interesting path forward for the optimization scheme.

However, the problem, in this case, is how to construct a suitable loss function. We proposed a systematic and system-specific optimization scheme in chapter 6 to find a suitable kernel for avoiding convergence to the wrong solution. This allows for the construction of kernels based on prior information in a systematic way by minimizing a loss function. In fact, by optimizing a kernel using the scheme laid out in section 6.4, we should, in principle, be able to get around the convergence to the wrong solution, and hence the sign problem, for systems in thermal Equilibrium. For these systems, we have prior knowledge such as symmetries of the theory, apriori known Eucledian correlation functions, and information about the boundary terms. However, we found that direct optimization of this loss function was problematic as the gradient is ill-defined at intermediate to long real-time extent.

The advantage of such a loss function is that we are much less susceptible to the kernel yielding a Fokker-Planck operator with positive eigenvalues. This can be achieved as we know the complex Langevin to yield correct convergence to some observables, like $\langle \phi(x) \rangle$ in the thermal ϕ^4 simulations. The loss function would strictly enforce that this stays true by, e.g., multiplying the L^{Sym} by a large factor.

We show in chapter 7 that to obtain a reliable gradient of the real-time complex Langevin, we need to overcome the obstacle of chaotic dynamical systems yielding ill-defined gradients. By computing the Lyapunov exponents, we showed that the real-time complex Langevin is a chaotic dynamical system. We saw that for increasing real-time extent that the Lyapunov exponent increased, and for the interacting theory, it became positive around the point where the simulations converged to the wrong solution. For a positive MLE, we know that the system is chaotic, meaning that it is very sensitive to changes in the initial conditions and parameters of the model. For systems with a positive Lyapunov exponent, we showed that computing the gradient numerically yielded an ill-defined gradient, such that any attempts to obtain a reliable gradient using standard methods, such as the adjoint sensitivity method, auto differentiation or finite-differences, would yield an incorrect gradient.

A possible way to get around this problem is to use a method to obtain the gradient of the loss with respect to the parameters, often called the sensitivity of the parameters, in a model by a method called least square shadowing (LSS) sensitivity method, which we discussed in section 7.4. These methods are capable of obtaining sensitivities of chaotic dynamical systems. I adopted a variant of this group of methods called the NILSAS (non-intrusive shadowing adjoint sensitivity) to be able to obtain sensitivities of stochastic differential equations. We use a new technique to solve the adjoint equation of the stochastic differential equation by interpolating the noise generated in the solution of the SDE. To our knowledge, this has not been used previously, and we see that it effectively stabilizes the adjoint equation for stochastic differential equations. The rest of the NILSAS algorithm was left unchanged. We tested the method using stochastic Lorenz equations, where we obtained correct gradients in contrast to the adjoint sensitivity method and forward auto differentiation, where the gradient error increased exponentially fast for increasing simulation time.

Although we have made progress toward direct real-time simulations, we are still far from the point where we can accurately read off the phenomenological relevant low-frequency part of the spectral function accurately. To quantify how far we need to go in real-time to get an accurate estimate of the spectral function, we will simulate the free theory using complex Langevin. By using the free theory propagator kernel, which we introduced in eq. (6.36), there is no restriction on the real-time extent from a convergence point of view. Therefore, we have tested with various real-time extents, namely 30, 100, 300, and 500, to see when we can read out the peak position of the spectral function within an error of 0.01. See details of the implementation in appendix B.

We saw that we were able at $mt^{\max} = 500$ to read off the spectral peak within an error of 0.01. While for the other $mt^{\max}$, the error amounted to more than 0.01. This is assuming we do not a priori know that the peak is a delta peak, as in this case, we could have fitted such a peak to the spectral function data. Another method would be to fit the real-time correlator, but this does not generalize to any real-time correlator, i.e., it again assumes the form of the spectral function. We need a direct simulation of real-time dynamics for systems where we do not know the spectral function's shape.

When simulating the complex Langevin for such a long real-time, discretization problems arise, which were not visible for shorter real-times, as is well known from the study of one-dimensional classical initial value problems[168]. This discrepancy is due to discretization in the time contour; in addition, the Langevin time discretization will also have an effect on the late time behavior. In the free theory, such discretization artifact arises as a scaling in the time direction, and we can show that it only amounts to a linear scaling in time. However, in interacting theories, this scaling arises both in the time direction and the correlator's magnitude. This means that much care must be taken for long real-time interacting theories to achieve high enough accuracy. The reason for this is the positive Lyapunov exponent, which tells us that the simulation requires high-accuracy solvers for the result to be reliable. If we apply the low-cost kernel optimization scheme to a real-time of $mt^{\max} = 5.0$, we see an indication of similar scaling artifacts as in the free theory. Although the optimized kernel-controlled CLE simulation converges to the wrong solution, we can recover the true solution by introducing a time-dependent scaling factor in the real-time correlator and time direction. It is clear that the CLE converges to the wrong result; however, it is surprising that scaling the CLE solution is sufficient to obtain the correct solution. As the complex Langevin at $tm^{\max} = 5.0$ have positive LEs after optimization, we can speculate that part of the failure is due to discretization artifacts either in Langevin time or in real-time. Future work into efficient high-order methods that can handle non-diagonal noise coefficients is necessary.

In conclusion, the kernel-controlled complex Langevin is a potentially effective way to resolve the sign problem in dynamic systems. By systematically incorporating prior knowledge through a kernel optimization scheme and a dependable gradient of the CLE, we can restore precise convergence and grant access to direct real-time simulations for more complex models.

A Implicit solvers for gauge theories

This section will briefly outline how to construct an implicit method for complex Langevin applied to gauge theories. In these models, the CLE is defined in terms of the gauge links U, such that the equation evolves on the corresponding manifold. The CLE for a $\mathrm{SU}(N)$ theory with the link variables $U_k \in \mathrm{SU}(N),\ k = 1, ..., N$, can we written as

$$dU_k = -\sum_a i\lambda_a \left\{U_k D_{a,k} S[U] dt + U_k \circ dW_{a,k}\right\} \tag{A.1}$$

where the group generators $\lambda_a,\ a = 1, ..., N^2 - 1$ satisfies $tr(\lambda_a \lambda_b) = 2\delta_{ab}$. The operator $D_{a,k}$ means the left Lie derivative operator, which can be computed as

$$D_{a,k} S[U] = \lim_{\epsilon \to 0} \frac{S[\tilde{U}^\epsilon] - S[U]}{\epsilon}, \quad \text{with} \quad \tilde{U}_l^\epsilon = e^{i\epsilon\delta_{kl}\lambda_a} U_l. \tag{A.2}$$

We emphasize that $\circ$ represents the Stratonovich interpretation of the stochastic integral. The Ito version can be written as

$$dU^j = -\sum_{a=1}^{n^2-1} i\lambda_a \left[U_k D_{a,k} S[U] dt + U_k dW_{a,k}\right] + \frac{1}{2} U_k dt \tag{A.3}$$

where we have used the conversion formula

$$\bar{A}(U) = A - \frac{1}{2} G(U) D_U G(U). \tag{A.4}$$

We can see that the continuum complex Langevin in the Ito sense can be rewritten as an exponential

$$dU_k = \exp\left(-\sum_a i\lambda_a \left\{D_{a,k} S[U] dt + dW_{a,k}\right\}\right) U_k \tag{A.5}$$

using the Ito calculus identities $dW_{a,k} dW_{b,j} = 2dt\ \delta_{a,b}\delta k, j$ and $dt^2 = dt\ dW_{a,k} = 0$. Now that we have the CLE written in Ito form, we can use numerical schemes for geometric SDEs in Ito form. At the point of writing no general scheme for Stratonovich integral form has been made.

The type of scheme we can construct falls under the stochastic Runge-Kutta Munte-Kaas (RKMK) scheme. RKMK schemes[165, 166] are used to solve ordinary differential equations on Lie groups by stepping along the time direction in the Lie Algebra and then using the exponential map to get it back to the group at each step. This limit the evolution of the links to the group. The main idea of these methods is to rewrite the evolution equation, eq. (A.5), in terms of the Magnus expansion. For a stochastic differential equation in the Ito sense, this is written as[167]

$$U_k(t) = e^{\Omega_k(t)} U_k(0). \tag{A.6}$$

where $\Omega(t)$ is given by a stochastic differential equation in the Lie algebra. The general form of this SDE is given by

$$\begin{aligned} d\Omega =& -\sum_a \{A_a dt + \Gamma_a dW\} && \text{(A.7)}\\ A_a =& \mathrm{dexp}_\Omega^{-1}\left(i\lambda_a D_{a,k} S[U]\right) + \mathrm{dexp}_\Omega^{-1}\left(C_a\right) && \text{(A.8)}\\ C_a =& \sum_{p=0}^{\infty}\sum_{q=0}^{\infty} \frac{1}{p+q+2}\frac{(-1)^p}{p!(q+1)!}\mathrm{ad}_\Omega^p\left(\mathrm{ad}_{\Gamma_a}\left(ad_\Omega^q\,\Gamma_a\right)\right) && \text{(A.9)}\\ \Gamma_a =& \mathrm{dexp}_\Omega^{-1}(i\lambda_a) && \text{(A.10)} \end{aligned}$$

where is derived in [167], see [169] for an introduction to the mathematical notation used in the equation above.

After the rewrite above, we can now discretize the Ω SDE in the same way as a normal non-geometric SDE, using already existing SDE schemes. We get the scheme used in most CL studies for gauge theories by evolving the SDE for Ω ones in the lowest order explicit discretization. This yields [167]

$$U_k^{(h+1)} = e^{-\sum_a i\lambda_a\left\{D_{a,k}S^{(h)}\Delta t + \Delta W_{a,k}^h\right\}} U_k^{(h)} \tag{A.11}$$

The lowest order implicit scheme with one step of the Ω SDE is given by

$$\begin{aligned} \Omega_1^{(h)} &= \sum_a i\lambda_a \left[\left(\theta D_a S[\exp(\Omega_1^{(h)}) U^{(h)}] \right.\right. \\ &\quad \left.\left. + \ (1-\theta) D_{a,k} S[U^{(h)}] \right) \Delta t + \Delta W_n^a \right] \\ U_k^{(h+1)} &= e^{\Omega_k^{(h)}} U_k^{(h)} \end{aligned} \tag{A.12}$$

where we first need to solve the non-linear equation to obtain $\Omega_1^{(h)}$, which we then apply directly to the update of the links in the next equation. We introduced the implicitness parameter θ in the same way as in the standard implicit Euler-Maruyama schemeeq. (4.10). For $\theta = 0$, we regain eq. (A.11) An efficient non-linear solver is necessary for this scheme to be sufficiently fast.

