Master Thesis

Numerical Analysis of Causal Fermion Systems on $\mathbb{R} \times S^3$

by

Niki Kilbertus

supervised by

Prof. Dr. Felix Finster

Faculty for Mathematics
University of Regensburg

November 2015

Zusammenfassung

Das Ziel dieser Arbeit ist die numerische Untersuchung eines kausalen Fermionsystems auf $\mathbb{R} \times S^3$. Kausale Fermionsysteme bezeichnen im Allgemeinen einen abstrakten mathematischen Rahmen, welcher einen potentiellen Kandidaten für eine vereinheitlichte physikalische Theorie darstellt. Wir formulieren ein Modell innerhalb dieses Rahmens, welches einerseits numerisch handhabbar ist und andererseits auf interessante physikalische Einblicke hoffen lässt.

Anschließend erklären und vergleichen wir unterschiedliche numerische Methoden und erläutern die Implementierung sowie Strategie unserer Analyse im Detail. Es folgen numerische Resultate für zwei Spezialfälle, die erste Einblicke in das Verhalten des Modells liefern. Diese Ergebnisse werden mit Hinblick auf analytische Resultate zu kausalen Fermionsystemen interpretiert. Wir schließen mit einem Ausblick auf weiterführende Fragestellungen und potentielle Schwierigkeiten für zukünftige numerische Untersuchungen.

Abstract

The aim of this thesis is to analyze a causal fermion system on $\mathbb{R} \times S^3$ numerically. Causal fermion systems generally provide an abstract mathematical framework, which constitutes a candidate for a unified physical theory. Within this framework we formulate a model, which is accessible to numerical treatment on the one hand, and expected to exhibit sufficient complexity to encompass interesting physical behavior on the other hand.

Subsequently, we explain and compare various numerical methods and elaborate on the implementation and strategy of our analysis in detail. We continue with numerical results for two special cases, which yield first insights into the behavior of the model. We interpret these findings in view of previous analytic results within the theory of causal fermion systems. Eventually, we provide an outlook on further questions and potential difficulties in future numerical analysis.

Contents

1 Introduction

1.1 Motivation and overview

In this thesis, we discuss a novel numerical approach to the *causal action principle* arising in the framework of *causal fermion systems*. Causal fermion systems provide a mathematically rigorous candidate for a fundamental theory of physics [12]. The theory of causal fermion systems was first developed in [10] as a generalization of the principle of the fermionic projector [5, 9]. We will treat neither the precise setup nor the numerous results about causal fermion systems in depth. However, to render this work more self contained by giving the reader a coarse idea of the bigger picture, we still – very briefly – introduce the general framework of causal fermion systems in section 1.2.

As we discuss there, from a causal fermion system we can construct an action principle. While a causal fermion system encodes all physical structures and objects, a minimizer of the causal action principle is supposed to single out physical reality. While the causal action principle has been extensively investigated analytically (see [8] and recent work by F. Finster et. al) and statements about the existence of minimizers have been established for certain classes of causal fermion systems, only for a few examples can we currently write down minimizers explicitly.

This is in part due to the generically high computational complexity of the causal action principle, which can make analytic calculations infeasible. In such a situation, numerical treatment immediately suggests itself and first attempts in this direction have been pursued in [14]. This work already elaborates on the numerical difficulties encountered in the numerical analysis of causal fermion systems. We will repeatedly come back to those and other related issues. A connection between the mathematical definitions and some widely used contemporary physical notions will be drawn in section 1.3 to round off the chapter.

In chapter 2, we introduce a special model of a causal fermion system – we call it the *Dirac sphere* – with accessibility to numerical treatment in mind. We state the causal action principle in section 2.2 and provide detailed step by step instructions on how to implement the mathematical objects of the Dirac sphere on a computer in section 2.3. Chapter 3 deals with the results of various simulations. First, we check the numerics by comparison to analytic computations in a vastly simplified setting, see section 3.2. We then intermingle just in time excursions on numerical methods (sections 3.3 and 3.5) with an explanation of the rescaling procedure in section 3.4 and further results for a less trivial setup in section 3.6.

1.2 A brief introduction to causal fermion systems

In this section, we will provide the general definition of causal fermion systems together with a brief schematic derivation of the causal action principle. This quick tour merely serves to illustrate the general underlying mathematical setting, in which our model is immersed. The interested reader is referred to [9] to begin with and we suggest to continue with references therein. There are also two books in preparation, one explaining the physical ideas and introducing the mathematical background in a rather non-technical way [13], while the other primarily establishes a rigorous connection between causal fermion systems and well known physical systems in Minkowski space [4].

Drawing this connection between the general framework and the Dirac sphere – ideally as a formal derivation of the latter from the prior – has not been done rigorously yet, thus is not part of the thesis. However, such a derivation – although it might contain some technical challenges – should be straight forward in principle and is subject of current research.

Definition 1. A **causal fermion system with spin dimension** $n \in \mathbb{N}$ is a triple $(\mathcal{H}, \mathcal{F}, \rho)$, where

- $\mathcal{H}$ is a separable complex Hilbert space with the scalar product $\langle \cdot | \cdot \rangle_{\mathcal{H}}$,
- $\mathcal{F} \subset \mathrm{L}(\mathcal{H}, \mathcal{H}) \equiv \mathrm{L}(\mathcal{H})$ is the set of all self-adjoint operators on $\mathcal{H}$ of finite rank, which (counting multiplicities) have at most n positive and at most n negative eigenvalues,
- ρ is a measure defined on a σ-algebra of subsets of $\mathcal{F}$. We call ρ the **universal measure**.

Remark. Often times we do not mention the spin dimension n explicitly when talking about causal fermion systems, but implicitly assume its existence. In the following, we equip $\mathcal{F}$ with the topology induced by the sup-norm on $\mathrm{L}(\mathcal{H})$. A causal fermion system is understood to encode a space-time together with all structures and objects therein – nothing else is needed in addition.

If we want to describe the universe as we experience it, we need to single out physically relevant causal fermion systems. It is customary to give conceivable names to the objects one is dealing with, because proper perception usually helps to understand mathematical relations. Of course, for a physical theory we also need to provide physical interpretation, which is done simplest in the usual jargon. In the following, we will very briefly touch upon the physical notions of several objects and simultaneously construct an action principle, which can be used to distinguish physically admissible causal fermion systems.

Definition 2. We call

$$M := \operatorname{supp} \rho \subset \mathcal{F} \tag{1.1}$$

the **space-time** of a causal fermion system. With the topology induced by $\mathcal{F}$ and the volume measure $\rho|_M$ on M, the space-time is a topological measure space. Accordingly, the elements of M are called **space-time points**.

Definition 3. The product xy of two space-time points $x, y \in \mathcal{F}$ is an operator on $\mathcal{H}$ of rank at most $2n$. This is clear from the fact that each of the operators has at most n positive and at most n negative eigenvalues. We denote the non-trivial eigenvalues of xy by $\lambda_1^{xy}, \ldots, \lambda_{2n}^{xy} \in \mathbb{C}$, where we count algebraic multiplicities. The two space-time points x and y are called

- **spacelike separated**, if all eigenvalues have the same absolute value, i. e. $|\lambda_i^{xy}| = \left|\lambda_j^{xy}\right|$ for all $i, j \in \{1, \ldots, 2n\}$.
- **timelike separated**, if all eigenvalues are real, but they are not all equal, i. e. $\lambda_i^{xy} \in \mathbb{R}$ for all $i \in \{1, \ldots, 2n\}$ and there exist $i, j \in \{1, \ldots, 2n\}$, such that $\lambda_i^{xy} \neq \lambda_j^{xy}$.
- **lightlike separated** in all other cases.

Since the space-time M is a subset of $\mathcal{F}$, we can establish a *causal structure* on our space-time by restricting the above definition to M.

Definition 4. We define the **spectral weight** $|\cdot|$ of an operator as the sum of the absolute values of its eigenvalues.

Example. For a product of space-time points $x, y \in M$ and the square thereof we find the spectral weights

$$|xy| = \sum_{i=1}^{2n} |\lambda_i^{xy}| \qquad \text{and} \qquad \left|(xy)^2\right| = \sum_{i=1}^{2n} |\lambda_i^{xy}|^2 \,. \tag{1.2}$$

Definition 5. The **Lagrangian** $\mathcal{L}$ and the **action** $\mathcal{S}$ are defined by

$$\mathcal{L} : \mathcal{F} \times \mathcal{F} \to \mathbb{R}_0^+ \,, \quad \mathcal{L}(x,y) := \left|(xy)^2\right| - \frac{1}{2n} |xy|^2 \tag{1.3}$$

and

$$\mathcal{S} : \mathbb{M}_0^+ \to \mathbb{R}_0^+ \,, \quad \mathcal{S}(\rho) := \iint_{\mathcal{F}\times\mathcal{F}} \mathcal{L}(x,y)\, d\rho(x)\, d\rho(y) \tag{1.4}$$

respectively. Here, $\mathbb{M}_0^+$ denotes the set of positive regular Borel measures on $\mathcal{F}$. For a motivation of this definition see [5, 13] and [6] for a mathematical point of view.

Definition 6. The **causal action principle** is to minimize $\mathcal{S}$ by varying the universal measure ρ in $\mathbb{M}_0^+$ under the following constraints:

$$\textbf{volume constraint} \qquad \rho(\mathcal{F}) = \text{const} \tag{1.5}$$

$$\textbf{trace constraint} \qquad \int_{\mathcal{F}} \mathrm{tr}(x)\, d\rho(x) = \text{const} \tag{1.6}$$

$$\textbf{boundedness constraint} \qquad \mathcal{T} := \iint_{\mathcal{F}\times\mathcal{F}} |xy|^2\, d\rho(x)\, d\rho(y) \leq C \tag{1.7}$$

Here, $C \in \mathbb{R}^+$ is a constant and tr denotes the trace of a linear operator on $\mathcal{H}$.

Remark. While it is easily shown that the volume and the trace constraints are needed to avoid trivial minimizers, it is not obvious, whether the boundedness constraint is truly necessary. While it is used in current versions of existence proves, it was not clear for a while, whether one could also find a proof that does not rely on it. In section 3.2, we will show with a specific example that the boundedness constraint is indeed necessary for the existence of minimizers.

Beware that the terminology at work here can be deceiving. It is all too common in physics to minimize the action, which is obtained by integrating a Lagrangian. While some parallels between these settings can definitely be drawn, they also differ vastly in other aspects. One is easily fooled into believing that "as usual" the only task left, is to choose a suitable Lagrangian and the rest is only a matter of simple number crunching. This is *not* the case here: The Lagrangian is fixed once and for all. We minimize by variation of the universal measure. Space-time, at this point, is of a fundamentally different nature with a different structure than the usual (smooth) Lorentzian manifold. We will encounter both, similarities and differences later.

All further physical objects and properties are now encoded in the causal fermion system together with the causal action principle. The particle content, their interaction, the geometry of space-time, all those contemporary physical notions can be extracted from the established setting.

However, by now one has probably realized that it is still far from obvious, how to do specific computations and obtain quantitative results that could be compared to experiments. Let us examine potential difficulties. First of all, to compute the Lagrangian, we find the eigenvalues of products of operators xy for $x, y \in M$. Since the Hilbert space $\mathcal{H}$ could be infinite-dimensional, it will come in handy to simplify this task by restricting our attention to finite-dimensional subspaces of $\mathcal{H}$.

Definition 7. We define $S_x := x(\mathcal{H})$ for every $x \in \mathcal{F}$ and call S_x the **spin space (at x)**. Because every $x \in \mathcal{F}$ has at most n positive and at most n negative eigenvalues, $S_x \subset \mathcal{H}$ is a subspace of dimension at most $2n$.

A map ψ on M that takes a space-time point $x \in M$ to $\psi(x) \in S_x$, is called a **one-particle wave function**.

Definition 8. For all $x, y \in \mathcal{F}$ we call

$$P(x,y) := \pi_x y|_{S_y} \,:\, S_y \to S_x \tag{1.8}$$

the **kernel of the fermionic operator**, where π_x is the orthogonal projection in $\mathcal{H}$ to the spin space S_x.

The kernel of the fermionic operator is now a map between finite-dimensional subspaces of $\mathcal{H}$ (of dimension at most $2n$ respectively). Therefore $P(x,y)$ is easier to handle in computations than the elements of $\mathcal{F}$ themselves. In the following, we will express all necessary quantities in terms of the kernel of the fermionic operator. Most importantly, let us address the eigenvalues of products of space-time points.

Definition 9. For all $x, y \in \mathcal{F}$ we call

$$A_{xy} := P(x,y)P(y,x) \,:\, S_x \to S_x \tag{1.9}$$

the **closed chain**.

The closed chain again is a map between finite-dimensional spaces and the eigenvalues of A_{xy} coincide with the non-trivial eigenvalues of xy for all $x, y \in \mathcal{F}$. To construct the Lagrangian – and thus the action – we now only have to compute the eigenvalues of a complex $2n \times 2n$ matrix instead of operator products on a potentially infinite-dimensional Hilbert space.

Next, let us add some further structure to the spin space, because it has just moved in the focus of our attention.

Definition 10. We equip the spin space S_x with an inner product to turn it into an indefinite inner product space of signature (p, q) with $p, q \leq n$. We choose the so called **spin scalar product**

$$\prec \cdot \,|\, \cdot \succ_x \,:\, S_x \times S_x \to \mathbb{C}\,, \quad \prec u \,|\, v \succ_x := -\langle u | xv \rangle_{\mathcal{H}}\,, \tag{1.10}$$

such that the kernel of the fermionic operator is symmetric with respect to the spin scalar product in the sense that

$$P(x,y)^* = P(y,x)\,. \tag{1.11}$$

Remark. Note that despite its name the spin scalar product is not positive definite, hence $(S_x, \prec \cdot \,|\, \cdot \succ_x)$ is an indefinite inner product space for all $x \in M$, not a Hilbert space.

To finish this section, we introduce the notion of *time direction* in causal fermion systems.

Definition 11. We define the anti-symmetric functional

$$\mathcal{C} \,:\, M \times M \to \mathbb{R}\,, \quad \mathcal{C}(x,y) := i\, \mathrm{Tr}(y\, x\, \pi_y\, \pi_x - x\, y\, \pi_z\, \pi_y)\,, \tag{1.12}$$

where we use the symbol Tr (as opposed to tr) to denote the trace over the finite-dimensional spinor space instead of the whole Hilbert space $\mathcal{H}$. We say

- y lies in the **future** of x, if $\mathcal{C}(x,y) > 0$.
- y lies in the **past** of x, if $\mathcal{C}(x,y) < 0$.

This definition distinguishes a *direction of time*. Note that this time ordering need not be transitive, which has curious implications. On the fundamental level, while we do have a causal structure, there is still no notion of causation, i. e. we cannot formulate statements like "something at x *caused* something later in time at y". There is no "global clock" allowing us to capture the system at one certain well-defined time.

1.3 A step towards contemporary physics

This section serves two purposes. For one, it demonstrates how the abstract formulation of causal fermion systems relates to the "usual" setting of contemporary field theories. Moreover, it also serves as an illustrative example to resort back to and draw parallels later, when we explore the Dirac sphere in chapter 2. Again, we try to keep this brief and leave out a whole lot of detail. Instead, we focus on the important concepts and relations.

For readers with a background in physics, this section will help understand and interpret the physical content. For all others, it can provide a hint towards the physical relevance of causal fermion systems. However, since the analysis in later chapters does not necessarily depend on any physical interpretation or prerequisites – which are to some extent assumed in this section – it might be skipped entirely. Keep in mind that all the symbols and definitions showing up shortly are a-priori independent from those we have encountered in section 1.2. Of course, we are using the same symbols for a reason. The respective relations will be made explicit by the end of the section.

Let $(\mathcal{M}, d\mu)$ be Minkowski space with the standard metric and signature $(+,-,-,-)$. On $\mathcal{M}$ we consider Dirac spinors ψ satisfying the Dirac equation

$$(i\gamma^k \partial_k - m)\psi = 0\,, \tag{1.13}$$

where m is a mass parameter. We define the following inner products on the solution space:

spin scalar product: For two spin one half Dirac spinors ψ, ϕ we define the spin scalar product pointwise as

$$\prec \psi \,|\, \phi \succ (x) := \overline{\psi}(x)\, \phi(x)\,, \tag{1.14}$$

where $\overline{\psi} = \psi^\dagger \gamma^0$ is the usual adjoint spinor. The spin scalar product has signature (p, q) with $p, q \leq 2$.

Hilbert space inner product: The inner product

$$\langle \psi | \phi \rangle := \int_{t=\text{const}} (\overline{\psi} \gamma^0 \phi)(t, x)\, dx \tag{1.15}$$

is positive-definite, i. e. a scalar product and thereby turns the solution space into a Hilbert space, which we denote by $(\mathcal{H}, \langle \cdot | \cdot \rangle)$. (The fact that the scalar product is well-defined, i. e. independent of the time hypersurface, follows from the existence and uniqueness theory for Cauchy initial value problems of partial differential equations.)

For each space-time point x one finds a linear operator F_x such that

$$\langle \psi | F_x \phi \rangle_{\mathcal{H}} = - \prec \psi \,|\, \phi \succ (x)\,. \tag{1.16}$$

One easily verifies in a direct calculation that F_x is self-adjoint with respect to the Hilbert space scalar product. Because of the signatures of the two inner products,

contemporary physics	causal fermion systems
solution space of the Dirac equation	abstract Hilbert space, not a-priori known
Minkowski space $\mathcal{M}$	$\text{supp}(\rho) = \{F_x \,|\, x \in \mathcal{M}\} \subset \mathcal{F}$
$x \in \mathcal{M}$	$F_x \in \mathcal{F}$
Dirac spinor ψ	$\psi : M \ni x \mapsto \psi(x) \in S_x$
$\overline{\psi}(x)\, \phi(x)$	$\prec \cdot \,|\, \cdot \succ_x : S_x \times S_x \to \mathbb{C}$, $\prec u \,|\, v \succ_x := -\langle u | F_x v \rangle_{\mathcal{H}}$

Table 1.1: An illustration of the correspondence between contemporary physics and causal fermion systems. The objects on the left hand side, which we are mostly familiar with, are reinterpreted as parts of a causal fermion systems. Hence, we show by "reverse engineering", that the contemporary model can be captured in the abstract framework.

we conclude that the operator F_x has at most two positive and at most two negative eigenvalues (counting multiplicities).

Let $\mathcal{F}$ be the set of all self-adjoint operators on $\mathcal{H}$ with at most two positive and at most two negative eigenvalues. Then we can push forward the measure μ on $\mathcal{M}$ along the map

$$F : \mathcal{M} \to \mathcal{F}, \quad x \mapsto F_x, \tag{1.17}$$

obtaining a measure $F_* \mu$ on $\mathcal{F}$. Going back to Definition 1, we realize that we have found all ingredients for a causal fermion system. The connection between the abstract framework and the "usual" concepts and notions of contemporary physics now becomes obvious. We summarize the findings of this example in Table 1.1, where we link the corresponding objects in tabular form. We reinterpret the "usual" notions on the left as part of the abstract framework on the right.

In this fashion, one can resume to define causality or time ordering and will not be surprised that they coincide with the "usual" notions and definitions. There is also a neat correspondence and formalism to construct wave functions in the abstract setting of causal fermion systems, which can be found in [9] and references therein.

Let us finally move on to the definition of the model we want to analyze numerically, the Dirac sphere. This time we start with the "usual" objects right away and –

as already mentioned – do not provide a one to one correspondence to the abstract framework like we did here. However, after this example, one will understand the basic idea of the connection anyway.

2 The Dirac sphere

The abstract formulation of causal fermion systems as in section 1.2 does not seem to allow for numerical treatment in a canonical way. Although the causal action principle in the end boils down to a constrained optimization problem and could hence be expected to be found among the class of problems extensively treated in the numerics literature, our setting is rather different from most optimization problems one usually encounters in computational physics and applied mathematics.

First of all, we vary the action with respect to a positive regular Borel measure on an operator space. The implementation of those objects on the computer is not straight forward. Moreover the integral in our objective function – the action $\mathcal{S}$ – is potentially difficult to carry out.

Thus the subject of this chapter is to find a model, which can be implemented numerically, but still exhibits enough complexity to encode interesting physics. After some motivational remarks in section 2.1, we will introduce our model in section 2.2, which contains the major results of this chapter. In section 2.3, we conclude with step by step instructions for an implementation of the model. This serves as the starting point for the simulations discussed in chapter 3.

2.1 Why we chose a model on $\mathbb{R} \times S^3$

The only mathematical objects computers can handle well inherently, are (finite precision) real numbers and the four fundamental arithmetic operations on them. Hence our goal is to basically parameterize complicated mathematical objects by a finite number of real parameters, thereby enabling us to use common minimization techniques. Recall that number representations on realistic computers are restricted in several ways. First, numbers can only be represented with finite precision and accuracy, i. e. their absolute values can neither be arbitrarily large nor arbitrarily small. Second, only finitely many different numbers can be represented, which is of course connected to the first restriction. Finally, one has to take into account round off errors and that even some of the most fundamental rules of calculus break down. For example, the addition of real numbers is not a transitive operation on computers. For further reading, number representations on computers and the pitfalls or issues related to them are treated in most introductory numerics textbooks.

