Numerical Analysis Using Generalised Pattern Search for a Discrete Fermionic Lattice Model of the Vacuum

DISSERTATION
ZUR ERLANGUNG DES DOKTORGRADES
DER NATURWISSENSCHAFTEN (DR. RER. NAT.)
AN DER NWF I – MATHEMATIK
DER UNIVERSITÄT REGENSBURG

vorgelegt von
Wätzold Plaum aus Regensburg

2009

Promotionsgesuch eingereicht am:	9. Juni 2009
Die Arbeit wurde angeleitet von:	Prof. Dr. Felix Finster
Prüfungsausschuss:	Prof. Dr. Felix Finster Prof. Dr. Georg Dolzmann Prof. Dr. Bernd Ammann Prof. Dr. Günter Tamme

für Laurin und Leonhard

It’s more fun to compute!
Kraftwerk – Computerwelt

Contents

Abstract

This thesis deals with the first numerical analysis of the variation principle concerning the theory of the Fermionic Projector.

A model for describing discrete fermionic systems is developed, whereas the case of vacuum is discussed. In the continuous case, vacuum systems can be described by the Fermionic Projector of the Dirac Sea. The discretisation of this concept allows the description of physical systems by the introduction of an action principle. In this thesis systems capable of configuring discretisations of continuous systems with one Dirac Sea are numerically analysed. For the purpose of a more easy numerical analysis spherical symmetry in momentum space is introduced.

The numerical problem of MINLP emerging from this setting is treated by the methods of complete enumeration and the MGPS algorithm, an extension of the method of *Generalised Pattern Search*.

The general hypothesis of this thesis is that there exist Dirac Sea like minimisers. This thesis could be confirmed by the numerical results. The model had to undergo some subtle modifications – which has to be considered technical in nature – to deliver the expected results. Finally several research programs for further research are addressed, which aim to bring forward the numerical treatment of this problem from a prototypical state to a state of high performance parallel computing.

Acknowledgments

I express my sincere appreciation to all those who have helped me through this very challenging ordeal within the time working on this thesis.

First of all, I wish to tank my advisor, Professor Felix Finster, who gave me on the one hand enough freedom to develop my own ideas, but on the other hand always kept an eye on the progress of my work. Second I wish to thank my colleques Stefan Hoch, Daniela Schiefeneder, Andreas Grotz, Hans Kronthaler and Marc Nardmann for their helpful comments and their amicable sociability during the time of my graduation.

Starting the work at this thesis, I had a strong need for support concerning the more technical aspect of numerical optimisation. I'm glad and thankful for the procurance of Prof. Dr. Jochem Häusser from the University of Applied Sciences Braunschweig/-Wolfenbüttel, who advocated me the help of Dr. Hans-Georg Paap. I thank Dr. Paap for his usefull hints and the time he spent to advise me.

Finally I have to thank my family especially my partner Justine for their patience and backing during the sometimes hard time of working at this thesis.

Regensburg, 2009 *Wätzold Plaum*

Declaration of Symbols

Symbol	Explanation
n	Number of occupied states
P	Fermionic Projector
$P(x, y)$	Integral Kernel of the Fermionic Projector
$A(x, y)$	Closed Chain
$L, \mathcal{L}$	Lagrangian
S	Action
ω	Vector of energy values with components either discrete or continuous.
$\omega_{\max}$	Border of allowed values for the components of ω: $0 \leq \omega_i \leq \omega_{\max}$ for all i.
τ	Vector of "angle parameters"
$\hat{\phi}$	Scalar component of the Fermionic Projector in momentum space
$\hat{v} = (\hat{v}_0, \hat{v_k})$	Two dimensional vector component of the Fermionic Projector in the spherical symmetric momentum space
ϕ	Scalar component of the Fermionic Projector in location space
$v = (v_0, v_k)$	Two dimensional vector component of the Fermionic Projector in the spherical symmetric location space
N	Lattice Factor
ν	Ordering number of optimisation runs
α_ω	Scattering Factor concerning ω
α_τ	Scattering Factor concerning τ
μ_r	Relative Qualification Parameter
μ_a	Absolute Qualification Parameter
$\|\| \cdot \|\|_p$	ℓ^p-norm

Finally we have to clarify some notation. Assuming a relation $\odot \subseteq M \times M$ with any set M written as $x \odot y$ if $(x, y) \in \odot$. Assuming further $n > 1$ a natural number and $u, v \in M^n$. Then we write

$$u \,.\odot\, v \quad \text{if} \quad u_i \odot v_i \;\forall\, i \text{ with } 1 \leq i \leq n. \tag{1}$$

Introduction

This thesis deals with the analysis of a variational principle connected with the theory of the *Fermionic Projector*. This theory is a proposal for modelling the physics of a discrete space-time (see [FIN2]). The assumption of a discreetness of space-time in the realm of the Planck scale

$$l_p = \sqrt{\frac{\hbar G}{c^3}} \approx 10^{-35} m \tag{2}$$

arises from solving the problems, which occur when one tries to merge General Relativity and Quantum Field Theory. For instance, spontaneous generation of black holes should be possible at this scale (see [CAL]). Perhaps the assumption of discrete space-time at the Planck scale gets more evidence from technically required regularisations related to the common renormalisation techniques. This means that as an ad hoc hypothesis there is made use of an finite energy and thus length scale cutoff (see [REB, chap. 20]).

Nowadays there exist quite a wide range of approaches to the physics of discrete space-time, e.g. Loop Quantum Gravity (see [ROV]) or non-commutative geometry ([MAJ]). A further approach, which provides the framework for our work, is the theory of the Fermionic Projector. This theory includes a variational principle, which can be used to formulate equations in the discrete space-time as well as to determine solutions.

Recently it was proved [FIN1], that this variational principle has minima. But there is not much known about their concrete structure. Calculations in the continuum lead to the hypothesis that for systems with many space-time points there exist minima, which are consistent with ad hoc discretisation of continuous Dirac Sea configurations. In this thesis, we will provide a first account to determine minima resulting from the variational principle numerically. So our problem comprehends the very first numerical exploration of the theory of the Fermionic Projector. For this purpose we deploy a model system. Hence we cannot estimate to touch topics which would be approachable for a concrete experimental test. But nevertheless, in the end of our work, we will answer the question if the variational principle of the theory of the Fermionic Projector does make sense physically. The physical impact of this numerical analysis is that by finding concrete minima

using numerical optimisation one can prove the theoretical prediction of minima and thus prove whether the theory of the Fermionic Projector leads to a stable vacuum, which could be for instance a base of operations for perturbation theory. Further we have to prove the general hypothesis that the minima preferred by the action principle (if they exist) are in some kind similar to the one expected from the continuous theory.

We first have to define a system for which the action can be calculated in an easy way. The discrete spacetime assumed for this model has not to be considered as the real structure of (discrete) spacetime. It forms nothing more than a model to get a first insight into the action principle of the fermionic projector. After the definition of the model we have to solve the resulting MINLP optimisation problem. Since the system size is not determined by the model we will start with the analysis of small systems and try to enlarge system size gradually. Different numerical strategies have to be tested and compared.

This thesis is organised as follows: First (cha. 1) we introduce a model based on the principle of the Fermionic Projector for a discrete space-time and work out the corresponding action. Then (cha. 2) we will generally discuss the possible approaches to the formulated problem from the numerical point of view. That is we have a look on the different algorithms to solve the numerical optimisation problem and argue why we chose the class of GPS algorithms as the main tool in this work. Chapter 3 discusses some features of the Lagrangian density of the action defined in Chapter 1. Chapter 4 provides some simple numerical results while chapter 5 discusses the Lagrange denisty of Dirac Sea like configurations for the purpose of comparison with the numerical determined minimiser. Chapter 6 will then discuss optimisation performed by a combination of GPS searches concerning the continuous part of the configuration space and complete enumeration concerning the discrete part. Chapter 7 has to be regarded as a preparation for the following two chapters, since some performance and quality comparisons are done here, which guide the further numerical analysis. Chapter 8 contains the most extensive calculations which aim on solid results for comparatively small systems. Chapter 9 contains attempts to get a numerical approach to larger systems by the cost of weaker evidence. Chapter 10 abstracts and discusses the results of this thesis.

Chapter 1

A new Model for a discrete Vacuum

1.1 Introducing Remarks to the Theory of the Fermionic Projector

It is generally believed that the concept of a space-time continuum (like Minkowski space or a Lorentzian manifold) should be modified for distances as small as the Planck length. The principle of the Fermionic Projector [FIN2] proposes a mathematical framework for physics on the Planck scale in which space-time is discrete. The physical equations are formulated via a variational principle for fermionic wave functions defined on a finite set of space-time points, without referring to notions like space, time or causality. The idea is that these additional structures, which are of course essential for the description of nature, arise as a consequence of the nonlinear interaction of the fermions as described by the variational principle. More specifically, it was proved that the original permutation symmetry of the space-time points is spontaneously broken by the fermionic wave functions [FIN3]. This means that the fermions will induce non-trivial relations between the space-time points. In particular, one can introduce the notion of a "discrete causal structure" (see the short review article [FIN4]). The conjecture is that for systems involving many space-time points and many particles, the fermions will group to a "discrete Dirac Sea structure", which in a suitable limit where the number of particles and space-time points tends to infinity, should go over to the well-known Dirac Sea structure in the continuum. Then the "discrete causal structure" will also go over to the usual causal structure of Minkowski space [FIN2].

Hints that the above conjecture is true have been obtained coming from the continuum theory. First, our variational principle has a well defined continuum limit [FIN2, Chapter 4], and we get promising results for the resulting effective continuum theory [FIN2,

Chapters 6-8]. Furthermore, rewriting certain composite expressions ad hoc as distributions in the continuum, one finds that Dirac Sea configurations can be stable minima of our variational principle [FIN2, Chapter 5.5]. The ad-hoc procedure of working with distributions is justified in the paper [FIN5], which also gives concrete hints on how the regularised fermionic projector should look like on the Planck scale. For a more detailed stability analysis in the continuum see [FH].

Despite these results, many questions on the relation between discrete space-time and the continuum theory remain open. In particular, it seems an important task to complement the picture coming from the discrete side; that is, one should analyze large discrete systems and compare the results with the continuum analysis. Since minimising the action for a discrete system can be regarded as a problem of non-linear optimisation, numerical analysis seems a promising method. Numerical investigations have been carried out successfully for small systems involving few particles and space-time points [FSD]. For large systems, however, the increasing numerical complexity would make it necessary to use more sophisticated numerical methods or to work with more powerful computers. Therefore, it seems a good idea to begin with simplified systems, which capture essential properties of the original system but are easier to handle numerically. In this chapter, we shall introduce such a simplified system. The method is to employ a spherically symmetric and static ansatz for the Fermionic Projector. This reduces the number of degrees of freedom so much that it becomes accessible to simulate systems which are so large that they can be compared in a reasonable way to the continuum.

The chapter is organised as follows. In Section 1.2 we review the mathematical framework of the Fermionic Projector in discrete space-time and introduce our variational principle. In Section 1.3, we take a spherically symmetric and static ansatz in Minkowski space and discretise in the time and the radial variable to obtain a two-dimensional lattice. In Section 1.4, our variational principle is adapted to this two-dimensional setting. In Section 1.5, we give a precise definition of our model and discuss its basic properties; for clarity this section is self-contained and independent of the rest of the chapter. In Section 1.6, the existence of minimisers is proved. The purpose of this chapter is to define the model and to discuss some basic properties. Numerical simulations of larger systems will be presented in the following chapters.

1.2 A Variational Principle in Discrete Space-Time

We briefly recall the mathematical setting of discrete space-time and the definition of our variational principle in the particular case of relevance here (for a more general introduction see [FIN1]). Let H be a finite-dimensional complex vector space endowed with a non-degenerate symmetric sesquilinear form $\langle .|. \rangle$. We call $(H, \langle .|. \rangle)$ an *indefinite inner*

product space. The adjoint A^* of a linear operator A on H can be defined as in Hilbert spaces by the relation $\langle A\Psi \,|\, \Phi\rangle = \langle \Psi \,|\, A^*\Phi\rangle$. A selfadjoint and idempotent operator is called a *projector*. To every element x of a finite set $M = \{1, \ldots, m\}$ we associate a projector E_x. We assume that these projectors are orthogonal and complete,

$$E_x E_y = \delta_{xy} E_x , \qquad \sum_{x \in M} E_x = \mathbb{1} . \tag{1.1}$$

Furthermore, we assume that the images $E_x(H) \subset H$ of these projectors are all four-dimensional and non-degenerate of signature $(2, 2)$. The points $x \in M$ are called *discrete space-time points*, and the corresponding projectors E_x are the *space-time projectors*. The structure $(H, \langle .|. \rangle, (E_x)_{x \in M})$ is called *discrete space-time*. Furthermore, we introduce the *Fermionic Projector* P as a projector on a subspace of H which is negative definite and of dimension f. The vectors in the image of P have the interpretation as the occupied quantum states of the system, and f is the number of particles. We refer to $(H, \langle .|. \rangle, (E_x)_{x \in M}, P)$ as a *fermion system in discrete space-time*.

When forming composite expressions in the projectors P and $(E_x)_{x \in M}$, it is convenient to use the short notations

$$\Psi(x) = E_x \Psi \qquad \text{and} \qquad P(x, y) = E_x P E_y . \tag{1.2}$$

Using (1.1), we obtain for any $\Psi, \Phi \in H$ the formula

$$\langle \Psi \,|\, \Phi \rangle = \sum_{x \in M} \langle \Psi(x) \,|\, \Phi(x) \rangle_{E_x(H)} , \tag{1.3}$$

and thus the vector $\Psi(x) \in E_x(H) \subset H$ can be thought of as the "localization" of the vector Ψ at the space-time point x. Furthermore, the operator $P(x, y)$ maps $E_y(H) \subset H$ to $E_x(H)$, and it is often useful to regard it as a mapping only between these subspaces,

$$P(x, y) \;:\; E_y(H) \to E_x(H) .$$

Again using (1.1), we can write the vector $P\Psi$ as follows,

$$(P\Psi)(x) = E_x P\Psi = \sum_{y \in M} E_x P E_y \Psi = \sum_{y \in M} (E_x P E_y)\, (E_y \Psi) ,$$

and thus

$$(P\Psi)(x) = \sum_{y \in M} P(x, y)\, \Psi(y) . \tag{1.4}$$

This relation resembles the representation of an operator with an integral kernel. Therefore, we call $P(x, y)$ the *discrete kernel* of the Fermionic Projector.

To introduce our variational principle, we define the *closed chain* A_{xy} by

$$A_{xy} \;=\; P(x,y)\,P(y,x) \;:\; E_x(H) \to E_x(H)\,. \tag{1.5}$$

Let $\lambda_1, \ldots, \lambda_4$ be the zeros of the characteristic polynomial of A_{xy}, counted with multiplicities. We define the *Lagrangian* by

$$\mathcal{L}[A_{xy}] \;=\; \frac{1}{8} \sum_{i,j=1}^{4} \Big(|\lambda_i| - |\lambda_j|\Big)^2 \tag{1.6}$$

and introduce the *action* by summing over the space-time points,

$$\mathcal{S}[P] \;=\; \sum_{x,y \in M} \mathcal{L}[A_{xy}]\,. \tag{1.7}$$

Our variational principle is to minimise this action under variations of the Fermionic Projector. We remark that this is the so-called critical case of the auxiliary variational principle as introduced in [FIN2, **?**].

1.3 The Spherically Symmetric Discretization

Recall that in discrete space-time, the subspace $E_x(H) \subset H$ associated to a space-time point $x \in M$ has signature $(2,2)$. In the continuum, this vector space is to be identified with an inner product space of the same signature: the space of Dirac spinors at a space-time point $x \in \mathbb{R}^4$ with the inner product $\overline{\Psi}\Phi$, where $\overline{\Psi} = \Psi^\dagger \gamma^0$ denotes the adjoint spinor. For any 4×4-matrix B acting on the spinors, the adjoint with respect to this inner product is denoted by $B^* = \gamma^0 B^\dagger \gamma^0$. Furthermore, the indefinite inner product space $(H, \langle .|. \rangle)$ in the continuum should correspond to the space of Dirac wave functions in space-time with the inner product

$$\langle \Psi \,|\, \Phi \rangle \;=\; \int \overline{\Psi(x)}\, \Phi(x)\, d^4x\,. \tag{1.8}$$

This resembles (1.3), only the sum has become a space-time integral integral. Likewise, in (1.4) the sum should be replaced by an integral,

$$(P\Psi)(x) \;=\; \int P(x,y)\, \Psi(y)\, d^4y\,,$$

where now $P(x,y)$ is the integral kernel of the Fermionic Projector of the continuum P. Since we assume that our system is isotropic, it follows that it is *homogeneous* in space. Furthermore, we assume that our system is *static*, and thus the integral kernel depends only on the difference $y - x$,

$$P(x,y) \;=\; P(\xi) \qquad \text{for all } x, y \in \mathbb{R}^4 \text{ and } \xi := y - x. \tag{1.9}$$

We take the Fourier transform in ξ,

$$P(\xi) \;=\; \int \frac{d^4k}{(2\pi)^4}\, \hat{P}(p)\, e^{ip\xi}\,, \tag{1.10}$$

where $p\xi$ denotes the Minkowsi inner product of signature $(+---)$. Let us collect some properties of $\hat{P}(p)$. First, the operator P should be symmetric (= formally self-adjoint) with respect to the inner product (1.8). This means for its integral kernel that

$$P(\xi)^* = P(-\xi)\,, \tag{1.11}$$

and likewise for its Fourier transform that

$$\hat{P}(p)^* \;=\; \hat{P}(p)\,.$$

Assuming as in [FIN2, §4.1] that the Fermionic Projector has a *vector-scalar structure*, $\hat{P}$ can be written as

$$\hat{P}(p) \;=\; \hat{v}_j(p)\,\gamma^j + \hat{\phi}(p)\,\mathbb{1} \tag{1.12}$$

with a real vector field $\hat{v}$ and a real scalar field $\hat{\phi}$. Moreover, the assumption of *spherical symmetry* implies that the above functions depend only on $\omega := p^0$ and on $k := |\vec{p}|$, and that the vector component can be written as

$$\hat{v}_j\gamma^j \;=\; \hat{v}_0\,\gamma^0 + \hat{v}_k\,\gamma^k \qquad \text{with} \qquad \gamma^k \;:=\; \frac{\vec{p}\,\vec{\gamma}}{|\vec{p}|}$$

and real-valued functions $\hat{v}_0$ and $\hat{v}_k$. Next we can exploit that the image of P should be *negative definite*. Moreover, since P should be a projector, it should have *positive spectrum*. Since in Fourier space, P is simply a multiplication operator, we can consider the operator $\hat{P}(p)$ for any fixed p. This gives rise to the conditions that the vector field $\hat{v}$ must have the same Lorentz length as $\hat{\phi}$ and must be past-directed,

$$\hat{v}_0 \;<\; 0 \qquad \text{and} \qquad \hat{v}_0^2 - \hat{v}_k^2 \;=\; \hat{\phi}^2\,,$$

and furthermore that $\hat{\phi}$ must be non-negative. Combining the above conditions, we conclude that $\hat{P}$ can be written in the form

$$\hat{P}(p) \;=\; \hat{\phi}(\omega,k)\Big(\mathbb{1} \;-\; \gamma^0\,\cosh\tau(\omega,k) \;+\; \gamma^k\,\sinh\tau(\omega,k)\Big) \tag{1.13}$$

with a non-negative function $\hat{\phi}$ and a real function τ. Note that we have not yet used that P should be idempotent, nor that the rank of P should be equal to the number of particles f. Indeed, implementing these conditions requires a more detailed discussion, which we postpone until the end of this section.

We next compute the Fourier transform of (1.13), very similar as in [FIN5, Lemma 5.1]. Introducing in position space the polar coordinates $\xi = (t, r, \vartheta, \varphi)$ and assuming that $r \neq 0$, the scalar component becomes

$$\begin{aligned} \phi(t,r) &= \frac{1}{(2\pi)^4} \int_{-\infty}^{\infty} d\omega \int_0^{\infty} k^2\, dk \int_{-1}^{1} d\cos\vartheta \int_0^{2\pi} d\varphi\; \hat\phi(\omega,k)\, e^{i\omega t - ikr\cos\vartheta} \\ &= \frac{1}{4\pi^3\, r} \int_{-\infty}^{\infty} d\omega\, e^{i\omega t} \int_0^{\infty} k\, dk\; \sin(kr)\; \hat\phi(\omega,k)\,. \end{aligned}$$

The zero component of the vector component is computed similarly,

$$v_0(t,r) = -\frac{\gamma^0}{4\pi^3\, r} \int_{-\infty}^{\infty} d\omega\, e^{i\omega t} \int_0^{\infty} k\, dk\; \sin(kr)\; \hat\phi(\omega,k)\; \cosh\tau(\omega,k)\,.$$

For the calculation of the radial component, we first need to pull the Dirac matrices out of the integrals,

$$\begin{aligned} &\int \frac{\mathrm{d}^4 p}{(2\pi)^4}\, \hat v_k \gamma^k\, e^{ip\xi} = \frac{\vec\gamma \vec\nabla_{\vec x}}{(2\pi)^4} \int \mathrm{d}^4 p\; \hat v_k\,(\omega,k)\; \frac{i}{k}\; e^{i\omega t - i\vec k \vec x} \\ &= i\vec\gamma\vec\nabla \left(\frac{1}{4\pi^3\, r} \int_{-\infty}^{\infty} d\omega\, e^{i\omega t} \int_0^{\infty} dk\; \sin(kr)\; \hat\phi(\omega,k)\; \sinh\tau(\omega,k) \right) \\ &= \frac{i\,\gamma^r}{4\pi^3\, r} \int_{-\infty}^{\infty} d\omega\, e^{i\omega t} \int_0^{\infty} k\, dk \left(\cos(kr) - \frac{\sin(kr)}{kr} \right) \hat\phi(\omega,k)\; \sinh\tau(\omega,k)\,, \end{aligned}$$

where we set $\gamma^r = (\vec\xi\vec\gamma)/|\vec\xi|$. Combining the above terms, we obtain

$$\begin{aligned} P(\xi) &= \frac{1}{4\pi^3\, r} \int_{-\infty}^{\infty} d\omega\, e^{i\omega t} \int_0^{\infty} k\, dk\; \hat\phi(\omega,k) \Big[\mathbb{1}\, \sin(kr) \\ &\quad - \gamma^0\; \cosh\tau(\omega,k)\; \sin(kr) \;+ i\gamma^r\; \sinh\tau(\omega,k) \Big(\cos(kr) - \frac{\sin(kr)}{kr} \Big) \Big]. \end{aligned} \tag{1.14}$$

Note that this formula has a well defined limit as $r \searrow 0$, and thus we set

$$P(t, r=0) = \frac{1}{4\pi^3} \int_{-\infty}^{\infty} d\omega\, e^{i\omega t} \int_0^{\infty} k^2\, dk\; \hat\phi(\omega,k) \left[\mathbb{1} - \gamma^0\; \cosh\tau(\omega,k) \right]. \tag{1.15}$$

In (1.13) and (1.14), the factors γ^k and γ^r involve an angular dependence. But all the other functions depend only on the position variables (t, r) and the corresponding momenta (ω, k). We now discretize these variables. In view of (1.11) it suffices to consider the case $t \geq 0$. The position variables should be on a finite lattice $\mathfrak{L}$,

$$(t,r) \in \mathfrak{L} := \Big\{ 0, \Delta_t, \ldots, (N_t - 1)\Delta_t \Big\} \times \Big\{ 0, \Delta_r, \ldots, (N_r - 1)\,\Delta_r \Big\}\,,$$

where N_t and N_r denote the number of lattice points in time and radial directions, and $\Delta_t, \Delta_r > 0$ are the respective lattice spacings. The momentum variables should be on the corresponding dual lattice $\hat{\mathfrak{Q}}$,

$$(\omega, k) \in \hat{\mathfrak{Q}} := \Big\{ -(N_t - 1)\,\Delta_\omega, \ldots, -\Delta_\omega, 0 \Big\} \times \Big\{ \Delta_k, \ldots, N_r\,\Delta_k \Big\}, \tag{1.16}$$

where we set

$$\Delta_\omega = \frac{2\pi}{\Delta_t\, N_t}, \qquad \Delta_k = \frac{2\pi}{\Delta_r\, N_r}.$$

We point out that the parameter ω in (1.16) is non-positive; this is merely a convention because we are always free to add to ω a multiple of $N_t\Delta_\omega$. Furthermore, note that the points with $k = 0$ have been excluded in $\hat{\mathfrak{Q}}$. This is because the integrands in (1.14) and (1.15) vanish as $k \searrow 0$, and thus it seems unnecessary to consider the points with $k = 0$. However, since $P(\xi)$ has a non-trivial value at $r = 0$ (see (1.15)), it seems preferable to take into account the points with $r = 0$ in the lattice $\mathfrak{Q}$. Replacing the Fourier integrals by a discrete Fourier sum, (1.14) and (1.15) become

$$\begin{aligned} P(\xi) &= \frac{\Delta_\omega \Delta_k}{4\pi^3\, r} \sum_{(\omega,k)\in\hat{\mathfrak{Q}}} e^{i\omega t}\, k\, \hat{\phi} \Big[(\mathbb{1} - \gamma^0 \cosh\tau)\, \sin(kr) \\ &\qquad + i\gamma^r \sinh\tau \Big(\cos(kr) - \frac{\sin(kr)}{kr} \Big) \Big], \quad \text{if } r \neq 0 \end{aligned} \tag{1.17}$$

$$P(t, r=0) = \frac{\Delta_\omega \Delta_k}{4\pi^3} \sum_{(\omega,k)\in\hat{\mathfrak{Q}}} e^{i\omega t}\, k^2\, \hat{\phi}\, (\mathbb{1} - \gamma^0 \cosh\tau), \tag{1.18}$$

with functions $\hat{\phi}$ and τ defined on $\hat{\mathfrak{Q}}$.

The points of the dual lattice $\hat{\mathfrak{Q}}$ have the interpretation as the quantum states of the system, which may or may not be occupied by fermionic particles. More precisely, if $\hat{\phi}(\omega, p) \neq 0$, a whole "shell" of fermions of energy ω and of momenta $\vec{k}$ with $|\vec{k}| = p$ is occupied. For most purposes it is convenient and appropriate to count the whole shell of fermions as one particle of our lattice model. Thus if $\hat{\phi}(\omega, p) \neq 0$, we say that the lattice point (ω, p) is *occupied by a particle*; otherwise the lattice point is not occupied. A system where n lattice points are occupied is referred to as an *n-particle system*. Each particle is characterized by the values of $\hat{\phi}$ and τ, or, equivalently, by the vector $(-2k\hat{\phi}\cosh\tau, 2k\hat{\phi}\sinh\tau)$. It is convenient to describe the fermion system by drawing these vectors at all occupied lattice points, as shown in Figure 1.1 for a three-particle system.