A.1 Simple example

The implicit solver was tested for a simple Polyakov chain model in $SU(2)$, where as expected, the solution follows on top of the explicit solver until it reaches a point where runaway trajectories occur. At this point, the explicit will diverges to infinity, while the implicit will, as seen in chapter 4, undershoot and avoid the runaway trajectory.

The model is given by the action [59]

$$S = -\frac{\beta}{2}\mathrm{Tr}[U_1 U_2 ... U_{N_l}] \tag{A.13}$$

where N_l is the number of links in the model. We use the same parameters as in [59], which is $\beta = \frac{1}{2}(1+\sqrt{3}i)$ and $N_l = 30$. When using the explicit scheme without adaptive step-size (gEM), this parameter has runaway trajectories. For adaptive step-size the CLE has no runaway solution but converges to the wrong solution. If we apply the Implicit scheme without adaptive step-size we also have no runaway trajectories, which means that it is stable. Still, the convergence to the wrong solution problem persists as expected[1].

[1] The code for the simulation can be found at GaugeSQM.jl

For this model, the advantage of the implicit scheme is limited since it does not improve the convergence rate. But since we see that it does, in fact, undershoot, which is the reason why it avoids the runaway trajectories, it can be useful in other problems. Examples are problems where a significantly smaller step-size is necessary in the adaptive scheme than the implicit scheme, which means that the implicit scheme can be faster. Another problem type is the real-time gauge theory models, where we saw in the scalar theories that the implicit scheme provided an effective regulator to the action.

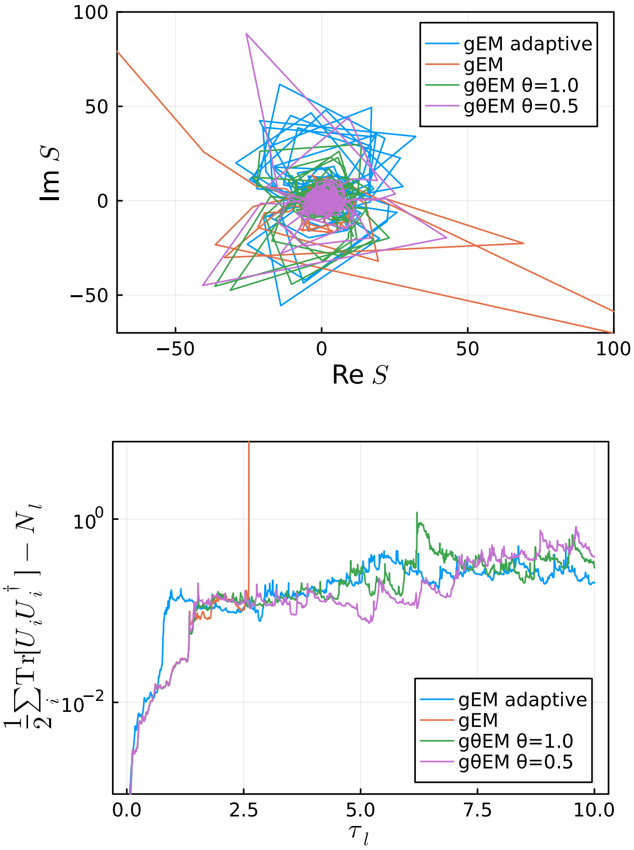

Figure A.1: Simulation of the $SU(2)$ Playakov chain with $N_l = 30$, and all except the adaptive scheme uses $dt = 10^{-3}$. (Top) The real and imaginary part of the action in Langevin time. We see the runaway trajectory of the explicit gEM scheme (orange). The other schemes avoid runaway trajectories but still accumulate boundary terms. (Bottom) The unitarity norm of the same schemes, where the explicit gEM diverges, and the other schemes stay close at a relatively large unitarity norm.

B Extracting the spectral function of free theory correlator

In this appendix, we will explore a real-time simulation up to $mt = 500$ of the free theory using the free theory propagator kerneleq. (6.36). This real-time extent is chosen to obtain a resolution of the spectral function around 0.01. The study of such kind of real-time simulations can be instructive in showing that the complex Langevin equation, with a suitable kernel, can correctly obtain the spectral function, which can explain all dynamical quantities.

We have used 5 points pr. unit real-time, such that a total of 2500 points is used in the forward path. Then an additional 2500 in the backward path, giving a total of 5004 points, including the 4 Eucledian points[1]. For the numerical setup, we use the implicit Euler-Maruyama schemeeq. (4.10), with the implicitness parameter θ set to $\theta = 0.5$. This is because we want as accurate a solver as we can. Since the introduction of a kernel both removed the need for a regulator and stabilized the simulation, the implicitness parameter only contributes to the accuracy. The scheme is similar to the Crank-Nicholsen scheme in ordinary differential equations.

In fig. B.1, we present two distinct segments of the forward correlator: the upper plot depicts the interval from $mt = 0$ to $mt = 50$, while the lower plot covers the range from $mt = 450$ to $mt = 500$. It is evident that within the initial segment ($mt = 0$ to $mt = 50$), the Complex Langevin Equation (CLE) converges to the accurate solution, as determined by solving the Schrödinger equation. However, a deviation between the CLE and Schrödinger solutions is observed at extended real-time intervals. This discrepancy grows linearly over time, affecting both the real and imaginary components. By applying a scaling factor of $mt' = 1.005mt$, the Schrödinger solution is recovered up to a real-time value of $mt = 500$.

[1] We use periodic boundary condition, which means that the last point at $-i\beta$ is the same as the first point

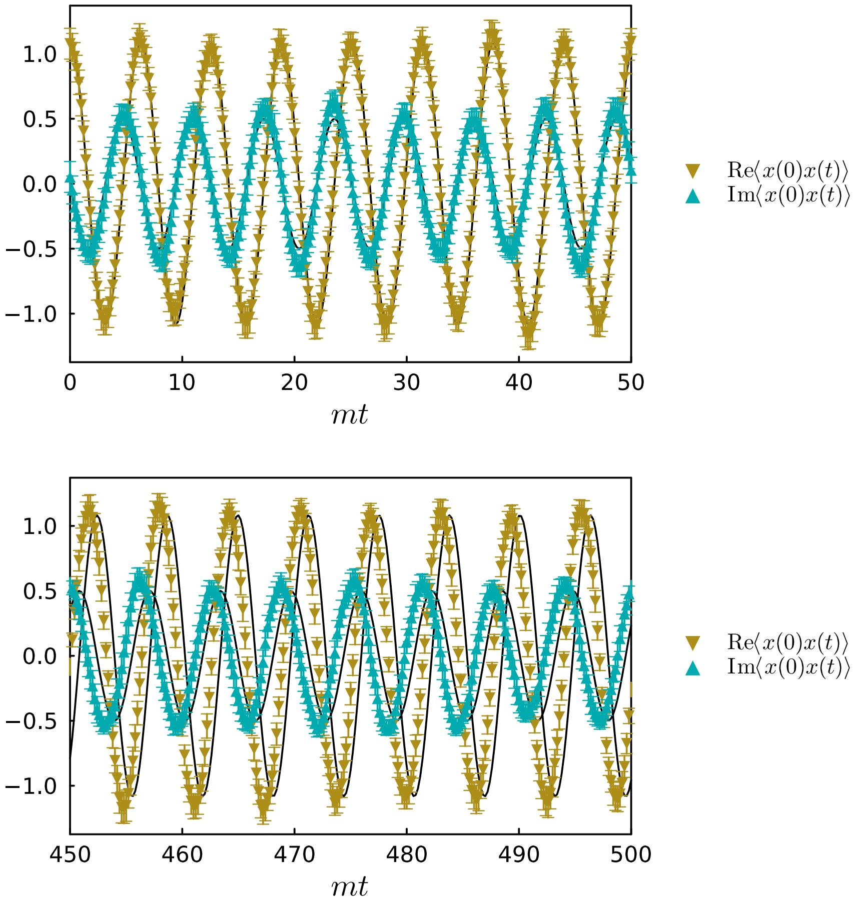

Figure B.1: Simulation of the free-theory harmonic oscillator on the canonical Schwinger-Keldysh contour up to $mt^{\text{max}} = 500$ in real-time. The top and lower panels show the correlator $\langle x(0)x(mt)\rangle$ at different times along the forward path; top from 0 to 50 and bottom from 450 to 500. We see a scaling in the CLE solution compared to the Schrödinger solution at large real-times.

The underlying cause of this observed shift is discretization artifacts, not the wrong convergence problem of the complex Langevin. There are two sources of such a discretization artifact; the dominant one is due to the discretization of the action, which means that by adding more points on the time contour, we reduce the deviation at large real-time. The second source of discretization artifacts is the numerical accuracy of the stochastic differential scheme. However, this is subdominant, and we have only seen a small improvement in accuracy in late real-time by increased solver accuracy. This deviation will be important in the interacting case, discussed in the next section, as similar scaling appears already in short real-time.

The spectral function is computed by the real-time correlator's discrete Fourier transform (DFT) from $mt = 0$ to $mt = 500$. To obtain a real output of the (DFT) we mirror it around zero and take the conjugate, such that $\langle(0)x(t)\rangle = \langle(0)x(-t)\rangle^*$. We also remove the last point to obtain a periodic signal. The DFT is done using the *FFTW* library, and the result is shown for the Scrödinger solution (QM) and the CLE solution in fig. B.2.

We see that the peak of the spectral functions is close to $\omega = 1.0$, which is the same as the Schödinger solution (QM). However, we get a wider peak for the complex Langevin version, indicating that the peak location is not exactly at 1.0. The reason for the peak being slightly off is due to the scaling we see in late real-time.

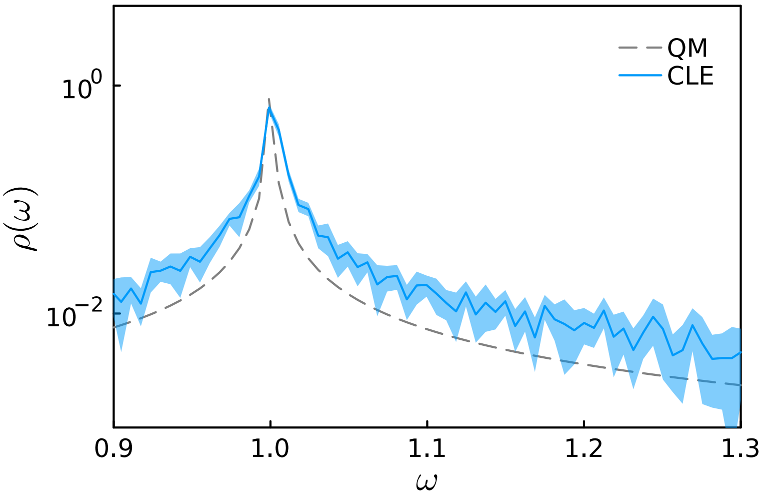

Figure B.2: The spectral function extracted from a real-time correlator simulated up to $mt^{\max} = 500$ in real-time. The gray dashed line is the Schrödinger solution with a peak position at $\omega = 0.999$, while the blue band in the result from the CLE simulation with a peak position at $\omega = 1.005$.