Most contemporary models of physical reality – at least on experienceable scales – usually assume a Lorentzian four-dimensional space-time manifold. Due to the natural restrictions of computers mentioned above, numerically, we can only handle finite domains. Note that for example even quadrature schemes for unbounded intervals in-

ternally shrink the interval to a bounded domain by a suitable coordinate transformation. Common choices are rectangular or circular regions and their higher-dimensional equivalents. Whenever an originally unbounded domain is replaced by a bounded part of it for numerical treatment, one has to worry about boundary effects.

There are several possibilities to avoid those perturbing effects. For domains which are a direct product of intervals, the most famous one is probably the implementation of periodic boundary conditions in each dimension separately. In order for this to work properly, one either has to confine oneself to periodic phenomena only, or one has to choose the computing domain sufficiently large and have functions fall off to zero well inside the domain. The determination of an appropriate size is a difficult task and heavily depends on the specifics of the problem. Heuristically, we need to make sure that "everything stays far away from the boundary". Thereby we already assume the observed phenomena to be local, meaning that we can determine, whether they behave differently in the finite region as compared to the originally unbounded region.

Causal fermion systems do not necessarily allow for such a localization. The integral in the action is over all of space-time. If we consider an open universe, we might run into problems, because we have no a-priori knowledge, whether and how the Lagrangian falls off approaching infinity. Choosing a closed universe with a finite volume frees us of these troubles and implicitly satisfies the volume constraint in Definition 6.

Along these lines it seems reasonable to choose a closed universe, so instead of the canonical choice $\mathbb{R}^4$, we will be working on $\mathbb{R} \times S^3$. In this way, we can parameterize the spatial domain S^3 in the standard way with spherical coordinates in the direct product of finite intervals. How we handle the currently unbounded temporal domain will be explained later.

To sum up, because numerical artifacts at the boundary were an issue in the first attempts of treating causal fermion systems numerically in [14], we have now switched to a three-dimensional sphere for the spatial domain and furthermore allow time to be unbounded. Moreover, we relax the strict space-time grid by allowing for more flexible discretizations. While these changes definitely resolve some issues – as always – they also introduce new ones, as we will discuss in chapter 3.

2.2 Mathematical setup

2.2.1 The underlying space-time

Let us now come to the specific setup. We decided to work with the space-time $M = \mathbb{R} \times S^3$, where S^3 – as a closed manifold – represents space and $\mathbb{R}$ represents the infinite lifetime of the universe. For later convenience, we introduce spherical coordinates on S^3

$$(r, \vartheta, \varphi) \in [0, \pi] \times [0, \pi] \times [0, 2\pi) \,, \tag{2.1}$$

where ϑ and φ are the polar and azimuthal angle coordinates on S^2 respectively. We think about r as the geodesic distance from the north pole $\mathfrak{n}$, as illustrated in Figure 2.1. Whenever necessary, we will implicitly assume the coordinate domain to be open, i. e.

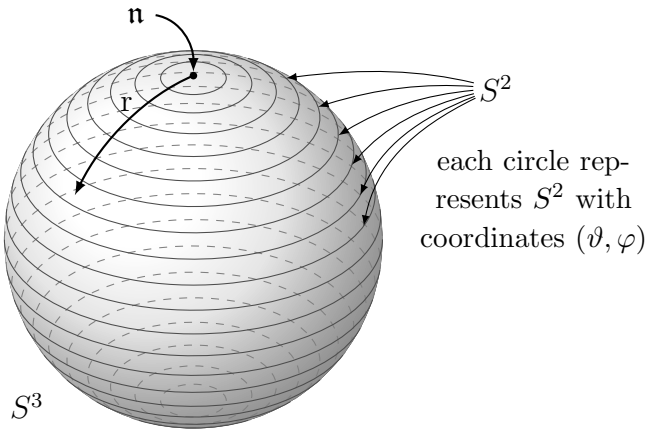

Figure 2.1: We illustrate the coordinates on S^3. The r coordinate can be interpreted as the geodesic distance from the north pole $\mathfrak{n}$, because S^3 has unit radius.

we exclude the end points in the intervals of (2.1). Embedding S^3 into $\mathbb{R}^4$, gives the coordinate transformations

$$\begin{aligned} x_1 &= R\, \cos(r) \\ x_2 &= R\, \sin(r)\, \cos(\vartheta) \\ x_3 &= R\, \sin(r)\, \sin(\vartheta)\, \cos(\varphi) \\ x_4 &= R\, \sin(r)\, \sin(\vartheta)\, \sin(\varphi)\,. \end{aligned} \tag{2.2}$$

We can now define a metric on M by the line element

$$ds^2 = dt^2 - R^2\, d\sigma^2_{S^3}\,, \tag{2.3}$$

where we inherit the line element

$$d\sigma^2_{S^3} = dr^2 + \sin(r)^2\, d\vartheta^2 + \sin(r)^2\, \sin(\vartheta)^2\, d\varphi^2 \tag{2.4}$$

on the three-dimensional unit sphere from the standard metric on $\mathbb{R}^4$. The parameter $R \in \mathbb{R}^+$ represents the "size" or "radius" of the spatial part of our space-time. The volume element on S^3 is given by the absolute value of the determinant of the Jacobian for the coordinate transformations in equation (2.2) and can also be read off directly from (2.4):

$$d\mu_{S^3} = \sin(r)^2\, \sin(\theta)\,. \tag{2.5}$$

Note that we start out with a smooth, four-dimensional, metric manifold from the very beginning. This should be compared to the Minkowski space example in section 1.3, where we explicitly point out the connection to the general definition of causal fermion systems.

A natural first goal is to model interacting spin one half particles in our space-time M, which is the subject of the following section.

2.2.2 Dirac spinors and the Dirac equation

Let us spend a few words on how four-component complex Dirac spinors are introduced in general, since especially among physicists this is often a point of great confusion. While we handle them with ease just like vectors in specific computations, it is still strictly forbidden to actually call them "vectors". They are spinors, not vectors or vector fields. We intend to show in a very short outline, at which level they differ from usual tensor fields. Details can be found in [19, 16].

To this end, we first have to construct a *spin structure* for M, e. g. a $\mathrm{Spin}(1,3)$ principal bundle Q over M together with a map F from Q to the oriented orthonormal frame bundle $\mathrm{SO}(M)$, which is itself a $\mathrm{SO}(1,3)$ principal bundle, such that the following compatibility criteria are satisfied. The projection of the principal bundle $\mathrm{SO}(M)$ after F is the same as the projection on P as maps from P to M, i. e. $\pi_{SO(M)} \circ F = \pi_P$. Second, we require that $F(pu) = F(p)\chi(u)$ for all $p \in P$ and all $u \in \mathrm{Spin}(1,3)$. Here χ is the universal two-fold covering map of $\mathrm{SO}(1,3)$ by $\mathrm{Spin}(1,3)$ and the multiplications are the group actions by $\mathrm{Spin}(1,3)$ and $\mathrm{SO}(1,3)$ respectively. The spin group $\mathrm{Spin}(1,3)$ is isomorphic to $\mathrm{SL}(2,\mathbb{C})$ as a Lie group.

Next, we define the *spinor bundle*. The spinor bundle is the associated complex bundle $P \times_\kappa \mathbb{C}^4$ to Q with respect to the spin representation $\kappa : \mathrm{Spin}(1,3) \to \mathrm{Aut}(\mathbb{C}^4)$. The sections of the spinor bundle eventually are the four-component complex *Dirac spinors* on M. Now we recognize, how transformations such as rotations ($\mathrm{SO}(1,3)$) make way for transformations according to the spin representation. The common, somewhat mysterious and definitely unsatisfying physical explanation that "spinors simply transform differently", is now understood through the fact that we do not consider tensor products of the (co-)tangent bundle, but an associated bundle with respect to the spin representation, a two-fold cover of the special orthogonal group.

In the following, we will not need the details of the construction, but treat the Dirac spinors more like physicists for computational simplicity. This includes denoting (and for most purposes also treating) spinors like four-dimensional vectors and the insertion of gamma (or Dirac) matrices in the Dirac representation

$$\gamma^0 = \begin{pmatrix} \mathbb{1}_{\mathbb{C}^2} & 0 \\ 0 & -\mathbb{1}_{\mathbb{C}^2} \end{pmatrix}, \quad \gamma^k = \begin{pmatrix} 0 & \sigma^k \\ -\sigma^k & 0 \end{pmatrix} \quad \text{for } k \in \{1,2,3\}\,, \tag{2.6}$$

with the Pauli matrices

$$\sigma^1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma^2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma^3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.7}$$

The gamma matrices fulfill the (defining) property

$$\gamma^j \gamma^k + \gamma^k \gamma^j = 2\eta^{jk} \tag{2.8}$$

for $\eta^{jk} = \mathrm{diag}(1,-1,-1,-1)$.

The space of complex four-component Dirac spinors can be endowed with an inner product of signature $(2,2)$, which we denote by $\overline{\psi}\phi$, where $\overline{\psi} := \psi^\dagger \gamma^0$ is the usual adjoint spinor.

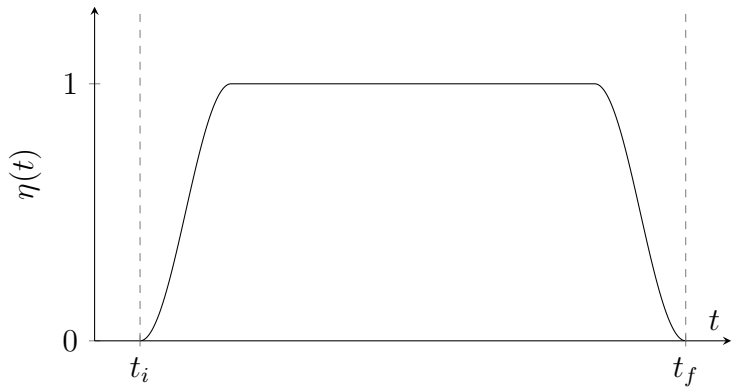

Figure 2.2: This is a possible cutoff function η for the lifetime of the universe from some initial time t_i to some final time t_f. As a Schwartz function it has to be smooth, so we cannot simply choose the indicator function $\chi_{[t_i,t_f]}$.

Integrating this inner product over space-time gives a bilinear form $\langle\cdot|\cdot\rangle$, which follows directly from the linearity of the multiplication by γ^0, the complex conjugation and the integral. In order to ensure that the temporal integral is finite, we choose a positive Schwartz function $\eta \in \mathcal{S}(\mathbb{R})$ and set

$$\langle\psi|\phi\rangle := \int_{\mathbb{R}} dt\, \eta\Big(\frac{t}{T}\Big) R^3 \int_{S^3} \overline{\psi}\phi\, d\mu_{S^3}\,, \tag{2.9}$$

where $d\mu_{S^3}$ is the volume element on S^3 from equation (2.5) and the parameter $T > 0$ describes the lifetime of the universe. Note that η is merely used to render the integration finite and is of no physical relevance. For practical purposes it is best to think of η as a smooth cutoff function, see Figure 2.2.

Let us now move on to the definition of the Dirac operator. Given the Levi-Civita connection on the tangent bundle of M, we can construct a connection on the spinor bundle $Q \times_\kappa \mathbb{C}^4$. This allows us to construct the Dirac operator as a linear, first order differential operator. For an extensive discussion of the general construction see [19, 16]. In [15, Section 2], it is shown that in our case the Dirac operator can be written as

$$\mathcal{D} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \otimes \mathbb{1}_{\mathbb{C}^2}\, i\,\partial_t + \frac{1}{R} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \otimes \mathcal{D}_{S^3}\,, \tag{2.10}$$

where $\mathcal{D}_{S^3}$ denotes the Dirac operator on S^3. Its explicit form is not relevant at the moment. We can then write down the Dirac equation in the vacuum

$$(\mathcal{D} - m)\psi = 0\,. \tag{2.11}$$

On the solution space of the Dirac equation, the probability integral gives rise to the scalar product

$$(\psi|\phi) := R^3 \int_{S^3} \psi^\dagger \phi\, d\mu_{S^3}\,. \tag{2.12}$$

This is well-defined, i. e. independent of t, for the same reasons as in the argument after (1.15).

Let us step back for a moment and see, where we are heading. We are looking for a simple, but non-trivial model that can be evaluated numerically to gain first insights about how minimizers of the causal action principle look like. In the beginning, we do not expect to be able to compute the action for the whole solution space of the Dirac equation, not even if we restrict ourselves to negative energy solutions. The latter case corresponds to the completely filled *Dirac sea*. The notion of the *Dirac sea* is that in the vacuum only the states with negative energy are populated – a sea of particles, as illustrated in Figure 2.3.

To model particles and anti-particles, a negative state can "jump out of the sea" and occupy a positive state, thereby creating a particle anti-particle pair. A detailed discussion of the meaning and the interpretation of the Dirac sea and its relation to the *principle of the fermionic projector* can be found in [5].

We rather want to restrict ourselves even more to a finite-dimensional subspace, so that we eventually end up with finitely many degrees of freedom.

A somewhat natural starting point to pick a finite subspace of the solution space of the Dirac equation, is to analyze its spectrum. Because the Dirac operator $\mathcal{D}_{S^3}$ is an elliptic operator on a compact manifold S^3, spectral theory yields that it is essentially self-adjoint with domain of definition $C^\infty(S^3)^2 \subset L^2(S^3)^2$. It has the purely discrete spectrum

$$\sigma(\mathcal{D}_{S^3}) = \left\{ \pm\frac{3}{2}, \pm\frac{5}{2}, \pm\frac{7}{2}, \dots \right\}, \tag{2.13}$$

see [19] and [15, Appendix A] for details. In these references, the dimension of the corresponding eigenspaces is found to be

$$\dim(\ker(\mathcal{D}_{S^3} - \mu)) = \mu^2 - \frac{1}{4}\,. \tag{2.14}$$

The basic idea is to restrict ourselves to a finite number of momentum modes. For a given momentum cutoff Λ and a number $N \in \mathbb{N}$ of momentum modes, we choose eigenvalues $\mu_1, \dots, \mu_N \in \sigma(\mathcal{D}_{S^3})$ with $|\mu_n| \leq \Lambda$. Later on we will simply pick the N smallest eigenvalues by absolute value.

2.2.3 Ansatz for the kernel of the fermionic operator

We have argued in section 1.2 that the object of interest should be the kernel of the fermionic operator. Before making an explicit ansatz, let us examine, which properties we require the kernel of the fermionic operator to exhibit.

First of all, we want it to be *static*, *spatially homogeneous* and *isotropic*. To a reader with a background in general relativity or cosmology, those terms probably sound familiar. However, the term *static* has a slightly different meaning here than in a *static universe*. We will explain the precise meaning of the assumptions as soon as we have written down an explicit ansatz.

Second, we want it to satisfy the Dirac equation – at least it should not be far from an exact solution. In light of the *regularization procedure* – described in detail in [7, 11] – we might want to allow for some additional freedom to perturb the kernel of the fermionic operator away from an exact solution of the Dirac equation. This will be made precise shortly.

While it is not in the scope of this thesis to explain the specifics of the regularization procedure, one can think about it in the following way: Since causal fermion systems are tailored to take into account physics beyond the standard model, we would like to be able to occupy states of different masses and arbitrarily change the frequency or spin orientation of the states. In the continuum limit, we do not expect the Minkowski vacuum to be an exact minimizer of the causal action principle, but only a suitably regularized version thereof. The class of possible regularizations is extremely big and finding "good" ones a difficult task. Thus, the additional freedom can be used to model some of the regularization effects discussed in [7, 11].

With these assumptions in mind, we make the following ansatz for the kernel of the fermionic operator

$$P(t,x;t',x') = \sum_{n=1}^{N} e^{-i\,\omega_n (t-t')} \begin{pmatrix} \Omega_n + m_n & K_n \\ -K_n & -\Omega_n + m_n \end{pmatrix} \otimes E_{\mu_n}(x,x')\,. \tag{2.15}$$

A few explanations and remarks are in order. By $E_{\mu_n}(x,x')$ we denote the integral kernel of the spectral projector of $\mathcal{D}_{S^3}$ corresponding to the eigenvalue μ_n. Recall that if E_{μ_n} is the spectral projector to the eigenvalue μ_n and ψ is a Dirac spinor, we can represent it by

$$\big(E_{\mu_n}\psi\big)(x) = \int_{S^3} E_{\mu_n}(x,x')\psi(x')\,d\mu_{S^3}(x')\,. \tag{2.16}$$

Remember that we found the purely discrete spectrum of the Dirac operator on S^3 in equation (2.13). Subsequently, we chose N of these eigenvalues as occupied momentum modes and restricted ourselves to the corresponding eigenspaces of those N eigenvalues. Each of these eigenvalues comes with a spectral projection operator to the corresponding eigenspace, which we henceforth represent by the integral kernel $E_{\mu_n}(x,x')$ as in (2.16). We will postpone an explicit formula for $E_{\mu_n}(x,x')$ until we have further simplified our initial ansatz. So far it involves the free parameters

$$m_n \in \mathbb{R}^+ \quad \text{and} \quad K_n, \omega_n, \Omega_n \in \mathbb{R} \qquad \text{for} \qquad n \in \{1,\ldots,N\}\,, \tag{2.17}$$

i. e. we are currently dealing with $4N$ degrees of freedom.

Let us examine, whether the ansatz satisfies our assumptions. Isotropy means that it is independent of ϑ, φ. Although $E_{\mu_n}(x,x')$ does depend on ϑ, φ (see σ^r in (2.48) and (2.49)), we find $\sigma^r(\sigma^r)^\dagger = (\sigma^r)^2 = \mathbb{1}_{\mathbb{C}^2}$, and just like the Pauli matrices themselves, σ^r has eigenvalues ± 1. Thus the closed chain and thereby also the Lagrangian are indeed independent of ϑ, φ after all.

Spatial homogeneity refers to translational invariance in the spatial domain. In other words, our ansatz ought to depend only on the difference $x - x'$. Later in this section we will construct E_{μ_n} based on this assumption, hence it is automatically built in.

Finally, by *static* we imply that the ansatz depends only on the difference $t - t'$, not on the individual values of t, t', which is obviously fulfilled. Note that this is a completely different notion than in the common term *static universe* (or *stationary universe*), which is used for a temporally (and spatially) infinite universe that is neither contracting nor expanding.

Next, we have to verify that it is "close to a solution of the Dirac equation". To this end, set

$$\begin{aligned} m_n &:= m \text{ for some } m \in \mathbb{R}^+ \\ K_n &:= \frac{\mu_n}{R} \\ \omega_n := \Omega_n &:= \pm\sqrt{m_n^2 + K_n^2}\,. \end{aligned} \tag{2.18}$$

We claim that for these values $P(t, x; t', x')$ is a solution of the Dirac equation.

We first compute the action of the Dirac operator $\mathcal{D}$ on P and suppress two-dimensional unit matrices for simplicity. Also, due to linearity, we need to consider only one term in the sum of the kernel of the fermionic operator (2.15) and thus drop the index n. We also leave out the arguments x, x' of E_μ. Using $\mathcal{D}_{S^3} E_\mu = \mu E_\mu$ we find

$$\begin{aligned} \mathcal{D}P &= \left(\begin{pmatrix} i\partial_t & 0 \\ 0 & -i\partial_t \end{pmatrix} + \frac{1}{R} \begin{pmatrix} 0 & \mathcal{D}_{S^3} \\ -\mathcal{D}_{S^3} & 0 \end{pmatrix} \right) e^{-i\,\Omega(t-t')} \begin{pmatrix} (\Omega+m)E_\mu & KE_\mu \\ -KE_\mu & (-\Omega+m)E_\mu \end{pmatrix} \\ &= e^{-i\,\Omega(t-t')} \begin{pmatrix} -\frac{K\mu}{R} + \Omega(m+\Omega) & \frac{(m-\Omega)\mu}{R} + K\Omega \\ -\frac{(m+\Omega)\mu}{R} + K\Omega & -\frac{K\mu}{R} - \Omega(m-\Omega) \end{pmatrix} \otimes E_\mu\,. \end{aligned} \tag{2.19}$$

The second part of the Dirac equation is simply

$$-m\,P = -m\,e^{-i\,\Omega(t-t')} \begin{pmatrix} \Omega+m & K \\ -K & -\Omega+m \end{pmatrix} \otimes E_\mu\,. \tag{2.20}$$

Adding (2.19) and (2.20) yields (omitting the common phase factor and the tensor product with E_μ)

$$\begin{aligned} \begin{pmatrix} \Omega^2 - m^2 - \frac{K\mu}{R} & \frac{(m-\Omega)(\mu-KR)}{R} + K\Omega \\ \frac{(m-\Omega)(\mu-KR)}{R} + K\Omega & \Omega^2 - m^2 - \frac{K\mu}{R} \end{pmatrix} &\overset{\mu=KR}{=} \begin{pmatrix} \Omega^2 - m^2 - K^2 & 0 \\ 0 & \Omega^2 - m^2 - K^2 \end{pmatrix} \\ &\overset{\Omega^2=m^2+K^2}{=} \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}. \end{aligned} \tag{2.21}$$

Now we see, how not fixing the parameters as in (2.18) right away gives some additional freedom. Only for the special values in (2.18) does our ansatz satisfy the Dirac equation. By allowing them to vary, we can "perturb P away" from the exact solution. This is illustrated by the shaded area in Figure 2.3, where we draw a typical picture of the Dirac sea. However, we shall not forget that eventually we will have to minimize the action with respect to all these parameters. In light of the complicated procedure of computing the action even for one fixed set of parameters, one might anticipate that we hit the limits of computation rather quickly. As a consequence, we should keep the number of parameters as large as necessary, but at the same time as small as possible.