We conclude this section by a discussion of what the parameter f and the idempotence condition $P^2 = P$ of discrete space-time mean in the setting of our lattice model. In discrete space-time, the number of particles f equals the trace of P. Computing the trace of P naively for our lattice model, our homogeneous ansatz (1.9) yields

$$f = \mathrm{Tr}P = \int_{\mathbb{R}^4} \mathrm{tr}P(x,x)\, d^4x = \mathrm{tr}P(\xi = 0) \cdot \infty, \tag{1.19}$$

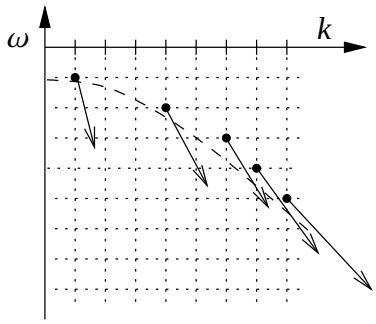


Figure 1.1: Example of a three-particle system on a 3×3-lattice

where "tr" denotes the trace of a 4×4-matrix. According to (1.18),

$$\mathrm{tr} P(\xi = 0) \;=\; \frac{\Delta_\omega \Delta_k}{\pi^3} \sum_{(\omega,k) \in \hat{\Omega}} k^2\, \hat{\phi}(\omega, k)\,, \tag{1.20}$$

showing that (1.19) is equal to $+\infty$ unless P vanishes identically. Here we used essentially that, although $\xi = y - x$ was discretized on a finite lattice, the space-time variable x itself is still an arbitrary vector in Minkowski space. In other words, our lattice system is a homogeneous system in infinite volume, and in such a system the number of particles is necessarily infinite. The simplest way to bypass this problem is to note that for a homogeneous system in discrete space-time [FIN1, Def. 2.4],

$$f \;=\; \sum_{x \in M} \mathrm{Tr}(E_x P) \;=\; m\, \mathrm{Tr}(E_1 P)\,,$$

and so the number of particles grows linearly with the number of space-time points. Due to this simple connection, we can disregard f and consider instead the local trace. This has the advantage that the local trace can be identified with the expression (1.20) of our lattice system. For the variational principle in discrete space-time (1.6, 1.7), it is important that variations of P keep the number of particles f fixed. This condition can be carried over to our lattice system, giving rise to the so-called *trace condition* (TC):

(TC) When varying the Fermionic Projector of the lattice system (1.17, 1.18), the local trace as defined by

$$f_{\text{loc}} \;:=\; \frac{\Delta_\omega \Delta_k}{\pi^3} \sum_{(\omega,k) \in \hat{\Omega}} k^2\, \hat{\phi}(\omega, k)$$

should be kept fixed.

We conclude that, although f is infinite for our lattice system, the local trace f_{loc} is well defined and finite. This all we need, because with (TC) we have implemented the condition corresponding to the condition in discrete space-time that f should be kept fixed

under variations of P. We point out that neither f nor f_{loc} coincides with the number of particles as obtained by counting the occupied states.

The idempotence condition $P^2 = P$ is satisfied if and only if the fermionic wave functions are properly normalized. As explained above, our lattice model is defined in infinite space-time volume, and thus a-priori the normalization integrals diverge. As shown in [FIN2, §2.6], a possible method for removing this divergence is to consider the system in finite 3-volume and to smear out the mass parameter. However, there are other normalization methods, and it is not clear whether they all give rise to the same normalization condition for our lattice model. The basic difficulty is related to the fact that each occupied lattice point $(\omega, p) \in \hat{\Omega}$ corresponds to a whole shell of fermions (see above). Thus the corresponding summand in (1.17, 1.18) involves an "effective wave function" describing an ensemble of fermions. But it is not clear of how many fermions the ensemble consists and thus, even if we knew how to normalize each individual fermion, the normalization of the effective wave function would still be undetermined. This problem becomes clear if one tries to model the same physical system by two lattice models with two different lattice spacings. Then in general one must combine several occupied lattice points of the finer lattice to one "effective" occupied lattice point of the coarser lattice. As a consequence, the normalization of the coarser lattice must be different from that on the finer lattice. This explains why there is no simple canonical way to normalize the effective wave functions.

Our method for avoiding this normalization problem is to choose the normalization in such a way that the Fermionic Projector of the continuum can be carried over easily to the lattice system: In Minkowski space, a Dirac Sea in the vacuum is described by the distribution (see [FIN2, §2.2])

$$\hat{P}(p) \;=\; (\not{p} + m)\, \delta(p^2 - m^2)\, \Theta(-p^0)\,. \tag{1.21}$$

Taking the Fourier transform and carrying out the angular integrals, we obtain again the expressions (1.14, 1.15), but now with $\hat{\phi}(\omega, k) = \delta(\omega^2 - k^2 - m^2)$. This allows us to carry out the k-integral,

$$\int_{-\infty}^{\infty} d\omega\, e^{i\omega t} \int_0^{\infty} k\, dk\; \delta(\omega^2 - k^2 - m^2)\; \cdots \;=\; \int_{\mathbb{R}\setminus[-m,m]} d\omega\, e^{i\omega t}\; \frac{1}{2}\; \cdots \Big|_{k=\sqrt{\omega^2-m^2}}\,.$$

The easiest method to discretize the obtained expression is to replace the ω-integral by a sum, and to choose for every $\omega \leq -m$ a lattice point $(\omega, k) \in \hat{\Omega}$ such that

$$0 \;\leq\; k - \sqrt{\omega^2 - m^2} \;<\; \Delta_k\,. \tag{1.22}$$

An example for the resulting discretized Dirac Sea is shown in Figure 1.2.

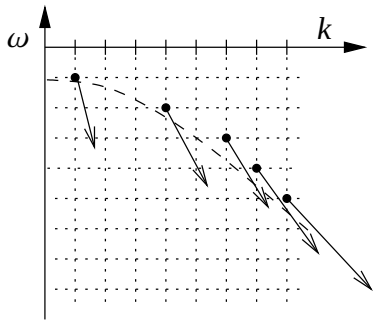


Figure 1.2: A discretized Dirac Sea

Note that for this configuration, $\hat{\phi}(\omega, k) = 1/(2k)$ at all occupied lattice points. Next we allow to modify this configuration, as long as the normalization integrals remain unchanged: First, changing τ corresponds to a unitary transformation of the corresponding state, without influence on the normalization. Second, hopping from a lattice point to another unoccupied lattice point with the same value of k changes the state only by the phase factor $\exp(-i(\omega - \omega')t)$, again without influence on the normalization. This leaves us with the so-called *normalization condition* (NC):

(NC) The function $\hat{\phi}$ in (1.17, 1.18) should only take the two values

$$\hat{\phi}(\omega, k) = 0 \qquad \text{or} \qquad \hat{\phi}(\omega, k) = \frac{1}{2k}\,.$$

We again point out that this normalization condition is not canonical. It could be modified or even be left out completely. It seems an interesting question to analyze how the behavior of the lattice model depends on the choice of the normalization condition.

1.4 The Variational Principle on the Lattice

The Lagrangian (1.6) is also well defined for our lattice model. Let us compute it in more detail. We decompose the fermionic projector (1.14, 1.15) into its scalar and vector components,

$$P(\xi) = \phi(t,r)\,\mathbb{1} + v_0(t,r)\,\gamma^0 + v_r(t,r)\,\gamma^r = \phi\,\mathbb{1} + v_j\,\gamma^j\,.$$

Furthermore, using that the functions $\hat{\phi}$ and τ are real, we find that

$$P(-\xi) = P(\xi)^* = \overline{\phi}\,\mathbb{1} + \overline{v_j}\,\gamma^j\,.$$

Thus, omitting the argument ξ, the closed chain (1.5) becomes

$$A \;=\; (\psi + \phi)(\overline{\psi} + \overline{\phi})\,.$$

For the computation of the spectrum, it is useful to decompose A in the form

$$A \;=\; A_2 + A_2 + \mu$$

with

$$A_1 \;=\; \frac{1}{2}\left[\psi, \overline{\psi}\right], \quad A_2 \;=\; \phi\,\overline{\psi} + \psi\,\overline{\phi}\,, \quad \mu \;=\; v_j \overline{v^j} + \phi\overline{\phi}\,.$$

A short calculation shows that the matrices A_1 and A_2 anti-commute, and thus

$$(A-\mu)^2 \;=\; A_1^2 + A_2^2 \;=\; D[A]\,\mathbb{1}\,, \tag{1.23}$$

where we set

$$D[A] \;=\; \frac{1}{4}\,\mathrm{tr}(A^2) - \frac{1}{16}\,(\mathrm{tr}A)^2 \;=\; (v_j\,\overline{v^j})^2 - |v_j\,v^j|^2 + (v_j\,\overline{\phi} + \phi\,\overline{v_j})\,(v^j\,\overline{\phi} + \phi\,\overline{v^j})\,. \tag{1.24}$$

The identity (1.23) shows that the characteristic polynomial of the matrix A has the two zeros

$$\lambda_\pm \;=\; v_j\,\overline{v^j} + \phi\overline{\phi} \pm \sqrt{D}\,. \tag{1.25}$$

If these two zeros are distinct, they both have multiplicity two. If the two zeros coincide, there is only one zero of multiplicity four. Hence the Lagrangian (1.6) simplifies to

$$\mathcal{L}[A] \;=\; (|\lambda_+| - |\lambda_-|)^2 \tag{1.26}$$

In order to further simplify the Lagrangian, we introduce a discrete causal structure, in agreement with [FIN4].

Definition 1.1: *A lattice point* $(t,r) \in \Omega$ *is called*

$$\begin{array}{ll} \textit{timelike} & \textit{if } D[A(t,r)] \;\geq\; 0 \\ \textit{spacelike} & \textit{if } D[A(t,r)] \;<\; 0\,. \end{array}$$

If (t,r) is spacelike, the $\lambda_\pm$ form a complex conjugate pair, and the Lagrangian (1.26) vanishes. If conversely the discriminant is non-negative, the $\lambda_\pm$ are both real. In this case, the calculation

$$\begin{aligned} \lambda_+\lambda_- \;&=\; (v\overline{v} + \phi\overline{\phi})^2 - \left[(v\overline{v})^2 - v^2\,\overline{v}^2 + (v\overline{\phi} + \phi\overline{v})^2\right] \\ &=\; 2\,(v\overline{v})\,|\phi|^2 + |\phi|^4 + v^2\,\overline{v}^2 - (v\overline{\phi} + \phi\overline{v})^2 \\ &=\; |\phi|^4 + v^2\,\overline{v}^2 - v^2\,\overline{\phi}^2 - \phi^2\,\overline{v}^2 \;=\; (v^2 - \phi^2)(\overline{v}^2 - \overline{\phi}^2) \;\geq\; 0 \end{aligned}$$

(where we omitted the tensor indices in an obvious way) shows that λ_+ and λ_- have the same sign, and so we can leave out the absolute values in (1.26). We conclude that

$$\mathcal{L}(t,r) \;=\; \begin{cases} 4D[A(t,r)] & \text{if } (t,r) \text{ is timelike} \\ 0 & \text{otherwise}\,, \end{cases}$$

where D is given by (1.24). Hence our Lagrangian is compatible with the discrete causal structure in the sense that it vanishes if (t,r) is spacelike.

Before we can set up the variational principle, we need to think about what the sum over the space-time points in (1.7) should correspond to in our lattice system. Since we are considering a homogeneous system, one of the sums simply gives a factor m, and we can leave out this sum. The other sum in the continuum should correspond to a space-time integral (see for example (1.8)). In our lattice system, the point (t,r) can be thought of as the 2-dimensional sphere $|\vec{\xi}| = r$ at time t. Therefore, we replace the spatial integral by a sum over the discretized radii, but with a weight factor which takes into account that the surfaces of the spheres grow quadratically in r. More precisely, we identify (t,r) with a shell of radius between $r - \Delta_r/2$ and $r + \Delta_r/2$. This leads us to the replacement rule

$$\int_{\mathbb{R}^3} d\vec{\xi} \cdots \;\longrightarrow\; \Delta_r^3 \sum_{n=0}^{N_r-1} \rho_r(n\Delta_r) \cdots$$

with the weight function ρ_r given by

$$\rho_r(n\Delta_r) \;=\; \frac{4\pi}{3} \cdot \begin{cases} 1/8 & \text{if } n = 0 \\ (n+1/2)^3 - (n-1/2)^3 & \text{if } n > 0\,. \end{cases} \tag{1.27}$$

When discretizing the time integral, we need to take into account that on the lattice $\mathfrak{L}$, the time parameter t is always non-negative. Since the Lagrangian is symmetric, $\mathcal{L}[A_{xy}] = \mathcal{L}[A_{yx}]$ (see [FIN2, §3.5]), this can be done simply by counting the lattice points with $t > 0$ twice. Thus we discretize the time integral by

$$\int_{-\infty}^{\infty} dt \cdots \;\longrightarrow\; \Delta_t \sum_{n=0}^{N_r-1} \rho_t(n\Delta_r) \cdots$$

with

$$\rho_t(n\Delta_t) \;=\; \begin{cases} 1 & \text{if } n = 0 \\ 2 & \text{if } n > 0\,. \end{cases} \tag{1.28}$$

Then the action becomes

$$\mathcal{S}[P] \;=\; \Delta_t\, \Delta_r^3 \sum_{(t,r)\in\mathfrak{L}} \rho_t(t)\, \rho_r(r)\, \mathcal{L}(t,r)\,. \tag{1.29}$$

Our variational principle is to minimise this action by varying the functions $\hat{\phi}$ and τ in (1.17, 1.18) under the constraints (TC) and (NC).

With the constructions of Sections 1.3 and 1.4 we successively derived our two-dimensional lattice model. Clearly, not all the arguments leading to the model were rigorous, and also we put in strong assumptions on the physical situation which we have in mind. More precisely, the main assumption was the spherically symmetric and static ansatz with a vector-scalar structure (1.10, 1.12); this ansatz was merely a matter of convenience and simplicity. Moreover, the choice of the weight function ρ involved some arbitrariness. However, we do not consider this to be critical because choosing the weight factors in (1.27) differently should not change the qualitative behavior of the model (except that for the existence of minimisers it is important that $\rho_r(0) \neq 0$; see Section 1.6). Finally, the normalization condition (NC) could be modified, as discussed in detail at the end of Section 1.3.

The main point of interest of our lattice model is that it allows to describe discretizations of Dirac Seas (1.22) but also completely different configurations of the fermions. Thus within the lattice model it should be possible to analyze in detail whether and how Dirac Sea configurations form as minimisers of our variational principle. Moreover, in our lattice model one can implement all the spherically symmetric regularization effects as found in [FIN5]. Hence our lattice model should make it possible to verify effects from [FIN5] coming from the discrete side and to analyze these effects in greater detail.

In the next section we shall define our lattice model once again more systematically, making the following simplifications:

- By scaling we can always arrange that Δ_ω and Δ_k have an arbitrary value. It is most convenient to choose
$$\Delta_\omega = 1, \qquad \Delta_k = 1.$$
Then
$$\Delta_t = \frac{2\pi}{N_t}, \qquad \Delta_r = \frac{2\pi}{N_r}.$$

- The formulas for P, (1.17, 1.18), only involve the two Dirac matrices γ^0 and γ^r, which satisfy the anti-commutation rules
$$(\gamma^0)^2 = \mathbb{1}, \quad (\gamma^r)^2 = -\mathbb{1}, \quad \left\{\gamma^0, \gamma^r\right\} = 0.$$
Since these anti-commutation rules can be realized already by 2×2-matrices, we may simplify the matrix structure by the replacements
$$\gamma^0 \longrightarrow \sigma^3, \qquad \gamma^r \longrightarrow -i\sigma^1,$$
where σ^i are the usual Pauli matrices. Modifying the definition of the discriminant (1.24) to
$$D[A] = \frac{1}{2}\,\mathrm{tr}(A^2) - \frac{1}{4}\,(\mathrm{tr}A)^2$$

(where now "tr" clearly denotes the trace of a 2×2-matrix), the Lagrangian remains unchanged.

- In order to simplify the normalization condition (NC), it is convenient to introduce the function
$$\Phi(k,\omega) \;=\; 2k\,\hat{\phi}(k,\omega)\,,$$
where for notational simplicity we also omitted the tilde.
- In order to simplify the prefactors, we multiply P by 4π, divide the Lagrangian by four, and divide the action by $(2\pi)^4\,\pi/6$. Furthermore, we multiply f_{loc} by a factor of π^3.

1.5 Definition of the Model and Basic Properties

For given integer parameters N_t, N_r and p we introduce the lattice $\mathfrak{L}$ and its dual lattice $\hat{\mathfrak{L}}$,

$$\begin{aligned}(t,r) \;\in\; \mathfrak{L} \;&=\; \Big\{0,\,\frac{2\pi}{N_t},\,\ldots,\,2\pi\,\frac{N_t-1}{N_t}\Big\}\times\Big\{0,\,\frac{2\pi}{N_r},\ldots,2\pi\,\frac{N_r-1}{N_r}\Big\}\\ (\omega,k) \;\in\; \hat{\mathfrak{L}} \;&=\; \Big\{-(N_t-1),\,\ldots,\,-1,\,0\Big\}\times\Big\{1,\,\ldots,\,N_r\Big\}\,.\end{aligned}$$

On $\hat{\mathfrak{L}}$ we choose a non-negative function Φ and a real function τ, which vanish except at p lattice points. We set

$$\begin{aligned}P(\xi) \;&=\; \frac{1}{r}\sum_{(\omega,k)\in\hat{\mathfrak{L}}} e^{i\omega t}\,\Phi\,\Big[(\mathbb{1}-\sigma^3\,\cosh\tau)\,\sin(kr)\\ &\qquad\qquad +\sigma^1\,\sinh\tau\,\Big(\cos(kr)-\frac{\sin(kr)}{kr}\Big)\Big], \qquad \text{if } r\neq 0 \quad (1.30)\\ P(t,r=0) \;&=\; \sum_{(\omega,k)\in\hat{\mathfrak{L}}} e^{i\omega t}\,k\,\Phi\,(\mathbb{1}-\sigma^3\,\cosh\tau)\,, \qquad (1.31)\end{aligned}$$

where σ^1 and σ^3 are two Pauli matrices. For any $(t,r)\in\mathfrak{L}$ we introduce the closed chain $A(t,r)$ by

$$A(t,r) \;=\; P(t,r)\,P(t,r)^*\,,$$

where the adjoint with respect to the spin scalar product is given by

$$P(t,r)^* \;=\; \sigma^3\,P(t,r)^\dagger\,\sigma^3\,,$$

and the dagger denotes transposition and complex conjugation. We define the discriminant $D[A]$ and the Lagrangian $\mathcal{L}[A]$ by

$$\begin{aligned}D[A] \;&=\; \frac{1}{2}\,\mathrm{tr}(A^2)-\frac{1}{4}\,(\mathrm{tr}A)^2 \qquad (1.32)\\ \mathcal{L}[A] \;&=\; D[A]\,\Theta(D[A])\,, \qquad (1.33)\end{aligned}$$

where Θ is the Heaviside function. The action is

$$\mathcal{S} = \frac{1}{N_t N_r^3} \sum_{(t,r)\in\mathfrak{Q}} \rho_t(t)\, \rho_r(r)\, \mathcal{L}[A(t,r)]\,,$$

where ρ_t and ρ_r are the weight functions

$$\rho_t\left(2\pi\,\frac{n}{N_t}\right) = \begin{cases} 1 & \text{if } n = 0 \\ 2 & \text{if } n > 0 \end{cases}$$

$$\rho_r\left(2\pi\,\frac{n}{N_r}\right) = \begin{cases} 1 & \text{if } n = 0 \\ (2n+1)^3 - (2n-1)^3 & \text{if } n > 0\,. \end{cases}$$

Our variational principle is to minimise the action, varying the functions Φ and τ under the following constraints:

(TC) The local trace

$$f_{\text{loc}} := \sum_{(\omega,k)\in\hat{\mathfrak{Q}}} k\, \Phi(\omega,k) \tag{1.34}$$

should be kept fixed.

(NC) The function Φ should only take the two values $\Phi(\omega,k) = 0$ or $\Phi(\omega,k) = 1$.

The last condition (NC) could be weakened or left out (see the discussion at the end of Section 1.3).

According to Definition 1.1, the functions Φ and τ induce on $\mathfrak{Q}$ a *discrete causal structure*. The Lagrangian is compatible with the discrete causal structure in the sense that it vanishes if (t,r) is spacelike. Furthermore, our lattice system has the following symmetries:

symmetry under parity transformations: The traces in (1.32) vanish unless an even number of matrices σ^1 appears. Therefore, the Lagrangian remains unchanged if the factor $\sinh\tau$ in (1.30) flips sign. Hence the action is symmetric under the transformation

$$\tau(\omega,k) \longrightarrow -\tau(\omega,k) \qquad \text{for all } (\omega,k) \in \hat{\mathfrak{Q}}. \tag{1.35}$$

This transformation changes the sign of the spatial component of P. The name "parity transformation" comes from the analogy to the usual parity transformation $\vec{x} \to -\vec{x}$.

gauge symmetry: We introduce on the dual lattice $\hat{\mathfrak{Q}}$ for any $\Omega \in \mathbb{Z}$ the translation respecting the periodic boundary conditions

$$\omega \longrightarrow \tilde{\omega} = (\omega + \Omega) \bmod N_t \tag{1.36}$$

and also translate the functions τ and Φ by setting

$$\tilde{\tau}(\tilde{\omega}, k) \;=\; \tau(\omega, k)\,, \quad \tilde{\Phi}(\tilde{\omega}, k) \;=\; \Phi(\omega, k)\,.$$

This translation in momentum space corresponds to a multiplication by a phase factor in position space,

$$\tilde{P}(\xi) \;=\; e^{i\Omega t}\, P(\xi)\,.$$

This phase factor drops out when forming the closed chain, and thus the Lagrangian remains unchanged. The transformation (1.36) are precisely those local gauge transformations which are compatible with our spherically symmetric and static ansatz.

1.6 Existence of Minimisers

In this section we prove an existence result, which is so general that it applies also in the case when the normalization condition (NC) is weakened.

Proposition 1.2: *Consider the variational principle of Section 1.5 with the trace condition (TC) and, instead of (NC), the weaker condition that that there is a parameter $\varepsilon > 0$ such that*

$$\Phi(\omega, k) \;=\; 0 \quad \text{or} \quad \Phi(\omega, k) \;>\; \varepsilon \qquad \text{for all } (\omega, k) \in \hat{\mathfrak{Q}}\,.$$

Then the minimum of the action is attained.

Proof. Since the Lagrangian is non-negative, we can estimate the action from above by the Lagrangian at the origin $t = r = 0$,

$$\mathcal{S} \;\geq\; \mathcal{L}[A(0,0)]\,. \tag{1.37}$$

At the origin, the Fermionic Projector takes the form (see (1.31))

$$P(0) \;=\; \sum_{(\omega,k)\in\hat{\mathfrak{Q}}} k\, \Phi\, (\mathbb{1} - \sigma^3\, \cosh\tau)\,.$$

This matrix can be diagonalized and has the two eigenvalues

$$\mu_\pm \;=\; \sum_{(\omega,k)\in\hat{\mathfrak{Q}}} k\, \Phi\, (1 \pm \cosh\tau)\,.$$

Thus the closed chain $A(0,0)$ has the two eigenvalues $\lambda_\pm = \mu_\pm^2$. As a consequence, using (1.32) and (1.34),

$$\begin{aligned} \mathcal{L}[A(0,0)] \;&=\; \frac{1}{4}\,(\lambda_+ - \lambda_-)^2 \;=\; \frac{1}{4}\,(\mu_+ + \mu_-)^2\,(\mu_+ - \mu_-)^2 \\ &=\; 4\, f_{\text{loc}}^2 \left(\sum_{(\omega,k)\in\hat{\mathfrak{Q}}} k\, \Phi\, \cosh\tau \right)^2 . \end{aligned} \tag{1.38}$$

Consider a minimal sequence. Then, according to (1.37), the expression (1.38) is uniformly bounded. If $f_{\text{loc}} = 0$, our system is trivial, and thus we may assume that f_{loc} is a positive constant. Using (1.34) and the fact that $k \geq 1$, we conclude that the functions Φ are uniformly bounded. The boundedness of (1.38) implies that there is a constant $C > 0$ such that

$$\sum_{(\omega,k)\in\hat{\Omega}} k\, \Phi\, \cosh\tau \;\leq\; C$$

for all elements of the minimal sequence. Whenever Φ vanishes, we can also set τ equal to zero. If Φ is non-zero, the inequality $\Phi \geq \varepsilon$ gives a uniform upper bound for $\cosh\tau$,

$$\cosh\tau \;\leq\; \frac{C}{\varepsilon}\,.$$

We conclude that the functions Φ and τ are uniformly bounded. Hence a compactness argument allows us to choose a convergent subsequence. Since our action is obviously continuous, the limit is the desired minimiser. □

We point out that this proposition makes no statement on uniqueness. There seems no reason why the minimisers should be unique. In Chapter 4 we shall see examples with several minimisers.

1.7 Conclusion

At the end of this chapter, we have to reflect on the general purpose of our model. The variational principle comprises the minimisation of the action (**??**) by varying the functions $\hat{\phi}$ and τ in (1.5) under the constraints (TC) and (NC). The way we derived the model linked to this action is not unique and rigidly determined. So we made some heristic assumptions to derive a well defined and numerically sufficient simple model of a fermionic vacuum system: (1) We assumed spherical symmetry in (1.12) and (1.10) for the matter of convenience and simplicity. (2) We justified the normalisation of the "effective" wave functions by the request of a most simple transformation from the continuous case. This was the origin of the normalisation (NC). (3) The choice of the weight functions ρ_r and ρ_t was done by some heuristics which doesn't come along with strict necessity.

The class of Fermionic Projectors described by (1.10) and (1.12) can now be studied on a discrete lattice model. This class includes Fermionic Projectors quite similar to the continuous Fermionic Projector of the vacuum, but also complete different ones. This enables us to analyse whether or not the action (**??**) prefers Dirac Sea like configurations as minimisers and how these minimisers look like.

The lattice model as developed in SectionS 1.1 to 1.6 as well as its presentation is identical to that one published in [FIN/PLA]. Nevertheless, throughout this thesis the model was

implemented with a slightly modifications. The programming code does not adopt the dropping of the prefactor $\frac{1}{(4\pi)^3}$ of the Fermionic Projector in Section 1.4.[1]

[1]The factor $\frac{1}{(4\pi)^3}$ was implemented due to a prior formulation of the model and profiling showed, that it's contribution to the numerical efford is negligible.

Chapter 2

The Numerical Challenge of Optimisation

While chapter 1 deals with the definition of the model analysed in this thesis numerically, this chapter has a purely numerical scope. From this point of view, the calculation of an action has to be considered as a blackbox

$$S = S(\omega, \tau) \tag{2.1}$$

only featured with the dependence on certain variables, possibly attached with some constraint conditions and a quite vague term of "runtime" belonging to the numerical calculation of the action.

2.1 General Introduction into the Problem of Mixed Integer Nonlinear Programming

Mixed-integer non linear programming problems (MINLP) is one of the broadest classes of optimisation problems. (For a general overview see [KAL] and [ABR1] and the references there.) Keeping the integer variables fixed, MINLP problems become simply non-linear problems (NLP) for which solving strategies as the gradient methods or Newton methods are well established (see [ALT]). The class of optimisation problems that also includes categorical variables is called *mixed variable programming* (MVP). Categorical variables are those that can only take values from a predefined list and thus have no ordering relationship. By mapping categorical variable injectively to a set of integers, one can transform a MVP problem into a MINLP problem, though the values do not conform to the inherent ordering, that the integer values come along with.