C Exploring long real-time behaviour

We saw in chapter 6 that the optimized kernel successfully regains the correct solution at intermediate real-time extent but falls short at real-times of $mt_{\max} = 2.0$. We see that the problems arise at the turning point at $mt_{\max}$, which can be explained by the contour turning back in the opposite direction. To avoid the sharp 180 turn, we change the time-contour to follow the real-axis up to $mt_{\max}$, and then tilt in the backward path such that it hits $-i\beta$ in the imaginary time path (see panel b in fig. 4.3). This path then consists only of two parts, the forward-path along real-time up to $mt_{\max}$ and then a second line from $mt_{\max}$ to $-i\beta$[1]. We call this contour the Schwinger-Kelysh right contour, which was used in ref. [135], not that we do not use a tilt in the forward path of the contour.

We follow the same steps as laid out in section 6.4.2 again using a real-time extent of $mt_{\max} = 5.0$, with 16 points per unit time in real-time. This corresponds to a lattice spacing of $a \approx 0.06$. The optimization is done by simulating 30 trajectories up to $\tau_{\mathrm{L}} = 10$ for every update step, and the final simulation is done with 100 trajectories up to 10 in Lengevin rime. Where we use the simulation with the smallest L^{prior} (eq. (6.47)).

The result for the forward correlator $\langle x(0)x(t)\rangle$ is shown in the upper panel of fig. C.1, which shows a correspondence with the Schrödinger solution only up to a short real-time and then start deviating. The later real-time is, however, still regaining the oscillatory pattern of a damped oscillator. We see a difference between this result and the one in section 6.4, where we are now able to correctly obtain the first point of the correlation, i.e., $\langle\phi^2(0)\rangle$, which was not the case in fig. 6.7. We observe that this difference is due to the change in contour from the canonical real-time contour in section 6.4 and the Schwinger-Keldysh right contour in fig. C.1.

Since we can correctly simulate up to intermediate real-times of $mt = 1.5$, as shown in section 6.4, we can use this to calibrate the longer real-time

[1] We could alternatively use the time-contour used in [116]. We find, however, that the tilted backward contour is more stable for the complex Langevin simulations.

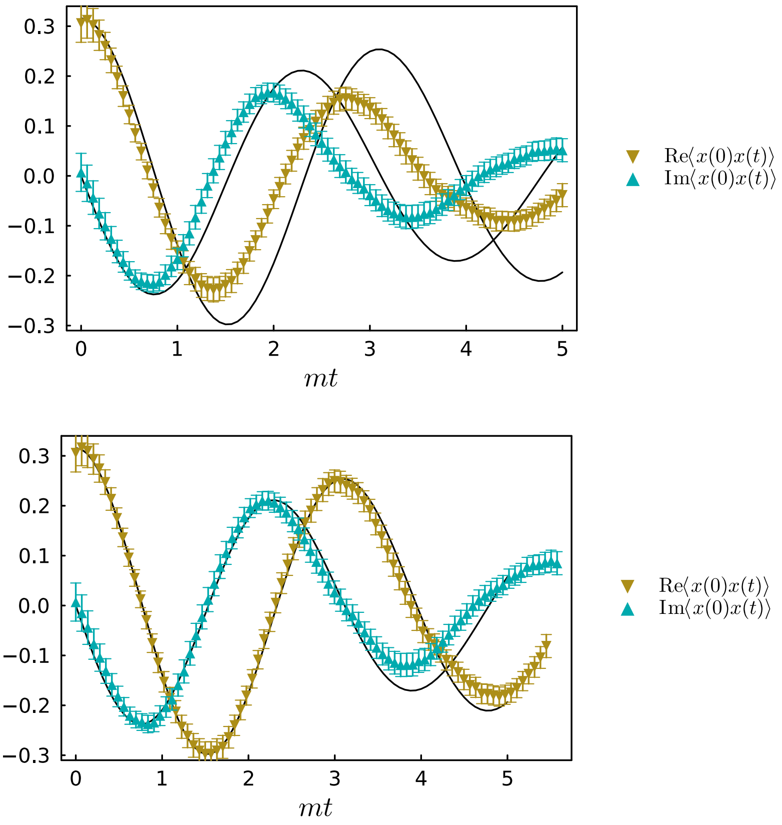

Figure C.1: Real-time simulation up to $mt^{\max} = 5.0$ on the Schwinger-Keldysh right contour (b in fig. 4.3) using the L_D loss function. (Top) The result after optimization, where we see the solution does not match the true solution (black line) given by the Scrödinger solution. (Bottom) Scaling the solution in mt direction and the correlator as explained in the text using based on the known time interval $mt = 0$ to $mt = 1.5$. We now see it matches close to the Scrödinger solution after the scaling.

simulation on the short real-time part. This is done in the lower panel of fig. C.1, where we have optimized 4 parameters to fit a linear interpolation of the solution to the $mt = 1.5$ simulation. The scaling is for the time $mt(1 + s_1)$ and scaling the correlator using $\langle x(0)x(t)\rangle(1 + mts_2)$.

The optimization uses a grid scan of the 4 parameters in a range between 0 and 0.4 for the four different parameters. The resulting parameters is $s_1^R = 0.09$, $s_2^R = 0.22$, $s_1^R = 0.115$ and $s_2^R = 0.13$.

Even though this scaling based on prior known real-time simulation does not rely on the true solution, we do not propose this update as a reliable scheme to be used for larger models. This is more of an interesting observation that we can obtain a simulation close to the true solution with only a simple scaling function in the real-time direction and the correlation magnitude.

The reason why such a scaling is possible is, however, unclear. We know from appendix B that scaling in the time direction can be caused by discretization artifacts due to lattice spacing and the accuracy of the numerical solver. We know from the discussion of fig. 4.2 that in the Interacting theory, discretization artifacts from the lattice spacing cause changes in the time direction and scaling in the correlator. This can therefore explain why we also need to scale the magnitude of the correlator.

D Generalized Thikonov

This appendix will delve into the fascinating topic of the inverse problem of spectral reconstruction in field theory. This is an area of active research with a broad range of theoretical and experimental physics applications. In this regard, we will apply adjoint methods to non-linear problems and not introduce the Bayesian spectral reconstruction method in detail. We refer the reader to ref. [2, 124], which provides an excellent and comprehensive review of this method.

I have added this appendix as an instructive application of the adjoint method to non-linear equations. This shows the power of the adjoint method, as we are able to optimize a matrix with $O(10^4)$ parameters by only solving one linear equation. See section 3.2 for the derivation of the method.

I will first introduce what spectral reconstruction is and the ill-posed problem that comes with it. Then I will define a new regulator, which we name the Generalized Thikonov regulator[16] that is based on a multivariate Gaussian, with a correlation matrix Σ. We will then define an optimization scheme to optimize the correlation matrix Σ, using the mock spectral function, and then define a new loss function that is used to optimize Σ that does not know about the mock spectral function.

D.1 Short introduction to the inverse problem of spectral reconstructions

The essence of this method lies in the fact that the Euclidean correlator shares vital information with the spectral representation of the real-time correlator; they both possess a common spectral function. The kernel function that facilitates the convolution from the spectral function to the Euclidean data is given by

$$K(\omega, \tau) = e^{-\omega\tau}. \tag{D.1}$$

where τ represents the Euclidean time index.

To express the inverse problem, we establish the connection between the discrete spectral function $\rho(\omega)$ over the Matsubara frequencies ω_l and the Euclidean correlator $D(\tau)$ through the discrete integral

$$D_i = \sum_{k}^{N_\omega} \Delta\omega_k K(\omega_k, \tau_i)\rho(\omega_k), \quad \text{where} \quad i \in [1, ..., N\tau] \tag{D.2}$$

with K denoting the Euclidean kernel (eq. (D.1)). Lattice computations in Euclidean time can only calculate a finite number of points, represented by N_τ, along the compact imaginary axis. The correlator points D_i are computed from the lattice data and are thus accompanied by a statistical error. Given a finite number of data points with a statistical error ΔD, determining the spectral function $\rho(\omega)$ becomes an inverse problem due to the lack of a one-to-one correspondence when inverting the equation, as $N_\tau \ll N_\omega$. Consequently, multiple spectral functions ρ can reproduce the data D_i within the statistical uncertainty. The task at hand, therefore, is to identify the most probable spectral function, considering both the available data and prior information about the system.

D.2 Bayesian spectral reconstruction

One approach to extract the spectral function from the Euclidean data via the inverse problem eq. (D.2) is to use *Bayesian inference* [13]. This approach uses the *Bayes theorem* to incorporate prior information as a regularization to construct a systematical prescription to tackle the ill-posed problem. Using the language of probabilities, we can formulate the problem in terms of the posterior distribution $P[\rho|D, I]$, which, given a spectral function ρ, denotes the probability to be correct based on the data D and some prior information I. Bayes theorem is defined as

$$P[\rho|D, I] = \frac{P[D|I]P[\rho|I]}{P[D|I]} \tag{D.3}$$

which relates the posterior distribution via the likelihood probability $P[D|\rho, I]$, the prior $P[\rho|I]$ and the evidence $P[D|I]$.

If we assume the input data to be normally distributed, we can define the likelihood function as

$$P[D|\rho, I] = Ne^{-L}, \quad L = \frac{1}{2}\sum_{jk}(D_j - D_j^{\rho})C_{jk}^{-1}(D_k - D_k^{\rho}) \tag{D.4}$$

where D^{ρ} is given by inserting ρ into eq. (D.2), D_i mean Euclidean data point at the ith time step and C_{ij} is the covariance matrix of the mean data points (D_i).

We diagonalize the covariance matrix such that the likelihood can be written in terms of the diagonal basis. The eigenvalues of the covariance matrix σ_k are used together with the transformed data $\tilde{D}_k$ and kernel $\tilde{K}$ to compute $\tilde{D}_k^{\rho}$. This yields the following likelihood, together with the other coordinate transformations [124]

$$L = \frac{1}{2}\sum_{k}\left(\tilde{D}_k - \tilde{D}_k^{\rho}\right)^2 / c_k^2, \quad \tilde{D}_k = \sum_j R_{kj}^{-1} D_j, \quad \tilde{K}_{jl} = \sum_k R_{jk}^{-1} K_{kl}. \tag{D.5}$$

We compute the maximum a posteriori distribution by finding the extremum of taking the log, which by defining the prior as $P[\rho|I] = e^S$ we get that we should maximize $-L + S = 0$. We find the maxima by multiplying by -1 and find where the derivative is 0, i.e., we solve the equation

$$\frac{\delta L}{\delta \rho} - \frac{\delta S}{\delta \rho} = 0, \tag{D.6}$$

which, in general, is a non-linear equation.