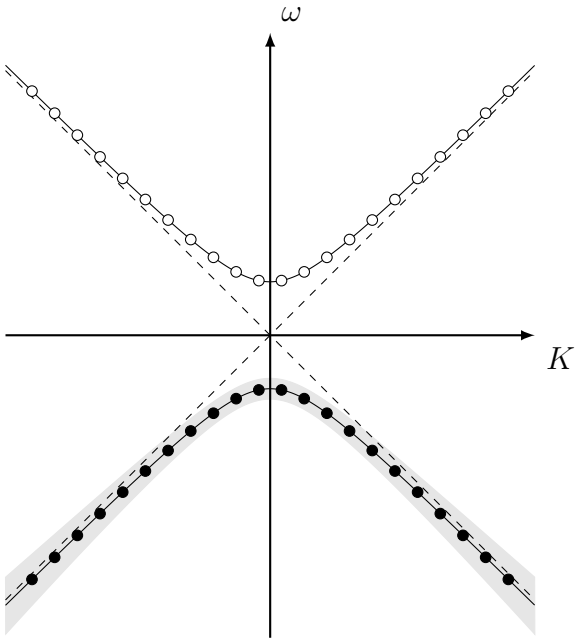

Figure 2.3: An illustration of the Dirac sea. The filled circles represent particles in negative energy states that are exact solutions to the Dirac equation. The shaded area suggests that we allow our ansatz to be perturbed away from the exact solution.

For mathematical simplicity and consistency, we make the following further assumptions:

negative-definite image: Considering $P(t,x;t',x')$ as the integral kernel of an operator P, we want the image of P to be negative-definite with respect to the inner product (2.9). This is equivalent to imposing

$$\Omega_n = -\sqrt{m_n^2 + K_n^2}\,. \tag{2.22}$$

Thereby, we reduce the number of free parameters from $4N$ to $3N$. This is the reason, why we have only drawn particles in the lower half of Figure 2.3. The black circles correspond to the shells we are successively occupying by increasing N in our ansatz for the kernel of the fermionic operator.

vector-scalar structure: In order to simplify the mathematical structure further, we impose a symmetry between the positive and negative momenta by demanding

$$P(t,x;t',x') = \frac{1}{2}\sum_{n=1}^{N} e^{-i\,\omega_n(t-t')} \sum_{\pm} \begin{pmatrix} \Omega_n + m_n & \pm K_n \\ \mp K_n & -\Omega_n + m_n \end{pmatrix} \otimes E_{\pm\mu_n}(x,x')\,. \tag{2.23}$$

In view of this symmetry, we assume without loss of generality that $\mu_1, \ldots, \mu_N \in \mathbb{R}^+$. While in principle one can choose freely from the infinitely many positive

eigenvalues, we will later simply pick the N smallest ones. In Figure 2.3, this amounts to occupying the uppermost two black spots for $N = 1$, adding the two neighboring ones for $N = 2$ and so on.

Remark. The vector-scalar structure does not directly reduce the number of free parameters. However, we get to higher-dimensional eigenspaces of $\mathcal{D}_{S^3}$ more quickly, because we always populate the shells of positive and negative eigenvalues simultaneously. For example, for $N = 2$ without the vector-scalar structure condition, we would choose the eigenvalues $\{\pm 3/2\}$. The corresponding eigenspaces are both two-dimensional. In contrast, with the vector-scalar structure, we choose $\{3/2, 5/2\}$ and the corresponding eigenspaces are two- and six-dimensional respectively.

The name *vector-scalar structure* is motivated by a general simplification in which the kernel of the fermionic operator is assumed to be of the form

$$P(t,x;t',x') = g_k(t,x;t',x')\gamma^k + h(t,x;t',x')\mathbb{1}_{\mathbb{C}^4}\,, \tag{2.24}$$

i. e. as a vector part g and a scalar part h, where g_k and h are complex-valued functions. Since the Pauli matrices (2.7), together with the unit matrix $\mathbb{1}_{\mathbb{C}^2}$ span the space of all hermitian 2×2 matrices, which in particular contains all orthogonal projections, and because of the form of the gamma matrices (2.6), we conclude that the kernel of the fermionic projector as in (2.23) is indeed of vector-scalar structure.

Setting $\rho_n = m_n/2$ and subsequently rescaling the momenta by $K_n \to K_n/m_n$ (which does not result in a loss of generality), we compute

$$\begin{aligned}
&\frac{1}{2}\begin{pmatrix} \Omega_n + m_n & \pm K_n \\ \mp K_n & -\Omega_n + m_n \end{pmatrix} = \frac{m_n}{2}\begin{pmatrix} \frac{\Omega_n}{m_n} + 1 & \pm\frac{K_n}{m_n} \\ \mp\frac{K_n}{m_n} & -\frac{\Omega_n}{m_n} + 1 \end{pmatrix} \\
&= \rho_n \begin{pmatrix} -\sqrt{1+\frac{K_n^2}{m_n^2}} + 1 & \pm\frac{K_n}{m_n} \\ \mp\frac{K_n}{m_n} & \sqrt{1+\frac{K_n^2}{m_n^2}} + 1 \end{pmatrix} \\
&\longrightarrow \rho_n \begin{pmatrix} 1 - \sqrt{1+K_n^2} & \pm K_n \\ \mp K_n & 1 + \sqrt{1+K_n^2} \end{pmatrix}
\end{aligned} \tag{2.25}$$

and can thus rewrite the ansatz (2.23) in the more convenient form

$$P(t,x;t',x') = \sum_{n=1}^{N} \rho_n\, e^{-i\,\omega_n (t-t')} \sum_{\pm} \begin{pmatrix} 1 - \sqrt{1+K_n^2} & \pm K_n \\ \mp K_n & 1 + \sqrt{1+K_n^2} \end{pmatrix} \otimes E_{\pm\mu_n}(x,x')\,. \tag{2.26}$$

In this form, it involves the $3N$ free parameters

$$\rho_n > 0 \quad \text{and} \quad \omega_n, K_n \in \mathbb{R} \qquad \text{for } n \in \{1,\ldots,N\}\,. \tag{2.27}$$

In light of the complexity of the action, this seems to be a reasonable dimension for the parameter space – at least for small N.

2.2.4 Computing the action

From here on, we simply follow the procedure outlined in section 1.2 to compute the action. We introduce the closed chain

$$A(t,x;t',x') = P(t,x;t',x')\; P(t,x;t',x')^* \,, \tag{2.28}$$

where the star denotes the adjoint with respect to the inner product on the spinors. We define this inner product pointwise, i. e. on each fiber of the spinor bundle, by inheritance of the inner product on $\mathbb{C}^4$. Recall that the spinor bundle is the associated bundle to the spin structure with respect to the spin representation on $\mathbb{C}^4$. Hence we simply use the standard scalar product on $\mathbb{C}^4$ together with the spin representation κ to define the inner product

$$\prec \psi \,|\, \phi \succ_x := -\big\langle \psi(x) \big| \gamma^0 \phi(x) \big\rangle_{\mathbb{C}^4} \tag{2.29}$$

for all spinors ψ, ϕ. Because the standard scalar product on $\mathbb{C}^4$ is positive definite and $\gamma^0 = \mathrm{diag}(1,-1) \otimes \mathbb{1}_{\mathbb{C}^2}$, we can directly conclude that the signature (p,q) of $\prec \cdot \,|\, \cdot \succ$ indeed fulfills $p, q \leq 2$.

The adjoint of the kernel of the fermionic operator is then given by

$$P(t,x;t',x')^* = \gamma^0 P(t,x;t',x')^\dagger \gamma^0 \,, \tag{2.30}$$

because

$$\begin{aligned} \prec \psi \,|\, P\phi \succ &= \big\langle \psi \big| \gamma^0 P \phi \big\rangle = \big\langle \gamma^0 \psi \big| P \gamma^0 \gamma^0 \phi \big\rangle = \big\langle \gamma^0 P^\dagger \gamma^0 \psi \big| \gamma^0 \phi \big\rangle & (2.31) \\ &= \big\langle P^* \psi \big| \gamma^0 \phi \big\rangle = \prec P^* \psi \,|\, \phi \succ \,, & (2.32) \end{aligned}$$

where we use $\gamma^0 \gamma^0 = \mathbb{1}_{\mathbb{C}^4}$ and $(\gamma^0)^\dagger = \gamma^0$. We leave out notational clutter for readability.

Before we move on to the explicit computation of the closed chain, we notice one important detail. In our ansatz for P in (2.26), we can consistently pull out one of the phase factors in the sum, for example

$$P(t,x;t',x') = e^{-i\,\omega_1 (t-t')} \sum_{n=1}^{N} \rho_n \; e^{-i\,(\omega_n - \omega_1)(t-t')} \sum_{\pm} \ldots \tag{2.33}$$

Because of (2.28) and (2.30), this phase factor is going to vanish in the closed chain and has therefore no effect on the Lagrangian and the action. Hence we can drop it already. Thereby, we reduce the number of the free parameters ω_n from N to $N-1$. In total, we now have $3N-1$ degrees of freedom. From here on, without loss of generality, we will simply assume $\omega_1 = 0$.

Denoting the eigenvalues of $A(t,x;t',x')$ by $\lambda_1,\ldots,\lambda_4$, we introduce the Lagrangian $\mathcal{L}$ as in Definition 5 (keeping in mind the remarks after Definition 9)

$$\begin{aligned}\mathcal{L}(t,x;t'x') &= \left|A(t,x;t',x')^2\right| - \frac{1}{4}\left|A(t,x;t',x')\right|^2 = \sum_{i=1}^{4}|\lambda_i|^2 - \frac{1}{4}\Big(\sum_{i=1}^{4}|\lambda_i|\Big)^2 \\ &= \sum_{i=1}^{4}|\lambda_i|^2 - \frac{1}{4}\sum_{i,j=1}^{4}|\lambda_i|\,|\lambda_j| = \frac{1}{8}\sum_{i,j=1}^{4}|\lambda_i|^2 - \frac{1}{4}\sum_{i,j=1}^{4}|\lambda_i|\,|\lambda_j| + \frac{1}{8}\sum_{i,j=1}^{4}|\lambda_j|^2 \\ &= \frac{1}{8}\sum_{i,j=1}^{4}\left(|\lambda_i|^2 - 2|\lambda_i|\,|\lambda_j| + |\lambda_j|^2\right) = \frac{1}{8}\sum_{i,j=1}^{4}\left(|\lambda_i| - |\lambda_j|\right)^2 . \end{aligned} \tag{2.34}$$

We will now show that due to the vector-scalar structure of P, the eigenvalues of the closed chain are always two-fold degenerate. Let us omit the arguments of P in the following computations and make use of the Feynman slash notation: $\not{a} := a_k\gamma^k$. Moreover, recall that $\gamma^0\gamma^k\gamma^0 = -\gamma^k = (\gamma^k)^\dagger$ for $k\in\{1,2,3\}$ and that $(\gamma^0)^2 = \mathbb{1}_{\mathbb{C}^4}$.

The vector-scalar structure implies

$$P = g_k\gamma^k + h\mathbb{1}_{\mathbb{C}^4} = \not{g} + h \tag{2.35}$$

for some g_k and h. Hence, we find

$$P^* = \gamma^0 P^\dagger\gamma^0 = \gamma^0(\not{g}^\dagger + \overline{h})\gamma^0 = \overline{g_k}\gamma^0(\gamma^k)^\dagger\gamma^0 + \overline{h} = \overline{g_k}\gamma^k + \overline{h} \tag{2.36}$$

and subsequently

$$\begin{aligned} A = P\,P^* &= (\not{g}+h)(\overline{\not{g}}+\overline{h}) = \not{g}\overline{\not{g}} + \not{g}\overline{h} + h\overline{\not{g}} + h\overline{h} \\ &= \underbrace{\frac{1}{2}\big[\not{g},\overline{\not{g}}\big]}_{=:A_1} + \underbrace{h\overline{\not{g}} + \overline{h}\not{g}}_{=:A_2} + \underbrace{g_j\overline{g^j} + h\overline{h}}_{=:\nu}, \end{aligned} \tag{2.37}$$

where we make use of the defining property of the gamma matrices (2.8). Next, using $\not{a}\not{a} = a_k a^k\mathbb{1}_{\mathbb{C}^4}$, we show that the matrices A_1 and A_2 anti-commute

$$\begin{aligned} A_1A_2 &= \frac{1}{2}\big[\not{g},\overline{\not{g}}\big](h\overline{\not{g}} + \not{g}\overline{h}) = \frac{1}{2}(\not{g}\overline{\not{g}} - \overline{\not{g}}\not{g})(h\overline{\not{g}} + \not{g}\overline{h}) \\ &= \frac{1}{2}(h\not{g}\overline{\not{g}\not{g}} - h\overline{\not{g}}\not{g}\overline{\not{g}} + \overline{h}\not{g}\overline{\not{g}}\not{g} - \overline{h}\overline{\not{g}}\not{g}\not{g}) = \frac{1}{2}(h\overline{\not{g}\not{g}}\not{g} - h\overline{\not{g}}\not{g}\overline{\not{g}} + \overline{h}\not{g}\overline{\not{g}}\not{g} - \overline{h}\not{g}\not{g}\overline{\not{g}}) \\ &= \frac{1}{2}(h\overline{\not{g}} + \overline{h}\not{g})(\overline{\not{g}}\not{g} - \not{g}\overline{\not{g}}) = -\frac{1}{2}(h\overline{\not{g}} + \overline{h}\not{g})(\not{g}\overline{\not{g}} - \overline{\not{g}}\not{g}) \\ &= -\frac{1}{2}(h\overline{\not{g}} + \not{g}\overline{h})\big[\not{g},\overline{\not{g}}\big] = -A_2A_1 . \end{aligned} \tag{2.38}$$

With $A_1A_2 + A_2A_1 = 0$ we find

$$(A-\nu)^2 = (A_1+A_2)^2 = A_1A_1 + A_1A_2 + A_2A_1 + A_2A_2 = A_1^2 + A_2^2 . \tag{2.39}$$

Let us write out the expressions for A_1^2 and A_2^2

$$\begin{aligned} A_1^2 &= \frac{1}{4}\big[\not{g}, \overline{\not{g}}\big]^2 = \frac{1}{4}\big((\not{g}\overline{\not{g}})^2 - \not{g}\overline{\not{g}}\not{g}\overline{\not{g}} - \overline{\not{g}}\not{g}\overline{\not{g}}\not{g} + (\not{g}\overline{\not{g}})^2\big) \\ &= \frac{1}{4}\big((g\overline{g})^2 - 2g^2\overline{g}^2 + (g\overline{g})^2\big) = \frac{1}{2}(g\overline{g})^2 - \frac{1}{2}g^2\overline{g}^2 \end{aligned} \tag{2.40}$$

and again with (2.8)

$$\begin{aligned} A_2^2 &= (h\overline{\not{g}} + \not{g}\overline{h})^2 = h^2\overline{\not{g}}^2 + \overline{h}^2\not{g}^2 + h\overline{h}\overline{\not{g}}\not{g} + h\overline{h}\not{g}\overline{\not{g}} \\ &= h^2\overline{g}^2 + \overline{h}^2 g^2 + h\overline{h}(\overline{\not{g}}\not{g} + \not{g}\overline{\not{g}}) = h^2\overline{g}^2 + \overline{h}^2 g^2 + h\overline{h}g_j\overline{g_k}\eta^{jk} \\ &= (h\overline{g} + \overline{h}g)^2 \,. \end{aligned} \tag{2.41}$$

Both A_1^2 and A_2^2 are scalar multiples of the unit matrix $\mathbb{1}_{\mathbb{C}^4}$. Hence we can interpret (2.39) as a quadratic equation for A. As a consequence, the roots $\lambda_\pm$ of this quadratic equation are the zeros of the characteristic polynomial for the closed chain A. From (2.39) we conclude directly

$$\lambda_\pm = \nu \pm \sqrt{A_1^2 + A_2^2}\,. \tag{2.42}$$

This finally yields the desired result that the eigenvalues of the closed chain are always two-fold degenerate.

We also see immediately that for $A_1^2 + A_2^2 < 0$ the eigenvalues form a complex conjugate pair and for $A_1^2 + A_2^2 > 0$ they are both real. Furthermore, in the latter case, they have the same sign since $\lambda_+\lambda_- > 0$ as one verifies in a short calculation.

These observations enable us to rewrite the Lagrangian in a way, even better suited for numerical computations. Take a look at the Lagrangian as in (2.34):

$$\mathcal{L}(t,x;t',x') = \sum_{i=1}^{4} |\lambda_i|^2 - \frac{1}{4}\left(\sum_{i=1}^{4} |\lambda_i|\right)^2 = \frac{1}{8}\sum_{i,j=1}^{4}\big(|\lambda_i| - |\lambda_j|\big)^2\,. \tag{2.43}$$

We show that we can rewrite (2.43) as

$$\mathcal{L}(t,x;t',x') = \max\left(0, \operatorname{Tr}(A^2) - \frac{1}{4}\operatorname{Tr}(A)^2\right). \tag{2.44}$$

To this end we distinguish the following two cases.

Case 1: The two-fold degenerate eigenvalues form a complex conjugate pair (in particular, assume their imaginary part is not zero). Then they do not contribute to the Lagrangian in (2.43), because all four eigenvalues have equal absolute values. In (2.44), this case is taken care of by the max function, because then

$$\operatorname{Tr}(A^2) - \frac{1}{4}\operatorname{Tr}(A)^2 = 4\operatorname{Re}(\lambda_+)^2 - 4\operatorname{Im}(\lambda_+)^2 - 4\operatorname{Re}(\lambda_+)^2 = -4\operatorname{Im}(\lambda_+)^2 < 0\,. \tag{2.45}$$

Case 2: Both eigenvalues are real. In particular, the square of the absolute value reduces to the square and hence

$$\sum_{i=1}^{4} |\lambda_i|^2 = \mathrm{Tr}(A^2), \qquad \text{and} \qquad \Big(\sum_{i=1}^{4} |\lambda_i|\Big)^2 = \mathrm{Tr}(A)^2\,. \tag{2.46}$$

In this case, our new expression (2.44) clearly coincides with (2.43) again.

Compare these two cases to Definition 3, where we have introduced a causal structure on $\mathcal{F}$. Spacelike separated points do not contribute to the action. Therefore the causal action principle is compatible with the causal structure. This gives rise to the name "causal action principle". Hence, all interactions are local at the most fundamental level in the sense that spacelike separated points do not interact. The representation of the Lagrangian in (2.44) is better suited for numerical treatment, because computing the trace of a matrix is usually less expensive than computing its eigenvalues.

As a last step, we have to integrate the Lagrangian to get the action. Instead of integrating over t, x and t', x', we make use of the required spatial and time symmetry in our ansatz of the kernel of the fermionic operator. They imply that P only depends on the difference of two space-time points. Hence we can arbitrarily fix one of the arguments of the Lagrangian and integrate only over the remaining one. We choose to set $t' = 0$ and $x' = \mathfrak{n}$, where $\mathfrak{n}$ is the north pole of S^3. We finally arrive at

$$\mathcal{S} = \int_{\mathbb{R}} dt\, \eta\Big(\frac{t}{T}\Big)\; R^3 \int_{S^3} \mathcal{L}(t, x; 0, \mathfrak{n})\, d\mu_{S^3}(x)\,. \tag{2.47}$$

At this point, we can fill in the last missing piece: Based on the same symmetry argument, in [15, Appendix A] we find an explicit formula for the integral kernel of the spectral projectors of $\mathcal{D}_{S^3}$. They are obtained by explicitly computing a basis of eigenvectors for each eigenspace. In the final form they read

$$E_{\pm(n+\frac{3}{2})}(x, \mathfrak{n}) = \frac{(n+2)!}{8\pi^{\frac{3}{2}}\,\Gamma(n+\frac{3}{2})}\Big(\cos(r/2)\; P_n^{(\frac{1}{2},\frac{3}{2})}(r) \mp i\sigma^r \sin(r/2)\; P_n^{(\frac{3}{2},\frac{1}{2})}(r)\Big)\,, \tag{2.48}$$

where

$$\sigma^r := \sin(\vartheta)\,\cos(\varphi)\,\sigma^1 + \sin(\vartheta)\,\sin(\varphi)\,\sigma^2 + \cos(\vartheta)\,\sigma^3 \tag{2.49}$$

and $P_n^{(\cdot,\cdot)}$ are the Jacobi polynomials, see [1]. Equation (2.48) holds for all $n \in \{0, 1, 2, \dots\}$. As we have already mentioned, we only consider the positive eigenvalues of $\mathcal{D}_{S^3}$ (due to the vector-scalar structure) and we will number them in increasing order

$$\mu_1 = \frac{3}{2}, \quad \mu_2 = \frac{5}{2}, \quad \mu_3 = \frac{7}{2}, \;\dots \tag{2.50}$$

Then we find the following relation to the integral kernel (2.48)

$$E_{\pm\mu_n}(x, \mathfrak{n}) = E_{\pm((n-1)+\frac{3}{2})}(x, \mathfrak{n}) \tag{2.51}$$

for all $n \in \{1, 2, 3, \dots\}$.

The causal action principle is to

$$\text{minimize } \mathcal{S} \text{ under variations of } P\,. \tag{2.52}$$

In particular, we are eventually interested in the limit $T \to \infty$. However, in chapter 3, we will discover that the scenarios where the limit $T \to \infty$ becomes relevant, currently lie beyond our numerical capabilities.