In practice the most conventional approach to MINLP problems with only a few discrete variables that can take only a few values is to enumerate the discrete variables exhaustively and to solve a series of NLP problems. The scope of this approach is very limited, since the costs usually raise for combinatorial reasons at least exponentially depending on the number of discrete variables. In the following we will survey three general, more sophisticated approaches to the topic of MINLP: *relaxation methods*, *search heuristics* and *pattern search methods*. (The main reference for this chapter are [ABR1] and [GRO], which are also recommended for further readings to this topics as well as the references there.)

2.1.1 Relaxation Methods

Relaxation methods are optimisation methods for MINLP problems that require relaxation of the objective function. For instance one has an objective which evaluates the total costs of a product depending of the number of workpieces of a special kind. Then in case of a polynomial objective, which is a reasonable assumption, it is exploitable for any real "number" of workpieces too. Hence only for the optimal solution the number of workpieces has to be integer and it can be real during the whole optimisation process. A further property of the relaxation methods reviewed here is, that they all make use of any derivative information for the NLP subproblems, hence differentiability of the objective while fixing the integer variables is required.

The first method to mention is *outer approximation* (OA). This method only applies to a very special class of MINLP problems, since the objective has to be of the form

$$f \;:\; R \times M \to \mathbb{R}, \qquad f(x,m) = g(x) + c^T m \tag{2.2}$$

where $R \subseteq \mathbb{R}^n$, $m \in M \subseteq \{0,1\}^p$, $p \in \mathbb{N}$, $g : R \to \mathbb{R}$ and $c \in \mathbb{R}^p$. Beside other restrictions and R has to be convex and compact, g as well as the constraint function has to be convex and continuous differentiable. The OA algorithm solves alternately two kinds of subproblems: First $m \in M$ is fixed and the MINLP problem reduces to a ordinary NLP problem, the so called *primal program*. After solving this with solution x' one gets an upper bound for the main problem. The convexity and the differentiability yield, that f and the constraint mapping can be approximated downward (in the "outer" region in relation to the convex graph of g) linearly by using the gradient. From this the second MINLP subproblem is obtained and solved, the so called *master program*, yielding also a lower bound for the objective. The obtained solution (x'', m') is now taken as the starting point for a further *primal program*, and so on. It can be shown, that the upper and the lower bound coincide after a finite number of iterations. *Outer approximation* was first introduced by Duran and Grossman in 1986 (see [DUR], and was extended to a broader

class of problems by Fletcher and Leyffer in 1996 (see [FLE]), whereas nevertheless the convexity conditions is needed.

Generalised Benders Decomposition, developed by Geoffrin (see [GEO]), works quite similar to the AO method, solves the same NLP primal program but a different MILP master program, which is obtained by linearising the Lagrangian function $L(x, \lambda) = f(x) + \lambda^T C(x)$ around the current point, where C denotes the constaint mapping and $\lambda \in \mathbb{R}^p$ is the vector of Lagrangian multipliers.

Branch and Bound Methods were originally developed for MINLP problems by Dakin (see [DAK]) and were generalized to non-linear problems by Gupta and Ravindran ([GUP]), Nabar and Schrage ([NAB]), Borchers and Mitchell ([BOR]), Stubbs and Mehrotra ([STU]) and Leyffer ([LEY]). This method starts with solving a relaxed MLP problem. If all discrete variables take integer values the search is ended. Otherwise a binary tree search is performed in the space of the integer variables by a implicit enumeration, where a subset of discrete variables is fixed at each node. For instance, if the NLP search yields for a discrete variable m the solution $m = 3.5$, then two branching problems emanate from that node by solving the NLP problem two times again, once adding the further constraint $m \leq 3$, once adding $m \geq 4$. Several efficiencies such as upper bounds for the solution are added to prevent testing unnecessary nodes.

The *Extended Cutting Plane Method*, introduced by Westerlund and Petterson (see [WES]) extended the Kelleys Cutting Plane algorithm (see [KEL]) for convex NLP to MINLP. It does not make use of solving a NLP subproblem, but generates a non-decreasing sequence of lower bounds by solving a sequence of MINLP problems, successively adding a linearisation of the most violated constraint at the previous previous suboptimal point. The algorithm terminates when the maximum constraint violation falls below a user specified tolerance.

The presented methods for solving MINLP problems have drawbacks that preclude their use for our optimisation problem. First the action cannot be assumed to be convex, a condition mostly required to achive convergence results. Although there exist heuristics to overcome the problem of non-convexity, no satisfying convergence theory has been developed for non-convex problems. Second, methods that linearise objectives make use of first-order Taylor Series, which requires differentiability of this function. Formally, this cannot be assumed in our problem, since absolute values in the Lagrangian (1.26) occur.

2.1.2 Search Heuristics

Search heuristics are weak optimisation methods in different ways: First they often do not make any use of derivative information and corresponding to this, they secondly do in general not come along with a strict local convergence theory. The best to achieve is

often convergence in a probabilistic sense. We will see that the most Search Heuristics are inspired by Physical of biological processes. (For a good overview to physical search heuristics see [HAR].) In spite to the theoretical point of view in practice the lack of convergence theory entails the advantage that according to experience these methods are numerically cheaper than deterministic methods. We will see in conjunction with discussing the GPS methods, that strict convergence results are obtained with the cost of raising numerical effort. Hence search heuristics are always a favourable choice when the effort of strict methods is excessively high or no strict methods are available, in particular at any kind of combinatorial optimisation where no polynomial-time algorithms appear to be at hand.

Most search heuristics also make use of some kind of stochastic process during the optimisation routine. Stochastic processes are indeed essential for two methods, that we will mention here: *Simulated Annealing*, a modified Monte Carlo search technique, and *Evolutionary Algorithms*. In contrast to these *Tabu Search* denotes a metaheuristical framework to enhance trajectory oriented search algorithms by the use of memory techniques, i.e. in principal the information of all previous iterations is exploited for finding the next improved iterate. This method holds for any kind of optimisation process

A widely practised method in particular for optimisation problems with many minima is *Simulated Annealing*. This method works for all kinds of optimisation problems and was first introduced by Metropolis *et al.* in 1953 (see [MET], for an example application for the travelling salesman problem see [SAN]). The special purpose of this method is to determine the ground configuration of a thermodynamical system at $T = 0$. At each step a randomly selected alternative point in the configuration space is evaluated by its temperature or action function. The essential point is that the algorithm also accepts pejorations of the action function with a certain probability due to the Boltzmann statistics

$$p(x \to x') = \begin{cases} 1 & \text{for} \quad \Delta\mathcal{H} \leq 0 \\ e^{-\frac{\Delta\mathcal{H}}{k_B T}} & \text{for} \quad \Delta\mathcal{H} > 0. \end{cases} \tag{2.3}$$

This helps the algorithm (see listing 2.1) to leave local minima towards more global ones. The crux of the matter is to find an appropriate *cooling schedule* $T(t)$ (whereas t denotes the computer time, e.g. number of main iterations) With the help of the theory of Markov processes the existence of cooling schedules has been shown, which guarantee for a finite problem the convergence to the global optimum. These cooling schedules have the form

$$T(t) = \frac{a}{b + \log(t)}, \tag{2.4}$$

where $a, b \in \mathbb{R}^+$ depend on the problem. Since the logarithmic dependency of t yields a quite slow cooling, which might be undesirable, there is in practice made use of other

cooling schedules, i.e. the *linear schedule* and the *exponential schedule*

$$\begin{aligned} T(t) &= a - bz && \text{with usually} \quad 0.01 \leq b \leq 0.2 \\ T(t) &= ab^t && \text{with usually} \quad 0.8 \leq b \leq 0.99, \end{aligned}$$

where $t \in \mathbb{N}^+$, a is the initial temperature and b is thestep size or cooling rate respectively.

Listing 2.1: A simple Simulated Annealing Algorithm

```
random choice of initial point x ∈ ℝ^n
while T > T_min do
    random choice of neighbor y of x
    if E(y) < E(x)
        x = y
    else
        x = y with probability e^(-(E(y)-E(x))/T)
        decrease T slightly (according to the
        cooling schedule)
    end
end
```

Although Simulated Annealing does in general not make use of derivative information, there are methods in use, which combine Simulated Annealing with gradient oriented methods, for instance [PLB, p. 74 et sqq.] demonstrates a combination of the Nelder and Meat-Algorithm (there called Downhill Simplex method) with simulated annealing with the purpose to force the algorithm to deliver more global solutions.

Another variation of Simulated Annealing is called *Parallel Tempering*, introduced by Hukushima and Nemoto (see [HUK]). This derivation of Simulated Annealing is specially aimed for systems with very low temperature dynamics, i.e. the structure of the problem is of such a kind that statistical trial points can only be exploited in the neighbourhood of the actual iterate. These systems show a dynamical behaviour similar to glasses. The general concept of parallel tempering is to perform several simulations parallely, which are all assembled by the same system but run at different temperatures. In the easiest case, where there are two systems, the algorithm interchanges the configurations of the two systems with the purpose that the higher temperated simulation helps the – previous – lower temperated configuration to overcome energy barriers.

The adaptive memory feature of *Tabu Search* aims to enhance economy of a local search process. It was introduced by Glover (see [GLO1], [GLO2], [GLO3] and [TAB]) as a metaheuristic for solving combinatorial optimisation problems. The main idea is to perform searches among a user defined set of neighbours $N(x_i)$ with x_i the position.of the i-th

step. Based on the actual choice, the *tabu list* is updated under special rules. Depending on the application *tabu strategy*, for instance the complement of a step is forbidden for a certain number of iterations. To obtain global solutions each local minimum is put on the tabu list and the algorithm moves to another point in an area of the domain that has not been searched yet.

The next class of methods spread a wide range of algorithms subsumed under the label of *evolutionary algorithms* (see [RUS]), namely *Evolution Strategies*, *Evolutionary Programming* and *Genetic Algorithms*. Since only the latter ones are aimed to solve discrete optimisation problems as well (the other two were designed primarily for continuous problems), we will focus in the discussion on this method only. In contrast to the preceeding methods, evolutionary algorithms are not *trajectory oriented*. I.e they do not calculate in each iteration one improved point from the latest forerunner, but at each iteration evolutionary algorithms deal simultaneously with finitely many points, called a *population*. In analogy to the biological concept of evolution, the main processes contained by evolutionary algorithms are:

- **Selection:** Process by which parents are selected for reproduction
- **Reproduction:** Process by which some kind of ”genetic” information is passed from parents to children
- **Mutation:** Random errors occur in the reproduction process
- **Competition:** Process by which children survive to the next generation

Listing 2.2: A simple Generic Algorithm

```
1  initialize population x_1, x_2, ..., x_M
2  select parents p_1, ..., p_m
3  for t := 1 to n_R
4     begin
5        create offsprings c_1, ..., c_n via crossover
6        perform mutations
7        eventually perform local optimisation
8        calculate fitness values for all crossovers
9        select new parents p_1, ..., p_{m_t}
10    end
11 end
```

Each point or *individual* in the population is evaluated by the objective function, which is called in the context of evolutionary algorithms *fitness function*. The main idea is that at each iteration a set of points, i.e. a population is generated from a subset of the forerunner population. Before this reproduction takes place, *selection* takes care that only the fittest

members of the population are allowed to reproduce. From this set of parents the new generation is created via crossover. The process creating crossovers varies, for instance special linear combinations of parent points are set up, optionally created with the help of some random variation process. Listing 2.2 shows a simple framework for genetic algorithms.

2.2 Intermediate Conclusion

The choice of a proper optimisation method is always hard and often only reasoned by the result. In our case the main problem of this work was *not* the implementation of the optimisation algorithm or the coding of the action evaluation, but to find a proper definition of the model as worked out in Chapter 1 and furthermore to find some subtle modifications of the model discussed below, not until which satisfying results where found. Furthermore, it was necessary to have an optimisation method implemented in a early state of our work to find the modifications mentioned before.

Up to the present day, MINLP is one of the most difficult tasks in numerical optimisation. Thus from a practical point of view, it is important to consider the latest developments in this area. Considerable advantages have recently been made concerning Generalised Pattern Search Methods. The main advantage of the class of GPS algorithms is that they apply to a broader range of problems compared to the relaxation methods discussed in subsection 2.1.1 and that GPS algorithms – in contrast to the heuristic methods discussed in subsection 2.1.2 – contain a quite elaborated convergence theory. Furthermore Search Heuristics are quite powerful in producing improvements to a given design, but usually perform the worse the closer the search algorithm has come to a relatively global minimum.

2.3 Generalised Pattern Search Methods [1]

The *Generalised Pattern Search* (GPS) labels a class of algorithms which are appropriate for different sophisticated optimisation problems. They are all extentions of the original GPS algorithm introduced by Torczon ([TOR]). Each iteration includes an obligatory POLL step and an optional global SEARCH step. In the SEARCH step, which is carried out first, the objective function f is evaluated at a finite number of points laying on an iteration dependend mesh M_k. for the purpose of minimising f. The choice of search

[1] The main reference for this class of algorithms is Mark Abramsons Thesis [ABR1] and many parts of this subsection are based on his text. For further information see also the Web page of the NOMADm software [NOMADm].

strategies is only restricted to the condition that it has to be finite, i.e. carried out on finite number of mesh points.

2.3.1 Positive Spanning Sets

The following terminology and theorem is due to [DAV]:

Definition 2.1: *A **positive combination** of the set of vectors $\{v_i\}_{i=1}^r$ is called a linear combination $\sum_{i=1}^r \alpha_i v_i$, where $\alpha_i \geq 0$, for $i = 1, 2, ..., r$*

Definition 2.2: *A finite set of vectors $W = \{w_i\}$ forms a **positive spanning set for** $\mathbb{R}^n$ if every vector $v \in \mathbb{R}^n$ can be expressed as a positive combination of vectors in W. The set of vectors W is said to **positively span** $\mathbb{R}^n$. The set W is said to be a **positive basis** for $\mathbb{R}^n$ if no proper subset of W positively spans $\mathbb{R}^n$.*

Davies [DAV] shows that for any positive basis $B \subset \mathbb{R}^n$, $1 \leq \sharp B \leq 2n$. I.e $[-\mathbb{1}, \mathbb{1}]$ and $[-\mathbb{1}, e]$ (with $e_i = 1 \; \forall \, 1 \leq i \leq n$) as well as for any basis V the columns of $[V, -V]$ and $[V, -Ve]$.

Theorem 1: *A set D positively spans $\mathbb{R}^n$ if and only if for all nonzero $v \in \mathbb{R}$ $v^t d > 0$ for some $d \in D$.*

2.3.2 The basic GPS Algorithm

In the POLL step, a positive spanning set $D_k \subseteq D$ is determined, from which the *poll set* is constructed, which may be represented by a matrix $D_k = D(k, x_k)$ whose columns are the member of the set. The *poll set* is defined as

$$P_k = \{x_k + \Delta_k d \in X \mid d \in D_k\}, \tag{2.5}$$

i.e.the poll set P_k is constructed as the neighbouring mesh points in each direction in D_k. f is evaluated at the points in P_k, either until the points have all been evaluated or until one with a lower function value is found.

If S_k is a finite set of mesh points evaluated during the SEARCH step, the set of trial points is defined as $T_k = S_k \cup P_k$. Each pair of SEARCH and POLL steps has two possible results.

Definition 2.3: *If $f(y) < f(x_k)$ for some y in T_k, then y is said to be an **improved mesh point.***

Definition 2.4: *If $f(x_k) \leq f(y)$ for all $y \in P_k$ then x_k is said to be a **mesh local optimiser.***

Listing 2.3: Basic GPS Algorithm

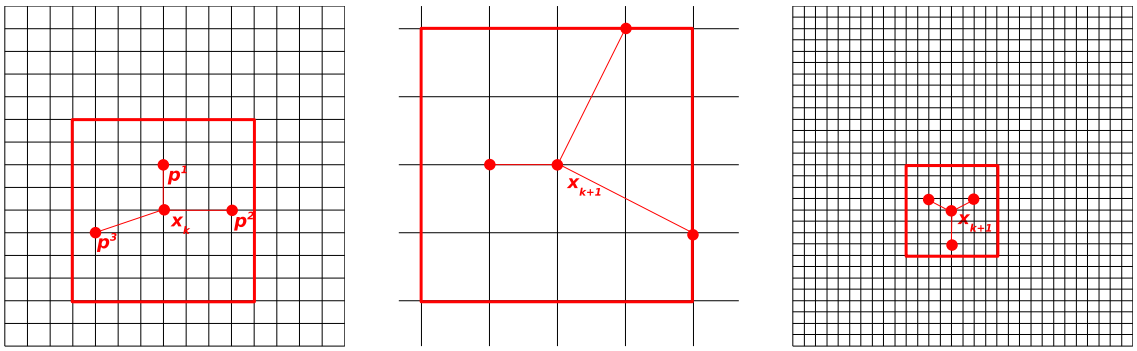
```
1  Initialisation: Let x_0 be so that f(x_0 is finite,
2  and let M_0 ∈ X be the mesh defined by Δ_0 > 0.
3
4  while ¬(break condition)
5          1. SEARCH step:
6          Employ some finite strategy
7          seeking an improved mesh point; i.e.
8          x_{k+1} ∈ M_k such that f(x_{k+1} < f(x_k).
9
10         2. POLL step:
11         If SEARCH step was unsuccessful, evaluate
12         f at points in the poll set P_k until
13         an improved mesh point x_{k+1} is found
14         (or until done).
15
16         3. UPDATE: If SEARCH or POLL finds an
17         improved mesh point, update x_k + 1 and set
18         Δ_k + 1 ≥ Δ_k according to (2.6).
19         Otherwise set x_{k+1} = x_k and set
20         Δ_k + 1 < Δ_k according to (2.7)
21
22         k = k+1
23 end
```

If either the SEARCH or the POLL step lead to an improved mesh point, it becomes the new incumbent x_{k+1}. In this case it is said that the iteration step was "successfull". In fact, repeated "unsuccessful iterations" lead to numerical convergence. To understand this, we have to take a look at the coarsening and refining rules.

The first states that if any $y \in T_k$ is a improved mesh point, then

$$\Delta_{k+1} = \tau^{m_k^+} \Delta_k, \tag{2.6}$$

where $\tau > 1$ is rational and fixed over all iterations and the integer m_k^+ is in the range $0 \leq m_k^+ \leq m_{max}$ for some fixed integer $m_{max} > 0$. The purpose of the coarensing rule is to allow the algorithm to skip over certain local minima and find a more global solution. It does not oppress convergence of the algorithm and makes it faster (acc. to [ABR1, p. 46]).



(a) Evaluating the Poll Set (b) Successfull Poll Step (c) Unsuccessful Poll Step

Figure 2.1: Poll Step of the Basic GPS Algorithm

If in the SEARCH and the POLL step no improved mesh point is found, the actual mesh point is according to Definition 2.4 a mesh local optimiser and remains unchanged. In this case the mesh is refined according to the rule

$$\Delta_{k+1} = \tau^{m_k^-} \Delta_k \tag{2.7}$$

where $\tau > 1$ is defined as in the context of formula 2.4, $\tau^{m_k^-} \in (0, 1)$, and the integer m_k^- fulfils the condition $m_{\min} \leq m_k^- \leq -1$ for one fixed integer $m_{\min}$. We will not discuss convergence results for the basic GPS algorithm, since we will mainly use a more sophisticated variant of the GPS algorithms.

2.3.3 GPS for MVP Problems

We will now discuss the class of Mixed Variable Generalised Pattern Search (MGPS) algorithms, which was mainly developed by Audet and Dennis [AD1]. The term "Mixed

Variable“ in this context means that beside continuous variables, there are discrete and in the more general sense categorical variables given, on which the objective depends. Categorical variables are often assigned numeric values, but these values are in general assumed to be not ”relaxable“. That is the objective cannot be embedded to a fully continuous function in a canonical way. Although in our problem this condition might be weakened, since all formulas necessary for the calculation of the action can also be evaluated for non discrete values of ω, it might happen that a naive relaxation leads to many solutions which are local minima only in the relaxed case. After all the discrete portion of the variables is called *categorical variables* and can be regarded to take values only from a predefined list.

The variables of mixed variable problems can naturally be divided into two parts: $x = (x^c, x^d) \in X = X^c \times X^d \subset \mathbb{R}^{n^c} \times \mathbb{Z}^{n^d}$. The continuous variable space is defined by a finite set of linear constraints, depending on x^d:

$$X^c(x^d) = \left\{ x^c \in \mathbb{R}^{n^c} \mid \ell(x^d). \leq A(x^d)x^c. \leq u(x^d) \right\} \tag{2.8}$$

with $A(x^d) \in \mathbb{R}^{n^c \times n^c}$ a real matrix, $\ell(x^d), u(x^d) \in (\mathbb{R} \cup \{\pm\infty\})^{n^c}$ and $\ell(x^d). \leq u(x^d)$ for all values of x^d. (In the further thesis we will omit the dependence on x^d since it is not relevant for our purpose.)

Mesh Construction and Local Optimality

To include the handling of discrete variables in the framework of the GPS algorithm, the definition of the mesh must be generalised in a way that the case of the basic GPS algorithm is included in the case of fixing the discrete variables.[2]

For each combination $i = 1, ..., i_{\max}$ of variables that the discrete variables may take, a set of positive spanning directions D^i (with δ to be the number of columns) is constructed via

$$D^i = G_i Z_i, \tag{2.9}$$

with $G_i \subset \mathbb{R}^{n^c \times n^c}$ a nonsingular generating matrix and $Z_i \in \mathbb{Z}^{n^c \times \delta}$. (We will sometimes write $D(x)$ instead of D^i to indicate that the set of directions is associated with the discrete variable values of $x \in X$.) We further want to define the set $D := \bigcup_{i_1}^{i_{\max}} D^i$.

We then define a mesh formed as the direct product of X^d with the union of a finite number of lattices in X^c:

$$M_k := X^d \times \bigcup_{i=0}^{i_{\max}} \left\{ x_k^c + \Delta_k D^i z \in X^c \mid z \in (\mathbb{Z}^+)^\delta \right\}. \tag{2.10}$$

[2] For more details concerning this paragraph see [ABR1, chap. 5].

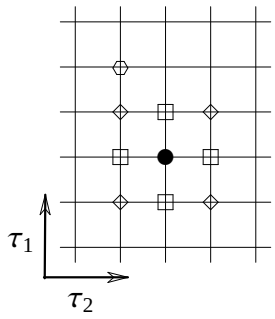


Figure 2.2: Different pseudo-topologies on a discrete lattice

This mesh is purely conceptual and is never explicitly created. Directions are only generated if the algorithm requires them.

For usual NLP problems, the common topology implicates a unique notion of local optimality. But in the discrete case of a lattice, there is no canonical analogon to the continuous notion of topology. For example if one looks at naive transformations of continuous neighbourhoods of the L_1- and L_∞-norms (marked by quadrats and diamonds around a point symbolised by the filled out circle in figure (2.2)), then we see that these would generate the discrete topology, which is of no use in our undertaking, since in this case every point would be a minimum. To get a discrete substitute of a topology we define

Definition 2.5: *For any set X we call a set valued function $\mathcal{N} : x \to \mathfrak{P}(X)$ a **pseudo-topology**, if for all $x \in X$ we have $x \in \mathcal{N}(x)$ and $\mathcal{N}(x)$ finite.*

In contrast to the continuous case, where the standard topologies are equivalent, in the discrete case no unique definition of a minimum can be given. Therefore every numerically determined minimum comes along with a certain insecurity, not only because of usual numerical insecurities, but also since the notion of minimum we choose depends on the pseudo-topologies indicated by figure (2.2)). For example a minimum determined by the pseudo L_1-Norm may not be one in the Pseudo-L_∞-norm, and one in this norm maybe voided if points one marked by the hexagon come into consideration.

Especially for the discussion of the convergence results we also need a definition of the term ”limit“.

Definition 2.6: *Let $X \subseteq \mathbb{R}^{n^c} \times \mathbb{Z}^{n^d}$ be a mixed variable domain. A sequence $(x_i)_{i=1}^\infty \subset X$ is said to **converge** if for every $\varepsilon > 0$ there exists a positive integer N such that $x_i^d = x^d$ and $\|x_i^c - x^c\| < \varepsilon$ for all $i > N$. The point x is said to be the **limit point** of the sequence $(x_i)_{i=1}^\infty \subset X$.*

We further meet a definition which is of technical importance in proving some convergence results

Definition 2.7: *The set-valued function* $\mathcal{N} : X \subseteq \mathbb{R}^{n^c} \times \mathbb{Z}^{m^d} \to \mathfrak{P}(X)$ *is said to be* ***continuous*** *at* $x \in X$ *if for every* $\varepsilon > 0$ *there exists* $\delta > 0$ *such that, whenever* $u \in X$ *satisfies* $u^d = x^d$ *and* $\|u^c - x^c\| < \delta$*, the following two conditions hold:*

(a) If $y \in \mathcal{N}(x)$*, then there exists* $v \in \mathcal{N}(u)$ *satisfying* $v^d = y^d$ *and* $\|v^c - y^c\| < \varepsilon$

(b) If $v \in \mathcal{N}(u)$*, then there exists* $y \in \mathcal{N}(x)$ *satisfying* $y^d = v^d$ *and* $\|y^d - v^c\| < \varepsilon$*.*

One common choice of pseudo-topologies $\mathcal{N}$, which we will use throughout this work, is given by

$$\mathcal{N} = \{y \in X^d \mid \|y - x\|_1 \leq 1\}. \tag{2.11}$$

With our concept of pseudo-topologies as a notion for "neighbourhood" we can now extend the classical definition of local optimality to the case of mixed variables.

Definition 2.8: *A point* $x = (x^c, x^d) \in X$ *is said to be a* ***local minimiser of*** f ***with respect to the pseudo-topology*** $\mathcal{N}(x) \subset X$ ***if there exists an*** $\varepsilon > 0$ ***such that*** $f(x) \leq f(v)$ ***for all*** v ***in the set***

$$X \cap \bigcup_{y \in \mathcal{N}} B_\varepsilon(y^c) \times y^d. \tag{2.12}$$

The MGPS Algorithm

Like the basic GPS algorithm, the MGPS algorithm implies a SEARCH and possibly a POLL step. The SEARCH step is almost identical to that one for the basic GPS algorithm. The only difference is that the applied search heuristic has now to deal with additional discrete variables.

The POLL step in the MGPS algorithm is performed with respect to three different sets: the poll set concerning the continuous variables, the discrete neighbour points described by the pseudo-topology and a optional set of points generated by an EXTENDED POLL step.

Let $D_k^i \subseteq D$ denote the set of poll directions corresponding to the i-th set of discrete variable values and $D_k = \bigcup_{i=1}^{i_{\max}} D_i^k$. (Again we still often write D_k^i as $D_k(x_k)$ to indicate that the polling directions depend on the discrete variable value x_k.) We then define the *POLL set*

$$P_k(x_k) := \{x_k + \Delta_k(d, 0) \in X \mid d \in D_k(x_k)\}, \tag{2.13}$$

(where the notion $(d, 0)$ is in accordance with the separation $\mathbb{R}^{n^c} \times \mathbb{Z}^{n^d}$, thus $x_k + \Delta_k(d, 0) = (x_k^c + \Delta_k d, x_k^d)$).