D.3 A generalized Tikhonov regulator

The next step is to define an appropriate regulator to incorporate prior information. There are three main regulates used for spectral reconstructions; these are the *Maximum Entropy Method* (MEM) [14, 13], the *Bayesian reconstruction* [15] (BR) method and the *Tikhonov regularization* [16]. The first two methods are the most prominent methods used for reconstructing positive spectral functions for lattice QCD, while the last one is popular in many different fields to solve inverse problems.

For testing purposes, we will use the Tikhonov regulator, which is defined by

$$S_{TK} = \frac{\alpha}{2}(\rho_l - m_l)^2 \tag{D.7}$$

where α is a hyperparameter and m_l is the default model which denotes the most probable apriori value of ρ_l. It stems from assigning the prior to be the product of normal distributions, such that $P[\rho|I] = \Pi_l^{N_\omega}\mathcal{N}[m_l, 1/\sqrt{\alpha}] \propto \exp[-S_{TK}]$. We omit the normalization as this can be factored into the α. A standard way to optimize the α parameter is by minimizing

$$J = (L - \frac{N_\tau}{2})^2 \tag{D.8}$$

where L is the likelihood function (eq. (D.5)).

This regulator has, however, some well-known drawbacks, the first of which is the oscillatory behavior (ringing effect) at small ω. This is a general artifact of spectral reconstruction due to a small number of input data. Since this regulator does not yield a positive spectral function by construct, it is more susceptible to this ringing effect. The second drawback is in reconstructing spectral functions from most lattice QCD data, as these are known to have positive spectral functions, which we cannot enforce with this regulator.

In this section, we will try to improve the Tikhonov method by generalizing the method to have correlations between the different ρ_ls. This is done by using the multivariate normal distribution as the prior, i.e.,

$$P[\rho|I] = e^{-(\rho_l - m_l)^T\Sigma_{lk}(\rho_k - m_k)} \tag{D.9}$$

where Σ is a positive semi-definite correlation matrix. The regulator now becomes

$$S_{GTK} = (\rho_l - m_l)\Sigma_{lk}(\rho_k - m_k). \tag{D.10}$$

The task at hand is now to optimize the correlation matrix Σ to incorporate more prior information than was possible with α. Note that Σ is a semi-positive definite matrix, so the optimization needs to consider this when updating the kernel's parameters. To enforce semi-positive definiteness, we use the Cholesky decomposition. In this way, the parameters of Σ are defined in a lower triangular matrix L, and then Σ is defined as $\Sigma = LL^T$. We have outlined the details of this in section D.A.

This is where the adjoint method comes in, as we can now use the adjoint sensitivity method for a non-linear equation to optimize such a matrix. For a matrix optimization of this size, with $O(10^4)$ parameters calculating gradients of a loss function using finite-difference or forward-difference would be very costly. We could have applied directly the reverse mode AD, but as we have discussed in chapter 3, using AD directly on the iterative method is not, in general, the best solution.

D.4 Optimization algorithm

We optimize a loss function $R(\Sigma, m)$ and minimize eq. (D.6) with S_{GKT}. Since L and the regulator S_{GKT} are linear terms after taking the derivative with respect to ρ, we can rewrite the minimization in terms of a linear system; $A(\Sigma)\rho = b(\Sigma, m)$. The optimization algorithm now works like this;

1. Initialize the $\Sigma^{\{0\}}$ matrix using a suitable parameterization, and set $m^{\{0\}}$ to zero. The parameterization of the two objects is merged into a vector $\mathbf{v}$.

2. Run a Bayesian reconstruction with the initial $\Sigma^{\{0\}}$ and $m^{\{0\}}$ to obtain $\hat{\rho}^{\{0\}}$, which in the Tikhonov case would be to solve the linear equation $A(\Sigma^{\{0\}})\rho = b(\Sigma^{\{0\}}, m^{\{0\}})$.

3. Calculate the adjoint linear system $\frac{\partial f}{\partial \rho}^T \lambda = \frac{\partial R}{\partial \rho}^T$, where $f(\hat{\rho}) = \frac{\partial L}{\partial \rho} - \frac{\partial S_{GKT}}{\partial \rho} = 0$. Insert the solution λ into the gradient of the loss function R with respect to the parameters of Σ and m; $\nabla_{\mathbf{v}} R|_{\rho=\hat{\rho}} = \frac{\partial R}{\partial v} - \lambda^T \frac{\partial f}{\partial v}$.

4. Update the parameters of α and m corresponding to the gradient calculated in the previous step using an optimization algorithm. This chapter will use the *ADAM* optimizer with an initial learning rate of $\eta = 0.01$. Then use the parameters to update Σ and m.

5. Rerun step 2 - 4, if no improvements have been made for the last 20 steps, decrease the learning rate by $5e-1$ and then continue the loop. Stop the optimization if the learning rate reaches the stopping criterion of $1e-5$.

D.4.1 Mock data

In this chapter, we will use mock spectral functions, constructing the spectral function using a distribution or a sum of distributions. We will be using relativistic Breit-Wigner distribution in different forms. A single relativistic Breit-Wigner distribution is defined by

$$\rho^{\mathrm{BW}}(\omega) = \frac{4A\Gamma\omega}{(M^2+\Gamma^2-\omega^2)^2+4\Gamma^2\omega^2}. \tag{D.11}$$

We construct the data from the mock spectral function

$$\tilde{D}_\tau = \sum_i^{N_\omega} d\omega_i K_{\tau,i}\rho(\omega_i). \tag{D.12}$$

To make the mock data $\tilde{D}$ generated from the mock spectral function ρ realistic, we also need to add noise. To do this we sample a set of configurations ($N_{\mathrm{Conf.}}$) with an added noise, i.e., the configuration γ is drawn using $D^\gamma = \tilde{D}+\xi$. The noise ξ is drawn from a normal distribution $\mathcal{N}(0, b_n^2)$ with variance given by $b_n = \sqrt{N_{\mathrm{Conf.}}} n\tilde{D}(p_\tau)$, where n is now a parameter to change the magnitude of the noise. The noise parameter will be set to $n = 10^{-4}$ for this test.

We use 3 different mock data B_1, B_2, B_3, which are all given by one or two BW peaks with the parameters;

- B_1: Single positive BW peak; $A_1 = 1.0$, $\Gamma_1 = 0.5$ and $M_1 = 3.0$.
- B_2: Two BW peaks, both positive; $A_1 = 0.8$, $\Gamma_1 = 0.5$ and $M_1 = 2.0$, and $A_1 = 1.0$, $\Gamma_1 = 0.5$ and $M_1 = 5.0$.
- B_3: Two BW peaks where one is positive and one negative; $A_1 = -0.15$, $\Gamma_1 = 0.8$, $M_1 = 1.5$ and $A_2 = 1.0$, $\Gamma_2 = 0.8$,$M_2 = 3.0$.

- B_4: Gluon-like spectra, with two equal sized BW peaks, one positive and one negative; $A_1 = -0.3$, $\Gamma_1 = 0.4$,$M_1 = 2.5$ and $A_2 = 1.0$, $\Gamma_2 = 0.8$,$M_2 = 4.0$.

For all the example below we use the same setup; $N_{\text{Conf.}} = 1000$, $n = 10^{-4}$, and $N_\tau = 30$, with $\delta\tau = 1$. We discretize the spectral function with $N_\omega = 500$ in the interval $\omega \in [0, 10]$.

D.5 Learning regulator from mock data

In this section, we will see what the Σ in the S_{GKT} looks like if we optimize it to an apriori-know spectral function. Although this scheme cannot be used for an unknown spectral function, it is interesting to see if we can recognize patterns in the Σ correlator, which we can then use in other examples. The optimization will use the loss function $R^{\text{True}}(\Sigma, m) = (\rho^{in} - \rho^{GKT}(\Sigma, m))$.

We will investigate a relativistic Breit-Wigner peak, with an additional negative BW peak in front to make the spectral function non-positive; this is the mock setup B_3. We have plotted the input peak, i.e., mock data (ρ^{in}), together with the optimized GKT and standard Tikhonov in fig. D.1. We see that we obtain, as expected, a spectral function that matches the mock spectral function.

In the lower panel of fig. D.1 we show Σ, which is a $N_\omega \times N_\omega$ matrix. This has a distinct structure for small ω. This pattern is undoing the oscillations occurring in the standard Tikhonov method. This means that to remove such oscillations, we need to correlate the different ω values in the regulator, and not only having a diagonal matrix or just a number α.

D.6 Loss function to optimize general Thikonov regulator

Instead of using the true mock function to learn the structure of Σ, let's see if we can find a different loss function that optimizes Σ to enforce other criteria. Examples of this are smoothness and positivity. I have

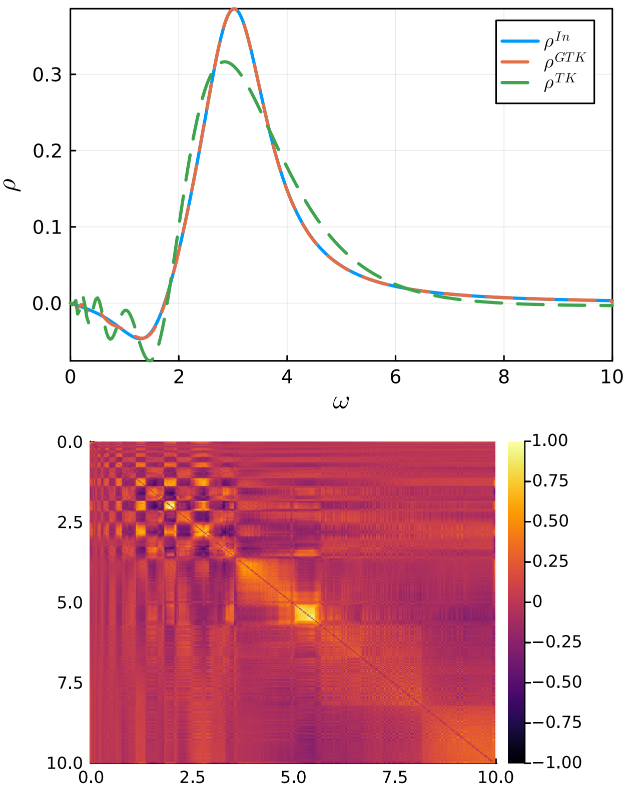

Figure D.1: (Top) Spectral reconstruction using the generalized Thikonov method (ρ^{GTK}) trained on the mock input spectral function ρ^{In}. We show the standard Thikonov method as an example of the oscillations appearing in small ω. (Bottom) A heatmap of the optimized Γ matrix kernel where the x- and y-axis represent the position in the matrix in terms of the corresponding ω value.

defined a set of unique loss functions that we will combine into the loss function R, on which we based the optimization.

$$R = R^{\square\rho} + R^{\text{Pos.}} + R^{\text{Linear}}. \tag{D.13}$$

The different terms are explained in the subsections below.