2.2.5 Bearing in mind the constraints

Now that we have set the objective, we still need to take into account the constraints. Recall Definition 6 of the causal action principle. The volume constraint is already incorporated in our model by the choices we have made considering the metric spacetime M and by working on the solution space of the Dirac equation.

Next, let us implement the trace constraint

$$\int_{\mathcal{F}} \mathrm{tr}(x)\, d\rho(x) = \text{const}\,. \tag{2.53}$$

In our model, the space $\mathcal{F}$ amounts to self adjoint operators on the solution space of the Dirac equation. We have restricted ourselves to a finite number of eigenvalues μ_n of the Dirac operator and the corresponding eigenspaces and weighted each of these eigenspaces by a parameter $\rho_n \in \mathbb{R}^+$. Therefore, we can replace the integral above in our scenario by a weighted sum over the N eigenvalues we have chosen. The traces are given by the dimension of the associated eigenspaces, which is given in (2.14) as $\mu_n^2 - 1/4$. Weighted by the mass parameters ρ_n, we find for the trace constraint

$$\sum_{n=1}^{N} \rho_n \Big(\mu_n^2 - \frac{1}{4}\Big) = \text{const}\,. \tag{2.54}$$

The left hand side is simply a linear combination of the mass parameters ρ_n. Hence we can readily replace one of the N free parameters by a combination of the others. Without loss of generality, let us choose the constant on the right hand side to be 1 and set

$$\rho_N := \frac{1}{\mu_N^2 - \frac{1}{4}} \left(1 - \sum_{n=1}^{N-1} \rho_n \Big(\mu_n^2 - \frac{1}{4}\Big)\right). \tag{2.55}$$

Hence we have eliminated another parameter, leaving us with only $3N - 2$ degrees of freedom. For the simplest case $N = 1$, this yields a single free parameter, which makes it very easy to analyze, see section 3.2.

There, we will also work out that the boundedness constraint is indeed necessary for the existence of minimizers. Hence we better think about how we can factor it in right now.

While the other two constraints could be integrated seamlessly into the model, we have to modify it a little bit to incorporate the last one. Recall the boundedness

constraint from (1.7)

$$\mathcal{T} := \iint_{\mathcal{F}\times\mathcal{F}} |xy|^2 \, d\rho(x)\, d\rho(y) \leq C \,, \tag{2.56}$$

and that the eigenvalues of the product xy coincide with the eigenvalues of the closed chain A_{xy}. Moreover, because we can express the spectral weight $|xy|$ by the trace of the closed chain $\mathrm{Tr}(A_{xy})$, we incorporate the boundedness constraint in our model by adding the term $\kappa \,\mathrm{Tr}(A)^2$ to the original Lagrangian for some $\kappa > 0$. The new Lagrangian reads

$$\mathcal{L} := \max\left(0, \mathrm{Tr}(A^2) - \frac{1}{4}\,\mathrm{Tr}(A)^2\right) + \kappa\,\mathrm{Tr}(A)^2 \,. \tag{2.57}$$

It is important to always keep in mind that κ is not yet another free parameter. We fix the value of κ, while the free parameters are varied in the causal action principle. Hence, κ is not a part of the minimization and entirely controlled by us. In chapter 3, we will use different values of κ, but we fix them once and for all in the beginning and use them only to consider the limit $\kappa \to 0$ after the minimization.

We will observe that adding the κ term indeed guarantees the existence of minimizers. However, we could not implement it as a legit constraint of the variation principle in a strict sense. We did not restrict the space in which we vary the free parameters, because it is not clear, how one could make these restrictions explicit such that they can be implemented on a computer. Moreover, we will see in section 3.5 that optimization with complicated constraints is much harder computationally than unconstrained optimization. Therefore, we "force" the boundedness constraint upon our problem by the modification in (2.57) and will try to revert that intervention by taking the limit $\kappa \to 0$ in a plausible way afterwards.

From here on, if not otherwise specified, we will always refer to definition (2.57), when talking about the Lagrangian. While this version ensures the existence of minimizers for $\kappa > 0$, eventually we want to take the limit $\kappa \to 0$. To do this in a sensible way, we introduce the so called rescaling procedure. Because the desire for this limit becomes more intuitive after we have seen some results for $N = 1$, we postpone the discussion of the rescaling procedure to section 3.4.

To sum up, the volume constraint does not need special attention for the Dirac sphere. The trace constraint is very easy to implement. Fortunately, it allows us to fix one free parameter by a linear combination of the others, thereby eliminating one dimension of the parameter space. The boundedness constraint is respected by explicitly inserting it into the model. We still need to review how to make sense out of the limit $\kappa \to 0$.

2.3 The numerical recipe

In the last section of this chapter, we rehash our efforts so far and view them as a starting point for a numerical study. We lay out the procedure step by step in an almost pseudo-code like manner.

1. We begin with the implementation of the kernel of the fermionic operator. The final version of our ansatz reads

$$P(t,x;0,\mathfrak{n}) = \sum_{n=1}^{N} \rho_n \, e^{-i\,\omega_n t} \sum_{\pm} \begin{pmatrix} 1-\sqrt{1+K_n^2} & \pm K_n \\ \mp K_n & 1+\sqrt{1+K_n^2} \end{pmatrix} \otimes E_{\pm\mu_n}(x,\mathfrak{n}) \,. \tag{2.58}$$

   - We have the $3N-2$ free parameters $\rho_n > 0$ for $n \in \{1,\dots,N-1\}$, $K_n \in \mathbb{R}$ for $n \in \{1,\dots,N\}$ and $\omega_n \in \mathbb{R}$ for $n \in \{2,\dots,N\}$. The last mass parameter ρ_N is fixed by the trace constraint in (2.55) and we set the first frequency parameter ω_1 to zero.
   - Our chosen eigenvalues μ_n of the Dirac operator on the unit sphere S^3 are given in (2.50).
   - The integral kernels of the spectral operators of $\mathcal{D}_{S^3}$ are explicitly given in (2.48). Since we know the Pauli matrices and the Jacobi polynomials, we can readily implement $E_{\pm\mu_n}(x,\mathfrak{n})$.
   - The last two missing symbols x and t represent a point in space-time and will be integrated over later. Note that we have fixed the second space-time point (t',x') at $(0,\mathfrak{n})$. This carries over to the closed chain, the Lagrangian and the action.

   As a conclusion, we can compute the kernel of the fermionic operator (2.58) as a complex 4×4 matrix depending on the space-time point (t,x) as well as on $3N-2$ free parameters. To perform this step on a machine, we basically only need to compute matrix multiplications. (Of course, we assume that all basic arithmetic operations on real numbers are no problem for a computer.)

2. The closed chain is easily computed as in (2.28) using (2.30) for P^*. From a computational perspective, we need to be able to perform multiplications and complex conjugation of 4×4 matrices.

3. By virtue of (2.57), all we need for the Lagrangian is again matrix multiplication and computing the trace of complex 4×4 matrices (as well as the max function for real numbers). Recall that the Lagrangian is a scalar real-valued function of the space-time point (t,x). Also it still depends on the free parameters and κ has entered as yet another variable of the model.

4. The action can then be computed as in (2.47). First of all, we can set $R = 1$ for all practical purposes, as it merely results in a scale factor. Note that the precise values of the action are of no interest. Only the position of the minimum in our parameter space is of significance.

   For the spatial integration we use the volume measure $d\mu_{S^3} = \sin(r)^2 \sin(\theta)$ in (2.5). The Lagrangian is independent of ϑ and φ, due to the isotropy assumption in our ansatz for P. Integrating over ϑ, φ yields a constant factor, which

can again be neglected. The remaining spatial integration is simply

$$\int_0^\pi \mathcal{L}(t, r; 0, \mathfrak{n}) \, \sin(r)^2 \, dr \, , \tag{2.59}$$

where we indicate that $\mathcal{L}$ depends only on r, not on (ϑ, φ).

The temporal integration is a little bit more tricky. How do we choose η and the lifetime T? In principle, one could choose η for example like in Figure 2.2. It has compact support, which renders the integral finite. Subsequently, one would find the minimum of the action for a sequence of increasing values for T and analyze the limit as T goes to infinity.

However, in chapter 3, we will only analyze the two scenarios $N = 1$ and $N = 2$. It turns out that in the first one the Lagrangian is independent of t and in the second one it is periodic in t with a certain period $T_0 > 0$. Thus we need to perform the temporal integration either not at all or only over one period. Therefore, we can get rid of the function η in the integrand. The action then reads

$$\mathcal{S} = \int_0^{T_0} dt \int_0^\pi dr \, \mathcal{L}(t, r; 0, \mathfrak{n}) \, \sin(r)^2 \, . \tag{2.60}$$

This is a two-dimensional integral of a real-valued function over a bounded rectangular region.

Although one finds plenty literature on numerical integration, depending on the integrand $\mathcal{L}$ this could potentially become a cumbersome step. Because it is hard to develop an intuition about how the Lagrangian looks like and how it behaves, we indeed spent a lot of time to make sure that we get the integration right. Gaining such insights is actually the main goal of this thesis. One unpleasant feature that we can detect already is the max function. It makes the Lagrangian non-differentiable. Integrands with kinks are rarely an advantage when it comes to numerics.

5. The last step is to minimize $\mathcal{S}$ by varying the free parameters. To this end we consider κ to be fixed as part of the model. We emphasize again that just like N, it is not a free parameter. For the case $N = 1$, there is only the free parameter K_1 left. In this case, the minimization can easily be done, see section 3.2. For $N = 2$, we have in principle four free parameters, namely $\rho_1, K_1, K_2, \omega_2$. However, since the Lagrangian in this case is clearly periodic with period $T_0 = 2\pi/\omega_2$, we can arbitrarily fix ω_2, as long as we properly adjust the integration bounds. Only ratios of frequencies have a substantial impact on the Lagrangian, which is why they only come into play for $N \geq 3$. We choose $\omega_2 = 1$ and $T_0 = 2\pi$. Eventually, we arrive at only three free parameters for $N = 2$.

   Exactly like numerical integration, minimization is discussed in great detail in the literature. However, just as before, finding and adjusting an efficient method for the problem at hand almost always requires some knowledge and understanding

of the target function. In particular, we need to keep in mind that each single evaluation of the action for a fixed set of parameters might already be very expensive. The methods we have ultimately chosen are discussed in section 3.3 and section 3.5 for integration and optimization respectively.

Now that we have established our model and verified that it fulfills our needs, we can begin with the numerics.

3 Numerical computations

In this chapter, we compare and explain several numerical techniques relevant for the treatment of the Dirac sphere and at the same time present the results they supply. We intend to stay close to the actual problem and discuss numerical methods just in time when they are needed, see sections 3.3 and 3.5.

We start out with a few remarks on the choice of appropriate tools in section 3.1. Afterwards, we proceed with an analytic treatment of the Dirac sphere with one shell, i. e. $N = 1$, as far as possible in section 3.2. This leads to the necessity of the boundedness constraint. After an excursion about numerical integration in section 3.3, we revisit the $N = 1$ case numerically. We explain the idea of rescaling in section 3.4. After a second excursion about numerical optimization in section 3.5, we eventually show results for $N = 2$ in section 3.6. We conclude with a brief outlook on larger N and the limiting case $N \to \infty$ in the final section 3.7.

3.1 Which tools should we use?

Let us start out with some remarks on the tools best suited for our task. While in theory only the applied methods matter for the final result, but not which programming language, libraries or frameworks one uses, in practice, this is a crucial decision. Unfortunately, the consensus among a rather large community is that C/C++ and Fortran are the only choices for "real numerics". There is no doubt that these languages usually dominate whenever it comes to highly parallelized large scale computing tasks on distributed memory machines, where performance is the number one priority.

However, although there are C/C++ packages for basically everything, one cannot deny that fast prototyping, experimenting with different implementations and quick visualizations are not exactly the strengths of C/C++ and Fortran compared to other languages and frameworks. But this is precisely what we demand for our problem. Albeit computation time might become an issue for larger N, flexibility and fast prototyping definitely outweigh the demands for the last bit of performance for $N = 1$ and $N = 2$.

We have little analytic knowledge about how the Lagrangian looks like, therefore it is hard to choose a proper integration scheme in advance and we probably need to experiment to find a good one. We do not know, how the action depends on the free parameters, therefore we cannot decide on a minimization technique right away, let alone, pick suitable start points for local methods. Again, trial and error will play an important role.

Quick, expressive and flexible visualizations prove helpful in all of the above mentioned. Finally, we would like to let the computer perform steps 1-3 in section 2.3 too, which are mostly symbolic modifications. For the cases $N = 1$ and $N = 2$ those could probably still be done with pencil and paper, but for larger N it gets quite cumbersome and error-prone.

For the symbolic computations to set up the model, the two standard commercial general purpose computer algebra systems are Maple™ (see [21]) and Mathematica™ (see [26]), which are not only most widely distributed, but also provide the largest functionality. Due to the author's preferences, Mathematica has been used for this work.

The integration and minimization are clearly non-symbolic numerical tasks. We have first considered Matlab™ (see [17]), and Python (under the Python Software Foundation license), but also newer approaches such as Julia (under the MIT/GPL v2 license) and Rust (under the MIT/Apache 2.0 license). Although we are in favor of free and open-source software, again for reasons of experience, we gave Matlab a try.

After a while, the process of transferring the output of symbolic computations in Mathematica to Matlab became rather annoying and since we crosschecked all Matlab results in Mathematica anyway, we completely switched to Mathematica. Unfortunately, it still comes as a surprise to most people that Mathematica's numerical capabilities are close to and often even beat those of Matlab in terms of performance for most benchmarks. In addition, the set of available algorithms is immense and Mathematica allows for manual fine tuning of all kinds of options. Once one gets used to Mathematica's syntax and peculiarities in the way numerical computations are input, it is a very powerful and convenient tool not only for symbolic manipulations.

As a final remark, let us state that all tools and languages mentioned above do a pretty good job, if they are used correctly. Usually, experience and a deep understanding of how to use a tool outweigh the inherent differences.

3.2 One shell ($N = 1$)

3.2.1 Necessity of the boundedness constraint

For the simplest case, $N = 1$, we can still perform most computations analytically. The only free parameter is $K := K_1$ and we set $\rho_1 = 1$ in the ansatz (2.58). For the time being, let us disregard the boundedness constraint, i. e. let us set $\kappa = 0$ in (2.57). Moreover, we will discover that K only appears as K^2 in all quantities. Therefore we can assume $K \geq 0$ without loss of generality.

First, we compute the eigenvalues of the closed chain

$$\lambda_\pm = \frac{1 + (1 + K^2)\cos(r) \pm \sqrt{(1 + K^2)(1 + \cos(r))(1 - K^2 + (1 + K^2)\cos(r))}}{\pi^4}. \quad (3.1)$$

Recall that they are always two-fold degenerate (which is verified by the numerical computation). Moreover, there is no time dependency for $N = 1$. The eigenvalues

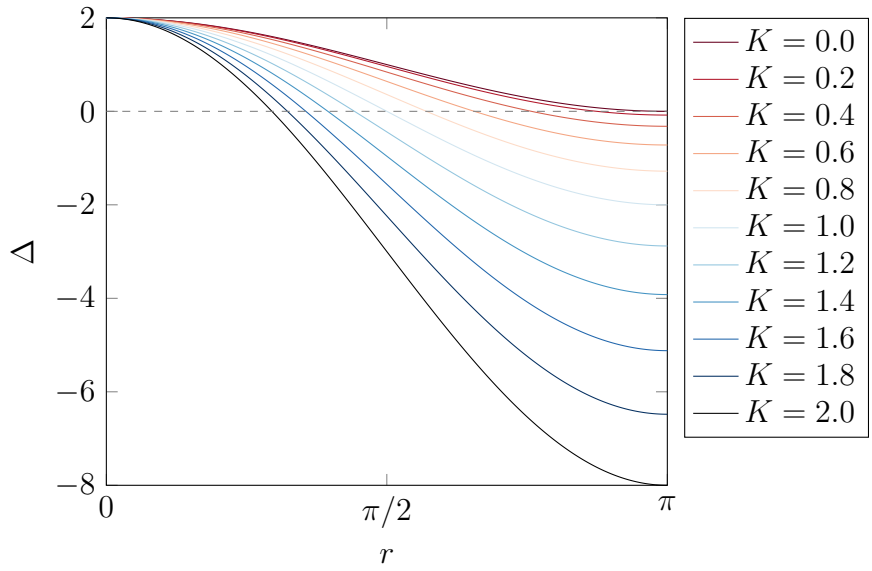

Figure 3.1: We show Δ as a function of r for various K. Clearly, the region where Δ is negative becomes larger with increasing K. At the same time the support of $\mathcal{L}$ in the r domain is shrinking.

are indeed of the form (2.42) and we can analyze the part under the square root to determine, for which combination of K and r they are going to contribute to the action. The best way to think about it is to ask: "For a given K, which part of $r \in [0, \pi]$ does contribute to the action". In other words: "Which r are not spacelike separated from the north pole $\mathfrak{n}$." The sign of the part under the square root is obviously determined by the sign of the expression

$$\Delta := 1 - K^2 + (1 + K^2)\cos(r)\,. \tag{3.2}$$

In Figure 3.1, we show Δ as a function of r for various K.

Solving $\Delta = 0$ for r yields $r_{\max}(K) = \arccos\big((K^2 - 1)(K^2 + 1)\big)$. We infer from Figure 3.1 that $\mathrm{supp}(\mathcal{L}_K) = [0, r_{\max}(K)]$, where we view $\mathcal{L}_K$ as a function of r only. The function $r_{\max}(K)$ decreases monotonically from $r_{\max}(0) = \pi$ to $r_{\max}(K) \to 0$ for $K \to \infty$. This means that for large K the eigenvalues of the closed chain become complex conjugate pairs almost everywhere on S^3 and hence the support of the Lagrangian shrinks to the empty set as K increases. However, we can not yet conclude that the action becomes zero for large K, because the Lagrangian might exhibit a δ distribution like behavior for $K \to \infty$.

We can write down $\mathcal{L}$ explicitly

$$\mathcal{L}(r) = \frac{1}{\pi^8}\,\max\Big(0, 8\,(1 + K^2)\cos\Big(\frac{r}{2}\Big)^2 \Delta\Big)\,. \tag{3.3}$$

In this simple setting, we can even perform the integral over space-time analytically, by making use of the fact that we know the support of the integrand. If we use $r_{\max}(K)$

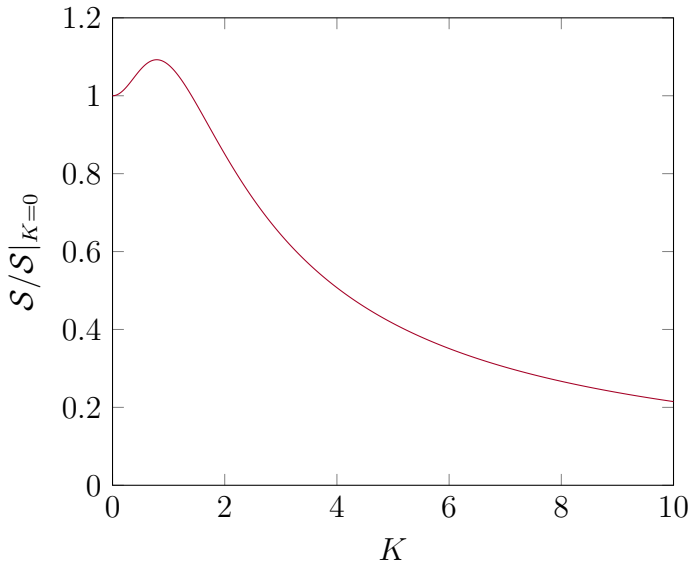

Figure 3.2: We show the (suitably normalized) action as a function of the only free parameter K. The plot already indicates that $\mathcal{S}$ decreases monotonically as $K \to \infty$.

as the upper integration bound, we can get rid of the max function in (3.3). This helps the computer algebra system, because the max function is generically hard to handle in symbolic integrations. Eventually, we find

$$\begin{aligned}\mathcal{S} &= \frac{4\pi}{3} \int_0^{r_{\max}(K)} \mathcal{L}(r)\, \sin(r)^2\, dr \\ &= \frac{2}{9\pi^7}\, \frac{2K(15+31K^2+9K^4+9K^6) + 3(1+K^2)^3(5-3K^2) \arccos\Big(1-\frac{2}{1+K^2}\Big)}{(1+K^2)^2}\,,\end{aligned} \tag{3.4}$$

which is shown in Figure 3.2. As already indicated by the plot, computing the series expansion

$$\mathcal{S} \propto \frac{1}{K} + \mathcal{O}\Big(\frac{1}{K^2}\Big)\,, \tag{3.5}$$

reveals that $\mathcal{S} \to 0$ for $K \to \infty$. Hence there exists no minimizer and we conclude that the boundedness constraint truly is a necessary condition for the existence of minimizers. Only at this point did we realize the resemblance of our model to example 2.8 and example 2.9 in [8]. This is how the "Dirac sphere" got its name.

3.2.2 Including the boundedness constraint

Now that we have shown the necessity of the boundedness constraint, let us take it into account. Using $\kappa > 0$ ensures that $\operatorname{supp}(\mathcal{L}) = [0, \pi]$ and hence we can not run into

the same problem as before. The Lagrangian in (3.3) only changes by the additional term

$$\kappa \operatorname{Tr}(A)^2 = \frac{16\kappa}{\pi^8}\big(1 + (1 + K^2)\cos(r)\big)^2 . \tag{3.6}$$

In Figure 3.3, we compare the Lagrangian for $\kappa = 0$ and $\kappa = 0.01$. One clearly recognizes how the κ term "kicks in" as K increases.