To produce more global solutions, the MGPS algorithm offers the opportunity to perform an EXTENDED POLL step, in which additional polling is done around every point in the set $\mathcal{N}(x_k)$, whose objective function value is sufficiently close to the incumbent value, i.e. around every point of the set

$$\widetilde{\mathcal{N}}(x_k) = \{y \in \mathcal{N}(x_k) \mid f(x_k) \leq f(y) < f(x_k) + \xi_k\} \tag{2.14}$$

with a user specified parameter, the so called *extended poll trigger* $\xi_k \geq \xi$. One after one for the points in $\widetilde{\mathcal{N}}$, there is started a polling sequence $(y_k^j)_{j=1}^{j_k}$ with respect to the continuous neighbours of y_k beginning with $y_k^0 = y_k$ and ending either when

$$F(y_k^{J_k} + \Delta_k(d, 0)) < f(x_k) \quad \text{for some } d \in D_k(y_k^{J_k}) \tag{2.15}$$

or when

$$f(x_k) \leq f(y_k^{J_k} + \Delta_k(d, 0)) \quad \text{for all } d \in D_k(y_k^{J_k}). \tag{2.16}$$

(For this discussion we let $z_k = y_k^{J_k}$ the last iterate or *endpoint* of the EXTENDED POLL step.)

In practice, the parameter ξ_k is usually defined via a fraction parameter $0 < p < 1$ and the extended poll trigger ξ via

$$\xi_k = \max\{\xi, p|f(x_k)|\}, \tag{2.17}$$

and i.e. $p = 0.05$. The higher the choice of ξ and p respectively is, the more EXTENDED POLL steps will be generated and thus the resulting final solution will usually be more global, but by the cost of more required function evaluations.

For the set of extended poll points denoted as $\mathcal{E}$ and created for a discrete neighbour $y \in \mathcal{N}(x_k)$ by some polling strategy, we have

$$\mathcal{E}(y) \subseteq \{P_k(y_k^j)\}_{j=1}^{J_k}. \tag{2.18}$$

At iteration k, the set of points evaluated at the EXTENDED POLL step is then given by

$$\chi_k(\xi_k) = \bigcup_{y \in \mathcal{N}_k^{\xi}} \mathcal{E}(y), \tag{2.19}$$

with

$$\mathcal{N} := \{y \in \mathcal{N}(x_k) \mid f(x_k) \leq f(y) \leq f(x_k) + \chi_k\}. \tag{2.20}$$

The set of trial points of the GPS algorithm containing the SEARCH step as well as the EXTENDED POLL step is defined as

$$T_k := S_k \cup P_k(x_k) \cup \mathcal{N}(x_k) \cup \chi_k(\xi_k), \tag{2.21}$$

with S_k denoting the finite set of mesh points calculated during the SEARCH step. In full analogy to the basic GPS algorithm we define

Definition 2.9: *If $f(y) < f(x_k)$ for some $y \in T_k$, then y is said to be an* ***improved mesh point*** *.*

Definition 2.10: *If $f(x_k) \leq f(y)$ for all $y \in P_k(x_k) \cup \mathcal{N}(x_k) \cup \chi_k(\xi_k)$, then x_k is said to be a* ***mesh local optimiser*** *.*

The MGPS algorithm is formally presented in listing 2.4.

Listing 2.4: MGPS Algorithm

```
1  Initialisation: Let x_0 satisfy f(x_0) < ∞. Set
2  Δ_o > 0 and ξ > 0.
3
4  while ¬(break condition)
5          1. Set extendend poll trigger χ_k > χ
6
7          2. SEARCH step:
8          Employ some finite strategy
9          seeking an improved mesh point; i.e.
10         x_{k+1} ∈ M_k such that f(x_{k+1} < f(x_k).
11
12         3. POLL step:
13         If SEARCH step was unsuccessful, evaluate
14         f at points in the poll set P_k ∪ N_k
15         until an improved mesh point x_{k+1} is found
16         (or until done).
17
18         4. EXTENDED POLL step:
19         If SEARCH and POLL steps does not find an
20         improved mesh point, evaluate f at points
21         in χ_k(ξ_k) until an improved
22         mesh point x_{k+1} is found (of until done).
23
24         5. UPDATE: If SEARCH POLL or EXTENDED POLL
25         finds an improved mesh point, update
26         x_k + 1 and set Δ_k + 1 ≥ Δ_k
27         according to (2.6) Otherwise set x_{k+1} = x_k
28         and set Δ_k + 1 < Δ_k according to
29         (2.7)
30
31         k = k+1
32 end
```

In contrast to common search heuristics presented in the beginning of this section, the class of algorithms derived from the basic GPS algorithm comes along with a quite substantial convergence theory. Hence it would be naturally at this point to refer some of the convergence results. The problem is that these results are developed in the theoretical

framework of the Clarke nonsmooth Calculus ([CLA]), which is too voluminous to be presented here in detail. Its main idea is to develop a generalisation of the concept of differentiability to steady functions, which are not differentiable in the usualy sense. With the notions of the Clarke calculus it is possible to generalise the Karush-Kuhn-Tucker (KKT) first-order necessary conditions for optimality ([KAR], [KUH], [BAZ/SHE]). Beside many other results it is possible to prove that under additional assumptions this usual necessary condition for optimality holds for the most variants of the GPS algorithms.

2.4 Conclusion

The survey of the usual methods of treating problems of mixed integer programming led to the conclusion that the framework of the GPS algorithm offers a viable and modern opportunity. Heuristic search algorithms as Simulated Annealing or Genetic Programming always come along with the disadvantage of poor numerical performance, especially near the minima. Moreover, an open source implementation for MATLAB is avaiable for the GPS algorithm, which will be used throughout this work.

Chapter 3

First Numerical Explorations

3.1 The General Assumption for the Numerical Analysis

Before we do any numerical analysis, we want to formulate the general hypothesis of this work. Qualitatively spoken this hypothesis comprehends that in our model there exist "Dirac Sea like" minimisers. But what are the features of such configurations? The term "Dirac Sea like" is quite vague, but some features should be made more explicit. The following definition is quite ad hoc and in no way canonical, but it is met because we want some mathematical exact notion of "Dirac Sea like".

Definition 3.1: *A configuration* (ω, τ). *is called* ***formal Dirac Sea like*** *up to* $k = m$, *if*

$$\omega_m > \omega_1 \quad \text{and}$$
$$\left| \sum_{l=1}^{m} \tau_l \right| \geq \frac{1}{2} \left| \sum_{l=1}^{m} \operatorname{atanh} \left(\frac{k_l}{\sqrt{\left(\frac{n}{2}\right)^2 + k_l^2}} \right) \right|. \tag{3.1}$$

The simple heuristics of this definition concerning the variable τ can be understood by comparing formula (3.1) with (5.4).

Allthough this is a quite weak notion of *"Dirac Sea like"* expecially concerning the variable ω it seems not appropriate to define a stronger notion, since this would cramp our terminologies, which is not adequate in consideration of our ignorance of the behaviour of the fermionic systems described by the action principle. Hence the qualitative notion of "Dirac Sea like" does not loose its relevance at all.

For the heuristical purpose it is adequate to formulate a general hypothesis for this thesis. Allthough it was mentioned above, we now have the notions to make this hypothesis as precise as possible. The further thesis will deal with the task, to what extent this hypothesis can be confirmed. The general hypothesis of this thesis is:

(GH) Discrete Fermionic Systems described by the model developed in Chapter 1 prefer Dirac Sea like minimisers.

Depending on whether the Dirac Sea like minima can be considered as *global* or *local* minima, we speak about a *global* or *local* confirmation of (GH). Finally we state:

Definition 3.2: *Considering a specific discrete fermionic system, a point* (ω, τ) *is called* ***numerical minimiser*** *if it is endpoint of a optimisation strategy applied* $\nu_{\max}$ *times. If a more optimal point is not known,* (ω, τ) *is called the* ***absolute numerical minimiser.***

3.2 Qualitative Results

To avoid programming errors it is quite useful to deliberate some tests to the numerical results. For example we have proved in Chapter 1 that the eigenvalues (1.25) are either complex conjugate – and for this reason the Lagrangian (1.26) vanishes – or both positive.

One might suggest that it would be a useful test to check if all real eigenvalues are positive. But this proof is not very useful, since the positivity of the eigenvalues $\lambda_\pm$ in the real case of (1.25) holds for every complex ϕ, v_0 and v_k and thus only falsifies the correct implementation of equation (1.25). But nevertheless: The question in which regions the discriminant

$$\mathcal{D}(r,t) = 4\Re^2\left(\phi\overline{v_0}\right) - 4\Re^2\left(\phi\overline{v_k}\right) - 4\Im^2\left(V_0\overline{V}_0\right) \tag{3.2}$$

is positive or negative, respectively, stays of interest. Spacetime points for which $\mathcal{D} \leq 0$ and thus $|\lambda_-| = |\lambda_+|$ do not contribute to our action. In reference to the relativistic causality principle it may be expected that this region is similar to the spacelike region of a Minkowski space-time. And contrary the region $\mathcal{D} > 0$ should match the timelike region. Figure 3.1 shows the function $\operatorname{sign}(\mathcal{D}(t,r))$. In the neighbourhood of the origin the light cone is observable as the border between positive and negative signum of the discriminant.

The calculations so far were done with an ad hoc discretisation only in the ω-direction. Furthermore the considered lattices were quite big, about 100 points in the ω-direction (referring to figure 3.2(a)). Obviousely the task of optimisation will start with considerable smaler systems of the kind of figure 3.2(b).

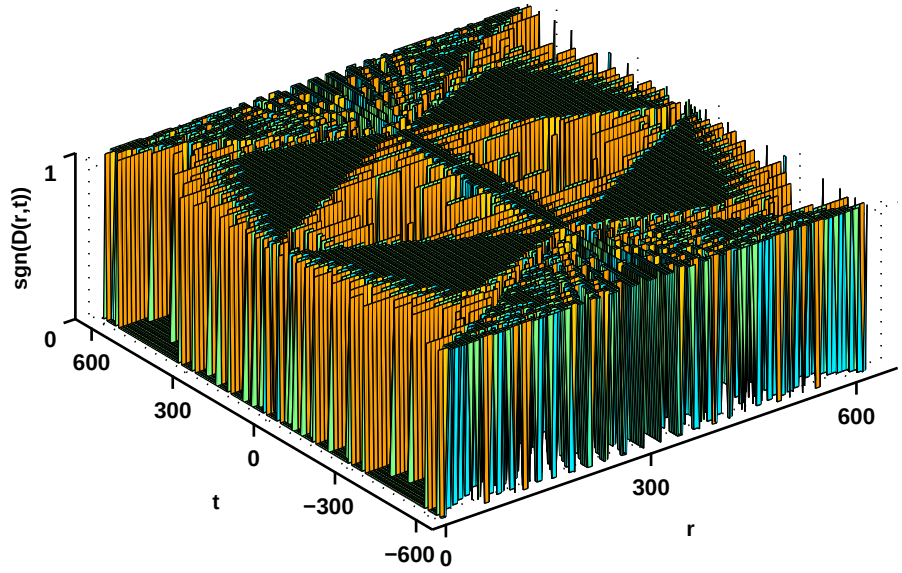
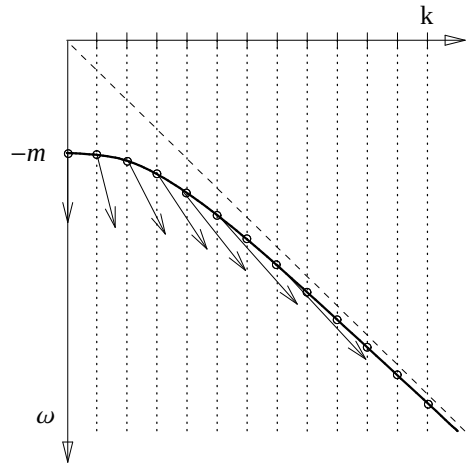
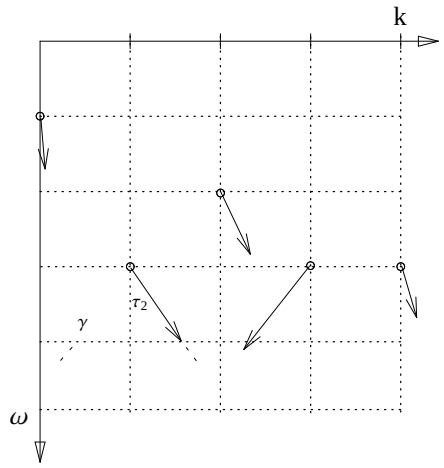


Figure 3.1: sign($\mathcal{D}$) calculated from a Fermionic Projector concerning to a system with 100×201 points in r-t-space and an arbitrary mass parameter



(a) Ad hoc discretisation of the continuous Dirac Sea with vectors $(v_\omega, v_k = (-\omega, k)$ pinned on each occupied state.



(b) An example of a vacuum configuration for the discrete space-time. The variation parameter can be considered as an angel rotating and stretching the the vector (v_k, v_ω) along the curve γ.

Figure 3.2: Different types of discretisation

3.3 General Remarks concerning the Task of Optimisation

We will now track a more numerical point of view. The general numerical problem is of the form

$$\begin{aligned} &\textbf{minimise} && S(\tau, \omega) \\ &\text{under the conditions} && \tau \in [-\tau^{\max}, \tau^{\max}]^n =: \mathrm{T} \\ &\text{and} && \omega \in [-\omega_{max}, 0]^n =: \Omega, \end{aligned} \tag{3.3}$$

where n is called the *system size*.

In complexity theory there is made a distinction between the classes of problems $\mathcal{P}$ and $\mathcal{NP}$. In general *mixed interger nonlinear problems* like ours are $\mathcal{NP}$-hard. (For a general introduction to the topic of *mixed integer optimisation* see [CAL].) For those readers, who are not familiar with complexity theory, just a few words. Problems in $\mathcal{P}$ are those decision and optimisation problems, which can be solved on a deterministic sequential machine in an amount of time that is polynomial in the size of the input. $\mathcal{NP}$-problems are those, whose positive solutions can be found in polynomial time on a non-deterministic machine, so clearly $\mathcal{P} \subset \mathcal{NP}$. Finally there is the class of $\mathcal{NP}$-complete problems ($\mathcal{NP}$-$\mathcal{C}$) which are in some kind the "hardest" $\mathcal{NP}$-problems. These problems are those, for which any problem in $\mathcal{NP}$ can be expressed as a problem in $\mathcal{NP}$-$\mathcal{C}$, only by some efficient – that means at most polynomial bounded number of steps – transformation. Therefore if one finds an efficient (that means polynomial) algorithm to solve any $\mathcal{NP}$-complete problem, then every problem in $\mathcal{NP}$ can be solved efficiently, and hence must be in $\mathcal{P}$. This would mean that $\mathcal{NP} = \mathcal{P}$, an assertion, which is still an open question of research.

In practice $\mathcal{NP}$-hard means that the costs of the numerics raises faster with the size of the initial data than any polynomial. In our case the size of the initial data is given so far by n. If we want to solve for every occupation the corresponding NLP-problem – which may be polynomial in the size of the initial data – then the costs of the numerics would raise at least like n^n in the case of *complete enumeration*, that is going through all combinatorial possible occupations and solving for each occupation the NLP-problem. Obviously this method is only practicable for small systems.

On the one hand our optimisation problem 3.3 has the nice property to be scalable by the lattice size n. On the other hand the complexity of the optimisation problem raises dramatically, while efficient methods (in the notion discussed above) are not available. In this scenario it is recommended to start with complete enumerations for small systems. Results obtained by this method could be used as a starting point for more sophisticated methods, which can then be applied to systems of a size, which are not treatable by

complete enumerations. That is more sophisticated methods should be tested on systems, which are already well understood by the analysis with complete enumeration.

3.4 Minima for Small Systems

3.4.1 Systems with one occupied state

The result obtained for the case $n = 1$ can easily be checked analytically. Note that in this case the lattice in the location space is just given by

$$\mathfrak{L} = \{0\} \times \{0\} \tag{3.4}$$

and thus one gets for the action a function of the form

$$S(\tau) = a + b(\cosh\tau)^2 \quad \text{with } a, b \in \mathbb{R}_+, \tag{3.5}$$

and one gets one minimum for $\tau = 0$ (see figure 3.4.1).

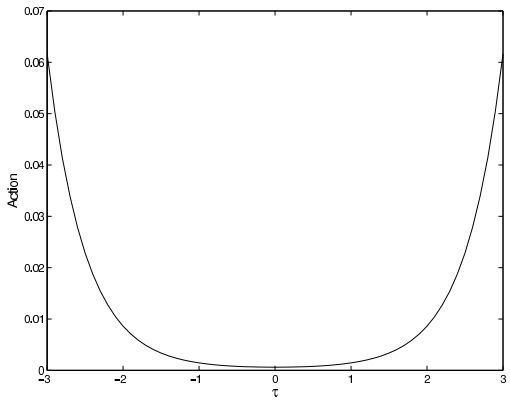


Figure 3.3: $S(\tau)$ for $n = 1$

We want to call minima like this *trivial minima*, i.e.

Definition 3.3: *A minimiser (τ, ω) is called **trivial**, if $\|\tau\|_\infty \leq \varepsilon$, with $\varepsilon = 0$ in the case of an analytically determined minimiser and ε to be the precision of the calculation in the case of a numerically determined minimiser.*

3.4.2 Systems with two varied states

We now have to face the question, if there are other minima then trivial ones. If this question would be answered negative, the variational principle of the Fermionic Projector

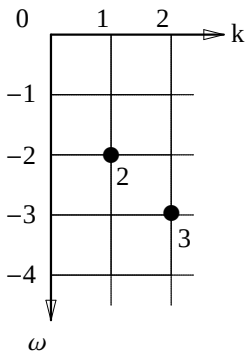


Figure 3.4: Example of an occupation (23) of the system with two varied states

would be proved to be physically moot. If otherwise the question would be answered positive then the questions arises, whether the not trivial minima are global or local and match the general hypothesis of this thesis.

The next analysis done for $n = 2$ was a complete enumeration concerning the discrete variables and a ”graphical analysis“ concerning the continuous variables. In the case of two continuous variables, it is possible to plot levelsets. In consideration of the more interesting results discussed in the following chapters, we will not discuss the results of the calculations into great detail. The only remarkable outcome of the numerical analysis is the numerical proof of the existence of non-trivial minima.

The main result for systems with $n = 2$ is that there exist non trivial minima, but that they are suboptimal compared to the trivial minimum. In general, there are three kinds of classes of occupations which differ in the kind of minima they feature. The first class holds only a trivial minimum. The second class holds trivial and non-trivial minima and accordingly the third class holds only non-trivial minima. Due to the symmetry $\tau_i \to -\tau_i$, for every non trivial numerical minimiser at the point (ω, τ) we have a corresponding second trivial minimiser at the point $(\omega, -\tau)$.

3.5 Conclusion

The calculations of this chapter show that there exist non trivial minima for the considered action principle for systems with $n > 1$. They come along with a spontaneous symmetry breaking. We will refer to the method used for $n = 2$ the *graphical method of optimisation*.

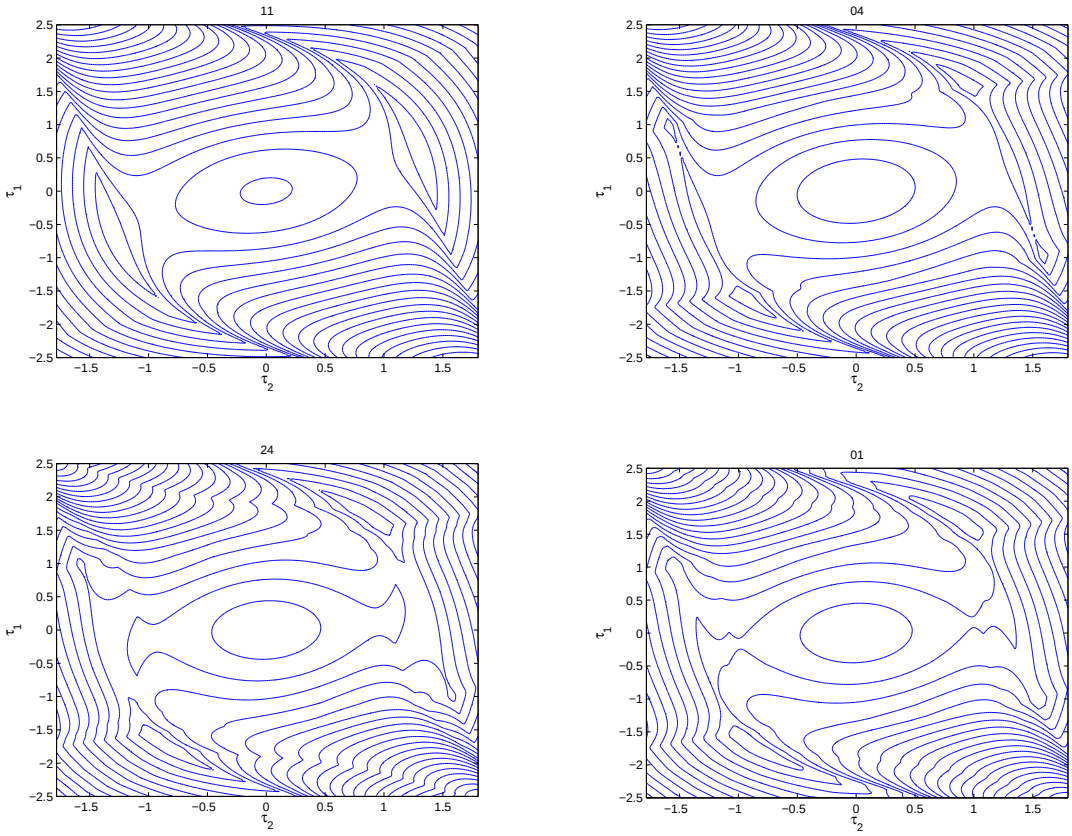


Figure 3.5: Characteristic level curves of the action for different types of occupations. (Note that these results anticipiate the use of the lattice factor $N > 1$ introduced in Chapter 4.)

Chapter 4

Complete Enumerations for more Complex Systems

4.1 Spontaneous Symmetry Breaking

The analysis of systems of small occupation numbers has to deal with a special problem. Looking at formulas (1.14) to (1.15) one sees that from the inverse dependencies of the form

$$\frac{\sin(r)}{r},$$

the action will be dominated by the contributions of the Lagrange density near the origin. E.g. the action of small systems is dominated by the contribution $\mathcal{L}(0,0)$. These contributions depend by terms proportional to $(\cosh \tau_i)^2$ on the variational parameters τ. Hence one gets for sufficiently coarse lattices an action which is highly dominated by trivial minima. It stands to reason that this will not lead to physically interesting minimisers. In other words: For small systems the resolution in the location space is too bad to include essential contributions to the action for $r \neq 0$. The problem is that, by this reason, small systems will not contain non trivial minima, but that it is essential to have such minima to construct propper start values for larger systems.

This problem can be resolved by considering larger lattices without increasing the number of occupied states. In the case of quadratic systems, i.e. systems with as many possible ω-values as k-values, the so called *lattice factor N* indicates the factor by which the number of possible (ω- and k-) values exceed n. (See figure 4.1.) In detail:

Definition 4.1: *Consider a discrete fermionic system with n occupied states. Then we call* $N \in \mathbb{N}$ *the* ***lattice factor*** *, if*

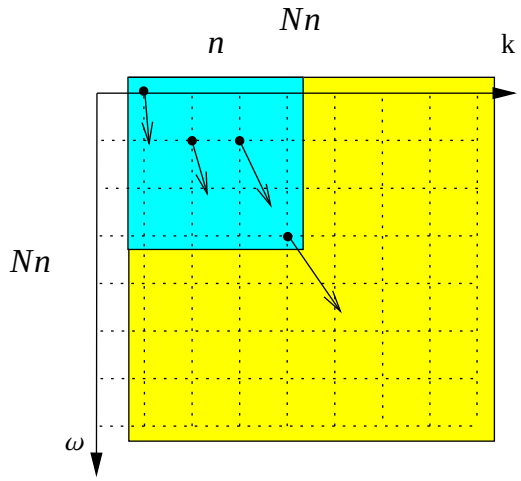


Figure 4.1: Illustration for the lattice factor N in the case $N = 2$.

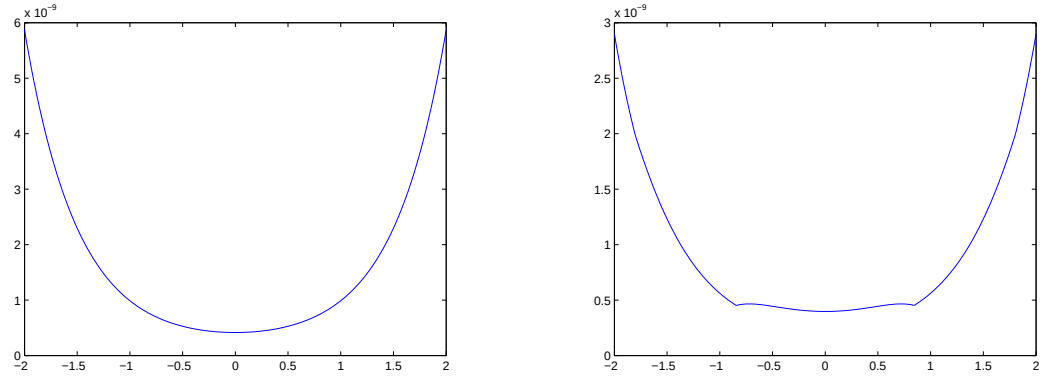
(a) N is the factor by which the analysed System is bigger than the subsystem of occupied states.

(b) $N_r = N_t = N$ (Quadratic Lattice)

(c) Only the lowest k-values are occupied.

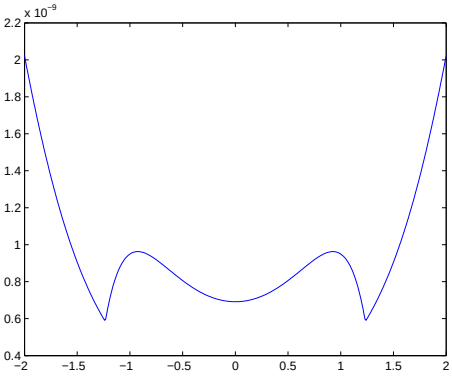
Introducing a lattice factor $N > 1$ can alternatively be interpreted as a system, which only considers states of low momenta and to leave states of the high energy region unoccupied.

Figure 4.2 shows the graphs of the action for $n = 1$, $\omega = 0$ and different values of N. For the smallest lattice ($N = 2$), the action has a quite simple structure with a trivial minimum at the origin. For $N > 3$ the symmetry breaks in the sense that there occur non trivial minima. A similar result is archieved for $n = 2$ (see figure 4.3). In this case, the structure of the action becomes more and more complex with increasing N. Hence for small systems non trivial results are only to expect if the considered lattice is considerably enlarged. One can expect that this modification of our model is only necessary for the analysis of small systems and can be omitted for larger systems. Thus an enlargement of the lattice of small systems has to be regarded as a heuristic tool to get an approach to the general problem, since it is very appropriate to start the numerical optimisation with preferably small systems. From this one can gradually enlarge the system size and develop certain optimisation strategies.