D.6.1 Smoothness

First out, we have a smoothness loss function, which minimizes the oscillations of the spectral function. This is based on the second-order derivative being small. This can potentially remove oscillations, but if this loss function is 0, we have a constant spectral function, which is also not what we want. We essentially want to tune the parameters such that this loss function is small but not zero. If we know the arc length of the spectral function, we could use this to constrain this loss function. The regulator is given by

$$R^{\square\rho} = \sum_{k=2}^{N_\omega} f(\omega_i)\Delta\omega(\rho_k - \rho_{k-1})^2 \tag{D.14}$$

where we choose the function $f(\omega_i) = \frac{1}{\omega_i}$, to make the focuse of this loss function at small ω where the oscillations are strongest.

D.6.2 Positivity

Next, we have a positivity loss function, which penalizes the negative parts of the spectral function. We have used a loss function that does not require the spectral function to be positive, i.e., which would be to use a log function or a step function at $\rho = 0$. I have found that a smooth function is preferable to avoid sudden changes in the optimization.

The loss function is a linear function for the spectral function less than some hyperparameter β; $\rho_i < \beta$, and exponential decaying when $\rho_i \geq \beta$, where the connection between the two is set such that the function becomes smooth, and differential at all points. The function is as follows

$$R^{\text{Pos.}} = \sum_{k=1}^{N_\omega} \tilde{f}(\omega_i) \begin{cases} -(\rho_i - \beta) + \gamma, & \rho_i < \beta \\ \gamma e^{-(\rho_i - \beta)/\gamma}, & \rho_i \geq \beta \end{cases} \tag{D.15}$$

where γ and β are two real parameters to tune. We will use $\gamma = 0.05$ and $\beta = 0.5$. The idea behind this regulator is that we want to push the spectral function to be positive without setting a strict boundary at 0. We use $\tilde{f}(\omega) = \exp[-\omega]$ which again focus the on the omegas close to 0. This, by definition, is normalized to 1 at $\omega = 0$.

D.6.3 Linear loss

The last regulator function is a linear loss function that penalizes high positive peaks, which is enforced at a large ω span. The linear loss function is

$$R^{\text{Linear}} = \sum_{k=1}^{N_\omega} \bar{f}(\omega_i) a \rho_i^2 \tag{D.16}$$

where $a = \Delta\omega$ in this section and $\bar{f}(\omega) = \omega^2$ is the a function prioritizing the intermediate to large ω values.

D.7 Learning General Tikhonov regulator

In this section, we present the result for the four mock spectra B_1,B_2,B_3 and B_4, using the loss function from the previous section and the optimization scheme from section D.4. The results, compared to the standard Thikhonov method, are shown in fig. D.2. We see that the ρ^{GKT} generally has a peak position and peak width closer to the mock spectral function for all the tested mock spectral functions.

In B_1 and B_2, which both have positive spectral functions, the method removed most of the oscillations for small ω. For B_3 and B_4, some small oscillations are left from small ω, but it is significantly reduced compared to the standard Tikhonov method, especially for B_4, which has very strong oscillations.

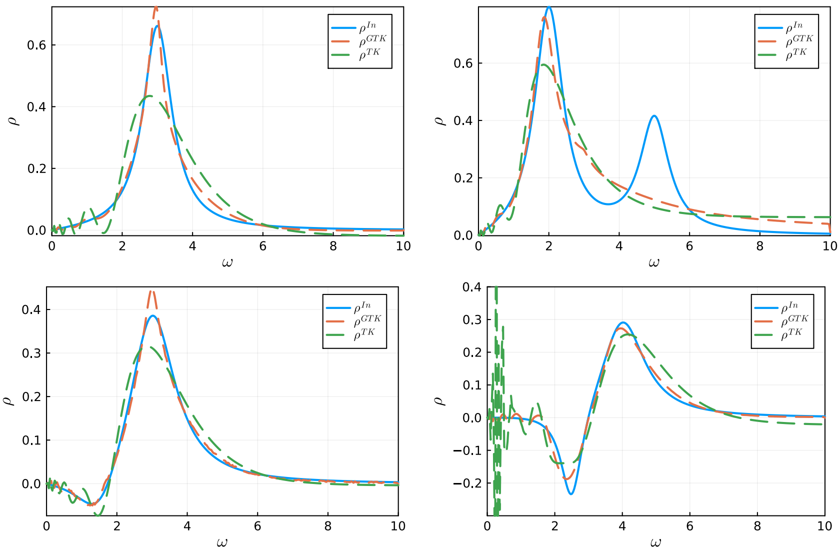

Figure D.2: Result using the Generlized Tikhonov (ρ^{GKT}) after optimizing the regulator parameters Σ and m, using the loss function R. The spectral function is compared to the mock spectral function ρ^{In} and to the stanrard Tikhonov ρ^{TK} where α is optimized using $L - \frac{N_\tau}{2} = 0$. We use the different BW mock data; top left is B_1, top right is B_2, bottom left is B_3 and bottom right is B_4.

D.8 Summary

In this appendix, we have introduced a generalized Thikhonov regulator based on the multivariate Gaussian distribution such that the different ω points are correlated in the regulator. We were then able to optimize a regulator such that the oscillations disappear, where we saw that the oscillation pattern was removed by a similar pattern in Σ. An interesting next step would be to find a functional parameterization of this kernel and see if this can be generalized to other mock spectral functions.

We then introduced an optimization scheme with a corresponding loss function that improved the Tikhonov method. Although the scheme, first and foremost, acts as a test of the adjoint sensitivity method, it is still a promising sign that we can improve the well-established Tikhonov method for the mock data used here, which is still used in some applications.

D.A Positive definite matrix optimization

In many applications, it is necessary to optimize a positive definite matrix. This can be done by minimizing the matrix to be positive definite, i.e., incorporating it into the loss function or forcing it to be positive definite by construction. The advantage of the latter one is that one does not need to use additional computational cost in the optimization scheme to also optimize for a positive definite matrix. But introducing the constraint on the matrix in the loss function might lead to a better-behaving optimization problem; this depends on the problem at hand.

In this appendix, we will show how one can force the matrix to be positive definite by construction. A positive definite matrix is defined by the matrix satisfying $x^T M x > 0$ for all $x \in \mathbb{R}^n \setminus \{0\}$, which is equivalent to the matrix being symmetric with positive eigenvalues. Since we want to have a positive definite matrix by construction we can use the Cholesky decomposition. This decomposes the matrix into a lower-triangular matrix L with a positive diagonal such that the positive definite matrix M is defined as $M = LL^T$. This means that the entries of the off-diagonal elements of L are only required to be real, while the diagonal must consist of positive entries.

The parameterization of $M \in \mathbb{R}^{N\times N}$ is now done via the lower-triangular entries of $L \in \mathbb{R}^{N\times N}$, such that the number of parameters $L_{ij} \in \mathbb{R}$ are $\frac{N^2+N}{2}$. To optimize the parameters of the matrix M, we, therefore, need to take the gradient of the matrix with respect to the lower-triangular entries of the matrix L, which is

$$\frac{\partial}{\partial L_{ij}} M_{kl} = \frac{\partial}{\partial L_{ij}} L_{km} L_{lm} = \delta_{ki}\delta_{jm} L_{lm} + L_{km}\delta_{il}\delta_{jm} = \delta_{ki} L_{lj} + L_{kj}\delta_{il} \tag{D.17}$$

for $i > j > 1$ and $i \in \{1, ..., N\}$. We can parameterize the entries with a log function to force the diagonal entries of L to be positive.

D.B Gradient of the generalized Thikonov regulator

To optimize the parameters of the general Thikonov regulator we need the gradient of the regulator with respect to the parameters and the spectral function. The partial derivative with respect to the spectral function is trivial as we are only left with the Σ matrix.

$$\frac{\partial}{\partial \rho_i} \frac{\partial S_{GKT}}{\partial \rho_j} = \Sigma_{ij} \tag{D.18}$$

For the parameters of the Σ matrix, which is required to be positive definite, we need to take the gradient of the parameters L_{ij}, and the resulting

$$\begin{aligned} \frac{\partial}{\partial L_{ij}} \frac{\partial S_{GKT}}{\partial \rho_k} &= \frac{\partial}{\partial L_{ij}} \Sigma_{kl}(\rho_l - m_l) = (\delta_{ki} L_{lj} + L_{kj}\delta_{il})(\rho_l - m_l) \\ &= \delta_{ki} L_{lj}(\rho_l - m_l) + L_{kj}(\rho_i - m_i). \end{aligned} \tag{D.19}$$

This can be implemented efficiently in an adjoint equation by first calculating the matrix-vector product $L(\rho - m)$, then identifying the $\lambda_i^T \delta_{ij}$ as a row-vector in the first term, calculating $\lambda^T L$ in the second term and then indexing from the vector jacobian products (VJP). See section 3.1.2 for more information.

We parameterize the elements with a log function to keep the diagonal positive. This means that the gradient for the diagonal elements of L is changed to be

$$\begin{aligned}\frac{\partial}{\partial L_{ii}}\frac{\partial S_{gKT}}{\partial \rho_k} &= \frac{\partial}{\partial L_{ii}}\Sigma_{kl}(\rho_l - m_l) = (\frac{\delta_{ki}}{L_{ii}}L_{li} + L_{ki}\frac{\delta_{il}}{L_{ii}})(\rho_l - m_l) \\ &= \frac{1}{L_{ii}}L_{li}(\rho_l - m_l) + L_{ki}\frac{1}{L_{ii}}(\rho_i - m_i).\end{aligned} \tag{D.20}$$

References

[1] Sz. Borsanyi, S. Durr, Z. Fodor, C. Hoelbling, S. D. Katz, S. Krieg, L. Lellouch, T. Lippert, A. Portelli, K. K. Szabo, and B. C. Toth. Ab initio calculation of the neutron-proton mass difference. *Science*, 347(6229):1452–1455, 2015.

[2] Alexander Rothkopf. Heavy quarkonium in extreme conditions. *Physics Reports*, 858:1–117, May 2020.

[3] Nora Brambilla, Simon Eidelman, Christoph Hanhart, Alexey Nefediev, Cheng-Ping Shen, Christopher E. Thomas, Antonio Vairo, and Chang-Zheng Yuan. The XYZ states: experimental and theoretical status and perspectives. *Phys. Rept.*, 873:1–154, 2020.

[4] K. Yagi, T. Hatsuda, and Y. Miake. *Quark-Gluon Plasma: From Big Bang to Little Bang*, volume 23. 2005.

[5] Panagiota Foka and Małgorzata Anna Janik. An overview of experimental results from ultra-relativistic heavy-ion collisions at the CERN LHC: Bulk properties and dynamical evolution. *Rev. Phys.*, 1:154–171, 2016.