When aiming for the action, we have to switch from symbolic evaluation to numerics. Because of the additional κ term, we cannot do the same trick as above to get rid of the max function anymore. Instead, we perform the integration numerically for various values of κ. The result is shown in Figure 3.4. Again, we have normalized all curves by $\mathcal{S}|_{\kappa=0,K=0}$. Clearly, the κ term prevents the action from monotonically approaching zero, but instead "bends" it up for large K, such that a minimum is acquired. This verifies numerically the existence of minimizers for $\kappa > 0$, i. e. when the boundedness constraint is taken into account.

Before we move on, let us spend a few words on the numerical integration.

3.3 Numerical methods I: Integration

3.3.1 Preliminary considerations

The issues one has to worry most about in numerical integration are either non continuous functions, highly localized features such as peaks or bumps up to singularities (generally speaking, large gradients are a disadvantage) and highly oscillating functions. For high-dimensional integrals one has to deal with yet another class of problems, which we are not going to discuss here. It is also not in the scope of this thesis to explain the basic quadrature rules and integration strategies. We assume that the reader is familiar with Gaussian quadrature and has a basic understanding of numerics.

So far, our integrands did not look too bad. For $\kappa = 0$ they are smooth and do not exhibit any localized peaks or bumps, let alone singularities, and are also not oscillating. We expect almost any quadrature scheme to do a splendid job. Let us look at how Mathematica handles those integrals. Without specifying any further options, Mathematica uses a lot of built in machinery to determine, which strategy and method performs best. It first tries to analyze the integrand analytically for a certain time of symbolic processing. It tries to find symmetries, singularities, localized features and rapid oscillations. For a performance and precision analysis it is thus important to make sure one knows exactly what Mathematica is doing behind the scenes and closely monitor the procedure.

Mathematica currently implements about a dozen quadrature rules together with numerous integration strategies and various methods for higher-dimensional integrals. In our examples, it almost always uses the Gauss-Kronrod rule with a global adaptive strategy. Also commonly used is the Clenshaw-Curtis rule with the same strategy. We will first discuss the choice of the quadrature rules and come back to integration strategies later. The following is not restricted to the one-dimensional case. By

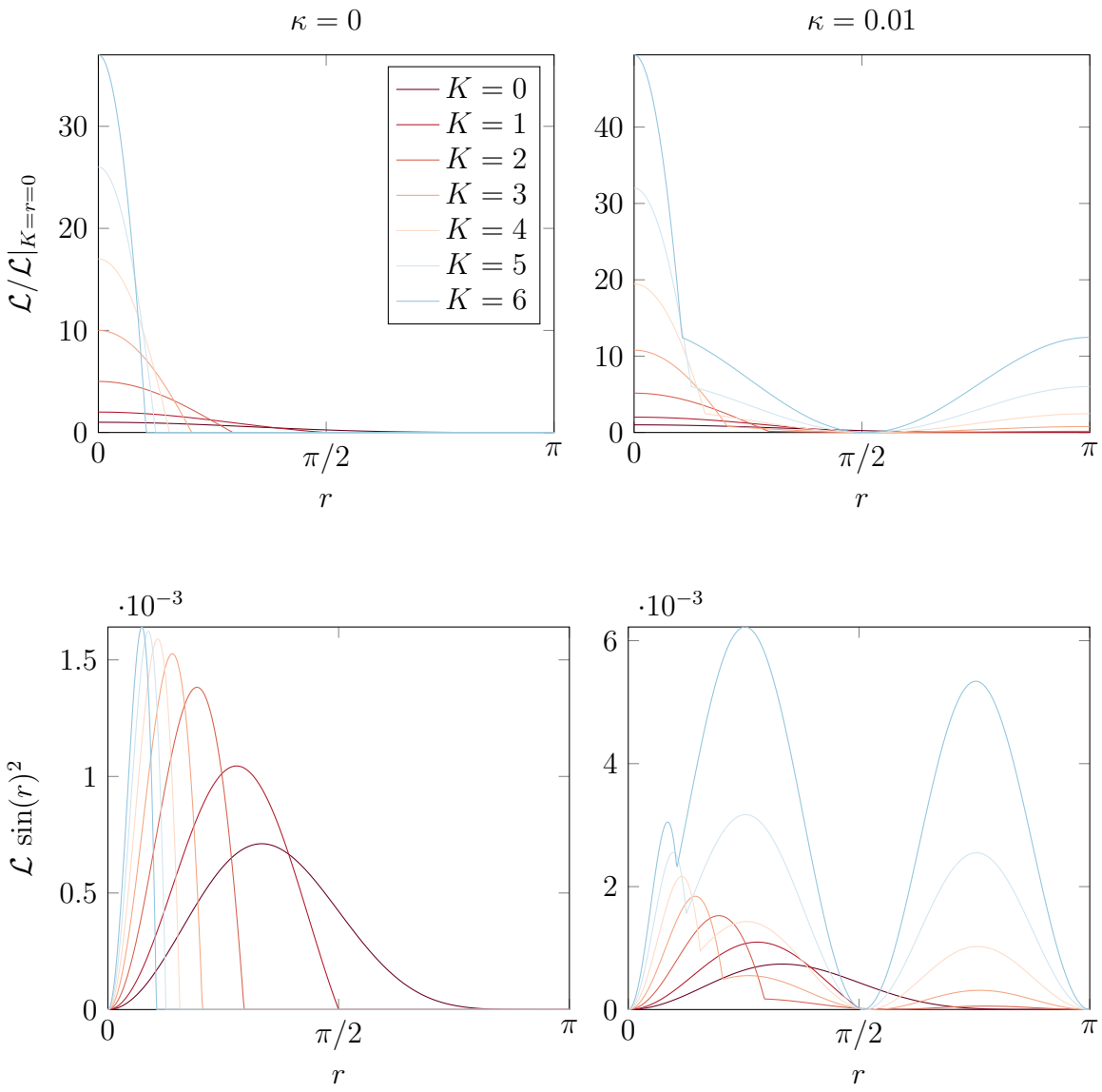

Figure 3.3: We plot (a suitably normalized version of) the Lagrangian as a function of r for several values of K in the first row, where in the left plot $\kappa = 0$ and in the right plot $\kappa = 0.01$. In the second row, we plot the actual integrand of the action again for several K and for $\kappa = 0$ (left) as well as $\kappa = 0.01$ (right). It is clearly visible, how a non-zero κ affects the support and smoothness of the Lagrangian and the integrand respectively. The legend in the first plot is valid for all four.

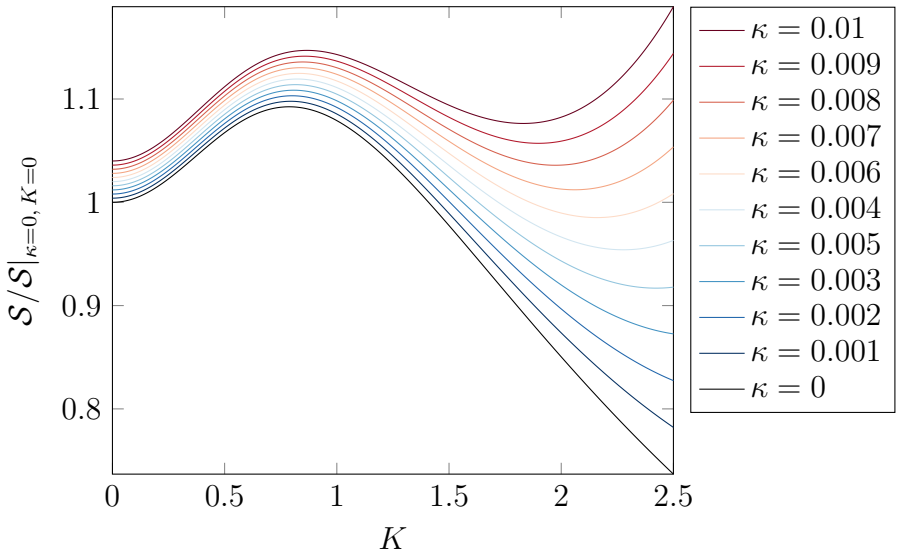

Figure 3.4: We plot (a suitably normalized version of) the action as a function of K for various values of κ. Note, how the κ term shifts the curve in the y direction, but most importantly, prevents a monotonic decrease as $K \to \infty$. In fact, for $\kappa > 0$ the action takes on a minimum and goes off to infinity for large K.

cartesian products of one-dimensional domains, all rules explained here also apply to higher-dimensional integrals.

The Gauss-Kronrod and Clenshaw-Curtis rule seem to be most adequate for our purposes, because they are used as so called *nested quadrature rules*. That means that we apply two quadrature schemes of different order, where we reuse the set of function evaluations of the lower order rule for the higher order rule. In order to construct an error estimate during the integration (when we obviously do not have the exact result at hand), one usually compares higher order estimates to lower order estimates.

Further, one specifies a goal for the absolute and/or relative difference of these comparisons and returns the current estimate when one has reached one of the two goals. We specify a *precision goal* and an *accuracy goal*. Both are given as positive real numbers, we call them g_p and g_a respectively. The integration stops as soon as the expression

$$(\text{error} \leq 10^{-g_p} \cdot \text{ integral value}) \quad \vee \quad (\text{error} \leq 10^{-g_a}) \tag{3.7}$$

evaluates to `true`.

Because the main computational effort in numerical integration usually lies in the frequent evaluation of the integrand, it is highly desirable to reuse prior function values. Instead of computing two completely independent sets of points and weights to obtain two approximations of different order, reusing the points of the lower order scheme and only adding some new ones for the higher order scheme can give a significant speedup.

The Gauss-Kronrod rule for example, does exactly that. It is a modified version of Gauss quadrature, in which the evaluation points are chosen precisely in such a

way, that they can be reused in a higher order approximation. For example, we can use a 7 point Gauss rule together with a 15 point Kronrod rule. Because the 7 grid points of the Gauss rule are also points of the Kronrod rule, one only has to perform 8 additional function evaluations in between the Gauss points to obtain the higher order Kronrod approximation. The Gauss-Kronrod rule is an open rule, i. e. it does not evaluate the integrand at the end points. While the Gauss rule with N points is exact for polynomials up to order $2N-1$, the Kronrod extension with an additional $N+1$ points ($2N+1$ in total) is exact for polynomials up to order $3N+1$ or $3N+2$ for even or odd N respectively, see [24]. For most Gauss rules, the weights are computed in $\mathcal{O}(N^2)$ time.

While for N evaluation points the Clenshaw-Curtis rule is only exact for polynomials up to order $N-1$ (as compared to Gauss quadrature schemes), it also has certain advantages. The Clenshaw-Curtis rule is based on an expansion of the integrand in Chebychev polynomials, hence the weights can be computed in $\mathcal{O}(N \log N)$ time via a discrete Chebychev transform by an algorithm analogous to the fast fourier transform. Moreover, it is naturally suited for nested quadrature, i. e. higher order approximations automatically contain the points of lower order approximations. One can still always use the same rule (as opposed to the Gauss-Kronrod rule).

In fact, since most integrands are not polynomials, the order to which a rule is exact is not the best criterion. It is known that an expansion in Chebychev polynomials converges rapidly for a large class of functions. Eventually, both, Clenshaw-Curtis and Gauss-Kronrod, are popular choices, which have proven to yield rapid convergence for a wide variety of integrands. In our simulations, we have compared the two along with several other rules for a set of representative integrands, i. e. a set of different combinations of parameters. In the end, it turned out that leaving the choice to Mathematica gave the best results. This was almost always the Gauss-Kronrod rule.

Now that we have a quadrature rule, we come to the integration strategy. In principle, one could use the above mentioned quadrature rules for the whole domain of integration and choose a huge number of points to make sure the estimate is good enough. However, this comes with an important disadvantage. Imagine our integrand is a low order polynomial for the bigger part of the domain, but at some small fraction of the integration region it exhibits highly oscillatory behavior with large amplitude. While the integral of the polynomial part is already exact for a reasonably low order approximation, to reach the precision and/or accuracy goals everywhere, one has to go to ever finer grids on the whole domain, until eventually, the errors are small enough also for the unpleasant part.

The immediate idea is to only refine the grid, where it is necessary. The two strategies we contemplated about are global and local adaptive strategies. Both, theoretical and practical considerations indicate that global adaptive strategies generally perform better than local adaptive strategies, see [18, 20]. Therefore, we focus on global adaptive strategies right from the start.

In a global adaptive strategy, we begin with the specified quadrature rule and a small number of grid points for the whole domain. If the termination condition (3.7) is not yet satisfied, the domain is divided into two halves and the integration is reevaluated

on each subdomain. For each of those integrals we check the precision and accuracy goal and recursively bisect the region further, if necessary. The new overall termination condition for a global adaptive strategy becomes

$$(\text{global error} \leq 10^{-g_p} \cdot \text{ global integral value}) \quad \vee \quad (\text{global error} \leq 10^{-g_a})\,, \tag{3.8}$$

i. e. we add up the errors of all subdomains to determine, whether the overall goal is satisfied.

The obvious advantage is that if the approximation on one subregion is good enough, the result is kept and the grid is not refined further on this part of the domain. However, all regions where the error estimate is still large are subdivided further. Thereby one reduces the overall number of function evaluations to a minimum. Specific implementations are rather complex, since one has to do a lot of book keeping for all the bisections and local error estimates. Otherwise we risk to miss local features that might be hiding in between the chosen points.

We make two final remarks before we dive into specific examples. First, note that the precision goal, i. e. the first condition in (3.8), involves the current value of the integral. If the actual value of the integral is zero, the precision goal can never be reached (in theory), because the error is positive and the right hand side of the inequality is zero. Therefore, adding a small positive constant to the integrand sometimes helps detect convergence. In the $N = 2$ case, we add $(2\pi^2)^{-1}$ to the integrand and subtract 1 from the action afterwards to compensate for it. (Recall that the integral is over $[0, 2\pi] \times [0, \pi]$.) Second, during the first bisections in the global adaptive strategy, the error can actually increase, as we resolve more localized features that were completely overseen in the previous resolution. However, at some point the error should decrease monotonically. One can specify, how many increases of the error one tolerates, before aborting the integration. As we do expect increasing errors, e. g. for the complex integrands in Figure 3.10, but also know that it will converge eventually, we allow for up to 10 000 increases.

3.3.2 Examples

Let us finally examine some examples. Again, we look at the integrand of the action for $N = 1$. First, we consider the simplest case, where $\kappa = K = 0$. In Figure 3.5, we compare the trapezoidal, the Clenshaw-Curtis and the Gauss-Kronrod rule for this simple example. The plot shows, how the domain $[0, \pi]$ has been sampled by each rule. On the y axis, we see the number of the evaluations in chronological order as carried out by the corresponding rule. The trapezoidal rule, marked by red circles, started with a first function evaluation at $r = 0$ and – as we expected – samples the region by equidistant points up to $r = \pi$. In this case, Mathematica uses 9 points to begin with. After those 9 evaluations, the error estimate was apparently already small enough and the trapezoidal rule returned its approximation without any further adaptive refinements.

The Clenshaw-Curtis rule in contrast, evaluated the function a total of 163 times. That is a factor of almost 20 compared to the trapezoidal rule. We also see some

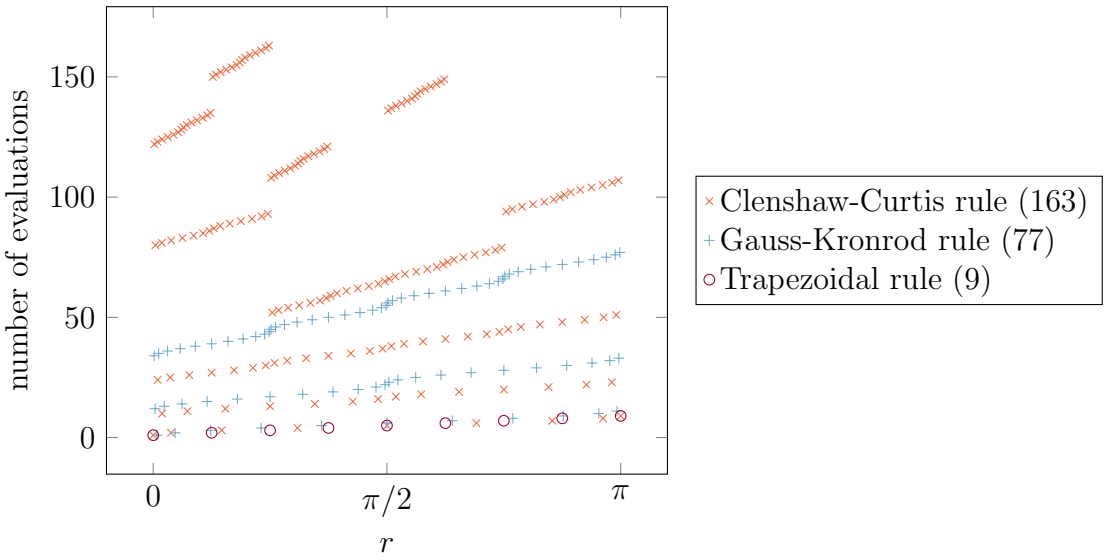

Figure 3.5: We compare the number of function evaluations for three different quadrature rules with $\kappa = K = 0$. We have used default settings for the number of starting points, the number of adaptive recursions and the same accuracy and precision goal. Symbolic preprocessing is switched off.

adaptive refinement. While it started out with 9 evaluations (not equidistant) from $r = 0$ to $r = \pi$ too, it probed the whole domain another two times with higher point density and finally increased the resolution even further for certain regions. The Gauss-Kronrod rule behaves similar. The points of the Gauss-Kronrod rule are closer near the end points as well, but do not include them. We see that it divides the whole domain into two parts after the first refinement step and then bisects once more creating a total of four subdomains. Comparing Figure 3.5 to the actual integrand for $\kappa = K = 0$ in the lower left plot of Figure 3.3, we see where the Clenshaw-Curtis and Gauss-Kronrod rule needed some refinement to correctly capture the integrand.

The results of all three methods agree up to the specified goal, i. e. within machine precision. We have worked with floating point numbers following the IEEE 754 standard (equivalent to the `double` data type in C) throughout the whole thesis (except for some testing). This suffices for our purposes.

Hence, the trapezoidal rule is the clear winner in this example. Why would we use something fancy, if the simple trapezoidal rule is better after all? Keep in mind that the integrand was very well behaved, $\mathcal{L}|_{\kappa=K=0} = 16\cos(r/2)^4/\pi^8$.

Figure 3.6 shows what happens, when we repeat the experiment for $K = 1$. We split the results in two plots for better visibility. Again, from the lower left plot in Figure 3.3, we realize that the integrand is not smooth anymore. Additionally, its gradients have become larger. To achieve the same accuracy, the trapezoidal rule now needs almost 50 times as many evaluations as the Gauss-Kronrod rule. In fact, the Gauss-Kronrod rule needs even less evaluations than before. Mathematica seems to "understand" quickly that the integrand vanishes on a large portion of the domain. If the lower and higher

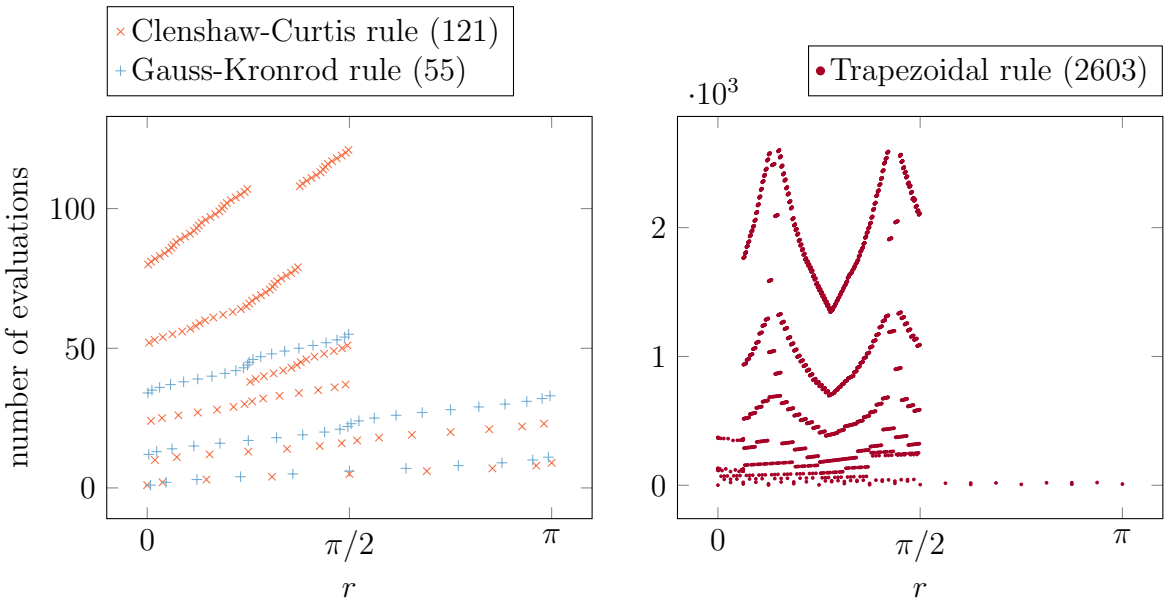

Figure 3.6: We repeat the experiment from Figure 3.5 with $K = 1$.

order approximation, which are used to determine the error estimate, coincide right away (for example because the integrand evaluates to zero everywhere), the termination conditions are immediately fulfilled (at least for the subdomain in question). Therefore, it makes sense to force Mathematica to perform a minimum number of recursive bisections before it stops, regardless of the error estimates.