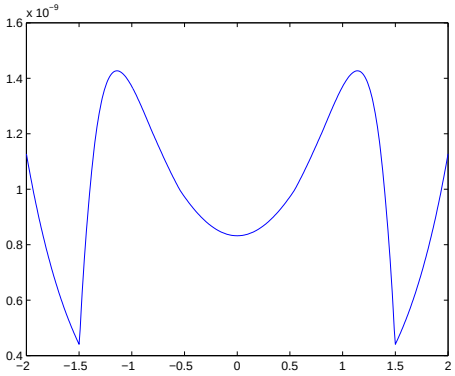


(a) $N = 2$

(b) $N = 3$

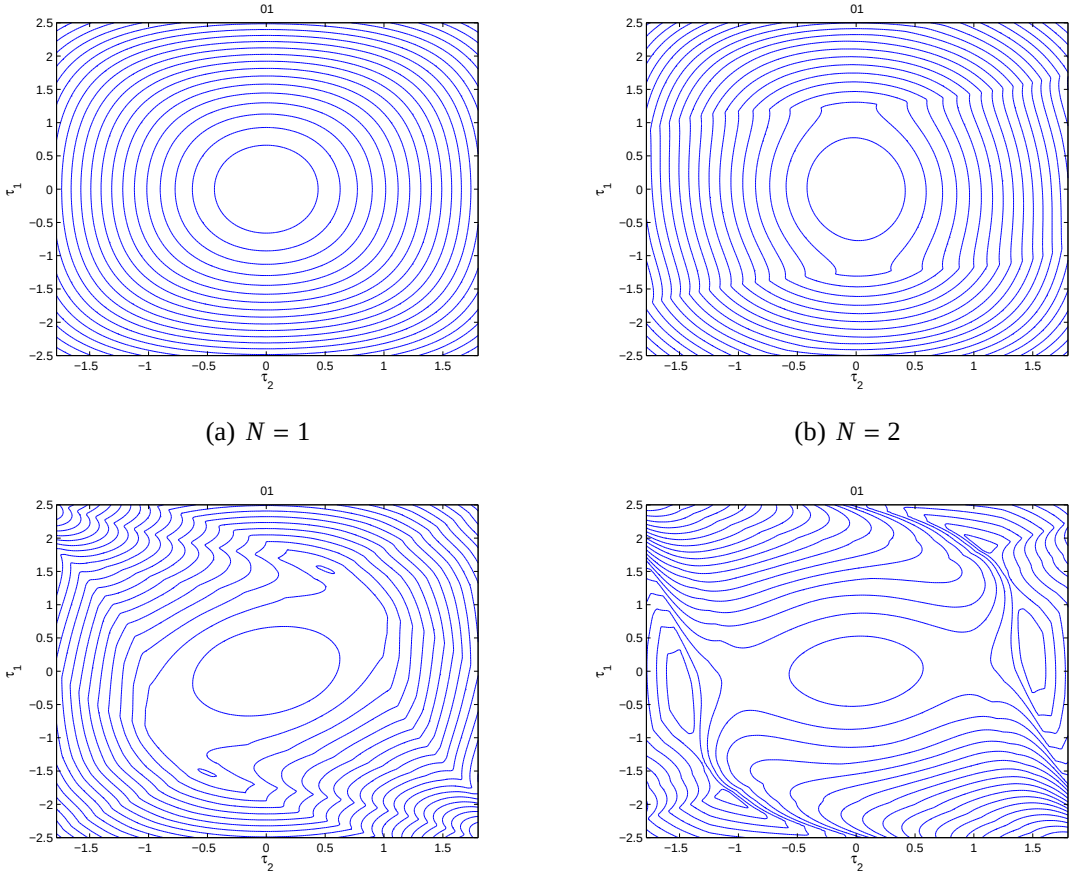


(c) $N = 4$



(d) $N = 5$

Figure 4.2: Spontaneous Symmetry Breaking caused by increasing the lattice for $n = 1$.



(a) $N = 1$ (b) $N = 2$

(c) $N = 5$ (d) $N = 7$

Figure 4.3: Spontaneous Symmetry Breaking caused by increasing the lattice for $n = 2$. For $N = 7$ the lower nontrivial minimum is absolute.

4.2 Enumerations for $n = 3$

4.2.1 Complete enumeration

The aim of the analysis discussed in this subsection is to get an accurate insight to the action in the case $n = 3$ and $N = 4$. The *graphical method* applied to the case $n = 2$ in chapter 3 was extended in the way that τ_3 was fixed and τ_1 and τ_2 were varied like in the two dimensional case. This led to a series of graphs numbered by τ_3, which now could

be merged to a "film".[1]. Four such films were produced, belonging to the occupations $\omega = -(1,2,4)$, $\omega = -(0,0,1)$, $\omega = -(0,3,1)$ and $\omega = (0,0,0)$. The resolution of the τ_i was set to

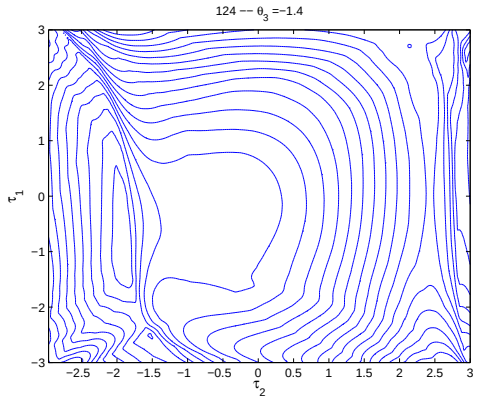
$$\begin{aligned}\tau_1 &\in \{\ -3 \leqslant 0.03 \cdot k_1 \leqslant 3 \,|\, k_1 \in \mathbb{Z}\ \}\\ \tau_2 &\in \{\ -3 \leqslant 0.03 \cdot k_2 \leqslant 3 \,|\, k_2 \in \mathbb{Z}\ \}\\ \tau_3 &\in \{\ -3.8 \leqslant 0.2 \cdot k_3 \leqslant 0 \,|\, k_3 \in \mathbb{Z}\ \}.\end{aligned}$$

Thereby τ_3 must only be analysed for negative values since the general symmetry of the action under the transformation $\tau \longrightarrow -\tau$.

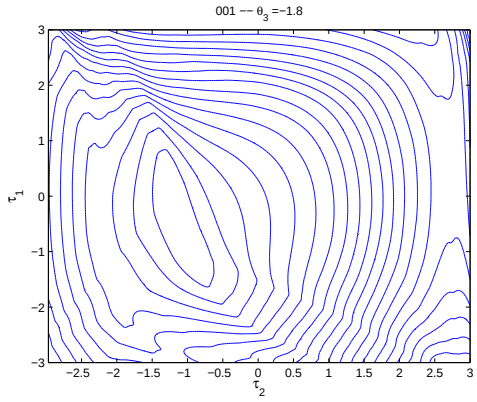
Generally speaking, the qualitative result was the same for all four occupations. As shown in figure 4.4 for the occupation $\omega = -(1,2,4)$, for $\tau_3 = 0$ the action is point symmetric about the origin. Obviously this symmetry is destroyed for $\tau_3 \neq 0$. For increasing k_3 the action gets dominated by two minima, one emerging from the trivial minimum for $\tau_3 = 0$, one for $\tau_1 \approx 0$ and $\tau_2 \approx -2$. With decreasing τ_3 these two minima merge and around the τ_3-value of the merging of these minima, the absolute minimum is attained, as a matter of course with respect to the chosen resolution (see figure 4.6). Furthermore figure 4.4 shows the sections referring to the absolute minimum of the examined occupation. From this one can qualitatively state that the action gets the more complex the more "complex" the occupation is. Also this is not a strict notion, the occupation $\omega = -(0,0,0)$ seems most simple, followed by $\omega = -(0,0,1)$ and the rest. According to this, the graph belonging to the occupation $\omega = -(0,0,0)$ shows only one minimum belonging to $\omega = -(0,0,1)$ contains two and the other at least three minima.[2]

[1] See the folder `/Complete_enumeration_3/`. The movies are named `cut.avi` and placed in the according directories on the data disc.

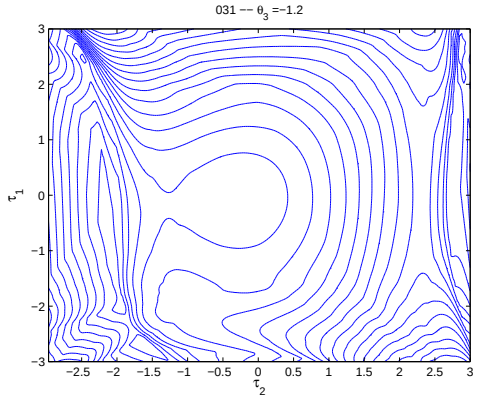
[2] The problem in the determination of the number of minima in the notion of the "graphical method" is quite raw and leads usually to much more determined numerical minimisers than there are real minima.



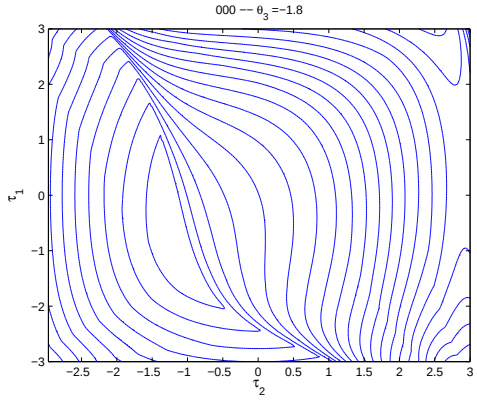
(a) $\omega = -(1, 2, 4)$, $\tau_3 = -1.4$



(b) $\omega = -(0, 0, 1)$, $\tau_3 = -1.8$



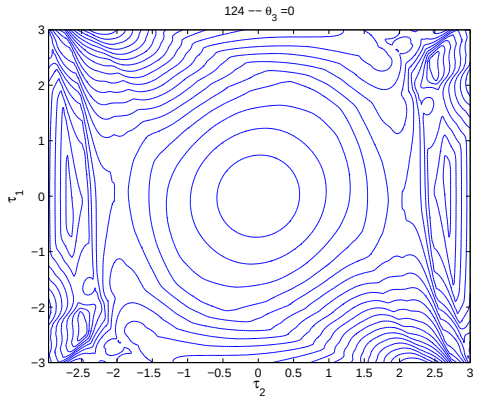
(c) $\omega = -(0, 3, 1)$, $\tau_3 = -1.2$



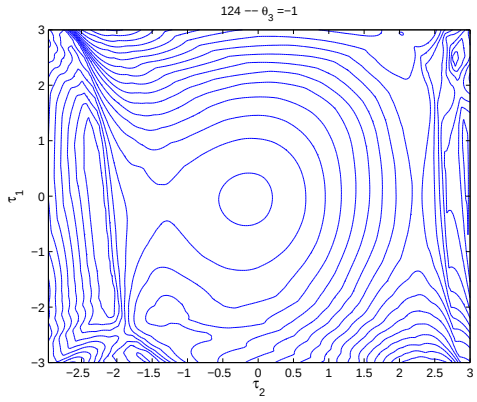
(d) $\omega = -(0, 0, 0)$, $\tau_3 = -1.8$

Figure 4.4: Sections of the action graph for fixed τ_3 and different occupations (cf. figure 4.6)

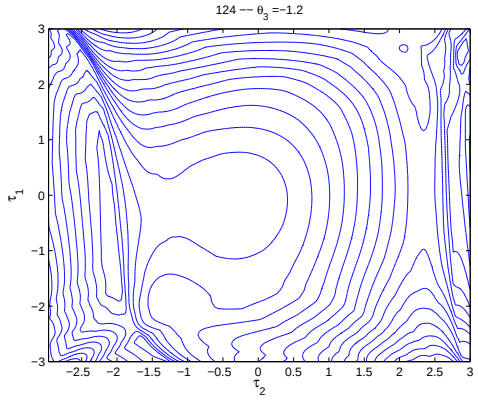
.



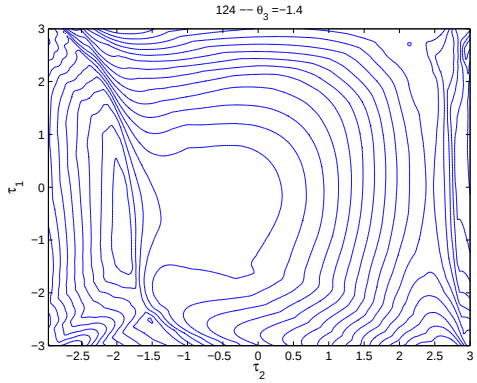
(a) $\tau_3 = 0$



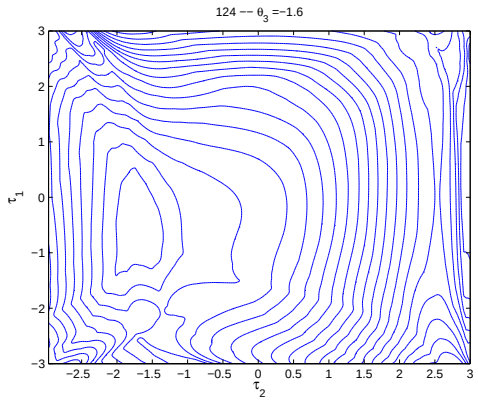
(b) $\tau_3 = -1.0$



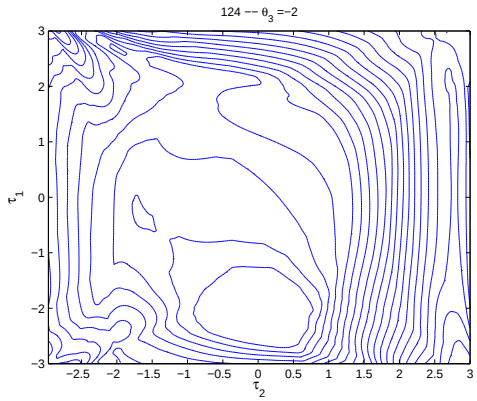
(c) $\tau_3 = -1.2$



(d) $\tau_3 = -1.4$ – absolute minimum



(e) $\tau_3 = -1.6$



(f) $\tau_3 = -2.0$

Figure 4.5: Sections of the action graph for different values of τ_3 and the fixed occupation $\omega = -(1, 2, 4)$.

ω	τ_1	τ_2	τ_3	S
$-(1,2,4)$	-0.9600	-1.8900	-1.4	$2.8144e-08$
$-(0,0,1)$	-0.6300	-1.0200	-1.8	$4.2677e-08$
$-(0,3,1)$	-2.5800	-1.6500	-1.2	$2.8750e-08$
$-(0,0,0)$	-0.8400	-1.0500	-1.8	$1.0534e-07$

Figure 4.6: Numerical minimiser according to the (resolution) of the "graphical" method

4.2.2 Combined Complete Enumeration and GPS-Search

The numerical costs of complete enumerations as performed for $n = 3$ in subsection 4.2.1 increase very rapidly with raising n. Hence it is nessesary to develop methods of analysis, which allow numerically a more effective account. The idea is to choose for every enumerable occupation an appropriate start value in the τ-space and to perform starting, from this start value, a trajectory oriented optimisation algorithm. Compared to the calculation of "graphs" of the action concerned to each occupation, this method should get by with much less action evaluations. We used the implementation of the GPS algorithm by Mark Abramson (see [ABR1], [ABR2], [AFIT]) NOMADm. Concerning the numerous options offered by the interface of NOMADm, we generally used the settings listed in Appendix C except other settings are mentioned explicitly. Particularly the abort condition was set to

$$\Delta\tau_i = 0.0001 \tag{4.1}$$

whereas $\Delta\tau_i$ denotes the Convergence Tolerance, which is a positive number such that the MADS run is terminated as soon as the mesh size parameter falls below this value. The abort condition concerning the discrete variable is determined by the algorithm, depending on the chosen polling strategy.

Single Search

In a first try we choose a standard MATLAB BFGS Quasi-Newton method with a cubic line search procedure, using an analytically given gradient in the τ-space, which works much more efficiently than using the numerically approximated gradient. Although the function cannot be assumed to be differentiable everywhere by the reason of absolute values occurring in (1.26), in practice gradient based methods can still deliver useful results. Nevertheless this method failed, which was easily brought to light by a comparison of the minimisation with the results of the complete enumeration of Section 4.2. The lack of differentiability becomes demonstrative with a look at figure 4.7. The algorithm steps to the

bottom of the main valley seen in this figure and stops, without descending within the valley. As a consequence we have to use an algorithm which is not restricted to differentiable functions.

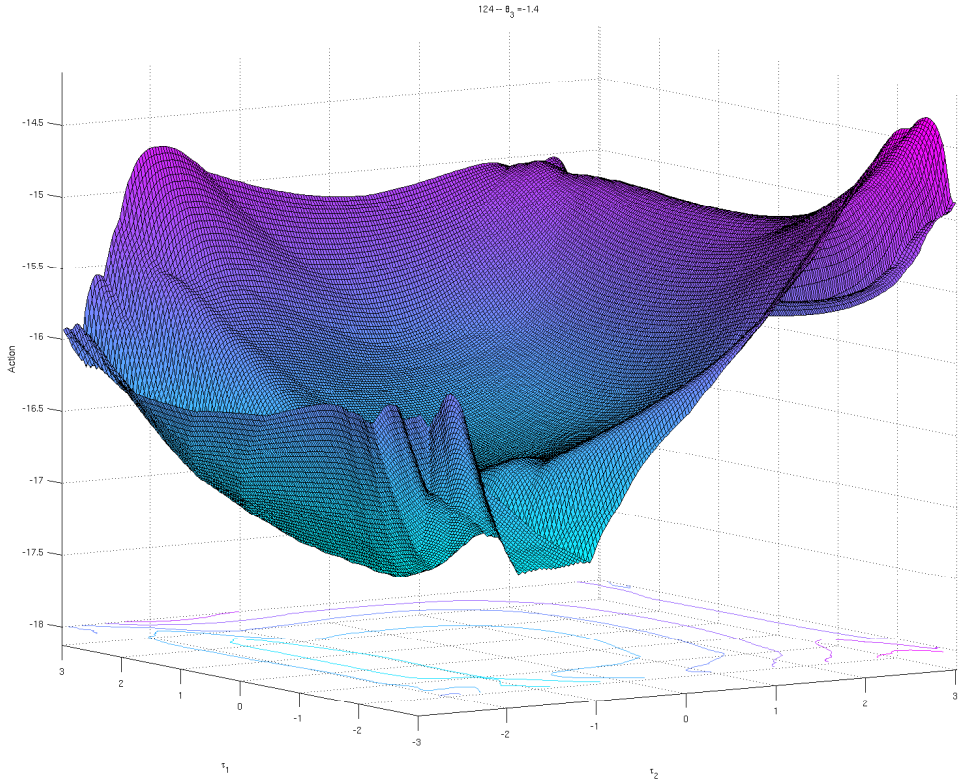


Figure 4.7: $S(\tau_1, \tau_2)$ of a system with $n = 3$, $N = 4$, $\tau_3 = -1.4$

As described in Section 2.2 the scope of the MADS algorithm in the original form is wider than that of differentiable functions. And in fact, the use of MADS resolved the struggle described above. We first performed standard MADS searches for starting-points derived from the results obtained in section 4.2.1. By averaging the results summarised in figure 4.6, we define the starting value

$$\tau_0^h = (-1.5282, -1.6188, -1.4044). \tag{4.2}$$

The minimisation enumerated all occupations ω within

$$\left\{\omega \in \mathbb{N}_0^3 \mid -2 \,.\leq \omega \,.\leq 0\right\}. \tag{4.3}$$

Again the minimisations where performed depending on the general lattice factor N introduced in definition 4.1. As a first result, we again get only non trivial minima for $N \geq 3$, whereas we get the same occupations for the minima with $N = 3$ and $N = 4$.

N	ω	τ_1	τ_2	τ_3	S
4	$-(0,1,2)$	$-1,00232109375$	$-1,87661250000$	$-1,40440000000$	$3.071438547435570e-08$
3	$-(0,1,2)$	$-0,290895312500$	$-1,54409296875$	$-1,12559140625$	$2.851709405942606e-08$
2	$-(1,0,2)$	-0.00036796875	-0.00014765625	-0.000103125	$1.7843076e-08$
1	$-(0,1,2)$	-0.00036796875	-0.00014765625	-0.000103125	$5.5970146e-08$

Figure 4.8: Numerical Minimisers for different values of N with combined complete enumeration and GPS search

Secondly, it is remarkable that we find for $N \geq 3$ for 100% of the enumerated occupations a non trivial minimum, whereas for $N \leq 2$ this fractions falls to 0% (see figure 4.9). This is consistent with the results achieved in Section 4.1.

N	ω	S	$S\|_{\tau=0}$	% of not trivial Minima
4	$-(0,1,2)$	$3.071438547435570e-08$	$5.0821808e-08$	100
3	$-(0,1,2)$	$2.851709405942606e-08$	$3.5771382e-08$	100
2	$-(1,0,2$	$1.7843076e-08$	$1.7843075e-08$	0
1	$-(0,1,2)$	$5.5970146e-08$	$5.597e-08$	0

Figure 4.9: Continuation of the table in figure 4.8

Third we also see in figure 4.9 that the not trivial minimum has lower action then for $\tau = 0$.

Multiple Random search – Global Search

The analysis so far raises the question if there is evidence that the chosen start value τ_0 is in fact close to τ-value of the absolute minimiser. Naturally numerical analysis here cannot give a definite answer. But if τ_0 as chosen above is a proper start value to reach the absolute minimiser of an occupation, it should be impossible to find more optimal minima by choosing different start values τ_0^r, whereby "r" denotes the fact, that we will make a random choice. More precisely we choose

$$\tau_{0i}^r \in [-1.7, 1.7] \tag{4.4}$$

by MATLAB's standard random function.

In appendix B.1.1 the results are listed. For each value of $1 \leq N \leq 4$ there were carried out 10 runs for which the minimum occupation was saved. (The tables in B.1.1 are sorted by

the value of the action.) Comparing B.1.1 with figure 4.8, one sees that the minima found from starting at the heuristic starting point τ_0^h does not lead to the absolute minimum. More precisely, for $N = 4$ the random search led to one more optimal solution and for $N = 3$ to two more optimal solutions.

Multiple Random Search – Local Search

The result of the last section leads to a further numerical task. It is possible that nearby the absolute minimum there are other minima which might be found by starting our minimising at points nearby τ_0^h. So we define new starting points by

$$\tau_0^{hr} = \tau_0^h + \alpha(\rho - 0.5), \tag{4.5}$$

where $\rho = \rho(n) \in [0, 1]$ describes a random noise term implemented as the standard MATLAB random function, n (here) denotes the absolute number of function evaluations or the natural time and $\alpha \in \mathbb{R}^+$ a scalar controlling the random spread. The results of these calculations are listed in the Appendix at B.1.1 in figure B.5 to B.6.

In summary we have only to discuss the results for $N = 3$ and $N = 4$ since the trivial results for $N \leq 2$ are of no interest. Surprisingly the results for $N = 3$ and $N = 4$ are different. For $N = 3$ the picture is quite unique. First the results in Figure B.8, B.2 and B.5 suggest that there is a symmetry via the permutation

$$(0, 1, 2) \quad \longleftrightarrow \quad (2, 1, 0), \tag{4.6}$$

since B.2 and B.6 contain numerous minima of both occupations with approximately equal action as the absolute numerical minimiser. This symmetry was numerically checked and confirmed. Modulo this symmetry, the best approximation of the minimum was reached by the local search listed in B.5. Remarkably the global search led to the same minimum as the local search. But a comparison of the global search B.2 and the local search listed in B.6 leads to the conclusion that it is a proper heuristic to search near by τ_0^h.

Although symmetries of the kind of (4.6) will be discussed in more detail later, we meet this

Definition 4.2: *Occupations ω^1 and $\omega^2 \in -\mathbb{N}_0^n$ are called **equivalent** if*

$$S(\tau, \omega^1) = S(\tau, \omega^2) \tag{4.7}$$

for all $\tau \in \mathbb{R}^n$.

Obviously this defines an equivalence relation and the referring equivalence classes are referred as the *classes of equivalent occupations*.

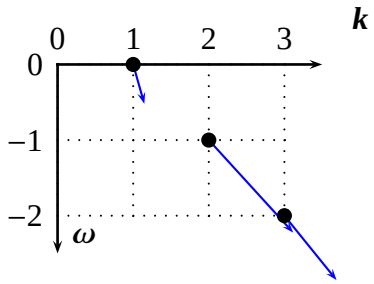


Figure 4.10: Best minimum for $n = 3$ and $N = 3$ (cf.B.6).

For $N = 4$ the picture is different and at first sight a little bit confusing. The most striking result for the case $N = 4$ comparing to $N = 3$ is that there seems no correspondence between the global search B.1 and the local search B.5 and B.7 respectively. The minima of both runs lack an intersection of solutions which likely belong to the same minimum. The reason for this is likely that the better "resolution" in the location space of the Lagrangian leads to an action, which contains more local minima. Hence at least for $n = 3$ the numerical task of optimisation is for $N = 3$ far better to handle as for $N = 4$.

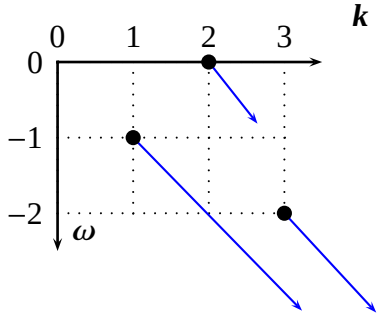


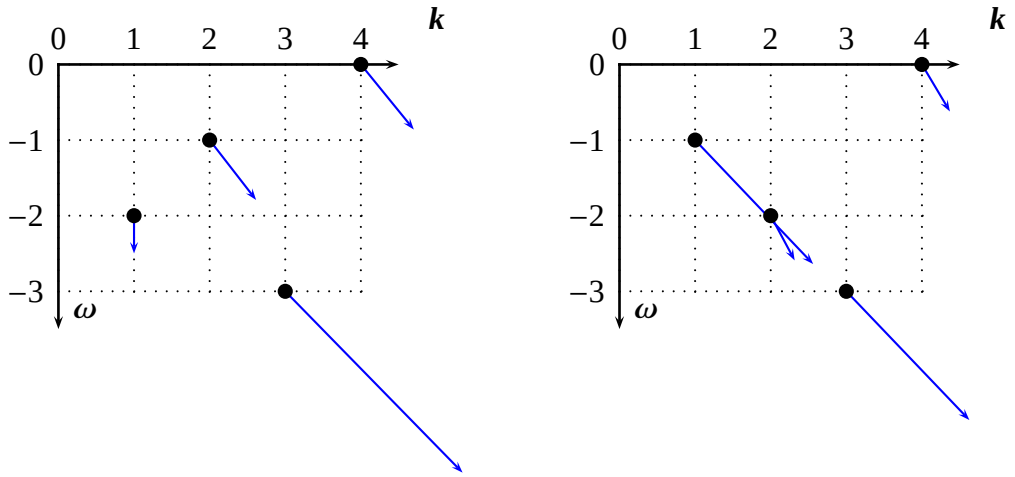
Figure 4.11: Best minimum for $n = 3$ and $N = 4$ (cf.B.7)

4.3 Enumerations for $n = 4$

At this point we do not want to put much attention on discussing the results for $n = 4$. The calculations we did were the same as for $n = 3$. Since the calculations at this point only serve as preliminary work for more sophisticated methods, we only notice the results. (Detailed data is found in appendix B.2)

4.4 Conclusion

In this chapter mainly three results were achieved.