[6] Chih-Chun Chien, Sebastiano Peotta, and Massimiliano Di Ventra. Quantum transport in ultracold atoms. *Nature Physics*, 11(12):998–1004, December 2015.

[7] J. Robert Schrieffer and James S. Brooks. *Handbook of High-Temperature Superconductivity : Theory and Experiment*. 2007.

[8] Kuang-chao Chou, Zhao-bin Su, Bai-lin Hao, and Lu Yu. Equilibrium and Nonequilibrium Formalisms Made Unified. *Phys. Rept.*, 118:1–131, 1985.

[9] N. P. Landsman and C. G. van Weert. Real and Imaginary Time Field Theory at Finite Temperature and Density. *Phys. Rept.*, 145:141, 1987.

[10] J. Smit. *Introduction to Quantum Fields on a Lattice: A Robust Mate*, volume 15. Cambridge University Press, January 2011.

[11] Christof Gattringer and Kurt Langfeld. Approaches to the sign problem in lattice field theory. *International Journal of Modern Physics A*, 31(22):1643007, August 2016.

[12] Gaopei Pan and Zi Yang Meng. Sign Problem in Quantum Monte Carlo Simulation. April 2022.

[13] Mark Jarrell and J. E. Gubernatis. Bayesian inference and the analytic continuation of imaginary-time quantum Monte Carlo data. *Physics Reports*, 269(3):133–195, May 1996.

[14] R. K. Bryan. Maximum entropy analysis of oversampled data problems. *European Biophysics Journal*, 18(3):165–174, April 1990.

[15] Yannis Burnier and Alexander Rothkopf. Bayesian Approach to Spectral Function Reconstruction for Euclidean Quantum Field Theories. *Phys. Rev. Lett.*, 111(18):182003, October 2013.

[16] A. N. Tikhonov. On the stability of inverse problems. *Proceedings of the USSR Academy of Sciences*, 39:195–198, 1943.

[17] Martin Hansen, Alessandro Lupo, and Nazario Tantalo. Extraction of spectral densities from lattice correlators. *Phys. Rev. D*, 99(9):094508, 2019.

[18] Jan Horak, Jan M. Pawlowski, José Rodríguez-Quintero, Jonas Turnwald, Julian M. Urban, Nicolas Wink, and Savvas Zafeiropoulos. Reconstructing QCD spectral functions with Gaussian processes. *Physical Review D: Particles and Fields*, 105(3):036014, 2022.

[19] Lukas Kades, Jan M. Pawlowski, Alexander Rothkopf, Manuel Scherzer, Julian M. Urban, Sebastian J. Wetzel, Nicolas Wink, and Felix P. G. Ziegler. Spectral reconstruction with deep neural networks. *Physical Review D: Particles and Fields*, 102(9):096001, 2020.

[20] C. E. Berger, L. Rammelmüller, A. C. Loheac, F. Ehmann, J. Braun, and J. E. Drut. Complex Langevin and other approaches to the sign problem in quantum many-body physics. *Physics Reports*, 892:1–54, January 2021.

[21] Mingpu Qin, Thomas Schäfer, Sabine Andergassen, Philippe Corboz, and Emanuel Gull. The hubbard model: A computational perspective. *Annual Review of Condensed Matter Physics*, 13:275–302, 2022.

[22] G. Parisi and Yong-shi Wu. Perturbation Theory Without Gauge Fixing. *Sci. Sin.*, 24:483, 1981.

[23] Naomi Rom, DM Charutz, and Daniel Neuhauser. Shifted-contour auxiliary-field Monte Carlo: Circumventing the sign difficulty for electronic-structure calculations. *Chemical physics letters*, 270(3-4):382–386, 1997.

[24] Andrei Alexandru, Gokce Basar, Paulo F. Bedaque, and Neill C. Warrington. Complex paths around the sign problem. *Reviews of Modern Physics*, 94(1):015006, 2022.

[25] Roman Orus. Tensor networks for complex quantum systems. *Nature Reviews Physics*, 1(9):538–550, September 2019.

[26] Johannes Hauschild and Frank Pollmann. Efficient numerical simulations with Tensor Networks: Tensor Network Python (TeNPy). *SciPost Physics Lecture Notes*, October 2018.

[27] Denis Boyda et al. Applications of machine learning to lattice quantum field theory. In *Snowmass 2021*, February 2022.

[28] M. S. Albergo, G. Kanwar, and P. E. Shanahan. Flow-based generative models for Markov chain Monte Carlo in lattice field theory. *Physical Review D: Particles and Fields*, 100(3):034515, 2019.

[29] Shuzhe Shi, Lingxiao Wang, and Kai Zhou. Rethinking the ill-posedness of the spectral function reconstruction — Why is it fundamentally hard and how Artificial Neural Networks can help. *Computer Physics Communications*, 282:108547, January 2023.

[30] Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: A survey, February 2018.

[31] Daniel Alvestad, Rasmus Larsen, and Alexander Rothkopf. Towards learning optimized kernels for complex Langevin. *JHEP*, 04:057, 2023.

[32] Matthias Troyer and Uwe-Jens Wiese. Computational complexity and fundamental limitations to fermionic quantum Monte Carlo simulations. *Phys. Rev. Lett.*, 94:170201, 2005.

[33] Mikko Laine and Aleksi Vuorinen. *Basics of Thermal Field Theory*, volume 925. Springer, 2016.

[34] Jürgen Berges. Introduction to nonequilibrium quantum field theory. *Aip Conference Proceedings*, 739(1):3–62, 2004.

[35] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[36] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid monte carlo. *Physics Letters B*, 195:216–222, 1987.

[37] Philippe de Forcrand. Simulating QCD at finite density. *PoS*, LAT2009:010, 2009.

[38] Daniel Alvestad, Rasmus Larsen, and Alexander Rothkopf. Stable solvers for real-time Complex Langevin. *JHEP*, 08:138, 2021.

[39] Gert Aarts, Frank A. James, Erhard Seiler, and Ion-Olimpiu Stamatescu. Complex Langevin dynamics: Criteria for correctness. *PoS*, LATTICE2011:197, 2011.

[40] Daniel Alvestad, Rasmus Larsen, and Alexander Rothkopf. Kernel controlled real-time Complex Langevin simulation. *PoS*, LATTICE2022:002, 2023.

[41] Daniel Alvestad, Rasmus Larsen, and Alexander Rothkopf. Kernel controlled real-time Complex Langevin simulation. *Epj Web of Conferences*, 274:08001, 2022.

[42] H. Huffel and H. Rumpf. Stochastic Quantization in Minkowski Space. *Phys. Lett. B*, 148:104–110, 1984.

[43] Giorgio Parisi. On complex probabilities. *Physics Letters B*, 131(4):393–395, 1983.

[44] J. R. Klauder. STOCHASTIC QUANTIZATION. *Acta Phys. Austriaca Suppl.*, 25:251–281, 1983.

[45] Gert Aarts, Pietro Giudice, and Erhard Seiler. Localised distributions and criteria for correctness in complex Langevin dynamics. *Annals of Physics*, 337:238–260, October 2013.

[46] John R. Klauder and Wesley P. Petersen. Spectrum of certain non-self-adjoint operators and solutions of Langevin equations with complex drift. *Journal of Statistical Physics*, 39(1-2):53–72, April 1985.

[47] Erhard Seiler. Status of Complex Langevin. *EPJ Web of Conferences*, 175:01019, 2018.

[48] Erhard Seiler, Dénes Sexty, and Ion-Olimpiu Stamatescu. Complex Langevin: Correctness criteria, boundary terms and spectrum. April 2023.

[49] Jan Ambjorn and S. K. Yang. Numerical Problems in Applying the Langevin Equation to Complex Effective Actions. *Phys. Lett. B*, 165:140, 1985.

[50] Jon Flower, Steve W. Otto, and Sean Callahan. Complex Langevin equations and lattice gauge theory. *Physical Review D*, 34(2):598–604, 1986.

[51] Jan Ambjorn, M. Flensburg, and C. Peterson. The Complex Langevin Equation and Monte Carlo Simulations of Actions With Static Charges. *Nucl. Phys. B*, 275:375–397, 1986.

[52] Gert Aarts, Frank A. James, Erhard Seiler, and Ion-Olimpiu Stamatescu. Adaptive stepsize and instabilities in complex Langevin dynamics. *Phys. Lett. B*, 687:154–159, 2010.

[53] M. Namiki, I. Ohba, K. Okano, Y. Yamanaka, A. K. Kapoor, H. Nakazato, and S. Tanaka. *Stochastic Quantization*, volume 9. 1992.

[54] H. Okamoto, K. Okano, L. Schulke, and S. Tanaka. The Role of a Kernel in Complex Langevin Systems. *Nucl. Phys. B*, 324:684–714, 1989.

[55] Keitaro Nagata, Jun Nishimura, and Shinji Shimasaki. Argument for justification of the complex Langevin method and the condition for correct convergence. *Physical Review D*, 94(11):114515, December 2016.

[56] Gert Aarts, Erhard Seiler, Denes Sexty, and Ion-Olimpiu Stamatescu. Complex Langevin dynamics and zeroes of the fermion determinant. *JHEP*, 05:044, 2017.

[57] Gert Aarts, Frank A. James, Erhard Seiler, and Ion-Olimpiu Stamatescu. Complex Langevin: Etiology and diagnostics of its main problem. *The European Physical Journal C*, 71(10):1756, October 2011.

[58] M. Scherzer, E. Seiler, D. Sexty, and I. O. Stamatescu. Controlling Complex Langevin simulations of lattice models by boundary term analysis. *Phys. Rev. D*, 101(1):014501, 2020.

[59] Erhard Seiler, Denes Sexty, and Ion-Olimpiu Stamatescu. Gauge cooling in complex Langevin for QCD with heavy quarks. *Phys. Lett. B*, 723:213–216, 2013.

[60] Lorenzo Bongiovanni, Gert Aarts, Erhard Seiler, Denes Sexty, and Ion-Olimpiu Stamatescu. Adaptive gauge cooling for complex Langevin dynamics. *PoS*, LATTICE2013:449, 2014.

[61] Keitaro Nagata, Jun Nishimura, and Shinji Shimasaki. Justification of the complex Langevin method with the gauge cooling procedure. *PTEP*, 2016(1):013B01, 2016.

[62] Gert Aarts, Lorenzo Bongiovanni, Erhard Seiler, Denes Sexty, and Ion-Olimpiu Stamatescu. Controlling complex Langevin dynamics at finite density. *Eur. Phys. J. A*, 49:89, 2013.

[63] Kirill Boguslavski, Paul Hotzy, and David I. Müller. Stabilizing complex Langevin for real-time gauge theories with an anisotropic kernel, December 2022.