As one can imagine from the shape of the integrands in the second row of Figure 3.3, when we also take $\kappa > 0$, the trapezoidal rule stands no chance against the Gauss-Kronrod rule. While these observations are only examples or heuristics and do not generally prove anything, they are extremely valuable in practice. Rigorous convergence analysis of integration strategies and rules for generic integrands is ongoing research, see [25] for a comparison of the Gauss-Kronrod and the Clenshaw-Curtis rule. At this point, one should be able to appreciate our decision to choose Mathematica over C.

By using very conservative initial conditions, such as a high number of sample points for the first scan of the domain and enforcing a minimum number of recursive refinements, we are confident to obtain accurate results. The methods generalize to two-dimensional integrals by a cartesian product of the one-dimensional procedure. For high-dimensional integrals one would use entirely different techniques such as stochastic methods, e. g. (pseudo) Monte Carlo, or sparse grid methods. Since we only need one- and two-dimensional integrals, the cartesian product rule is still the strategy of choice.

3.4 Rescaling

3.4.1 Why do we need that?

After this excursion, let us come back to the Dirac sphere for $N = 1$. We have discovered the necessity of the boundedness constraint and took it into account by adding the term $\kappa \operatorname{Tr}(A)^2$ to the Lagrangian, where we assume κ to be an arbitrary real positive parameter.

For relatively large κ, we find the interesting feature that the minimum "jumps" at some point. For example, consider the line for $\kappa = 0.01$ in Figure 3.4. The minimum is clearly at $K = 0$ and it will apparently stay there for $\kappa > 0.01$. In contrast, for $\kappa = 0.005$ it is roughly at $K = 2.3$. The lines in between indicate that this is not a continuous transition. Indeed, we have found the jump to occur at $\kappa \approx 0.009$ from $K \approx 1.9$ down to $K = 0$. However, since we are eventually interested in the limit $\kappa \to 0$, this behavior for large κ is nothing more than a curious feature and of no particular interest. In the limit $\kappa \to 0$, the position of the minimum depends continuously on κ.

Let us explore this continuous behavior in more detail. In the one-dimensional case, we can still employ the simplest imaginable method of minimizing the action for fixed κ. We evaluate the action on a dense grid for K. We choose 501 equidistant grid points in the interval $[0, 50]$ including the endpoints. As a first approximation, we can simply pick the grid point with the smallest action as our minimum. With a grid spacing of Δt, this procedure is accurate up to $\Delta t/2 = 0.05$ in our example. To add some precision, we additionally interpolate the action by cubic B-splines and minimize the interpolating function. Given that the position of the minimum depends reasonably smoothly on κ, which it does in our case for small κ, this method is already highly accurate. The minimization procedure for the interpolating functions will be discussed in section 3.5.

We then also sample κ in the interval $[10^{-7}, 10^{-3}]$. We have already observed that the position of the minimum depends continuously on κ in this range. For κ we again choose 501 grid points and space them logarithmically in the specified interval. The results are shown in Figure 3.7.

This analysis verifies that by simply taking $\kappa \to 0$, we get back the same results as without the boundedness constraint, see Figure 3.4, which was already clear from the analytic findings in section 3.2. The action approaches zero, as $K \to \infty$, because the eigenvalues (3.1) form complex conjugate pairs on the whole domain $r \in [0, \pi]$. For $\kappa = 0$ we can easily find a minimizing sequence, for example $(K_n)_{n \in \mathbb{N}} = n$, see Figure 3.2 and Figure 3.7. In addition, the absolute value of the eigenvalues approaches infinity as $K \to \infty$, which can be read off equation (3.1).

Therefore, we want to modify the limit by what we call the *rescaling procedure*. Because we will also apply the procedure later on for $N = 2$, let us go back to the abstract formulation of causal fermion systems and briefly outline the general idea.

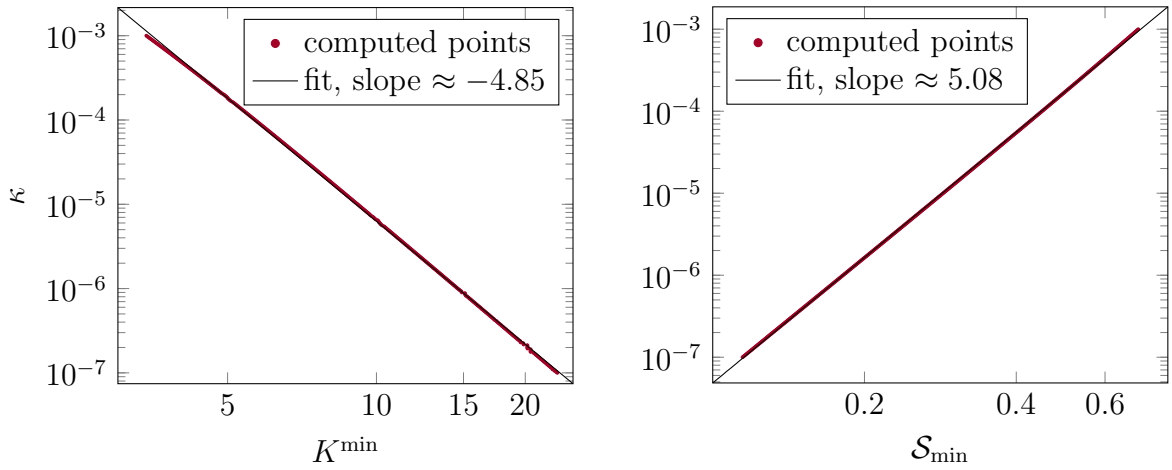

Figure 3.7: We show the position $K^{\min}$ of the minimum of the action (left) and the minimal value of the action (right) as $\kappa \to 0$. There are 501 logarithmically spaced points for κ between 10^{-7} and 10^{-3}. For each one, we obtain $K^{\min}$ by minimizing a cubic B-spline interpolating function of the action based on 501 linearly spaced values for K from 0 to 30. The data is well approximated by power laws $K^{\min} \propto \kappa^{-0.206}$ and $\mathcal{S}_{\min} \propto \kappa^{0.197}$.

3.4.2 The rescaling procedure

Let $(\mathcal{H}, \mathcal{F}, \rho)$ be a causal fermion system and $M := \operatorname{supp}(\rho)$ the corresponding space-time. We will assume, that all $x \in M$ are *isospectral*, i. e. $\sigma(x) = \sigma(x')$ for all $x, x' \in M$, we write $x \simeq x'$. Note that the Dirac sphere satisfies this assumption by construction. If we choose as a norm on M

$$\|\cdot\| \,:\, M \to \mathbb{R}\,, \quad x \mapsto \|x\| := \max_{\lambda \in \sigma(x)} |\lambda|\,, \tag{3.9}$$

apparently all $x \in M$ have the same norm and we define $c := \|x\|$. Furthermore, because $P(x,x) = \pi_x x = x|_{S_x}$, we find $P(x,x) \simeq x$. In particular, for the Dirac sphere $\|P(\mathfrak{n},\mathfrak{n})\| = c$.

The idea of the rescaling procedure is to rescale the causal fermion system such that $\|x\| = 1$. In the general setting, this amounts to the map

$$\Phi \,:\, \mathcal{F} \to \mathcal{F}\,, \quad y \mapsto \frac{y}{\|y\|}\,. \tag{3.10}$$

We have illustrated the action of the map Φ in Figure 3.8. By virtue of the map Φ, we define a new rescaled measure by the push forward measure $\rho^{\mathrm{r}} := \Phi_* \rho$ and the rescaled space-time by $M^{\mathrm{r}} := \operatorname{supp}(\rho^{\mathrm{r}}) = \{x/c \,|\, x \in M\}$. The kernel of the fermionic operator is also rescaled by $1/c$ and one easily verifies the following formulas for the emerging rescaled quantities

$$\int \operatorname{tr}(x)\, d\rho^{\mathrm{r}}(x) = \frac{1}{c} \int \operatorname{tr}(x)\, d\rho(x)\,, \qquad A^{\mathrm{r}}_{xy} = \frac{A_{xy}}{c^2}\,, \qquad \mathcal{L}^{\mathrm{r}} = \frac{\mathcal{L}}{c^4}\,, \qquad \mathcal{S}^{\mathrm{r}} = \frac{\mathcal{S}}{c^4}\,. \tag{3.11}$$

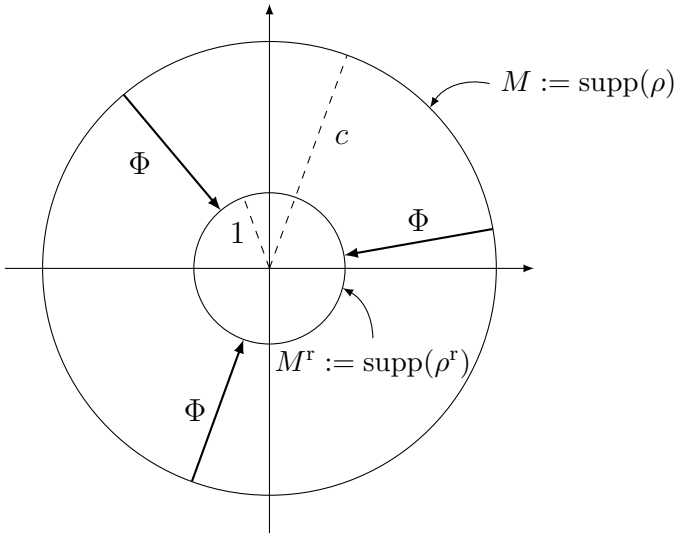

Figure 3.8: We show an illustration of the rescaling procedure. It basically shrinks the space-time to "radius" one. Of course, we can not give reasonable names to the axes in this picture.

Let us now assume that ρ_n is a minimizing sequence for the action. Then ρ_n^{r} converges as a measure to some ρ^{limit} and, as usual, we define $M^{\mathrm{limit}} := \mathrm{supp}(\rho^{\mathrm{limit}})$. At the same time, by the rescaling procedure, we ensure that $\|x\| = 1$ stays constant for all $x \in M$.

What does this mean for the Dirac sphere with $N = 1$? Again, assume we have a minimizing sequence of the action for $\kappa = 0$. We have argued that for such a sequence $c_n \to \infty$ and no minimum is obtained. However, for all $\kappa > 0$, we do find a minimum of the action. We can now keep track of the minimizers while taking $\kappa \to 0$ and perform the rescaling procedure along the way. Thereby, $\|x\|$ stays constant when taking the limit.

In practice, the rescaling procedure is performed in the following steps:

1. Take a finite decreasing sequence $\kappa_1 > \kappa_2 > \cdots > \kappa_M > 0$ for some $M \in \mathbb{N}$.

2. For each κ_i, find the minimizer of the causal action principle. This will result in a set of fixed parameter values for the $3N - 2$ free parameters. For example, for $N = 1$ it yields $K^{\min}$, which we have already seen in Figure 3.7. For $N = 2$, it yields a set of values $\rho_1^{\min}, K_1^{\min}, K_2^{\min}$.

3. For each κ_i, use the corresponding minimizing parameter values to compute $c_i := \|P(\mathfrak{n}, \mathfrak{n})\|$ as in (3.9).

4. For each κ_i, rescale the minimum of the corresponding action by $\mathcal{S}_{\min}^{\mathrm{r}} := \mathcal{S}_{\min}/c_i^4$.

In Figure 3.9, we show the result of this rescaling procedure for $N = 1$ and again 501 logarithmically spaced grid points for κ from 10^{-7} to 10^{-3}. The action decreases almost linearly as $\kappa \to 0$. The limit is totally space-like in the sense that $\mathcal{L}(x, y) = 0$ for all $x, y \in M^{\mathrm{limit}}$.

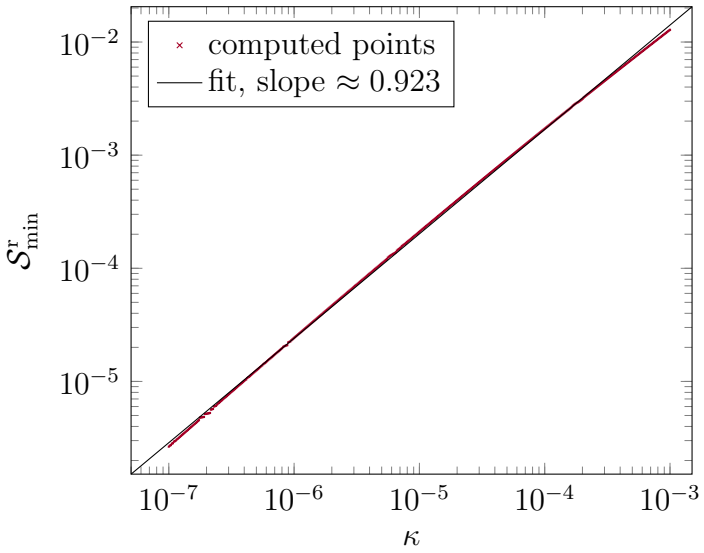

Figure 3.9: We plot the rescaled minimal action as a function of κ. Instead of a moderate power of roughly 0.2 as in Figure 3.7, the rescaled action scales almost linearly $\mathcal{S}^{\mathrm{r}}_{\mathrm{min}} \propto \kappa^{0.923}$.

Before we go on to the Dirac sphere with two shells, let us again review some numerical concepts.

3.5 Numerical methods II: Finding a minimum

Numerical optimization is a huge field on its own and there are plenty classification criteria for numerous methods. We will only briefly touch upon two criteria to restrict the pool of possibilities for our application. We first take a look at our parameter space. For $N = 1$ we had $K \in [0, \infty)$ (recall that only K^2 entered the equations). For $N = 2$ we found $\rho_1 \in \mathbb{R}^+$, $K_1, K_2 \in \mathbb{R}$. For the same reason as above, we can restrict ourselves to non-negative values for K_1, K_2. The trace constraint for $N = 2$ reads $2\rho_1 + 6\rho_2 = \text{const}$. We choose $\rho_1 + 3\rho_2 = 1$, hence the parameter ρ_1 is restricted to $(0, 1)$. The constraints of our optimization problem are encoded in the boundaries of the parameter space.

This suggests that we are dealing with a constrained optimization problem. However, once we have obtained a reasonable understanding of the Lagrangian, we do not explicitly take these constraints into account. As we discuss shortly, this allows us to use faster methods for unconstrained optimization.

Second, let us compare global and local methods. At a first glance, one might argue that we certainly want to find the global minimum. Why would one even bother for local methods? It is not that simple. Intuitively, it is clear that finding a global optimum is much harder than finding a local one. In fact, the difference can hardly be overestimated. As a consequence, global optimization methods are generically more

expensive than local ones. Given the computational complexity of the action, it is well worth the effort to invest some time in getting local methods to work.

So far, we have narrowed down our possibilities to unconstrained local optimization. We can further categorize into gradient based methods and direct search methods. Prior generally use first and second derivatives (or approximations thereof) to gradually move to lower/higher values from one or multiple start points. Direct search methods do not need derivatives. Famous examples are the Nelder-Mead algorithm, stochastic methods such as simulated annealing, or genetic algorithms and differential evolution.

For the minimization in the $N = 1$ case, we employ a line search using a Quasi-Newton method. Again, it is beyond the scope of this thesis to introduce basic techniques such as (Quasi-)Newton methods, gradient descent, conjugate gradient descent or the Levenberg-Marquardt algorithm. The following not only holds true for one-dimensional optimization problems, but generalizes also to higher-dimensional settings.

The specific Quasi-Newton method used here is the Broyden-Fletcher-Goldfarb-Shanno algorithm (BFGS), see for example [2] and references therein for details. Usually, the search direction p_n in the n-th step is determined by solving $B_n\, p_n = \nabla \mathcal{S}(x_n)$, where B_n is the Hessian of $\mathcal{S}$ and x_n is the current position in the parameter space. The gradient ∇ is with respect to x_n, i. e. with respect to the free parameters. For $N = 1$, this is just K, for $N = 2$, it contains the parameters ρ_1, K_1, K_2. Basically, the BFGS algorithm determines the search direction not by computing the exact Hessian at each step, but using an iteratively updated approximation. Moreover, Mathematica does not even compute the Hessian B_n itself, but only the Cholesky factors. Thereby the above equation for the search direction p_n can be solved in $\mathcal{O}(k^2)$ time, if k is the number of parameters, see [3].

While Quasi-Newton methods can not hold up with the quadratic convergence of Newton's method, with an adequate line search strategy it still converges superlinearly [23]. The savings from not having to compute the Hessian (or a finite difference approximation thereof) from scratch in each step and the robustness against the scale problem outweigh the slightly inferior convergence rate in most applications.

Since we do not have an analytic expression for the gradient, it is approximated by second order central finite differences. This becomes expensive as the dimension of the parameter space increases. Hence we now have the ingredients to find the search direction p_n at the point x_n in the n-th step. Keep in mind that the direction is not exact, but only an estimate. For an update to the next point in parameter space $x_n \to x_{n+1}$, we still need the step size.

The idea of a line search is to determine an appropriate step size α by minimizing $\epsilon(\alpha) := \mathcal{S}(x_n + \alpha p_n)$. In this equation, we only minimize with respect to α, i. e. it is a simple one-dimensional optimization problem independent of the dimension of the original parameter space.

Note that $\epsilon'(\alpha) = (\nabla \mathcal{S}(x_n + \alpha p_n))^T p_n$, i. e. it amounts to a one-dimensional gradient based optimization problem. We are not interested in solving this problem exactly, but only need an approximation that lies close to the real minimum. For one, recall that the search direction itself is not exact and furthermore, this step only serves to find a reasonable step size for the original minimization.

For the Quasi-Newton method, as suggested in [23], we employ the Wolfe conditions with a decrease factor of 10^{-4} and a curvature factor of 0.9 to decide, whether the approximation is good enough. For the minimization of $\epsilon(x_n + \alpha p_n)$, we follow a line search strategy due to More and Thuente described in [22]. It uses quadratic and cubic interpolations in seeking to fulfill both Wolfe conditions.

Why would we accept the overhead of an additional one-dimensional optimization in each step only to determine the step size? The biggest advantage is that thereby we do not rely on the assumption that the model is close to the real behavior. In Newton's method (and most other gradient based local methods), one works with a quadratic model, i. e. $\mathcal{S}(x_n) + (\nabla\mathcal{S}(x_n))^T p_n + \frac{1}{2} p_n^T B_n p_n$. Hence the model captures the true function only up to second order so to speak. The resulting step size is only as good as this representation. However, it is not at all guaranteed in general that a normal step in Newton's method indeed brings us closer to the optimum. Since we only seek a crude estimate for the minimum of a rather simple one-dimensional optimization problem, the line search strategy does eventually pay off.

For one shell, i. e. $N = 1$, it does not really matter, which minimization procedure we employ. We have tested various techniques ranging from a simple Newton method up to a line search with different Quasi-Newton methods. All of them finish within seconds for 501 different values of κ and gave the same results. Note that we already had a very good idea of the positions of the minima from sampling an appropriate interval as in Figure 3.7. For all local methods we tested, it suffices to choose any value $K > 2$ as a start point, which we read off Figure 3.4. Providing a good start point is crucial. For a three-dimensional parameter space, visualizing the action for different κ is not that simple anymore and we expect the minimization to be orders of magnitude more expensive. Therefore, we went through great lengths to get a feeling for where the minima might lie, which is the subject of the following section.

3.6 Two shells ($N = 2$)

We have already developed all necessary concepts based on the example $N = 1$ in the previous sections. Thus in this section, we will merely repeat the analysis for $N = 2$ and present the results. Additionally, we will emphasize some problems arising only in the $N = 2$ case.

Computing the Lagrangian should be straight forward. We simply follow the recipe given in section 2.3. The expressions become a little bit lengthy, but are still easily handled symbolically by Mathematica. After tweaking the integrand, by letting Mathematica simplify the expression first and then adding some further simplifications by hand, we end up with an expression that takes up around 10 to 20 lines. It is symmetric around 0 for both K_1 and K_2, hence we only consider $K_1, K_2 > 0$. Moreover, as mentioned before, it is periodic in t. We have chosen $\omega_2 = 1$, hence the period is 2π.

From the ansatz of the kernel of the fermionic operator, we infer that it only involves the four basic mathematical operations as well as sin, cos, max and the square root. This is important, because it allows us to use Mathematica's `Compile` function. As a

symbolic computer algebra system, Mathematica always deals with all kind of input. Basically every expression is allowed as input to any function. A lot of checks and interpretation has to be performed for this to work. By "compiling" a function, we specify that the only input we will provide are always real numbers following the IEEE 745 norm. Mathematica can then transform the function implementation from the usual Wolfram language to perfectly valid C code. (To find a C implementation the expression must not contain complex functions or operations not present in the C standard libraries.) A C-compiler produces code that is very close to actual machine code and links the resulting object file back to Mathematica. This can give an enormous performance boost for applications like ours, which rely on a whole lot of basic number crunching.

Furthermore, we demand the compilation to be optimized for speed (Mathematica looks for recurring subexpressions and only computes them once and uses available compiler optimizations), to suppress all symbolic evaluation and to produce thread safe code for parallel execution.

As a second step, we need to get some intuition about the integrand. If it behaves smoothly, we might only need a few function evaluations for a good approximation of the integral in the action. If it shows kinks, large gradients high frequency oscillations or strongly localized features, we might want to be very conservative during the integration and choose many points and/or adaptive refinements, see section 3.3.