(a) $N = 4$ (b) $N = 3$

Figure 4.12: Absolute numerical minimisers according to the methods of this chapter for $n = 4$

The first one is that the use of derivatives is not adequate for our problem. One might think that this is a trivial statement, since the formal nature of the Lagrange density 1.26 does not allow to expect a differentiable function. Nevertheless, even in cases of formally lack of global differentiability it might be possible to calculate a correct gradient in almost any regions of the variables. And in special cases using these gradients for optimisation purposes lead to good results anyhow. But this is not the case in our scenario. The action is especially in this regions not differentiable, which is important for trajectory oriented search algorithms. For this reason, the following analysis will not make use of the partial gradient calculated in this chapter.

The second result is that small systems as considered in this chapter require the introduction of the lattice factor N and choosing of $N \geq 3$. This can be considered as a slight change of our model. For considerably larger systems as analysed in this chapter, one should get non trivial results as well for $N = 1$. The results of subsection 4.2.2 suggest to choose for the further work $N = 3$.

The third result is that the existence of non trivial minima could be confirmed for the system sizes $n = 3$ and $n = 4$. For larger systems the method of complete enumeration will fail due to the exponentially raising numerical effort. Hence for larger systems, we have to apply more advanced methods.

Chapter 5

Causal Structure

To evaluate the solutions of any optimisation algorithm we have to understand what they might look alike if the general hypothesis of this work is right. As mentioned above the configurations should be in some way similar to the Dirac-sea configuration known from the continuous case. One aspect of this is to evaluate the *causal structure* which is given by

Definition 5.1: *The function $\mathcal{L}(t,r)$ as defined in (1.26) is called the* ***causal structure*** *of the configuration* (ω, τ).

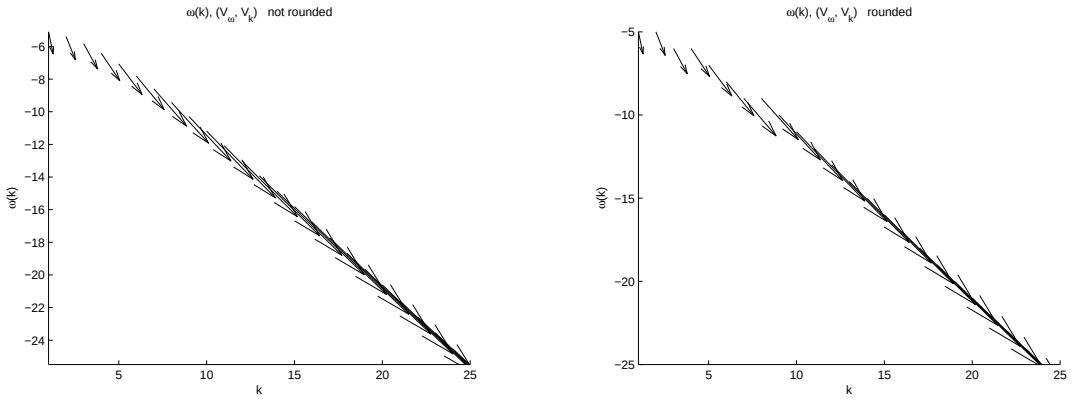
Note that this definition differs from Definition 1.1, where we used the notion "discrete causal structure". It is connected to Definition 1.1 by the fact that

$$\mathcal{L}(t,r) > 0 \qquad \Longleftrightarrow \qquad D[A(t,r)] > 0. \tag{5.1}$$

But the formal notion of Dirac Sea like configurations defined in Chapter 3 does not make any statement about the causal structure. Hence we have to make some considerations about causal structures, which are heuristically judged to be Dirac Sea like. The limitation of this is that the expected solutions from our discrete variational problem should be similar to corresponding continuous system only for small momenta.

5.1 Varying the System Size

The causal structure of any configuration can easily be plotted as a 2-dimensional graph. We have to elaborate which properties causal structures of Dirac Sea like configurations



(a) without discretisation in ω-direction (b) with discretisation in ω-direction

Figure 5.1: Dirac Sea like configurations for 25 occupied states.

have. For this purpose we have to regard Dirac-Sea like configurations

$$k \in \{1, 2, ..., N_k\}\,, \qquad \omega = -\sqrt{m^2 + k^2} \qquad \text{and} \tag{5.2}$$

$$k \in \{1, 2, ..., N_k\}\,, \qquad \omega = \left[-\sqrt{m^2 + k^2} + \frac{1}{2}\right], \tag{5.3}$$

whereas [...] denotes the Gauss bracket.

In (5.2) we consider configurations with continuous ω-values (see figure 5.1(a)) and in (5.3) we consider those with ω on a lattice as well as k (see figure 5.1(b)). In both cases we set

$$\tau_i = -\text{atanh}\left(\frac{k_i}{-\sqrt{m^2 + k_i^2}}\right). \tag{5.4}$$

Thus there is no other difference between (5.2) and (5.3) than the discretisation of the energy values.

The plots of causal structures of a specific configuration in the two cases is shown in figure 5.2 and 5.3. As a common property both configurations show the maximum Lagrange density at $r = 0$, with the absolute maximum at $t = 0$. Further both causal structures show essential contributions around the hypersurface $t = r$ which approximates the light cone. A similar structure around the hypersurface $t = 2\pi - r$ effects from the periodic boundary conditions. Apart from this, the causal structure of the continuous system can be regarded as approximately causal, since the Lagrangian vanishes nearly completely in spacelike regions not affected by the mentioned symmetry.

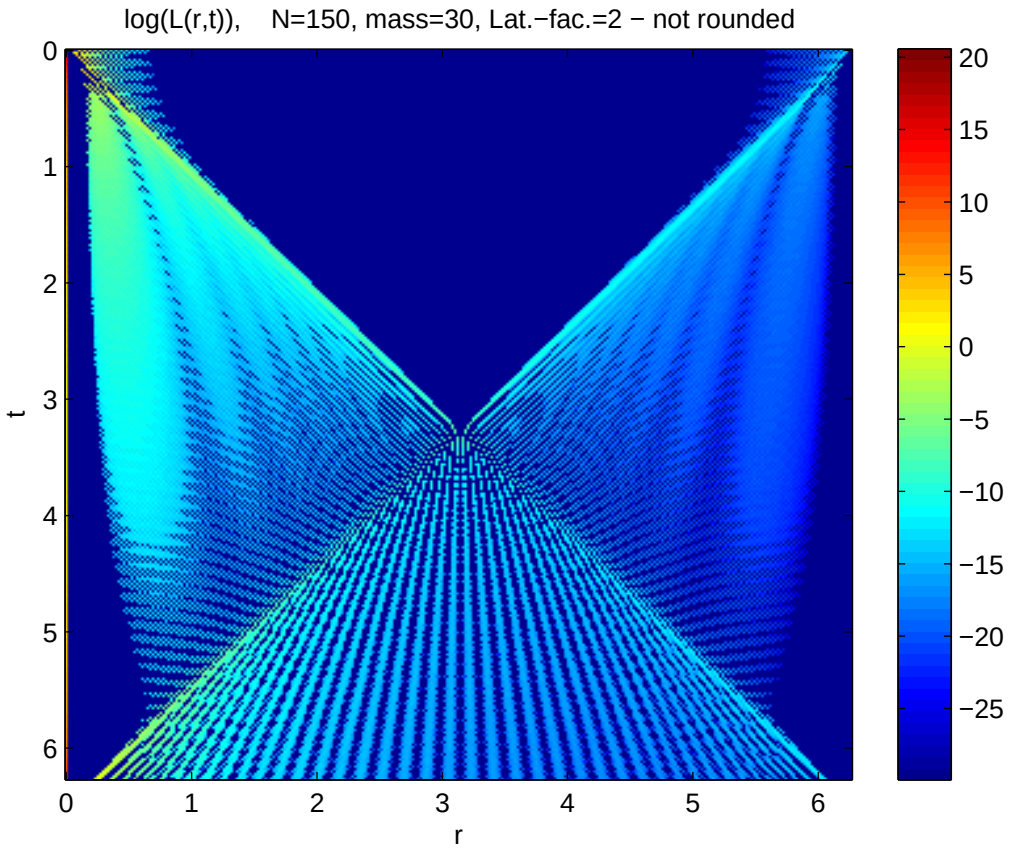


Figure 5.2: Causal Structure (logarithmic scale for the value of $\mathcal{L}(t,r)$ ($n = 150$, $m = 30$, $N = 2$).

The second causal structure shows two striking differences: First, there appear contributions even in a predominant part of the time-like space-time. In this sense, the Lagrangian of the second system can be called *non causal*. Secondly, an additional symmetry appears. The causal structure gets symmetric under reflections at the hypersurface $t = \pi$. Hence the transformations

$$\begin{aligned} t &\longrightarrow t + 2\pi \qquad \text{and} \\ t &\longrightarrow -t \end{aligned} \tag{5.5}$$

leave the Lagrangian unchanged

The last property might become useful if the numerics of large systems should be enhanced.

To get a qualitative understanding of causal structures which might be to expect in the

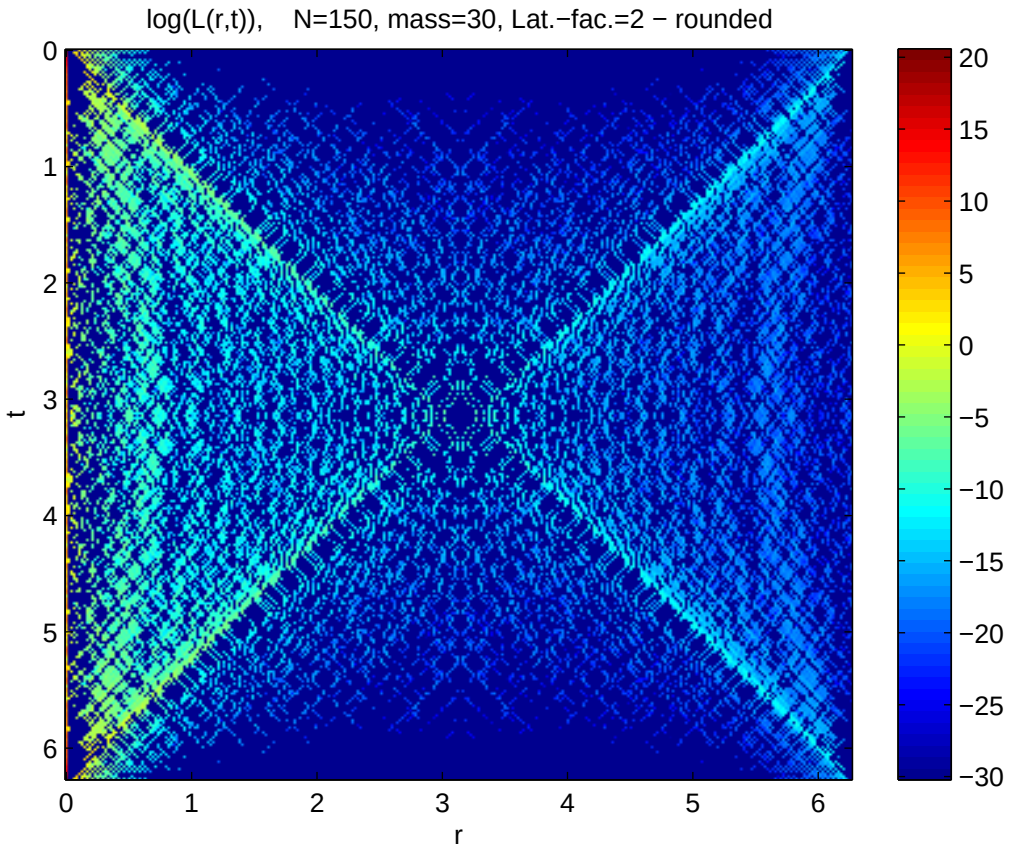


Figure 5.3: The same system as in figure 5.2 with the only difference that the ω-value was rounded to the integer lattice

notion of a confirmation of the main hypothesis of this thesis, we plotted several causal structures. Figure 5.4 shows a comparison between configurations (5.2) and (5.3) for different system sizes. The results illustrate that it cannot be expected that the non-causal contributions to the Lagrangian in (5.3) vanish when $n \to \infty$. Figure 5.5 shows configurations of the type (5.3) for different mass parameters and especially for small n. Since in this thesis we will mainly deal with systems with $n \leq 12$, the plots illustrates that the "causality" of the causal structure for small systems is very hard to judge, since configurations that are derived from continuouse Dirac Sea configurations via rounding do not show clearly something similar to a light cone. One might object that these causal structures does not correspond to a minimum of the action, but this is not the point here. We want to get a notion of discrete configurations which can be considered as discrete analoga of continuouse Dirac Sea configurations. If minima of discrete configurations are in some way similar to those analoga, the main hypothesis of this thesis would be

confirmed. In comparison to figure 5.5, figure 5.6 shows the same systems as figure 5.5 configured randomly.

5.2 Varying the Mass

The first calculation presented in this section only illustrates how the causal structure of Dirac Sea like discrete configurations as discussed in the last section behave when varying the mass. Figure 5.8 show configurations with $n = 25$, $N = 3$ and different mass parameters. All these configurations show some kind of causal structure, even though higher mass parameters seem to distort this structure slightly. On the other hand choosing higher mass parameters may be considered as focusing on smaller energies. Remarkably in this case the portion of spacelike points near $r = 0$ decline. Further the set of points contributing to the action and expected to be spacelike raises with raising the mass parameter.

The second calculation presented in this section discusses a method of analysis which might inspire a complete new research program beside the program of this work. Thus, the calculations here are of no systematical nature and aim only on illustrating the idea.

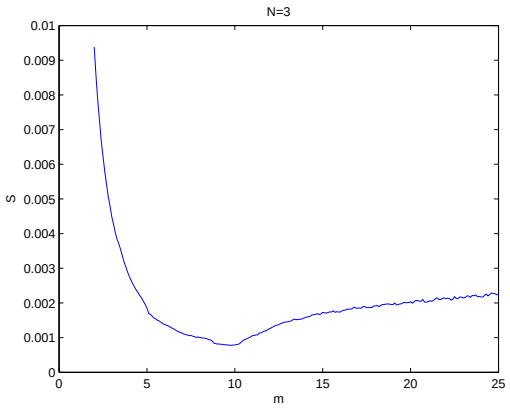


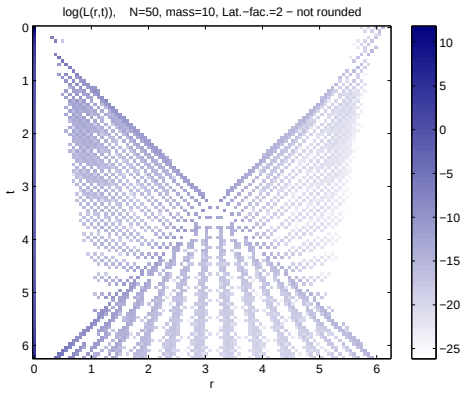
Figure 5.7: $S(m)$

Assuming that indeed the action principle of the Fermionic Projector as applied in this work leads to Dirac Sea like minimisers, one could find a way to parametrise sets of Dirac Sea like configurations and do optimisation on the parameters. In this chapter we discussed Dirac Sea like configurations parametrised by the mass parameter m, and figure 5.7 shows the graph of $S(m)$ for a system with $n = 25$, $N = 3$ and $m \in [3, 25]$. The action has one minimum quite exactly for $m = 10$. In the framework of the theory of the

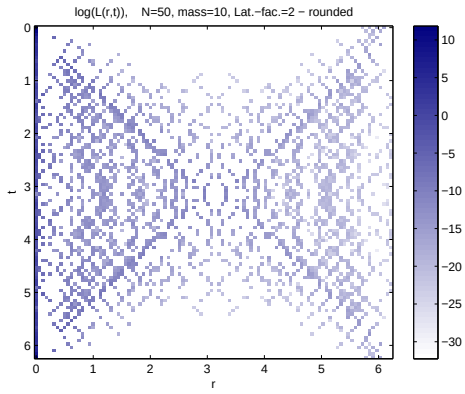
Fermionic Projector one would actually expect not one but three Dirac Seas according to the three generations of elementary particles. This would lead to a multidimensional optimisation program on the action of the Fermionic Projector of such systems.

5.3 Conclusion

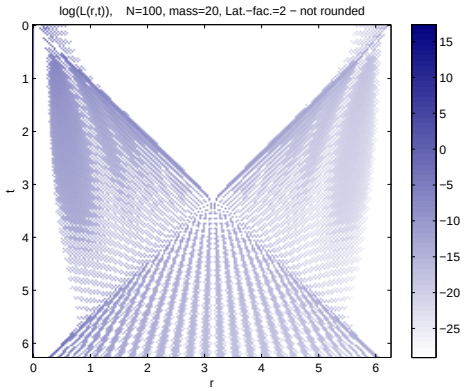
This chapter gives a concrete idea of the causal structures, which are to be due if the variational principle has some physical relevance. It was also made clear that the estimation of causal structures of small systems will be difficult. Unique results are to be due not until system sizes $n \approx 50$.



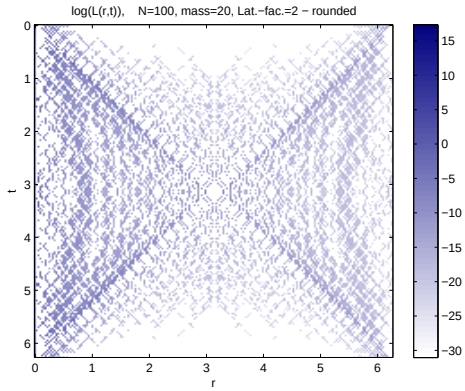
(a) $n = 50$, $m = 20$, $N = 2$, cont.



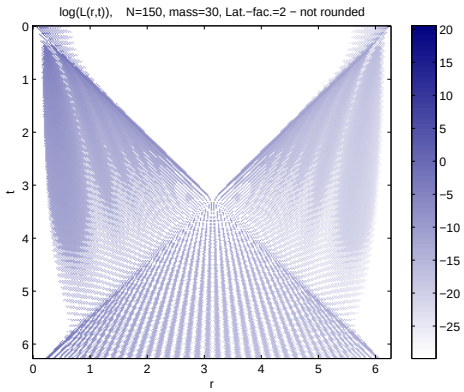
(b) $n = 50$, $m = 20$, $N = 2$, disc.



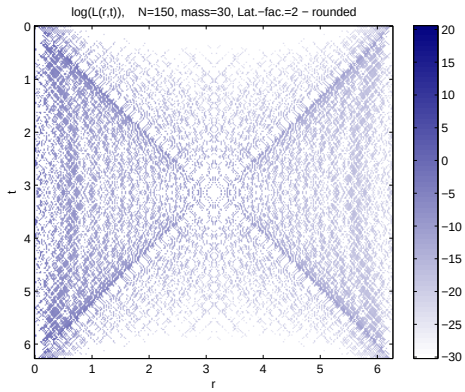
(c) $n = 100$, $m = 20$, $N = 2$, cont.



(d) $n = 100$, $m = 20$, $N = 2$, disc.

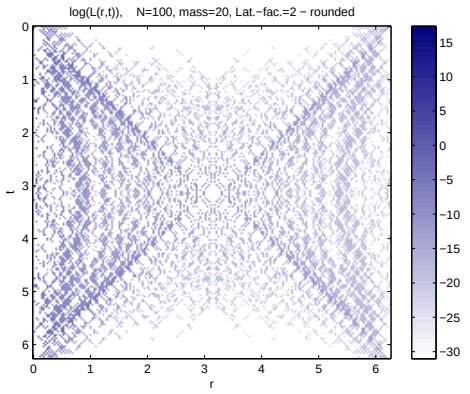


(e) $n = 150$, $m = 30$, $N = 2$, cont.

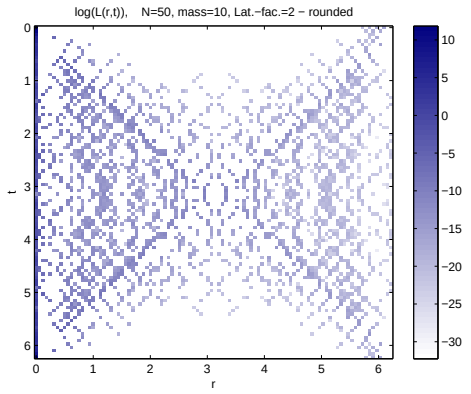


(f) $n = 150$, $m = 30$, $N = 2$, disc.

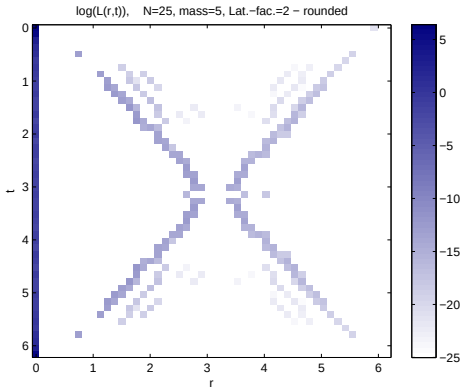
Figure 5.4: Causal Structures of Dirac-Sea like configurations



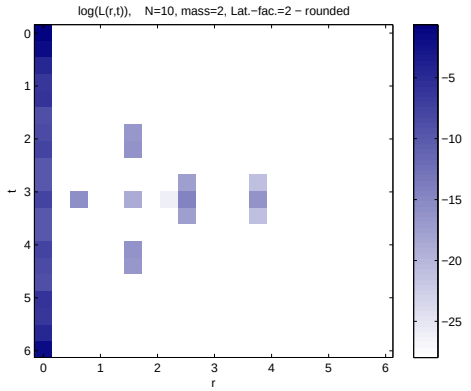
(a) $n = 100, m = 20, N = 2$



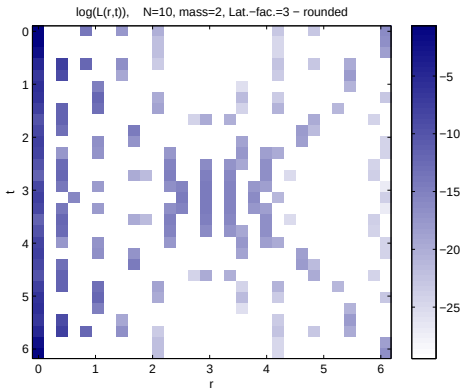
(b) $n = 50, m = 10, N = 2$



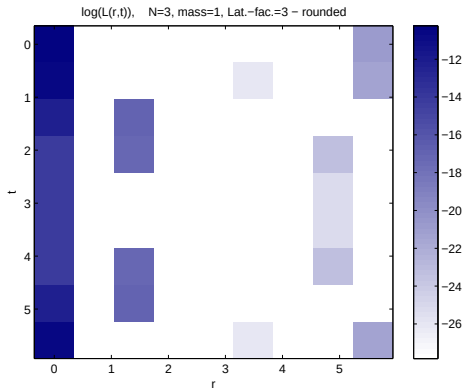
(c) $n = 25, m = 5, N = 2$



(d) $n = 10, m = 2, N = 2$

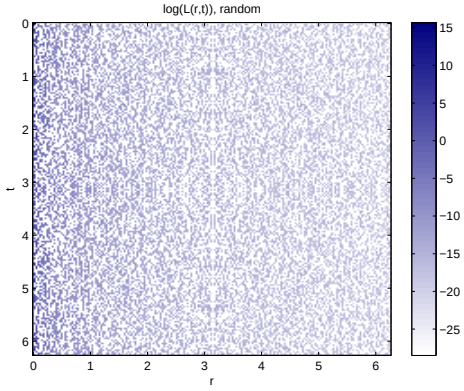


(e) $n = 10, m = 2, N = 3$

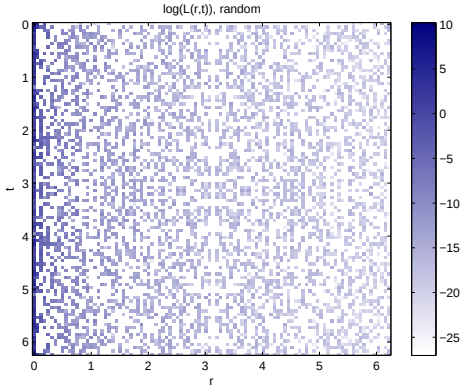


(f) $n = 3, m = 1, N = 3$

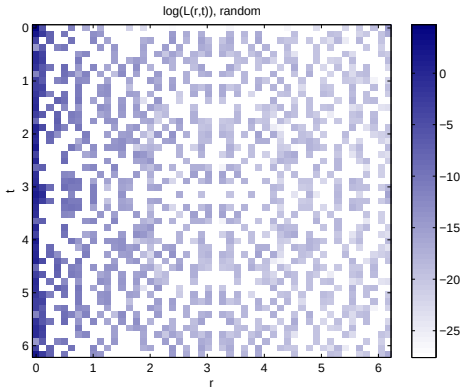
Figure 5.5: Causal Structures of Dirac-Sea like configurations