[64] Felipe Attanasio and Benjamin Jäger. Dynamical stabilisation of complex Langevin simulations of QCD. *The European Physical Journal C: Particles and Fields*, 79(1):16, 2019.

[65] Felipe Attanasio, Benjamin Jäger, and Felix P. G. Ziegler. QCD equation of state via the complex Langevin method. March 2022.

[66] Andrew C. Loheac and Joaquín E. Drut. Third-order perturbative lattice and complex Langevin analyses of the finite-temperature equation of state of nonrelativistic fermions in one dimension. *Phys. Rev. D*, 95(9):094502, May 2017.

[67] Zhenning Cai, Yang Kuang, and Hong Kiat Tan. Regularization of Complex Langevin Method. *Physical Review D*, 105(1):014508, January 2022.

[68] Manuel Scherzer, Erhard Seiler, Dénes Sexty, and Ion-Olimpiu Stamatescu. Complex Langevin and boundary terms. *Phys. Rev. D*, 99(1):014512, January 2019.

[69] T. C. de Aguiar, N. F. Svaiter, and G. Menezes. Stochastic Quantization of Real-Time Thermal Field Theory. *Journal of Mathematical Physics*, 51(10):102304, October 2010.

[70] K. Okano, L. Schülke, and B. Zheng. Kernel-controlled complex Langevin simulation. Field dependent kernel. *Physics Letters B*, 258(3-4):421–426, April 1991.

[71] Michael S. Albergo, Denis Boyda, Daniel C. Hackett, Gurtej Kanwar, Kyle Cranmer, Sébastien Racanière, Danilo Jimenez Rezende, and Phiala E. Shanahan. Introduction to Normalizing Flows for Lattice Field Theory. 1 2021.

[72] Jarrett Revels, Miles Lubin, and Theodore Papamarkou. Forward-mode automatic differentiation in julia. *CoRR*, abs/1607.07892, 2016.

[73] Seppo Linnainmaa. Taylor expansion of the accumulated rounding error. *BIT Numerical Mathematics*, 16(2):146–160, June 1976.

[74] Michael Innes. Don't unroll adjoint: Differentiating SSA-Form programs. *CoRR*, abs/1810.07951, 2018.

[75] William Moses and Valentin Churavy. Instead of rewriting foreign code for machine learning, automatically synthesize fast gradients. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12472–12485. Curran Associates, Inc., 2020.

[76] Michael B. Giles and Niles A. Pierce. An Introduction to the Adjoint Approach to Design. *Flow, Turbulence and Combustion*, 65(3):393–415, December 2000.

[77] Ronald M. Errico. What Is an Adjoint Model? *Bulletin of the American Meteorological Society*, 78(11):2577–2591, 1997.

[78] R.-E. Plessix. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophysical Journal International*, 167(2):495–503, November 2006.

[79] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. *CoRR*, abs/1806.07366, 2018.

[80] Xuechen Li, Ting-Kam Leonard Wong, Ricky TQ Chen, and David K Duvenaud. Scalable gradients and variational inference for stochastic differential equations. In *Symposium on Advances in Approximate Bayesian Inference*, pages 1–28. PMLR, 2020.

[81] Patrick Kidger, James Foster, Xuechen (Chen) Li, and Terry Lyons. Efficient and accurate gradients for neural SDEs. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 18747–18761. Curran Associates, Inc., 2021.

[82] Gert Aarts. Lefschetz thimbles and stochastic quantisation: Complex actions in the complex plane. *Physical Review D*, 88(9):094501, November 2013.

[83] Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, and Ali Ramadhan. Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*, 2020.

[84] Ch. Tsitouras. Runge–Kutta pairs of order 5(4) satisfying only the first column simplifying assumption. *Computers & Mathematics with Applications*, 62(2):770–775, 2011.

[85] Harvey B. Meyer. A Calculation of the shear viscosity in SU(3) gluodynamics. *Physical Review D: Particles and Fields*, 76:101701, 2007.

[86] Alessandro Amato, Gert Aarts, Chris Allton, Pietro Giudice, Simon Hands, and Jon-Ivar Skullerud. Electrical conductivity of the quark-gluon plasma across the deconfinement transition. *Physical Review Letters*, 111(17):172001, 2013.

[87] Bastian B. Brandt, Anthony Francis, Benjamin Jäger, and Harvey B. Meyer. Charge transport and vector meson dissociation across the thermal phase transition in lattice QCD with two light quark flavors. *Physical Review D: Particles and Fields*, 93(5):054510, 2016.

[88] Heng-Tong Ding, Olaf Kaczmarek, and Florian Meyer. Thermal dilepton rates and electrical conductivity of the QGP from the lattice. *Physical Review D: Particles and Fields*, 94(3):034504, 2016.

[89] N. Astrakhantsev, Viktor Braguta, and Andrey Kotov. Temperature dependence of shear viscosity of SU(3)–gluodynamics within lattice simulation. *JHEP*, 04:101, 2017.

[90] N. Yu. Astrakhantsev, V. V. Braguta, and A. Yu. Kotov. Temperature dependence of the bulk viscosity within lattice simulation of SU(3) gluodynamics. *Physical Review D: Particles and Fields*, 98(5):054515, 2018.

[91] Nikita Astrakhantsev, V. V. Braguta, Massimo D'Elia, A. Yu. Kotov, A. A. Nikolaev, and Francesco Sanfilippo. Lattice study of the electromagnetic conductivity of the quark-gluon plasma in an external magnetic field. *Physical Review D: Particles and Fields*, 102(5):054516, 2020.

[92] Sz. Borsányi, Z. Fodor, M. Giordano, S. D. Katz, A. Pásztor, C. Ratti, A. Schäfer, K. K. Szabó, and B. C. Tóth. High statistics lattice study of stress tensor correlators in pure SU(3) gauge theory. *Physical Review D: Particles and Fields*, 98(1):014512, July 2018.

[93] R. Bellwied, S. Borsanyi, Z. Fodor, J. Günther, S. D. Katz, C. Ratti, and K. K. Szabo. The QCD phase diagram from analytic continuation. *Physics Letters B*, 751:559–564, 2015.

[94] A. Bazavov et al. The QCD equation of state to o(μ_b^6) from lattice QCD. *Physical Review D: Particles and Fields*, 95(5):054504, 2017.

[95] Szabolcs Borsanyi, Zoltan Fodor, Jana N. Guenther, Sandor K. Katz, Kalman K. Szabo, Attila Pasztor, Israel Portillo, and Claudia Ratti. Higher order fluctuations and correlations of conserved charges from lattice QCD. *JHEP*, 10:205, 2018.

[96] M. Scherzer, D. Sexty, and I. O. Stamatescu. Deconfinement transition line with the complex Langevin equation up to $\mu/T \sim 5$. *Physical Review D: Particles and Fields*, 102(1):014515, 2020.

[97] Yuta Ito, Hideo Matsufuru, Yusuke Namekawa, Jun Nishimura, Shinji Shimasaki, Asato Tsuchiya, and Shoichiro Tsutsui. Complex Langevin calculations in QCD at finite density. *JHEP*, 10:144, 2020.

[98] Felipe Attanasio, Benjamin Jäger, and Felix P. G. Ziegler. Complex Langevin simulations and the QCD phase diagram: Recent de-

velopments. *The European Physical Journal A: Hadrons and Nuclei*, 56(10):251, 2020.

[99] Frédéric Chevy and Christophe Mora. Ultra-cold polarized fermi gases. *Reports on Progress in Physics*, 73(11):112401, oct 2010.

[100] Jens Braun, Jiunn-Wei Chen, Jian Deng, Joaquín E. Drut, Bengt Friman, Chen-Te Ma, and Yu-Dai Tsai. Imaginary Polarization as a Way to Surmount the Sign Problem in Ab Initio Calculations of Spin-Imbalanced Fermi Gases. *Physical Review Letters*, 110(13):130404, March 2013.

[101] K.B. Gubbels and H.T.C. Stoof. Imbalanced fermi gases at unitarity. *Physics Reports*, 525(4):255–313, 2013. Imbalanced Fermi Gases at Unitarity.

[102] Marco Cristoforetti, Francesco Di Renzo, Luigi Scorzato, and Aurora-Science Collaboration. New approach to the sign problem in quantum field theories: High density QCD on a Lefschetz thimble. *Physical Review D*, 86(7):074506, 2012.

[103] Erhard Seiler and Jacek Wosiek. Positive representations of a class of complex measures. *Journal of Physics A: Mathematical and General*, 50(49):495403, 2017.

[104] Poul H. Damgaard and Helmuth Hüffel. Stochastic quantization. *Physics Reports*, 152(5):227–398, August 1987.

[105] G. G. Batrouni, G. R. Katz, A. S. Kronfeld, G. P. Lepage, B. Svetitsky, and K. G. Wilson. Langevin simulations of lattice field theories. *Physical Review D*, 32(10):2736–2747, 1985.

[106] J. Berges, Sz. Borsányi, D. Sexty, and I.-O. Stamatescu. Lattice simulations of real-time quantum fields. *Physical Review D*, 75(4):045007, February 2007.

[107] Andreas S. Kronfeld. Dynamics of langevin simulations. *Progress of Theoretical Physics Supplements*, 111:293–312, 1993.

[108] Gert Aarts and Frank A. James. Complex Langevin dynamics in the SU(3) spin model at nonzero chemical potential revisited. *Journal of High Energy Physics*, 2012(1):118, 2012.

[109] Jan M. Pawlowski and Christian Zielinski. Thirring model at finite density in 0+1 dimensions with stochastic quantization: Cross-check with an exact solution. *Physical Review D: Particles and Fields*, 87(9):094503, 2013.

[110] Peter E Kloeden and Eckhard Platen. Numerical solution of stochastic differential equations. pages 1–50, 1992.

[111] Marija Milošević. Almost sure exponential stability of solutions to highly nonlinear neutral stochastic differential equations with time-dependent delay and the Euler–Maruyama approximation. *Mathematical and Computer Modelling*, 57(3):887–899, 2013.

[112] Andreas Rößler. Runge–Kutta methods for the strong approximation of solutions of stochastic differential equations. *SIAM Journal on Numerical Analysis*, 48(3):922–952, 2010.

[113] Christopher Rackauckas and Qing Nie. Differentialequations.jl–a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software*, 5(1):15, 2017.

[114] Christopher Rackauckas and Qing Nie. Adaptive methods for stochastic differential equations via natural embeddings and rejection sampling with memory. *Discrete and continuous dynamical systems. Series B*, 22(7):2731, 2017.

[115] Chris Rackauckas and Qing Nie. Stability-optimized high order methods and stiffness detection for pathwise stiff stochastic differential equations. In *2020 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–8. IEEE, 2020.