The analytic expression of the integrand reveals that there are no high frequency oscillations in neither the t nor the r direction. However, we can not exclude the other potential issues. Quite the contrary, from the $N = 1$ case we even anticipate them to occur. To this end we wrote an interactive tool that lets us change all three free parameters and κ simultaneously and updates a plot of the corresponding integrand in real time.[1]

We further augmented this tool to perform the integration in the background, plot the evaluation points that were used and return the value of the action as well as the number of evaluation points. We show the output of the tool for two exemplary points in parameter space in Figure 3.10. Even for the complex structure shown there, the integration took less than a second.[2] The number of function evaluations varies greatly from less than 2000 up to about 150 000. At this point, an adaptive integration strategy is indispensable.

Let us point out a few observations we made, while playing around with the tool, trying to get a feeling for how the action depends on the parameters. As expected, for all combinations of $\kappa > 0$ and $\rho_1 \in [0, 1]$ did we find an increasing action for both $K_1 \to \infty$ and $K_2 \to \infty$. Hence, sufficiently increasing either one of the two (while keeping the other fixed at an arbitrary value), eventually results in a monotonic increase of the action. Due to the form of the κ term, this does not come as a surprise.

[1] All code is available upon request from niki.kilbertus@physik.uni-regensburg.de.
[2] These computations were performed on an i7-4470HQ at 2.2 GHz with 16 GB RAM.

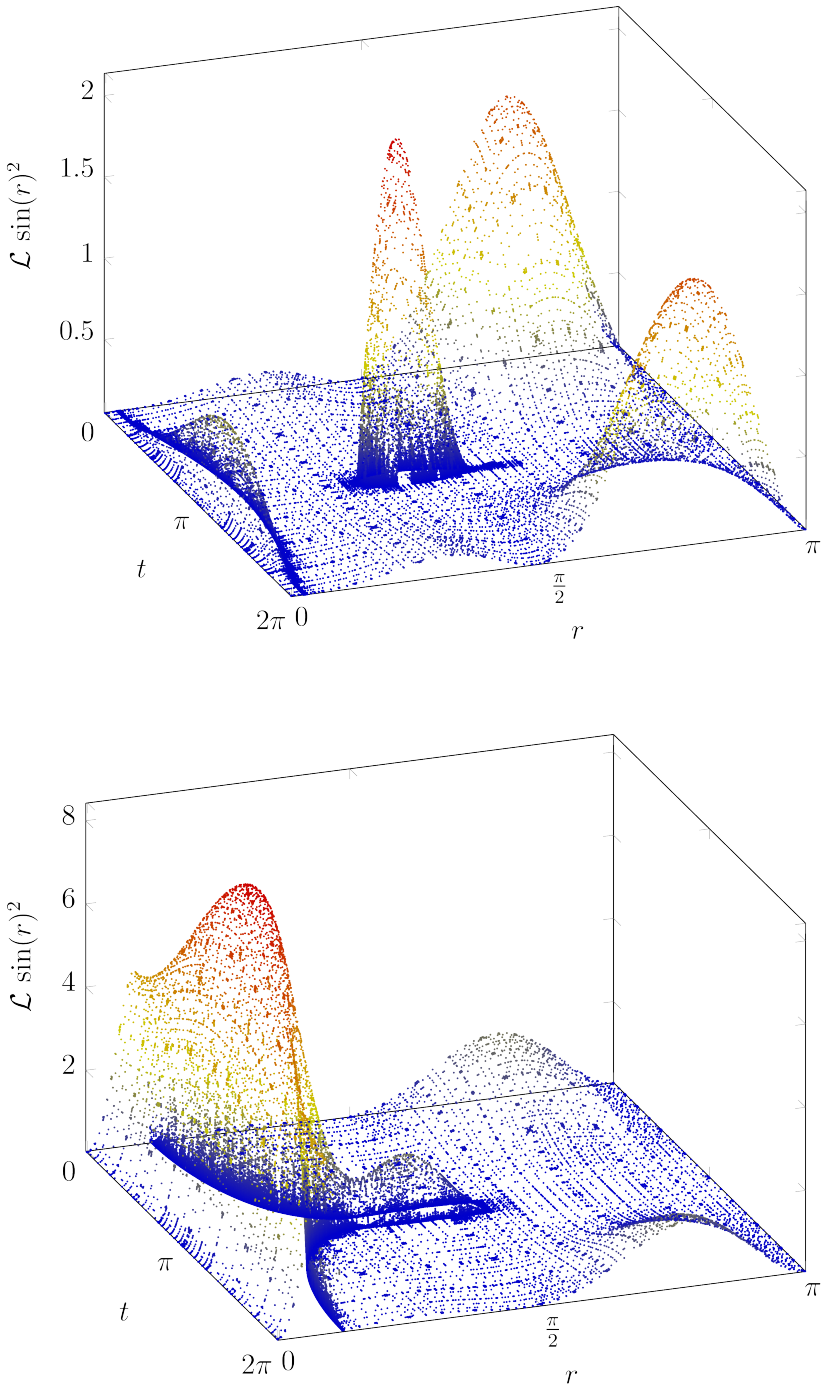

Figure 3.10: We show the evaluation points of the integrand in a scatter plot for the parameter values $\kappa = 10^{-4}$, $\rho_1 = 0.3$, $K_1 = 10$, $K_2 = 2$ (top), and $\kappa = 10^{-4}$, $\rho_1 = 0.8$, $K_1 = 4$, $K_2 = 6.5$ (bottom). The integration routine performed 70 992 and 110 942 function evaluations respectively until it reached the specified precision goal. Those are examples with a particularly rich structure.

The observations led us to believe that once the action has increased beyond a certain threshold for large K_1, K_2, it will keep on growing and not return to smaller values again for even larger Ks. This is backed by the analytic form of $\mathrm{Tr}(A)^2$. Hence, for fixed κ and ρ_1, we should be able to find a minimum by starting sufficiently far from the origin in the K_1-K_2-plane.

The main task is to specify good start points for the minimization. In the beginning, we followed the same approach as for the one shell case. We took out a three-dimensional cube of the parameter space, e. g.

$$(\rho_1, K_1, K_2) \in [0.01, 0.99] \times [0, 10] \times [0, 10] \,, \tag{3.12}$$

discretized it by a grid (equidistant or logarithmically spaced in each direction) and looked for the minimal value as a good candidate for a start point. Then we repeated this for several values of κ, say in the interval $[10^{-7}, 10^{-3}]$. This was feasible for $N = 1$, because we only had one free parameter and the integration was very quick. Computing a few hundred points for the single parameter K and repeating that for again a few hundred different values of κ was a matter of minutes, even including interpolation and minimization of the interpolant for each κ.

For the three-dimensional parameter space, the largest grid we used had $27 \times 51 \times 51$ points and we computed it for 31 values of κ. This is a total of about 2 million evaluations of the action. (Keep in mind that each evaluation of the action costs about 2000 to 150 000 evaluations of the integrand.) On a machine with 16 CPUs of the Xeon E5-2650 model at 2.5 GHz with 66 GB RAM this took about a day and a half. Unfortunately, we waited for so long only to find that the minima were almost always located at the boundary of our computational domain. This indicates that we did not choose the domain large enough in the K_1 and K_2 direction. Note that the ρ_1 domain is constrained anyway and is therefore under control. Moderate extensions still led to the same results. Since the resolution is limited in terms of computing time, we need to find a different method.

In order not to hit the boundaries, we want to adjust the K_1, K_2 domain depending on the values of κ and ρ_1. Due to the trace constraint $\rho_1 + 3\rho_2 = 1$, the parameter ρ_1 determines the relative contributions of the two shells. In the limiting cases $\rho_1 = 1$ and $\rho_1 = 0$, we have $\rho_2 = 0$ and $\rho_2 = 1/3$ (the maximum value for ρ_2) respectively. In the first case, we get back the $N = 1$ model. In the second case, we find a very similar model, because then again only one shell is occupied. This is seen from the general ansatz for the kernel of the fermionic operator in (2.58). In Figure 3.11, we show the action along the K_1 and K_2 axis respectively for various values of ρ_1. We are interested in the position of the minima. Therefore, we normalize all curves by $\mathcal{S}_{K_i=0}$ for $i \in \{1, 2\}$ respectively.

While we only show those plots for $\kappa = 10^{-4}$ the qualitative behavior is representative for the whole range of $\kappa > 0$ below about 10^{-2}. Note that in the two plots of Figure 3.11, the colors correspond to the same values of ρ_1. In the upper plot, the minima of the action move away from zero as ρ_1 increases. On the contrary, in the lower plot, the minima approach zero as ρ_1 increases. While the absolute position of the minima changes slightly, this behavior is universal for all relevant κ.

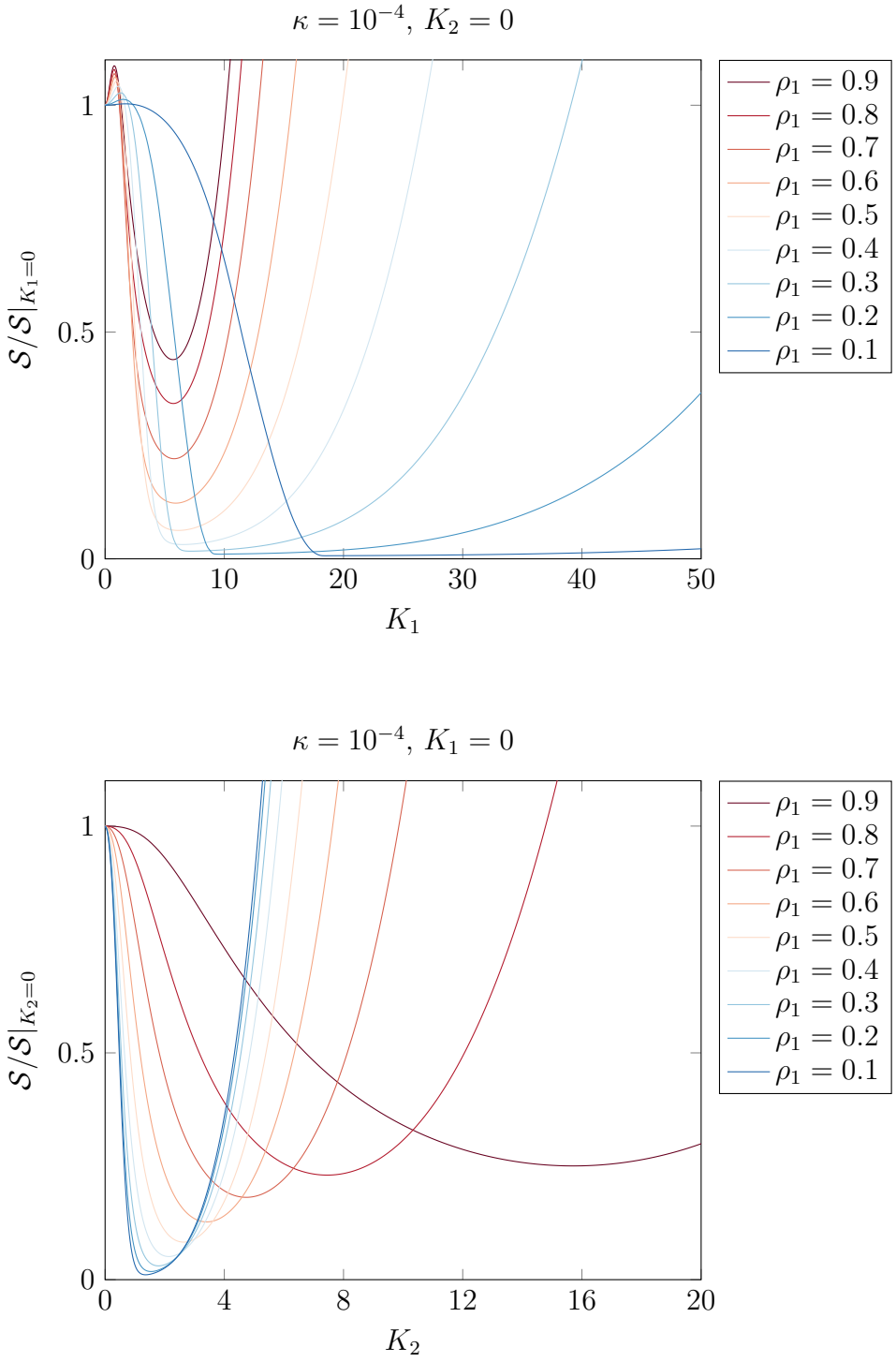

Figure 3.11: We plot (a suitably normalized version) of the action for $\kappa = 10^{-4}$ and several values of ρ_1. In the upper plot, we observe the behavior along the K_1-axis, i. e. $K_2 = 0$. In the lower plot, we observe the behavior along the K_2-axis, i. e. $K_1 = 0$.

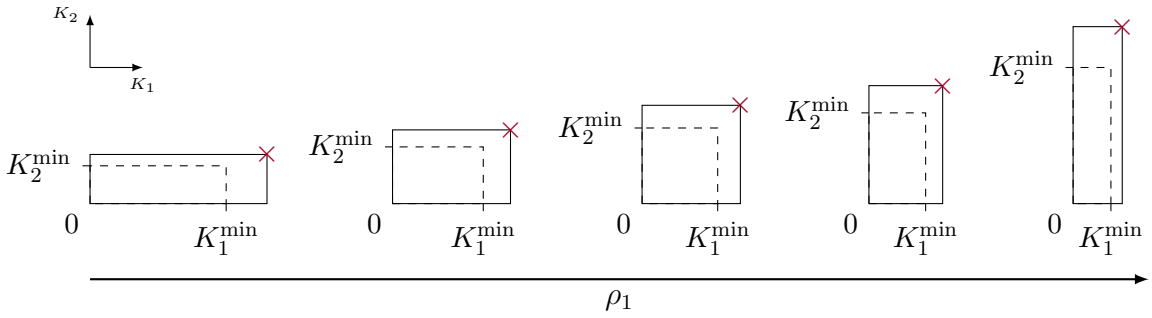

Figure 3.12: For each ρ_1, $K_1^{\min}$ and $K_2^{\min}$ set the scale for the choice of the computational domain (shown as dashed rectangles). Extending those bounds by a reasonable scale factor (for the picture we have used $\xi = 1.3$), we expect the minimum of the action to lie within the solid rectangles. Hence the red $\times$ symbols indicate our candidates for start points of the minimization procedure. As we shift the weight from one shell to the other, by increasing ρ_1 from 0 to 1, the height and width of the domain scale inversely to each other.

Let us denote the position of the minima as a function of ρ_1 by $K_1^{\min}(\rho_1)$ (where $K_2 = 0$) and by $K_2^{\min}(\rho_1)$ (where $K_1 = 0$), which are different functions for eachκ. We will only make the dependence on κ explicit in the notation, when necessary. The above observation amounts to the statement that $K_1^{\min}$ is monotonically decreasing and $K_2^{\min}$ is monotonically increasing. This can be understood in light of the fact that small values of ρ_1 put more weight on the second shell, whereas large values of ρ_1 (close to 1) enforce the effect of the first shell.

Our basic assumption in choosing plausible start points for the local gradient based optimization is that $K_1^{\min}$ and $K_2^{\min}$ set the scale for the thresholds from which on the action will only increase. We choose a factor $\xi \gtrsim 1$ and postulate that for a given ρ_1 (and a given κ of course), the minimum of the action lies within $(K_1, K_2) \in [0, \xi\, K_1^{\min}(\rho_1)] \times [0, \xi\, K_2^{\min}(\rho_1)]$. In Figure 3.12, we illustrate how the computational domain is now altered depending on ρ_1. One immediately recognizes, why our first approach with a fixed computational domain for all combinations of parameters and all κ could not work. For most cases it was rather generous for one direction but way too narrow for the other one. For example, for $\kappa = 10^{-3}$ and $\rho_1 = 0.01$ we have found $K_1^{\min} \approx 200\, K_2^{\min}$.

Now we have a strategy to choose a starting point in the K_1-K_2-plane, once ρ_1 is given. Note that so far we have always assumed that ρ_1 is fixed and we are only minimizing with respect to K_1 or K_2. While this does not come near the actual problem, it gives us a neat heuristic on how to tackle it. Because we still have no idea how to choose a reasonable start value for ρ_1, we will simply scan it again, where we take a smaller spacing near the endpoints 0 and 1.

We again discretize the new adaptive domain

$$(\rho_1, K_1, K_2) \in (0,1) \times [0, \xi K_1^{\min}(\rho_1)] \times [0, \xi K_2^{\min}(\rho_1)] \tag{3.13}$$

by a lattice with a fixed number of grid points. Indeed, for all κ, the minimal value of those evaluations came to lie inside the computational domain. This encouraged us to proceed with the suggested approach. Unfortunately, an interpolation between the established grid points and subsequent minimization of the interpolating function, which can be evaluated much faster than the true action, turned out to not work at all. As soon as we went beyond linear interpolation (which yields the same results as just picking the minimal value on the grid), the interpolating function took on negative values. Because the action is strictly positive, the interpolant clearly does not represent the true behavior of the action properly. To get a trustworthy interpolant, we would need more grid points than we could compute in reasonable time, in particular also more than the minimization procedure.

A canonical choice for the start point is the minimal value on the grid. However, whenever we want to test a new κ, we would then have to sample the whole parameter space again, which is incredibly expensive for reasonably fine grids. Moreover, it yields only one start point for each κ, which we feel slightly uncomfortable with, because it bears a great risk of getting trapped in a local minimum. Starting at multiple points in parameter space for each κ adds some valuable redundancy that mitigates this risk. Hence, our new overall strategy is given by the following multistart method:

1. Take a finite decreasing sequence $\kappa_1 > \kappa_2 > \cdots > \kappa_M > 0$ for some $M \in \mathbb{N}$, e. g. $\kappa_i = 10^{-i}$ for $i \in 2, 4, \ldots, 6$.

2. Scan the interval $(0,1)$ by finitely many values ρ_1^j for $j \in \{1, \ldots, L\}$.

3. For each κ_i and each ρ_1^j, find $K_1^{i,j} := K_1^{\min}(\kappa_i, \rho_1^j)$ and $K_2^{i,j} := K_2^{\min}(\kappa_i, \rho_1^j)$. (Here, we indicate the dependence of the $K^{\min}$ on κ explicitly.)

4. For each κ_i and each ρ_1^j, minimize the action as outlined in section 3.5 with the start point $(\rho_1^j, \xi K_1^{i,j}, \xi K_2^{i,j})$.

5. For each κ_i, find among the L minima from the previous step the smallest one and save it together with the position in parameter space.

After we have obtained the minima for a decreasing sequence κ_i, we can proceed with the rescaling procedure as in section 3.4.

Performing $2 \cdot M \cdot L$ one-dimensional optimizations for each new run only to get hold of $K_1^{\min}, K_2^{\min}$ for start points in a reasonable range still seems like a large overhead. Therefore, we precompute a model for the start points in the K_1-K_2-plane depending on ρ_1 and κ. We compute the minima once for $\kappa_i = 10^{-i}$ with $i \in \{2, \ldots, 6\}$ and 27 values for ρ spread over the interval $(0,1)$ with more points near the boundaries. For each κ_i, we found the position of the minima to be reasonably well approximated by $K_1^{\min}(\rho_1) = a_1 + a_2 \cdot \rho_1^{a_3}$ and $K_2^{\min}(\rho_1) = b_1 + b_2 \cdot (b_3 - \rho_1)^{b_4}$ with appropriate values for the a_i and b_i.

The least squares fits of these models to the data are shown in Figure 3.13. Note that while the approximation of $K_1^{\min}$ is off for small values of κ, especially around $\rho_1 \approx 0.2$, the fits mostly overestimate the true value. This amounts to a larger scaling factor ξ in Figure 3.12 and shifts the start point further into the region, where the action is definitely increasing with increasing K_1 and K_2. That might cause the optimization procedure to take a few more steps, but should not distort the quality of the start point in view of the found minimum. Keep in mind that we only need to estimate the scale of $K_i^{\min}$ for the choice of our start points. To this end, the quality of the fits is more than enough.

Finally, we again use a least squares fit for the parameters a_i, b_i of the fits above. Those are of similar quality. Using these models, we can estimate $K_1^{i,j}$ and $K_2^{i,j}$ for any κ_i and ρ_1^j (in a reasonable range) in practically no time, without having to optimize along the K_1 and K_2 axes over and over again. After all, the scaling factor ξ is the pivotal tuning parameter for the ultimate choice of the start points. We have used values ranging from $\xi = 0.3$ up to $\xi = 2$.

In Figure 3.14, we analyze the behavior of the action as $\kappa \to 0$ before and after the rescaling procedure. With the rescaling, we find an almost linear dependence, just as for $N = 1$, see Figure 3.9. Thus, we again find a totally spacelike solution, where $\mathcal{L}(x, y) = 0$ and $\|x\| = 1$ for all $x, y \in M$. Moreover, as κ goes to zero, we find $\rho_1^{\min} \to 0$, $K_1^{\min} \to \infty$ and $K_2^{\min} \to 0$. Hence, in the limit, it seems that only the second shell is occupied, i. e. a different diverging minimizing sequence is found here than for $N = 1$, which would amount to $\rho_1 \to 1$, $K_1 \to \infty$.