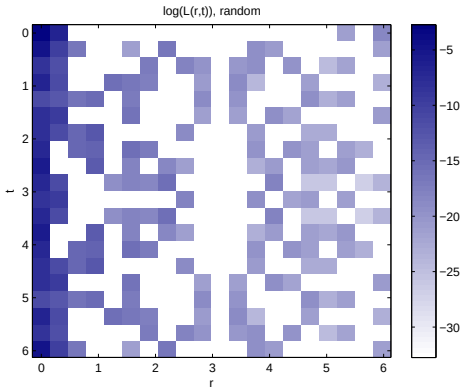
(a) $n = 100, m = 20, N = 2$



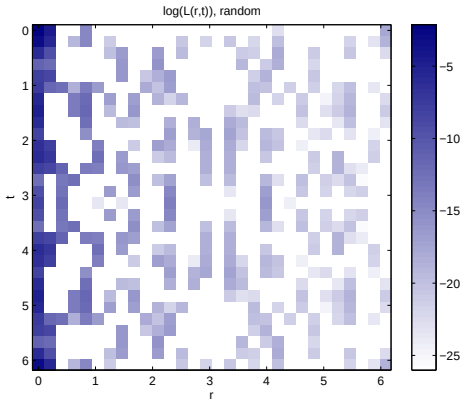
(b) $n = 50, m = 10, N = 2$



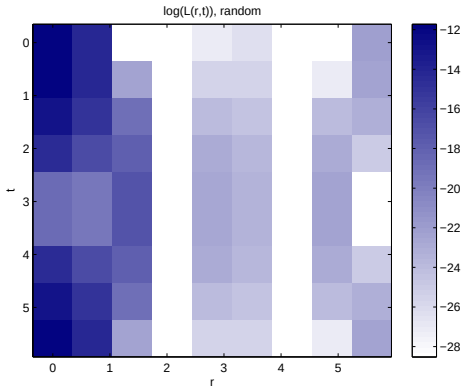
(c) $n = 25, m = 5, N = 2$



(d) $n = 10, m = 2, N = 2$

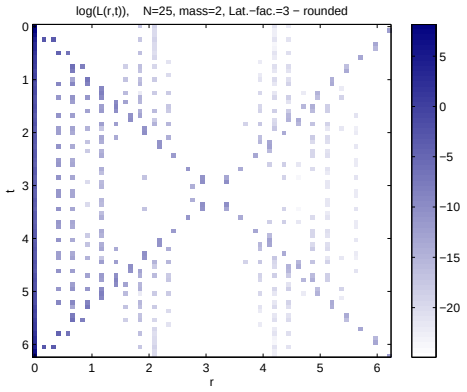


(e) $n = 10, m = 2, N = 3$

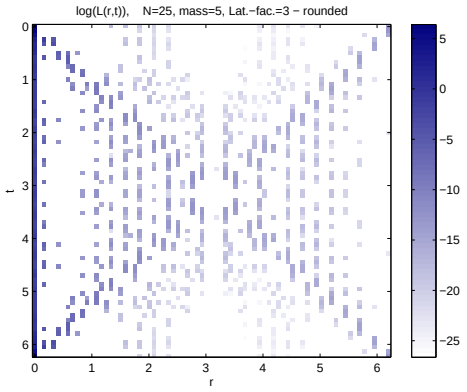


(f) $n = 3, m = 1, N = 3$

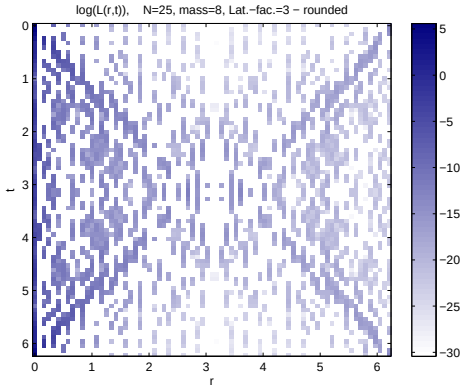
Figure 5.6: Causal Structures of Random Configurations



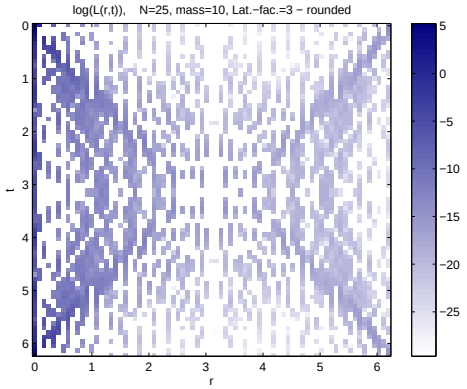
(a) $m = 3$



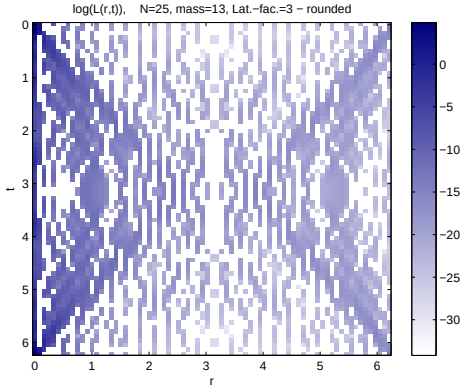
(b) $m = 5$



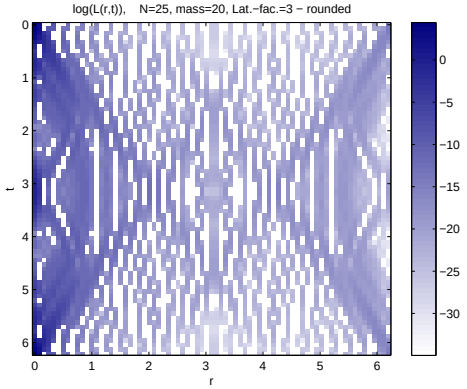
(c) $m = 8$



(d) $m = 10$



(e) $m = 13$



(f) $m = 20$

Figure 5.8: Causal Structures for Configurations (5.2) with $N = 3$, $n = 25$ and different mass parameters

Chapter 6

Multiple GPS Search

The calculations discussed in this chapter differ from those in chapter 4 by using the GPS algorithms for both variables τ and ω. Hence according to chapter 2 we now apply the MGPS variant of the GPS algorithm. We will also pay attention to the discussion of the *causal structure* as introduced in chapter 5. Further we will choose for thr further work according to Section 4.4 $N = 3$. We choose the convention for the following work to declare instead of ω the vector $(|\omega_i|)_{i=1}^n$.

6.1 $n = 3$

For the first calculations discussed in this chapter we chose

$$\omega \in \{0, ..., \omega_{\max}\}^n =: \Omega \tag{6.1}$$

with $n = 3$ and $\omega_{\max} = 2$. 1000 runs of the standard MGPS algorithm were performed. The start values were selected randomly

$$\omega^0 \in \Omega \quad \text{and} \quad \tau^0 \in [-\tau_{\max}, \tau_{\max}] \tag{6.2}$$

The analysis started by sorting the solutions by the final value of the action. During this work this will often be a useful tool of analysis. As a result we get a function $S(\nu)$ $(1 \leq \nu \leq \nu_{\max})$ as plotted in figure 6.1. This diagram suggests to classify the solutions by the value of the action. As charted in figure 6.1 8 domains can be distinguished.

Before we discuss these domains we have to explain another method of displaying the results from numerous optimisation runs. Let $N \subset \{1, ..., \nu_{\max}\}$ and $\tau_{\min}$ belonging to the

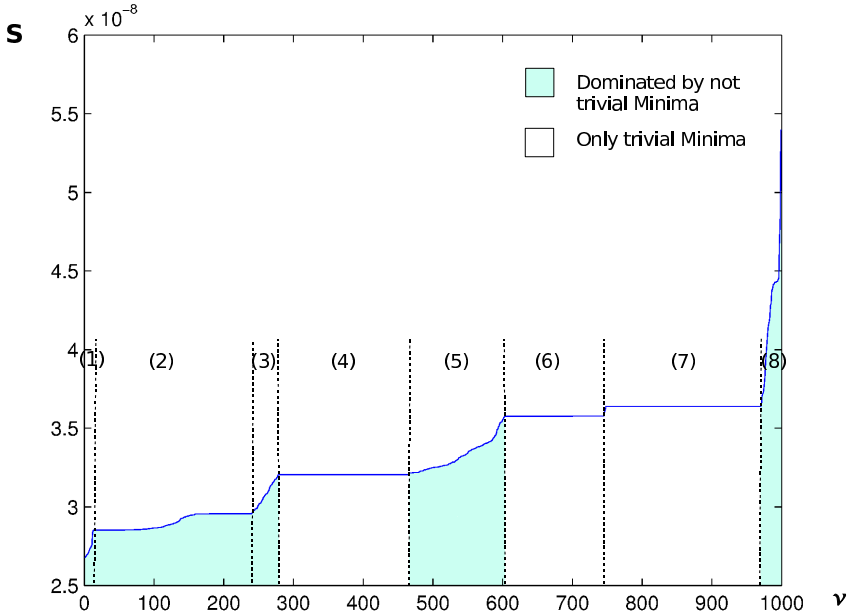


Figure 6.1: $S(\nu)$ ($n = 3$, $N = 3$, $\tau^0 \in [-0.85, 0.85]^3$ and $\omega^0 \in [0, 2]^3$), 1000 runs.

minimal solution on N, then we define

$$\Delta\tau^r(\nu) := \frac{\|\tau_{\min} - \tau(\nu)\|_2}{\|\tau_{\min}\|_2} \tag{6.3}$$

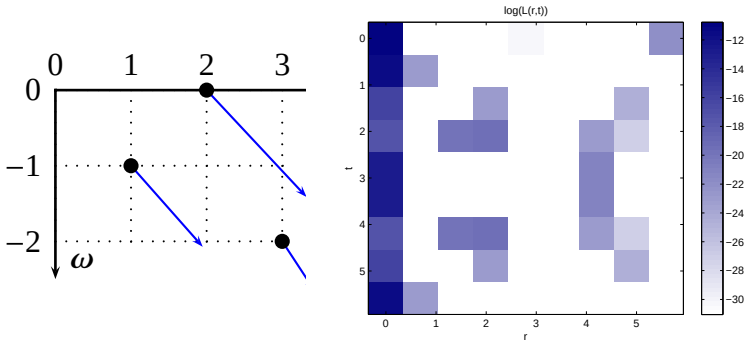
as a quantity, which should give information about the variation of an vector $\tau(\nu)$ from $\tau(\tilde{\nu})$ of the relative numerical minimiser $S(\tilde{\nu}) = \min_{\nu \in N} S(\nu)$, and which is normalised in just a canonical way. A two dimensional plot of the graph $(S(\nu), \Delta\tau^r(\nu))$ will be called a *convergence diagram*. Naturally convergence diagrams are only a usefull tool for analysing optimisation results, if the number of optimisation runs is sufficiently high.

Figure 6.6 illustrates some convergence diagrams, which are discussed in detail in the next paragraph. So far we want to discuss in general the information convergence diagrams provide. Subfigure (a) of Figure 6.6 charts the convergence result of a whole 1000-run calculation using GPS, while (b) to (d) chart details of (a). Diagrams (c) and (d) are likely to belong to numerical minimisers, which approximate the same minimiser. On the one hand $\max \Delta\tau^r \ll 1$ and on the other hand $\rho \gg 0.5$, whereas ρ denotes the coefficient of correlation. In other words the variation of τ throughout the considered numerical minimisers is small whereas there is a strong correlation between the variation of τ and the value of the action, which should correspond to real local minimisers. Contrary to this the case $\rho \lesssim 0.5$ and $\max \Delta\tau^r \gtrsim 1$ as in subfigure (a) and (b) should refer to subsets

N which count numerical minimisers belonging corresponding to more than one local minimiser. Note that the illustrations 6.1 and 6.6 do not give any information about the occupations $\omega(\nu)$.

We now want to discuss the domains labeled in figure 6.1:

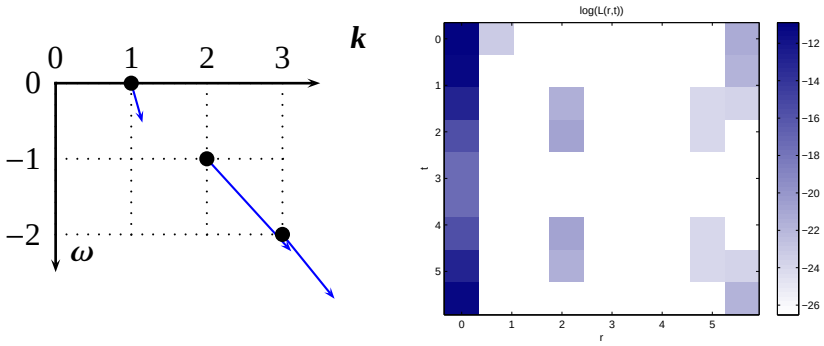
(1) This domain ($1 \leq \nu \leq 12$) belongs to optimisation endpoints with $\omega = (1, 0, 2)$ (see figure 6.2). The convergence diagram of this domain is shown in figure 6.6 (a).



(a) Configuration (b) Causal Structure

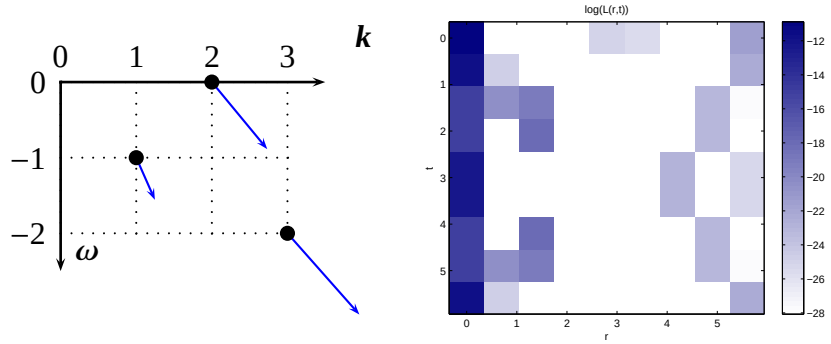
Figure 6.2: Numerical Minimiser ($\nu = 1$) with $\tau = (-1.3967, -1.6998, -0.7858)$ and $S = 2.6772007e - 08$.

(2) This domain contains two plateaus. The lower is highly dominated by occupations belonging to $\omega = (0, 1, 2)$ (see figure 6.3) while the upper is dominated by occupations belonging to $\omega = (1, 0, 2)$. The convergence diagram of the lower plateau is shown in figure 6.6 (d).



(a) Configuration (b) Causal Structure

Figure 6.3: First Numerical Minimisers of the domain (2), $\nu = 13$, $\tau = (-0.2908, -1.5439, -1.1258)$, $S = 2.8515584e - 08$.



(a) Configuration (b) Causal Structure

Figure 6.4: Second Numerical Minimisers of the domain (2), $\nu = 145$, $\tau = (-0.475, -1.1758, -1.3994)$, $S = 2.9359701e-08$.

A closer look at figure 6.6 (d) raises the question why quite numerous numerical solutions (end points of an optimisation run due to 4.1) are placed close to a straight line in the $(S, \Delta\tau^r)$-space. Comparing with figure 4.7 we can qualitatively understand the action as a sum of a convex differentiable function with the only minimum at $\tau = 0$ and negative contributions which lower the actions in certain regions of the τ-space in such a way that in the end we have a steady but non differentiable function. The non-trivial contributions causes non-trivial minima, which are located in some kind of "canyons". That is the action increases in almost every direction around the minimum fast but only in one or a few directions slowly. Once reached the bottom of the canyon, it is quite hard for the GPS algorithm to find improving points, especially near the minimum, as illustrated in figure 6.5. Thus it is reasonable to suppose that the numerical solutions located at the mentioned lines correspond to end points located in a "canyon" which also contains a minimiser. As already suggested by figure 4.7 *trivial minima* seem to have a very good convergence behaviour thus trivial minima – as seen below – are attained with high precision by the GPS algorithm. Contrary *non-trivial minima* have worse convergence behaviour, hence we get sets of end-points approximating "canyon-like" non trivial minimisers. As a consequence, convergence diagrams focus on the representation of non trivial minima, since trivial numerical minimisers belonging to the same value of the action even if numerous should concentrate at *one* point in the $(S(\nu), \Delta\tau^r(\nu))$-plane.

(3) This domain is still dominated by occupations of the class $\omega = (1, 0, 2)$ and also the τ-values are in the surrounding of these in domain (2) belonging to the same equivalence class. But the variation of the final τ-values increases considerably, so these solutions can be considered belonging to runs, which failed to reach the minimiser approximated in domain (2).

(4) This domain belongs to the trivial minimum with occupations in $\omega = (1, 0, 2)$.

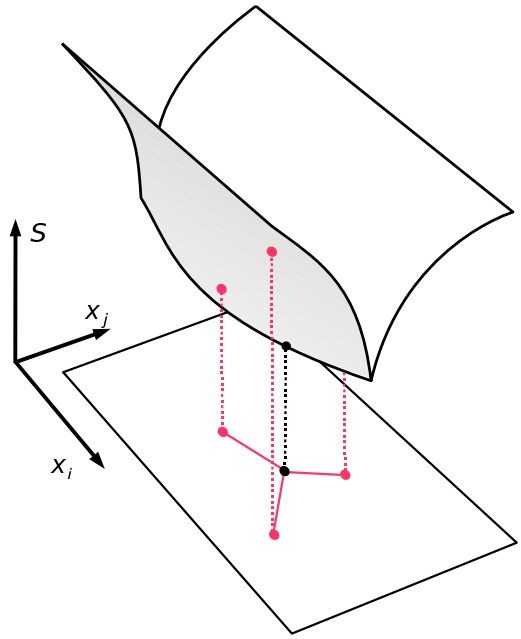
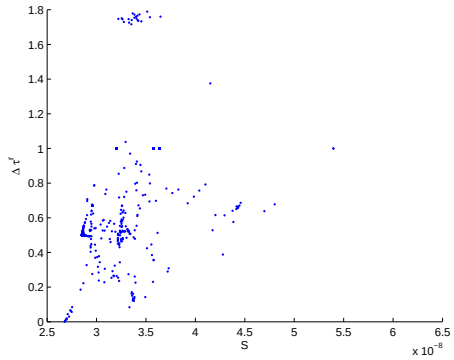


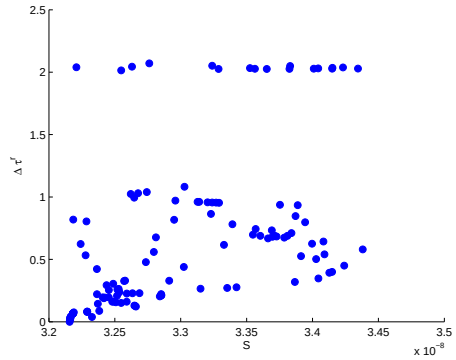
Figure 6.5: Shematic illustration of the action near a minimiser with an incumbent mesh point and a set of positive spanning mesh points subset of the poll set.

(5) This domain is quite similar to domain (3) also the variety of occupations is much wider.

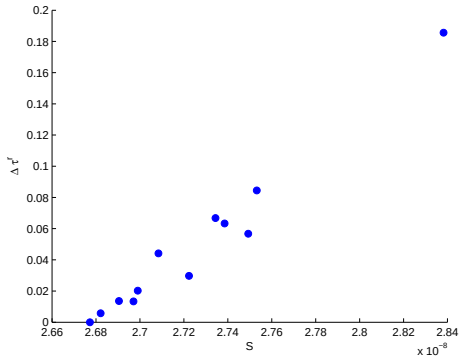
Domain (6) and (7) belong again to trivial minima and domain (8) is again similar to domain (3) and (5).



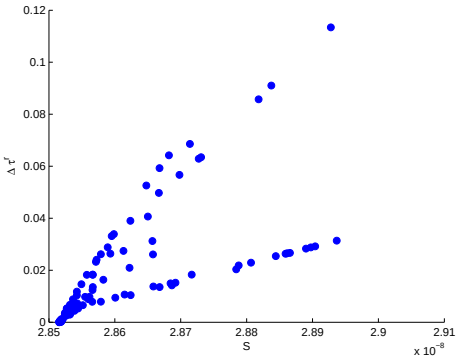
(a) $1 \leq \nu \leq 1000$, ($\sigma = 0.481$)



(b) $466 \leq \nu \leq 587$, ($\sigma = 0.501$)



(c) $1 \leq \nu \leq 12$, ($\sigma = 0.981$)



(d) $(13 \leq \nu \leq 104) \vee (106 \leq \nu \leq 128)$, ($\sigma = 0.707$)

Figure 6.6: $\tau_r(S)$ as defined in (6.3) for different domains.

To sum up: For the considered system there were numerically found three non-trivial minima, all pictured in this subsection. Aside there where also found three trivial minima, which all have higher action than the non-trivial minima. With the exception of some solutions with $\nu \lessapprox 1000$ all final occupations – not only that of the numerical approximations of the minimisers – are bijective (interpreted as mappings $\omega : \{0, 1, 2\} \to \{0, 1, 2\}$). And nevertheless all τ_i have the same sign. Compared to the combination of complete enumeration (in the discrete variables) and GPS-search (in the continuous variables) the method of multiple GPS-searches in both variables works much better. The optimal solution found by the combined method could not only be reproduced and with respect to the value of the action be enhanced, but one more optimal numerical minimiser was found.

One could ask why no systems with $\omega_{\max} > 2$ ($n = 3$) were analysed. There are mainly two reasons: First, all naively discretised Dirac Seas lie are contained in the class $\omega_{\max} =$

$n-1$. As a consequence of the symmetry

$$\omega_i \to \omega_i + \Delta\omega \tag{6.4}$$

we can always set $\omega(1) = 0$. It is reasonable not to enlarge the class of varied configurations without any striking reason to keep the numerical afford as small as possible.

For the second reason we have to consider the discrete symmetries concerning the variable ω. As figured out in figure 6.7 the number of symmetries increases dramatically with raising n. Note that for this table $\omega_{\max} = 2n-1$. For combinatorial reasons the numerical determination of the symmetry classes gets very fast out of reach. Most symmetries are understood well like the symmetry (6.4) and the symmetry

$$\omega_i \to -\omega_i + \Delta\omega. \tag{6.5}$$

But as one can see with a look at the example listed in the Appendix B.4 not all symmetries determined numerically are of the form (6.4) or (6.5). For the purpose of this thesis it is important to determine if a occupation belongs to a class, which also includes Dirac Sea like occupations or not. The pure result of an optimisation run is not significant and if the classes of equivalent occupations of a system is not known a final occupation cannot be judged, whether it represents a Dirac Sea like occupation or not. Note that the discrete symmetries in the ω-space makes our definition 3.1 ambiguous. In the further thesis this notion should be understood in such a way, that we call a occupation, i.e. equivalence class of occupations *Dirac Sea like* if it contains a Dirac Sea like pair (ω, τ). Further: since we want to step forward to more complex systems with $n = 5$ and $n = 6$ by a systematic strategy to consider a fixed class of systems only distinguished by the lattice size n (see chap. 8 and 9) it is reasonable to concentrate even for $n = 3$ on a system, whose enlarged versions are numerically controllable as far as possible.

n	$\omega_{\max}$	4	6	8	10	12	14	16	20	22	24	26	28	32	56	Frac.
3	5		4			6		3			3					100 %
4	7	115		89	67	73		54		7						100 %
5	9	360		480	496	1035	240	1050	1020		405	360	120	75	15	89.7 %

Figure 6.7: Number of classes with equivalent occupations (determined numerically) ordered by cardinal number. The last column names the fraction of occupations which belong to classes with higher cardinal number then 1.

6.2 $n = 4$

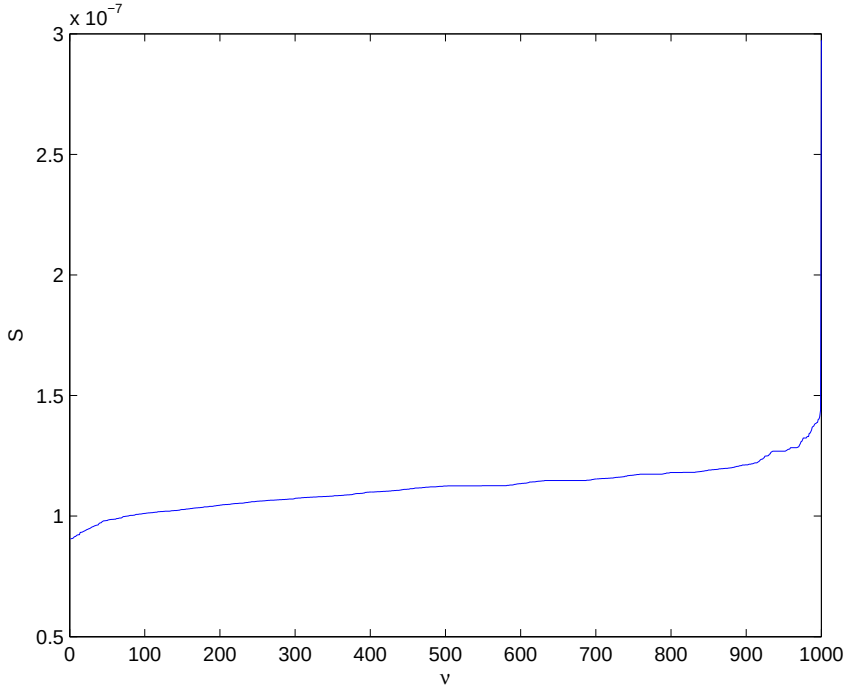
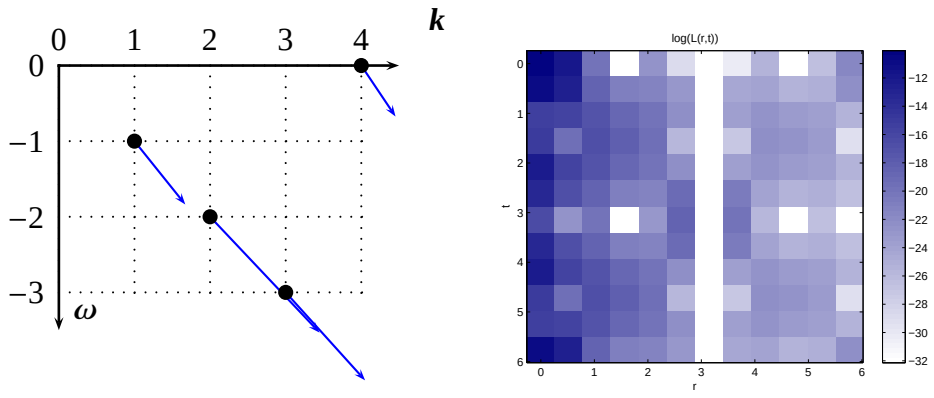


Figure 6.8: $S(\nu)$, ($n = 4$, $N = 3$, $\omega_{\max} = 3$)

In contrast to the system discussed above (sec. 6.1) the plot $S(\nu)$ (fig. 6.2) does not lead to a distinct classification into several domains. In fact the different equivalence classes of the end points (ordered by ν) are much more mixed up. The numerical minimiser is shown in figure 6.9.



(a) Configuration (b) Causal Structure

Figure 6.9: Numerical Minimiser for $n = 4$, $N = 3$, $\omega_{\max} = 3$.

6.3 Conclusion

We have not discussed yet the causal structures of the numerical determined minima of this chapter. All configurations discussed in this chapter (cf. fig. 6.2, 6.3, 6.4 and 6.9) correspond to causal structures, which qualitatively confirm the general hypothesis of this thesis. According to chapter 5 the relatively higer values of the Lagrangian in the region $r \neq 0$ of figure 6.9 (case $n = 4$) can be interpreted in the way that the "mass" of this system is higher, allthough a strong notion of mass is not a priori given for discrete fermionic systems. In this chapter there was no analysis done which brought us to the analysis of larger systems. Instead the new method of MVP was applied to our optimisation problem.

This chapter also shows that it is not trivial to handle the struggles of optimisation for our problem. Even the transition from $n = 3$ to $n = 4$ changes the complexity of the problem quite considerable. While for $n = 3$ $S(\nu)$ shows clearly the minima as plateaus, for $n = 4$ these plateaus vanish mostly, which raises the question, how to judge the quality of an numerical minimiser, since there has to be done some consideration concerning the problem of "false minima", i.e. endpoints of numerical optimisation algorithms, which has to be considered as failed attempts to archive a minimum.

Chapter 7

Performance and Quality Comparison between discretised and relaxed Search

In the remaining work we will numerically analyse systems with larger n. Since therefore runtime becomes a critical parameter, we will discuss some profiling results, which help us selecting the proper optimisation strategy. Particularly we will compare relaxed and non-relaxed handling of our optimisation problem.