[116] Andrei Alexandru, Gokce Basar, Paulo F. Bedaque, Sohan Vartak, and Neill C. Warrington. Monte Carlo study of real time dynamics. *Physical Review Letters*, 117(8):081602, August 2016.

[117] Hiromichi Nakazato and Yoshiya Yamanaka. Minkowski stochastic quantization. *Physical Review D*, 34(2):492–496, July 1986.

[118] J. Berges and I.-O. Stamatescu. Simulating Nonequilibrium Quantum Fields with Stochastic Quantization Techniques. *Physical Review Letters*, 95(20):202003, November 2005.

[119] Jean Zinn-Justin. *Quantum Field Theory and Critical Phenomena; 4th Ed.* International Series of Monographs on Physics. Clarendon Press, Oxford, 2002.

[120] Daniel Alvestad. CLSolvers.jl: Stable solvers for real-time Complex Langevin, June 2021.

[121] Gert Aarts, Frank A. James, Jan M. Pawlowski, Erhard Seiler, Denes Sexty, and Ion-Olimpiu Stamatescu. Stability of complex Langevin dynamics in effective models. *Journal of High Energy Physics*, 2013(3):73, March 2013.

[122] John R. Klauder and Wesley P. Petersen. NUMERICAL INTEGRATION OF MULTIPLICATIVE NOISE STOCHASTIC DIFFERENTIAL EQUATIONS. *SIAM J. Num. Anal.*, 22:1153–1166, 1985.

[123] Wit Busza, Krishna Rajagopal, and Wilke van der Schee. Heavy Ion Collisions: The Big Picture, and the Big Questions. *Annual Review of Nuclear and Particle Science*, 68(1):339–376, October 2018.

[124] Alexander Rothkopf. Bayesian inference of real-time dynamics from lattice QCD. *Frontiers in Physics*, 10, August 2022.

[125] Shailesh Chandrasekharan and Uwe-Jens Wiese. Meron-cluster solution of fermion sign problems. *Physical Review Letters*, 83(16):3116, 1999.

[126] Ydalia Delgado Mercado, Hans Gerd Evertz, and Christof Gattringer. QCD Phase Diagram According to the Center Group. *Physical review letters*, 106(22):222001, 2011.

[127] Thomas Kloiber and Christof Gattringer. Dual methods for lattice field theories at finite density. *PoS*, LATTICE2013:206, 2014.

[128] Philippe de Forcrand and Owe Philipsen. Constraining the QCD Phase Diagram by Tricritical Lines at Imaginary Chemical Potential. *Physical Review Letters*, 105(15):152001, October 2010.

[129] Jens Braun, Joaquín E. Drut, and Dietrich Roscher. Zero-Temperature Equation of State of Mass-Imbalanced Resonant Fermi Gases. *Physical Review Letters*, 114(5):050404, February 2015.

[130] J.N. Guenther, R. Bellwied, S. Borsányi, Z. Fodor, S.D. Katz, A. Pásztor, C. Ratti, and K.K. Szabó. The QCD equation of state at finite

density from analytical continuation. *Nuclear Physics A*, 967:720–723, November 2017.

[131] Fugao Wang and David P Landau. Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical review letters*, 86(10):2050, 2001.

[132] Kurt Langfeld, Biagio Lucini, and Antonio Rago. Density of States in Gauge Theories. *Physical Review Letters*, 109(11), September 2012.

[133] Christof Gattringer and Pascal Törek. Density of states method for the $\mathbb{Z}_3$ spin model. *Physics Letters B*, 747:545–550, 2015.

[134] Dénes Sexty, Erhard Seiler, Ion-Olimipu Stamatescu, and Michael Westh Hansen. Complex Langevin boundary terms in lattice models. *PoS*, LATTICE2021:194, 2022.

[135] Jürgen Berges and Dénes Sexty. Real-time gauge theory simulations from stochastic quantization with optimized updating. *Nuclear Physics B*, 799(3):306–329, August 2008.

[136] Gert Aarts, Felipe Attanasio, Benjamin Jäger, and Dénes Sexty. Complex Langevin in Lattice QCD: Dynamic stabilisation and the phase diagram. *Acta Phys. Polon. Supp.*, 9:621, 2016.

[137] Gert Aarts, Chris Allton, Ryan Bignell, Timothy J. Burns, Sergio Chaves García-Mascaraque, Simon Hands, Benjamin Jäger, Seyong Kim, Sinéad M. Ryan, and Jon-Ivar Skullerud. Open charm mesons at nonzero temperature: Results in the hadronic phase from lattice QCD. September 2022.

[138] Kirill Boguslavski, Paul Hotzy, and David I. Müller. A stabilizing kernel for complex Langevin simulations of real-time gauge theories. In *39th International Symposium on Lattice Field Theory*, October 2022.

[139] Andrei Alexandru, Gokce Basar, Paulo F. Bedaque, and Gregory W. Ridgway. Schwinger-Keldysh formalism on the lattice: A faster algorithm and its application to field theory. *Physical Review D: Particles and Fields*, 95(11):114501, 2017.

[140] Daniel Alvestad. alvestad10/KernelCL: Towards learning optimized kernels for complex Langevin, November 2022.

[141] Pietro Giudice, Gert Aarts, and Erhard Seiler. Localised distributions in complex Langevin dynamics. *PoS*, LATTICE2013:200, 2014.

[142] Yuya Abe and Kenji Fukushima. Analytic studies of the complex Langevin equation with a Gaussian ansatz and multiple solutions in the unstable region. *Physical Review D: Particles and Fields*, 94(9):094506, 2016.

[143] L. L. Salcedo. Positive representations of complex distributions on groups. *Journal of Physics A: Mathematical and General*, 51(50):505401, 2018.

[144] Simon Woodward, Paul M. Saffin, Zong-Gang Mou, and Anders Tranberg. Optimisation of Thimble simulations and quantum dynamics of multiple fields in real time. *JHEP*, 10:082, 2022.

[145] Nobuyuki Matsumoto. Comment on the subtlety of defining a real-time path integral in lattice gauge theories. *PTEP*, 2022(9):093B03, 2022.

[146] Davan Harrison. A brief introduction to automatic differentiation for machine learning. *arXiv preprint arXiv:2110.06209*, 2021.

[147] Yang Cao, Shengtai Li, Linda Petzold, and Radu Serban. Adjoint sensitivity analysis for differential-algebraic equations: The adjoint DAE system and its numerical solution. *SIAM journal on scientific computing*, 24(3):1076–1089, 2003.

[148] Frank Schäfer, Michal Kloc, Christoph Bruder, and Niels Lörch. A differentiable programming method for quantum control. *Machine Learning: Science and Technology*, 1(3):035009, 2020.

[149] Qiqi Wang, Rui Hu, and Patrick Blonigan. Least squares shadowing sensitivity analysis of chaotic limit cycle oscillations. *Journal of Computational Physics*, 267:210–224, 2014.

[150] Angxiu Ni and Qiqi Wang. Sensitivity analysis on chaotic dynamical systems by non-intrusive least squares shadowing (NILSS). *Journal of Computational Physics*, 347:56–77, October 2017.

[151] DANIEL J. LEA, MYLES R. ALLEN, and THOMAS W. N. HAINE. Sensitivity analysis of the climate of a chaotic system. *Tellus Series A-dynamic Meteorology and Oceanography*, 52(5):523–532, 2000.

[152] Angxiu Ni and Chaitanya Talnikar. Adjoint sensitivity analysis on chaotic dynamical systems by Non-Intrusive Least Squares Adjoint Shadowing (NILSAS). *Journal of Computational Physics*, 395:690–709, October 2019.

[153] Qiqi Wang. Convergence of the least squares shadowing method for computing derivative of ergodic averages. *SIAM Journal on Numerical Analysis*, 52(1):156–170, 2014.

[154] John Thuburn. Climate sensitivities via a Fokker–Planck adjoint approach. *Quarterly Journal of the Royal Meteorological Society*, 131, 2005.

[155] F. C. Cooper and Peter H. Haynes. Climate sensitivity via a nonparametric Fluctuation–Dissipation theorem. *Journal of the Atmospheric Sciences*, 68:937–953, 2011.

[156] Kyriakos D Kantarakias and George Papadakis. Sensitivity analysis of chaotic systems using a frequency-domain shadowing approach. *Journal of Computational Physics*, 474:111757, 2023.

[157] G. L. Eyink, T. W. N. Haine, and D. J. Lea. Ruelle's linear response formula, ensemble adjoint schemes and Lévy flights. *Nonlinearity*, 17(5):1867–1889, September 2004.

[158] John Craske. Adjoint sensitivity analysis of chaotic systems using cumulant truncation. *Chaos, Solitons & Fractals*, 2019.

[159] Wei Fang and Mike B Giles. Importance sampling for pathwise sensitivity of stochastic chaotic systems. *SIAM/ASA Journal on Uncertainty Quantification*, 9(3):1217–1241, 2021.

[160] Karlheinz Geist, Ulrich Parlitz, and Werner Lauterborn. Comparison of different methods for computing lyapunov exponents. *Progress of Theoretical Physics*, 83:875–893, 1990.

[161] Giancarlo Benettin, Luigi Galgani, Antonio Giorgilli, and Jean-Marie Strelcyn. Lyapunov Characteristic Exponents for smooth dynamical systems and for hamiltonian systems; a method for computing all of them. Part 1: Theory. *Meccanica*, 15(1):9–20, March 1980.

[162] L. Arnold. *Random Dynamical Systems*. Springer Monographs in Mathematics. Springer Berlin Heidelberg, 2013.

[163] Sergei Yu. Pilyugin. Shadowing near an invariant set. In Sergei Yu. Pilyugin, editor, *Shadowing in Dynamical Systems*, Lecture Notes in Mathematics, pages 1–101. Springer, Berlin, Heidelberg, 1999.

[164] Edward N. Lorenz. Deterministic nonperiodic flow. *Journal of Atmospheric Sciences*, 20(2):130–141, 1963.

[165] Hans Munthe-Kaas. Runge-Kutta methods on Lie groups. *BIT Numerical Mathematics*, 38(1):92–111, March 1998.

[166] Hans Munthe-Kaas. High order Runge-Kutta methods on manifolds. *Proceedings of the NSF/CBMS Regional Conference on Numerical Analysis of Hamiltonian Differential Equations*, 29(1):115–127, January 1999.

[167] Goran Marjanovic and Victor Solo. Numerical methods for stochastic differential equations in matrix lie groups made simple. *IEEE Transactions on Automatic Control*, 63:4035–4050, 2018.

[168] Alexander Rothkopf and Jan Nordström. A new variational discretization technique for initial value problems bypassing governing equations. *Journal of Computational Physics*, 477:111942, 2023.

[169] Arieh Iserles, Hans Munthe-Kaas, Syvert Nørsett, and Antonella Zanna. Lie-group methods. *Acta Numerica*, 2005.