In Figure 3.15, we show the trajectory of the minima in the three-dimensional parameter space as $\kappa \to 0$ together with the projection to each of the two-dimensional planes. In addition, we plot each single free parameter as a function of κ with logarithmic axes. Even though the curves exhibit a plateau, we have no reason to believe that any of them will converge to a finite value greater than zero for $\kappa \to 0$. This is also expected since the decreasing action is not flattening out for almost four decades.

3.7 Outlook: Many shells

When we began to feel comfortable with the specification of the Dirac sphere as a numerically accessible model, we originally planned on exploring the $N \to \infty$ (and subsequently also the $T \to \infty$) limit. The $N \to \infty$ limit would correspond to the completely filled Dirac sea. Our motivation is to find out, if and how the minimum is perturbed away from the vacuum. Such a deviation amounts to a regularized version of the vacuum state and potentially encodes physics beyond the results of classical field theory.

Soon it became clear that this will remain an open task beyond the designated six months of a master's thesis. Let us try to estimate, how much harder the problem will become for larger N. It did not come as a surprise that the transition from one to three parameters complicated the situation quite a bit. For $N = 1000$, the number of

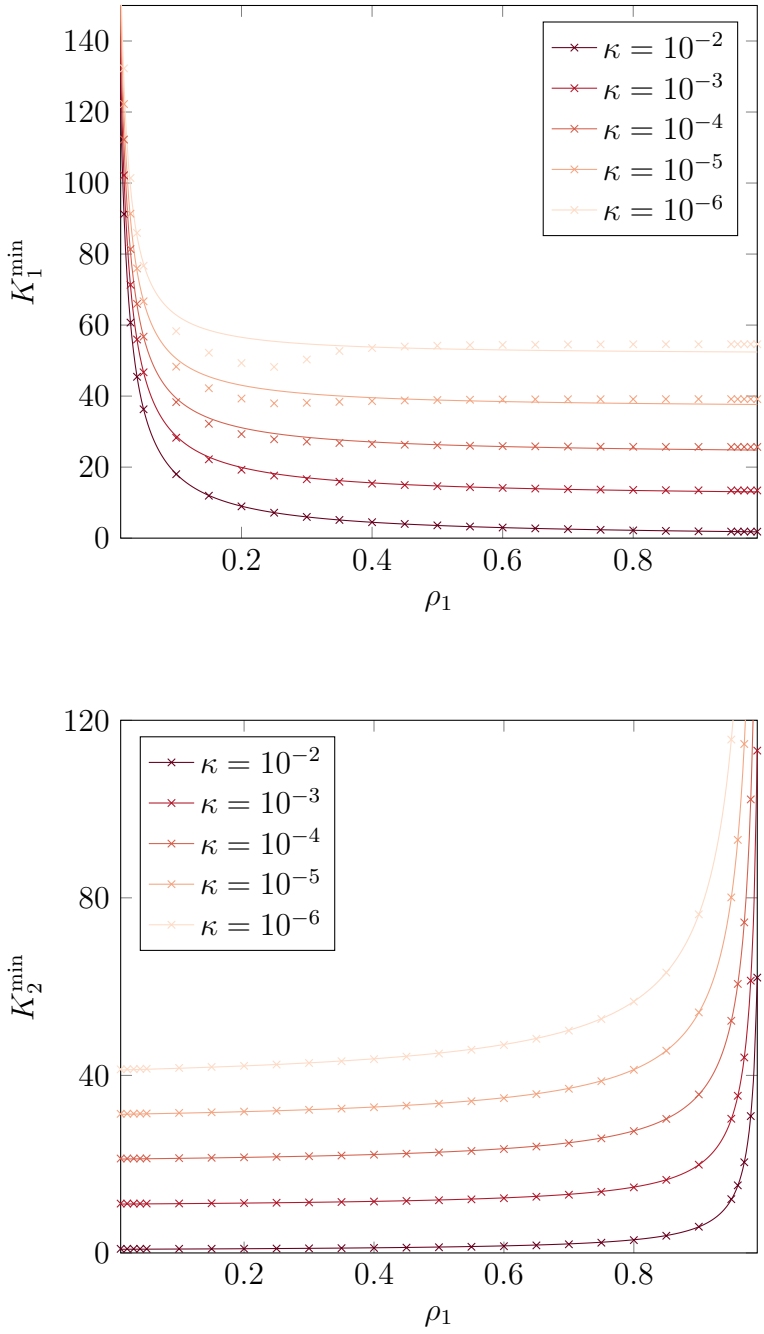

Figure 3.13: We plot $K_1^{\min}$ (top) and $K_2^{\min}$ (bottom) as a function of ρ_1 for several values of κ together with the corresponding least squares fits. In both plots, the curves for different κ have been offset vertically by 10 from each other for better visibility. The true curves lie much closer together.

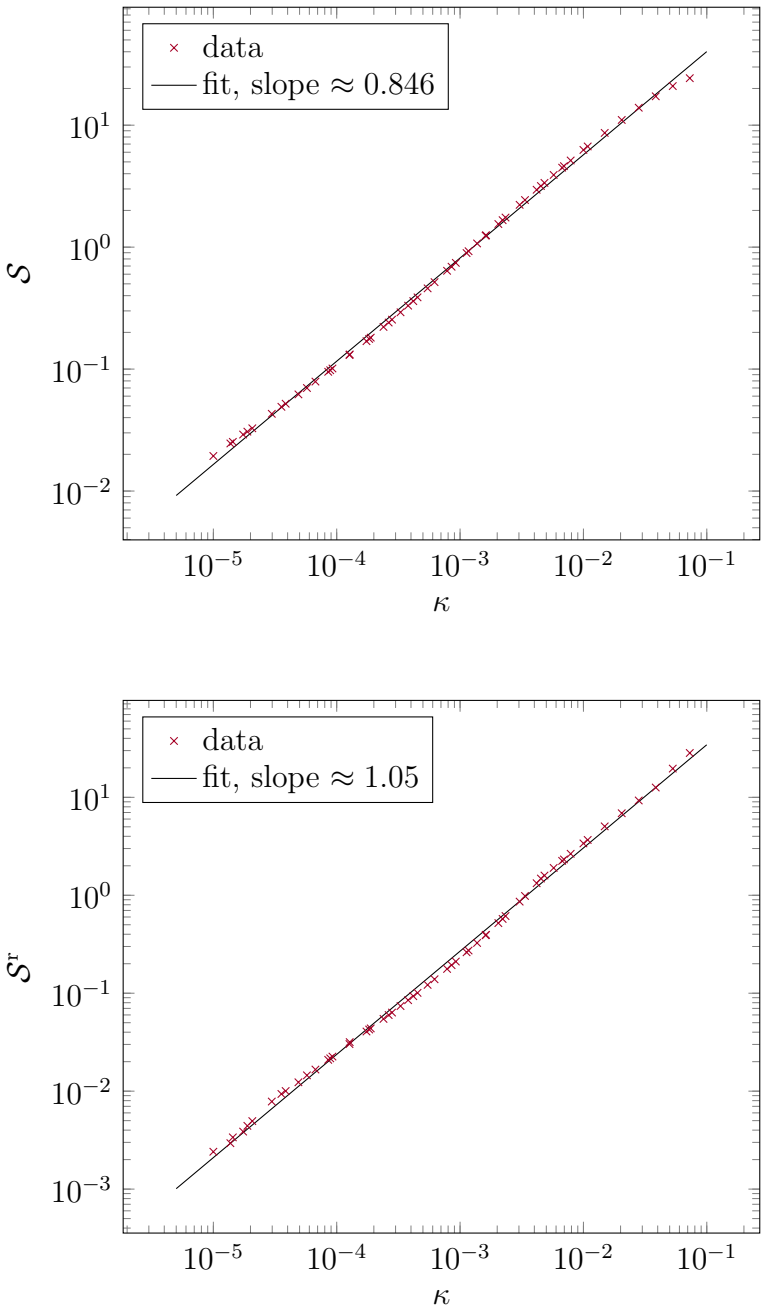

Figure 3.14: We plot the action as a function of κ over almost four decades before the rescaling procedure (top) and after (bottom). After the rescaling procedure we find an almost linear decay of the action, as $\kappa \to 0$. The points here are the combined results of three different runs with different values for ξ from 0.3 to 2 and also different sequences of ρ_1^j, i. e. completely different start points. The consistent overall alignment gives us confidence about the numerical correctness.

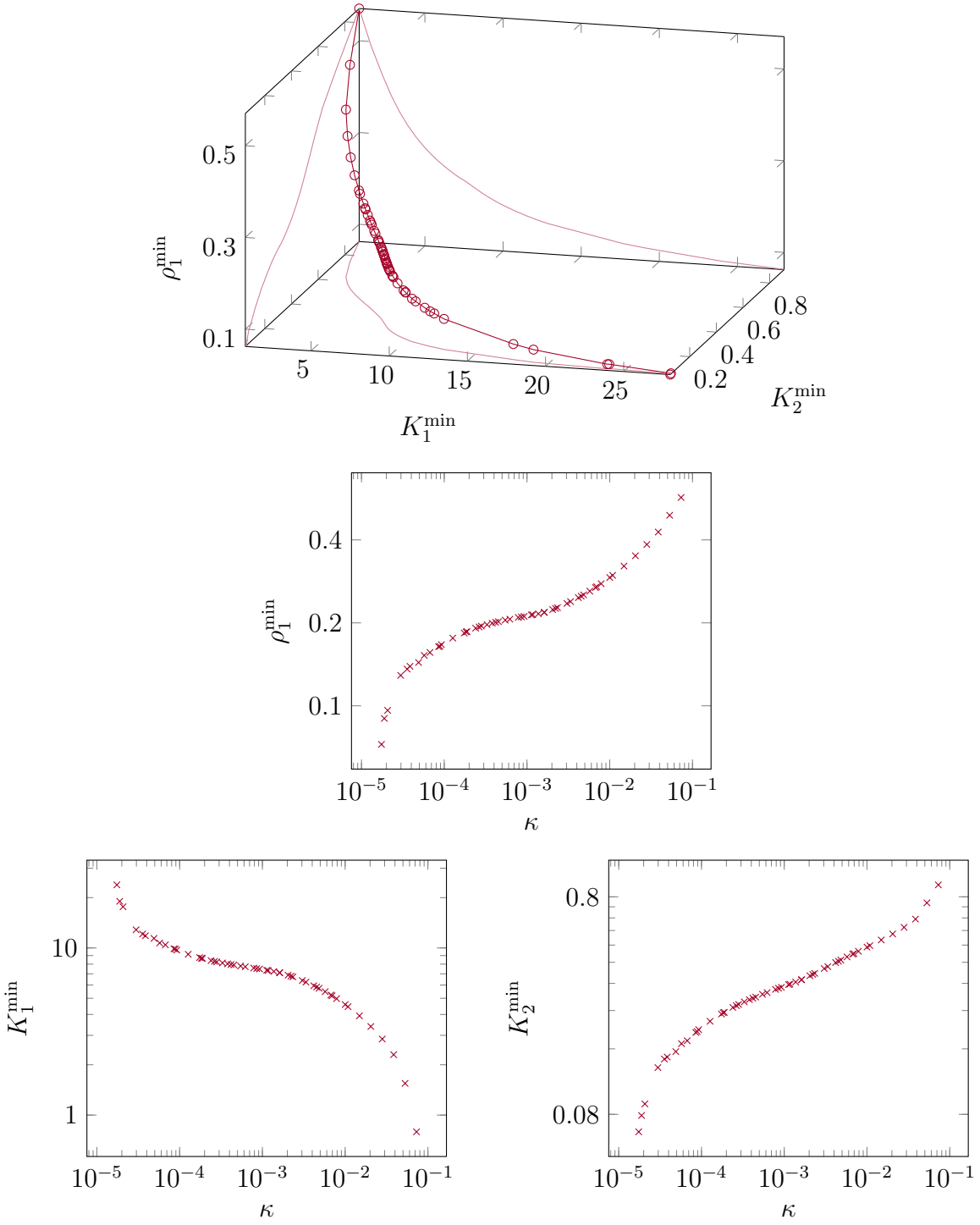

Figure 3.15: In the first plot, we show the trajectory of the minima in the three-dimensional parameter space as $\kappa \to 0$ together with the projections to each of the three two-dimensional planes. The uppermost point corresponds to large κ. In the remaining three plots, we show each of the three free parameters of the minimum as a function of κ separately.

free parameters grows to roughly 3000, which seems almost impossible to handle by the same brute force method we employed here. Needless to say, we would have to redo the cumbersome task of finding reasonable start points all over again. For $N = 2$, we temporarily fixed one parameter ρ_1 and explored what happens in the K_1-K_2-plane. We developed little intuition about the dependence of the minimum on ρ_1, but sampled the whole domain for ρ_1. An analogous procedure is infeasible already for $N = 3$.

Moreover, recall that for $N = 2$ we enjoyed the luxury of a periodic integrand in the temporal direction. Already for $N = 3$, we have two free frequency parameters ω_2, ω_3. For an irrational ratio of those two, the Lagrangian is not periodic. For this reason, we have to factor in the temporal cutoff function η, see Figure 2.2. This means that choosing one fixed domain for the temporal integration is not sufficient anymore, but one has to perform the whole task for several lifetimes of the universe. Since we still expect highly localized features, we will also have to increase the resolution for a growing temporal domain. While we ultimately felt that we had the integration under control for $N = 2$, this is predicted to change drastically for $N = 3$ and higher.

If one were to manage $N = 3$, it would be very interesting to check the following conjecture: It is believed that the minimizer will be periodic, i. e. the frequency parameters ω_i will take on values with pairwise rational ratios. Thereby, the lowest frequency would determine the overall period of the Lagrangian in the temporal direction.

Before one can explore an adequate approximation of the Dirac sea, we expect some fundamental analytic insights to be necessary, in order to overcome the numerical difficulties. However, up to now, we only took some first steps and aimed at gaining a feeling and some intuition about the Dirac sphere. There is a lot of room for improvement in the numerical part too, hence we do not exclude further fruitful numerical analysis even for larger N.

4 Conclusion

This thesis consists of two major parts. For one, we specify the Dirac sphere as a numerically accessible model. Second, we implement the model and provide results for two special cases. Let us rehash the main findings.

In analogy to the well understood Minkowski example in section 1.3, we start from the "usual" notions right away, see Table 1.1. Because of the lessons learned in previous work, see [14], we chose a closed space-time manifold and let time be unbounded. A natural choice therefore is $M := \mathbb{R} \times S^3$. The main task is to make a reasonable ansatz for the kernel of the fermionic operator, since all further quantities, such as the closed chain, the Lagrangian and eventually the action are derived from it.

A good starting point is to model spin one half particles, i. e. we construct four-dimensional Dirac spinors on M. We can write down a solution to the Dirac equation on M directly, see (2.18)-(2.21). Following the idea of causal fermion systems and the related regularization, we relax the conditions (2.18) and allow for more freedom in the choice of the parameters. At the same time, we need to ensure that the number of degrees of freedom does not get too large. Otherwise, we would not be able to minimize the action numerically.

Under the assumptions that P is homogeneous, isotropic and static, as well as that the image of P is negative-definite (2.22) and it has vector-scalar structure (2.23), we can eventually write the Lagrangian as in (2.44). In this simple form, it is perfectly suited for numerical evaluation. Due to [15], we also have the explicit expression (2.48) for the integral kernel of the spectral projectors of the Dirac operator on S^3. Taking into account the constraints has two effects. First, we can eliminate one free parameter using the trace constraint. Second, the boundedness constraint requires us to alter the model by adding the term $\kappa \operatorname{Tr}(A)^2$ to the Lagrangian. In section 2.3, we conclude with a positive answer to our first major question. The Dirac sphere constitutes a model with $3N - 2$ free parameters, which is straight forward to implement on a computer. Moreover, it encodes fermions on a closed, four-dimensional space-time and could hence also reproduce the notion of the Dirac sea.

Chapter 3 is devoted to the actual analysis of the Dirac sphere. The case $N = 1$ in section 3.2 can almost entirely be computed analytically and yields the necessity of the boundedness constraint as a first interesting result. Disregarding the boundedness constraint, we find a diverging minimizing sequence. Therefore, we have to add the κ term, which ensures the existence of minimizers for positive κ, see Figure 3.4.

Introducing the rescaling procedure to sensibly take the limit $\kappa \to 0$ after the optimization is another essential part of this work. For both cases, $N = 1$ and $N = 2$, we find an almost linear dependence of the minimal action on κ. It would be interesting to investigate, whether this behavior is also recovered for larger N. The limit $\kappa \to 0$

for $N = 2$ is summarized in Figure 3.14 and Figure 3.15.

Of course, the most time consuming part of the whole thesis was code development. Extensive testing, trial and error, as well as literature studies led us to the conclusions in sections 3.3 and 3.5 about which methods to use.

Ultimately, the paramount conclusion is that numerical treatment of causal fermion systems will remain a tedious task, but moderate advance in minor steps is definitely possible. The code and considerations developed throughout this thesis, form a solid basis for future numerical analysis of causal fermion systems. However, for physically relevant results, we anticipate further analytic insight to be indispensable. A long-term goal is to compute the Dirac sphere for N on the order of 10^3 to 10^5 and investigate, whether one observes a completely filled Dirac sea with three generations and if or how it deviates from the exact vacuum solution.

We also have short-term goals in mind. First, deriving the Dirac sphere rigorously from the abstract framework of causal fermion systems should in principle be straight forward. Second, the temporal periodicity of the minimizers is another question that poses itself for any $N > 2$. Ultimately, we want to understand in detail the connection of the Dirac sphere to so called *critical minimizers* that arise in the causal action principle of certain models.

References

[1] Milton Abramowitz and Irene A. Stegun, *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, no. 55, Courier Corporation, 1964.

[2] John E. Dennis, Jr and Jorge J. Moré, *Quasi-newton methods, motivation and theory*, SIAM review **19** (1977), no. 1, 46–89.

[3] John E. Dennis Jr and Robert B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*, vol. 16, Siam, 1996.

[4] Felix Finster, *The continuum limit of causal fermion systems*, in preparation.

[5] ———, *The principle of the fermionic projector*, American Mathematical Society, 2006.

[6] ———, *A variational principle in discrete spacetime – existence of minimizers*, Calc. Var. **29** (2007), no. 4, 431–453.

[7] ———, *On the Regularized Fermionic Projector of the Vacuum*, J. Math. Phys. **49** (2008), 032304.

[8] ———, *Causal Variational Principles on Measure Spaces*, J. reine angew. Math. **646** (2010), 141–194.

[9] ———, *Causal Fermion Systems – An Overview*, ArXiv e-prints (2015).

[10] Felix Finster, Andreas Grotz, and Daniela Schiefeneder, *Causal fermion systems: A quantum space-time emerging from an action principle*, Quantum Field Theory and Gravity (Felix Finster, Olaf Müller, Marc Nardmann, Jürgen Tolksdorf, and Eberhard Zeidler, eds.), Springer Basel, 2012, pp. 157–182.

[11] Felix Finster and Stefan Hoch, *An Action principle for the masses of Dirac particles*, Adv. Theor. Math. Phys. **13** (2009), 1653–1711.

[12] Felix Finster and Johannes Kleiner, *Causal Fermion Systems as a Candidate for a Unified Physical Theory*, J. Phys. Conf. Ser. **626** (2015), no. 1, 012020.

[13] Felix Finster, Johannes Kleiner, and Jan-Hendrik Treude, *An introduction to the fermionic projector and causal fermion systems*, in preparation.

[14] Felix Finster and Watzold Plaum, *A Lattice model for the fermionic projector in a static and isotropic space-time*, Math. Nachr. **281** (2008), 803–816.

[15] Felix Finster and Moritz Reintjes, *The Dirac equation and the normalization of its solutions in a closed Friedmann Robertson-Walker universe*, Classical and Quantum Gravity **26** (2009), no. 10.

[16] Thomas Friedrich and Andreas Nestke, *Dirac operators in riemannian geometry*, vol. 25, American Mathematical Society Providence, 2000.

[17] The MathWorks Inc., *Matlab*, Natick, MA.

[18] Arnold R. Krommer and Christoph W. Ueberhuber, *Computational integration*, Siam, 1998.

[19] H. Blaine Lawson and Marie-Louise Michelsohn, *Spin geometry*, Princeton Mathematical Series, vol. 38, Princeton University Press, 1989.

[20] Michael A. Malcolm and R. Bruce Simpson, *Local versus global strategies for adaptive quadrature*, ACM Trans. Math. Softw. **1** (1975), no. 2, 129–146.

[21] Maplesoft, a division of Waterloo Maple Inc., *Maple*, Waterloo, Ontario.

[22] Jorge J. Moré and David J. Thuente, *Line search algorithms with guaranteed sufficient decrease*, ACM Trans. Math. Softw. **20** (1994), no. 3, 286–307.

[23] Jorge Nocedal and Stephen Wright, *Numerical optimization*, Springer Science & Business Media, 2006.

[24] Robert Piessens and Maria Branders, *A note on the optimal addition of abscissas to quadrature formulas of gauss and lobatto type*, Math. Comp **28** (1974), no. 125, 135–139.

[25] Lloyd N. Trefethen, *Is gauss quadrature better than clenshaw-curtis?*, SIAM review **50** (2008), no. 1, 67–87.

[26] Inc. Wolfram Research, *Mathematica*, Champaign, Illinois.

Index

Eidesstattliche Erklärung

Ich habe die Arbeit selbstständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und bisher keiner anderen Prüfungsbehörde vorgelegt.

Palo Alto (CA), am 23. November 2015

Niki Kilbertus