7.1 Performance Comparison

In this section we will discuss the practical issue of runtime concerning the two search strategies using the GPS MVP algorithm and relaxing the problem and using the GPS NLP algorithm. The following table shows the comparison of runtimes using the NLP algorithm, the MVP algorithm without EXTENDED SEARCH step and the MVP algorithm with EXTENDED SEARCH step.

The heuristic idea for relaxing our optimisation problem is to make the problem achievable for NLP algorithms, which usually allow the application of much more powerful optimisation methods. But most of the advantage of applying usual NLP algorithms get lost by the fact that the action is not differentiable, so the large class of powerful optimisation methods using derivative information cannot be applied.

For $n = 3$ to $n = 6$ (and $N = 1$) the profiling leads to the result, that relaxation does not generate a significant profit of numerical performance. Contrariwise the relaxed run often

took the longest runtime (see figure 7.1)[1].

n	Method	Function Name	Calls	Total Time
3	MVP	mads	100	1404.280 s
		ferm_proj_4	38089	424.213 s
	MVP with ex. Poll	mads	100	1371.311 s
		ferm_proj_4	39178	437.556 s
	NLP	mads	100	1284.498 s
		ferm_proj_5	34286	419.638 s
4	MVP	mads	100	2282.850 s
		ferm_proj_4	61963	813.260 s
	MVP with ex. Poll	mads	100	2624.779 s
		ferm_proj_4	71539	894.685 s
	NLP	mads	100	3217.906 s
		ferm_proj_5	63623	858.592 s
5	MVP	mads	100	3451.981 s
		ferm_proj_4	98213	1301.683 s
	VP with ex. Poll	mads	100	3394.048 s
		ferm_proj_4	105438	1275.569 s
	NLP	mads	100	4866.393 s
		ferm_proj_5	97317	1471.130 s
6	MVP	mads	100	7004.313 s
		ferm_proj_4	146115	2830.529 s
	MVP with ex. Poll	mads	100	8111.797 s
		ferm_proj_4	159538	2808.306 s
	NLP	mads	100	8976.190 s
		ferm_proj_5	152753	3127.654 s

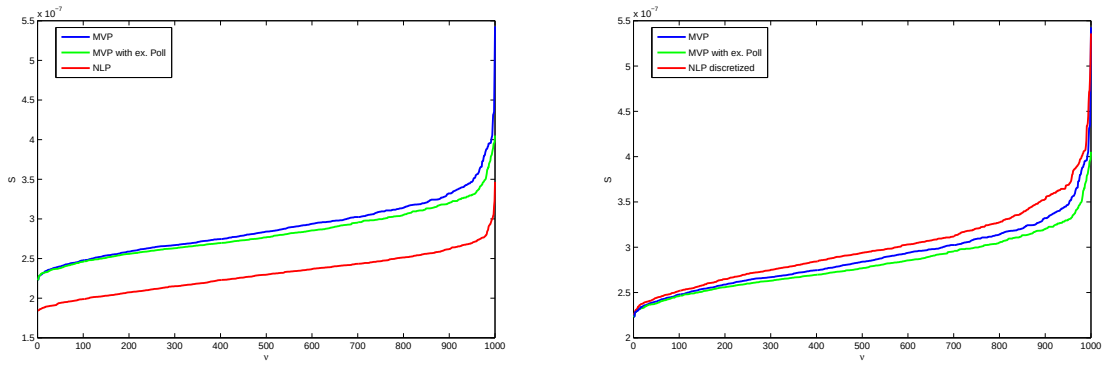
Figure 7.1: Profiling data for different methods ($n = 3$, $N = 1$, 100 runs)

7.2 Quality Comparison

Runtime optimisation is only one aspect of improving numerical optimisation methods. Another crucial aspect is the quality of the resulting solutions. To figure out this aspect

[1] mads is the core optimisation algorithm (without GUI) and ferm_proj_4 and ferm_proj_5 are the main function evaluation functions. The starting points were chosen randomly and for each run separately.

1000 runs of a system ($n = 5$, $N = 1$) were taken out with different variants of the GPS algorithm. First the problem without relaxation was calculated with the standard MVP algorithm. Second this algorithm was extended by the EXTENDED POLL STEP. Third the problem was relaxed and run by the NLP variant of the GPS algorithm. The result was discretised by rounding, since a systematic search at all corners of the unit cube around a numerical solution of the relaxed problem would raise the additional numerical cost exponentially with system size. The results are plotted in figure 7.2.



(a) NLP results not discretised

(b) NLP results discretised

Figure 7.2: $S(\nu)$ for different optimisation methods

7.3 Conclusion

Although the results of the relaxed runs are much lower then that of the other methods, this advantage dissappears with discretisation and the NLP method turns out as the poorest and the MVP method with EXTENDED POLL STEP as the best method. This result still holds for $\nu = 1$ which is not matched properly by the resolutuion of the figure. Nevertheless this result still depends on the randomly chosen starting points as well as the specific system parameters, expecially the system size. It might be appropriate to reply this quality check and also the runtime check in the last subsection if considerable different systems are analysed. But until further notice the MVP algorithm with EXTENDED POLL STEP has to be considered as the best choice.

Chapter 8

Local Search with slowly increasing n

8.1 Preparation

As we have seen above the complexity of the systems raises drastically with increasing n so that a global understanding even of systems about $n = 5$ to $n = 10$ seems out of reach. The general idea of the calculations in this section is to do the minimisation for some low n, in our case $n = 3$, and to use the global numerical optimiser to construct a proper starting point for a system with one more occupied state. The new starting point is then used for a local search so that it might be possible to construct local numerical minimisers starting from a well understood small system.

8.1.1 "Adding a Particle"

To do the transformation $n \to n + 1$ we make the settings

$$\hat{\tau}_1 = \tau_1, \quad \hat{\tau}_{n+1} = \tau_n$$
$$\hat{\tau}_k = \left(1 - \tfrac{k-1}{n}\right)\tau_{k-1} + \tfrac{k-1}{n}\,\tau_k \quad \text{for} \quad 1 < k < n+1$$

and

$$\hat{\omega}_1 = \left\lfloor \tfrac{n}{n-1}\,\omega_1 + \tfrac{1}{2} \right\rfloor, \quad \hat{\omega}_{n+1} = \left\lfloor \tfrac{n}{n-1}\,\omega_n + \tfrac{1}{2} \right\rfloor$$
$$\hat{\omega}_k = \left\lfloor \tfrac{n}{n-1}\left(\left(1 - \tfrac{k-1}{n}\right) + \omega_{k-1} + \tfrac{k-1}{n}\,\omega_k\right) + \tfrac{1}{2} \right\rfloor, 1 < k < n+1$$

whereas $[\cdot]$ denotes the Gauss brackets, (τ, ω) the absolute numerical minimiser for the system of size n and $(\hat{\tau}, \hat{\omega})$ the scattering centre for the system of size $n + 1$.

It has to be mentioned that this choice of "adding a particle" is only canonical to a certain point. It's advantage is that it is unique and conserves the "global shape" of the configuration in some way, since it is similar to scaling a digitalised image to a bigger size. Another method would be just simply to add a particle at $k = n + 1$, but then ω_{n+1} and τ_{n+1} has to be determined in some way.

8.1.2 The Method of Scattering

For the purpose of local search it is nessesary to find a method to scatter random starting points around an initial point. In the following we will concentrate on the scattering process for the τ_i, regarding that the process is completely analogue for the ω_i except the fact that usually the scattering has to be completed by rounding to the ω-lattice.

All scattering methods which could be implemented have to depend on a numerical random functions. These functions can formally described as mappings

$$\begin{array}{ccc} \mathbb{N} & \longrightarrow & [0,1] \\ n & \longmapsto & \rho(N_0 + n) \end{array} \tag{8.1}$$

whereas n is the actual counting index of the algorithm and N_0 is an offset corresponding to natural time.[1] The obvious choice of scattering methods as used in chapter 4 to 6 would be to choose a range $\Delta\tau$ and to set

$$\tau_i^{(n)} = \tau_i^0 + (2\rho(n) - 1)\Delta\tau. \tag{8.2}$$

But this approach has several disadvantages: First the used implementation of the GPS algorithm requires the discrete and the continuous variables to be bounded, i.e. for the variables τ_i

$$\mathcal{l}_\cdot \leq A\tau_\cdot \leq u \tag{8.3}$$

with the invertible matrix A and the lower and upper boundary vectors $\mathcal{l}$ and u. (By the way we set generally

$$A = \mathbb{1} \qquad \text{and} \qquad \mathcal{l} = \tau_l \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \quad \text{as well as} \quad u = \tau_u \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}.) \tag{8.4}$$

[1] Naturally ρ has to fullfill some homogeneity condition making it a real random function, e.g.

$$\forall\, a, b \in [0.1], \quad a < b \quad \lim_{N\to\infty} \frac{\sharp\{\rho(n) \in (a,b) \,|\, n \leq N\}}{N} = b - a.$$

Due to condition (8.3) the scattering method (8.2) has to be modified in some way near the boundaries. Second for the purpose of local search and under the condition that the variational principle is not local it seems to be appropriate not to disturb the global structure of the initial starting points too much but on the other hand not to be too restrictive in selecting random starting points. In practice a small value of $\Delta\tau$ in (8.2) might lead to a very restricted set of numerical minimisers while a too big value of $\Delta\tau$ might destroy the global structure too much. Unlike this two border cases it might be interesting for small systems only to alter one value τ_i considerably or only one value ω_i. These variations are not well modelled by method (8.2).

These reasons lead us to another method of scattering around some given pair (τ, ω). The advantage is that the range of the scattering can very easily be controlled via one parameter per dimension without getting in conflict with any boundary condition. With the decision functions

$$\Gamma^{-}(n) := \tfrac{1}{2}\left(1 - \operatorname{sign}\left(\tau^l + \rho(n)(\tau^u - \tau^l) - \tau^0\right)\right) \tag{8.5}$$

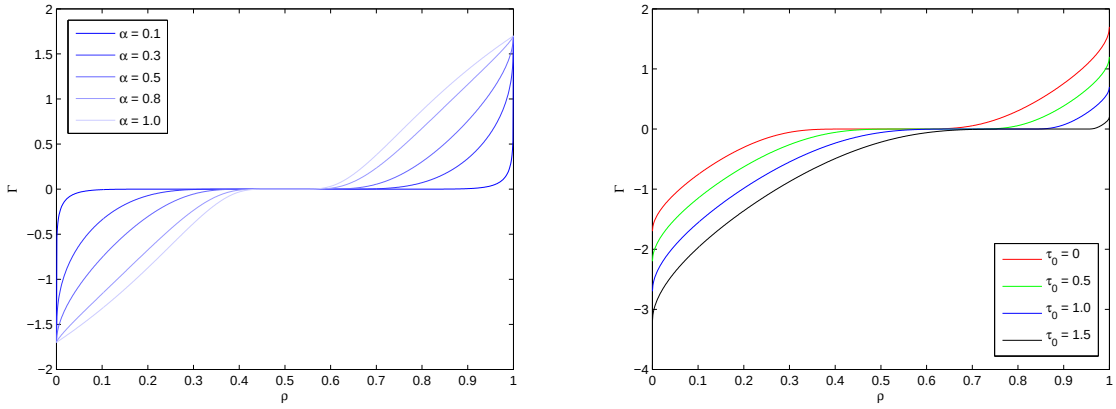
and

$$\Gamma^{+}(n) := \tfrac{1}{2}\left(1 + \operatorname{sign}\left(\tau^l + \rho(n)(\tau^u - \tau^l) - \tau^0\right)\right) \tag{8.6}$$

we set

$$\begin{aligned} \tau(n) &= \tau^0 + \Gamma^{-}(n)\,(\tau^l - \tau^0)\,\exp\left(1 - \left(1 - \frac{\rho(n)}{\frac{\tau^0-\tau^l}{\tau^u-\tau^l}}\right)^{-\alpha}\right) + \\ &\qquad \Gamma^{+}(n)\,(\tau^u - \tau^0)\,\exp\left(1 - \left(1 - \frac{1-\rho(n)}{\frac{\tau^u-\tau^0}{\tau^u-\tau^l}}\right)^{-\alpha}\right) \\ &\equiv \tau^0 + \Gamma(n) \end{aligned} \tag{8.7}$$

If we consider Γ not as a function from $\mathbb{N} \to \mathbb{R}$ but from $[0,1] \to \mathbb{R}$ declaring the values of Γ for all possible random numbers $\rho(n)$ we can easily plot it (see fig. 8.1).



(a) varying α, $\tau^0 = 0$ (b) varying τ^0, $\alpha = 0.5$

Figure 8.1: $\Gamma(\rho)$ for $\tau^l = -1.7$ and $\tau^u = 1.7$

Obviously the scattering method is designed in a way that always the full domain of a variable given by the boundary conditions is covered (see figure 8.1(b)). The degree of scattering is controlled by the parameter α (see figure 8.1(a)).

8.1.3 Tuning the Scattering Factors α_ω and α_τ

During the calculations it turned out that the optimality of the numerical minimisers might heavily depend on the proper choice of α_ω and α_τ. To avoid here any arbitrariness we performed test calculations for $n = 5$, $N = 3$ and

$$\alpha_\omega \in \{0.2, 0.4, ..., 1\} \quad \text{and} \quad \alpha_\tau \in \{0.2, 0.4, ..., 1\} \tag{8.8}$$

From these calculations the graphs of $S(\nu)$ where plotted and qualitatively the "best" combinations where selected. (See figure 8.2. All plots are found in the appendix B.3.) We get the result that the numerical minimisers are best for $0.6 \leq \alpha_\omega \leq 0.8$ and $\alpha_\tau \leq 0.4$. Hence we will choose in the following calculations $\alpha_\omega = 0.7$ and $\alpha_\tau = 0.3$.

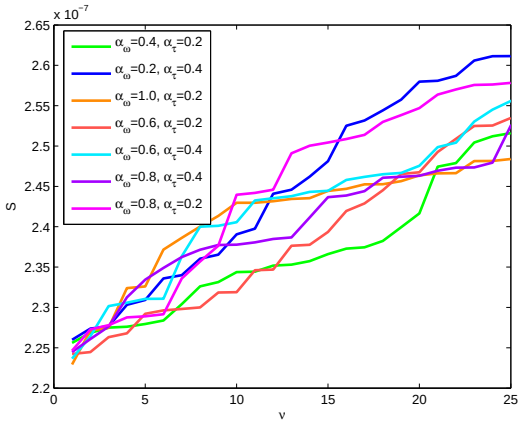


Figure 8.2: $S(\nu)$ for different combinations of α_ω and α_τ

8.2 The Calculations

As above the Calculations in this section will all be performed with the lattice factor $N = 3$.

8.2.1 $n = 3$

Evaluation of the Numerical Minimiser

The calculation done in this section is just similar to that in section 6.1 and it leads to the same numerical minimiser. This was taken as basis for the further calculations with slowly increasing n.

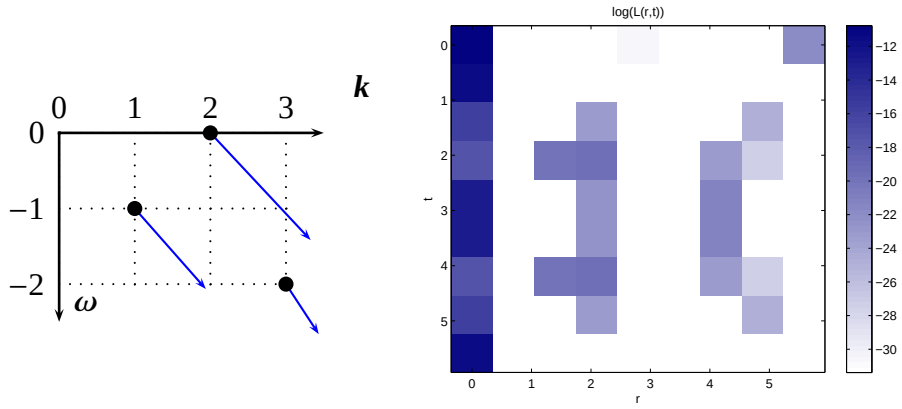


Figure 8.3: Numerical Minimiser for $n = 3$

The Action around the Numerical Minimiser

To illustrate the qualitative behaviour of the action it is appropriate to consider sections of the action around the numerical minimiser of this subsection. In figure 8.2.1 graphs of the function

$$S(t) = S(\omega^{\min}, \tau^{\min} + t\tau^{\text{var}}) \qquad \text{with} \quad t \in [-0.5, 0.5]. \tag{8.9}$$

and τ^{var} fulfilling the condition $||\tau^{\text{var}}||_2 = 1$.

The plots in figure 8.2.1 do not only demonstrate again the non differentiable character of the action, but also depict the fact that concerning the variations in the τ-space there exist preferred directions, which contain variations of multiple τ-values. While for the black plot only τ_1 was varied, the green plot shows variations basically of τ_1 and τ_3, which in some kind seem to "compensate" each other in some way.

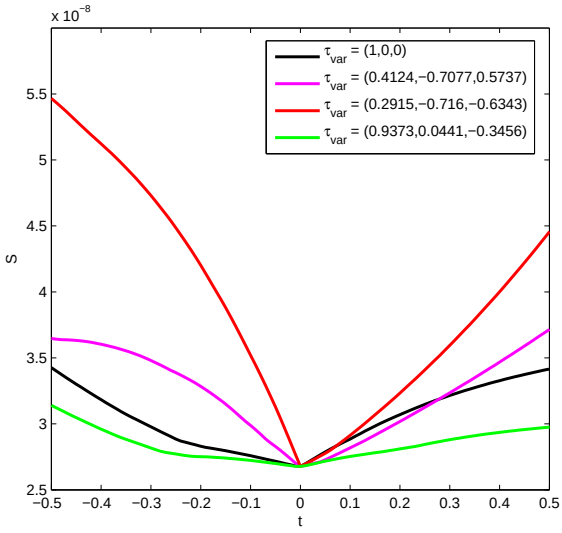
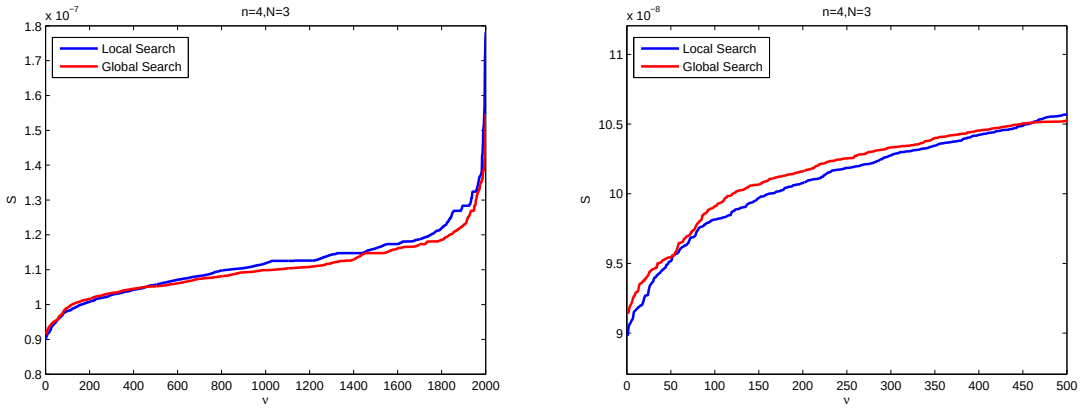


Figure 8.4: Sections of the action around the numerical minimiser plotted in figure 8.3

8.2.2 $n = 4$

Since the complexity of the systems increases with raising n the number of optimisation runs is increased in this whole section via the formula $\nu_{\max} = 1000\,(n-2)$. Thus starting from the solution of subsection 6.1 there were first created a starting configuration via the mechanism described in 8.1.1 and illustrated in figure 8.2.2. The starting points of the optimisation runs were scattered around this configuration as described in subsection 9.2 with the scattering factors $\alpha_\omega = 0.7$ and $\alpha_\tau = 0.3$ as reasoned in Section 8.2.2

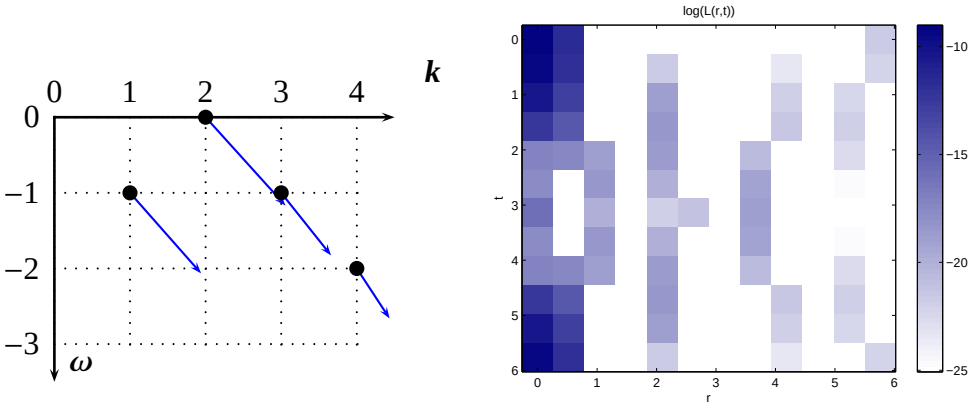
The runs scattered around the starting configuration are referred as the *Local Search* whereas a set of runs with also $\nu_{\max} = 2000$ was done as in subsection 8.2.1 with global random reach referred as *Global Search*.



(a) $1 \leq \nu \leq 2000$ (b) $1 \leq \nu \leq 500$

Figure 8.5: Runoptimality graphs ($S(\nu)$) for the local and the global search

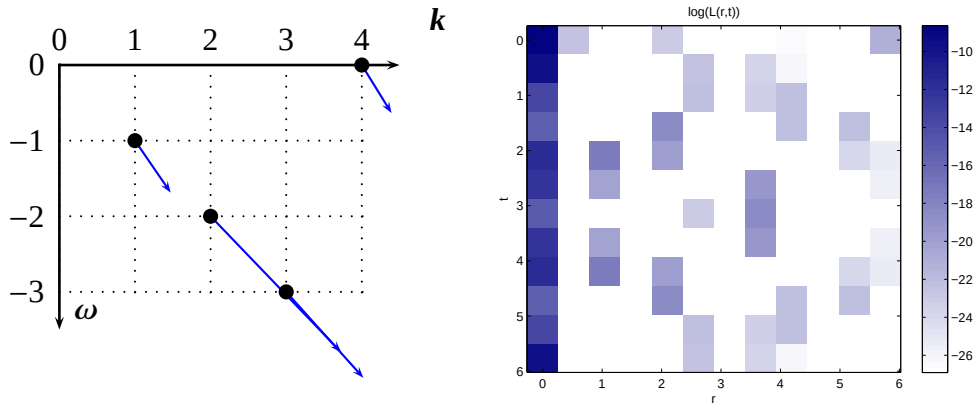
To discuss the optimisation runs from the numerical point of view we have a look at figure 8.5. The plots show clearly that the Local Search is better for small ν which justifies the heuristics of local scattering.



(a) Configuration (b) Causal Structure

Figure 8.6: starting configuration for Local Search, $n = 4$

The numerical minimiser is quite different compared to the starting configuration 8.2.2. The result for the causal structure is inconclusive compared to the results in chapter 6. The assertion that the Lagrange density of points, which are spacelike in the continuous sense ($r > t$), tend to vanish compared to timelike points cannot be confirmed from the numerical result. Nevertheless the occupation itself has indeed a Dirac Sea like shape up to $k = 3$. After all the best numerical minimiser for $n = 4$ (see Fig. 8.9) was determined by a local search starting from the extrapolated start value.



(a) Configuration (b) Causal Structure

Figure 8.7: Best numerical minimiser for $n = 4$

8.2.3 $n = 5$

In this section three major runs with $\nu_{\max} = 3000$ where done. The first one (Local Search I) was based on the best numerical minimiser according to subsection 8.2.2 and transformed to the scattering centre calculated via the formulas in subsection 8.1.1. The second run was a global Search with same $\nu_{\max}$.

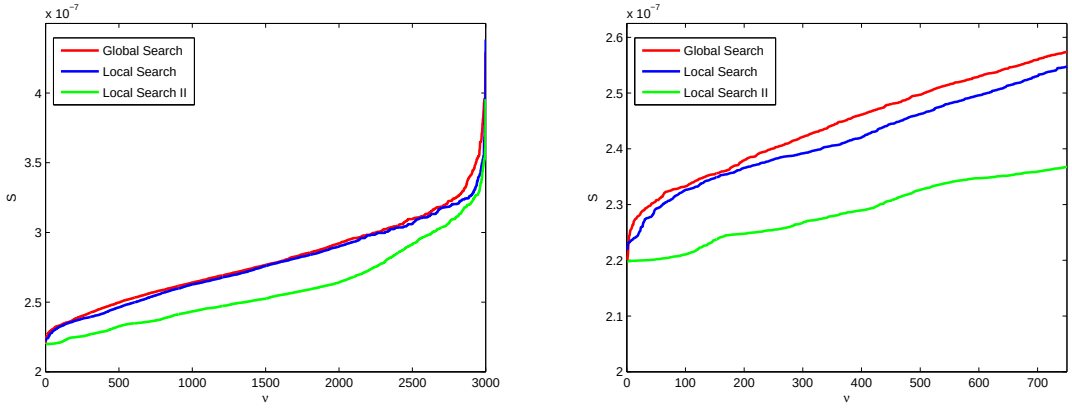
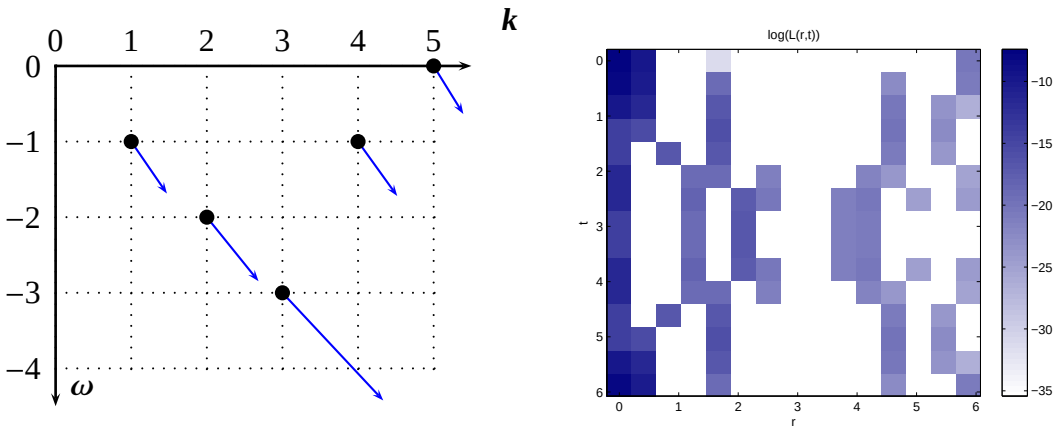


Figure 8.8: $S(\nu)$

Allthough the graphs of $S(\nu)$ as figured in 8.8 is for Local Search I almost everywhere better than that one for the global search, just for $\nu = 1$ the Global Search is better than the local. For this reason there was performed a second local search (Local Search II) around the numerical minimiser of the Global Search.

Local Search I

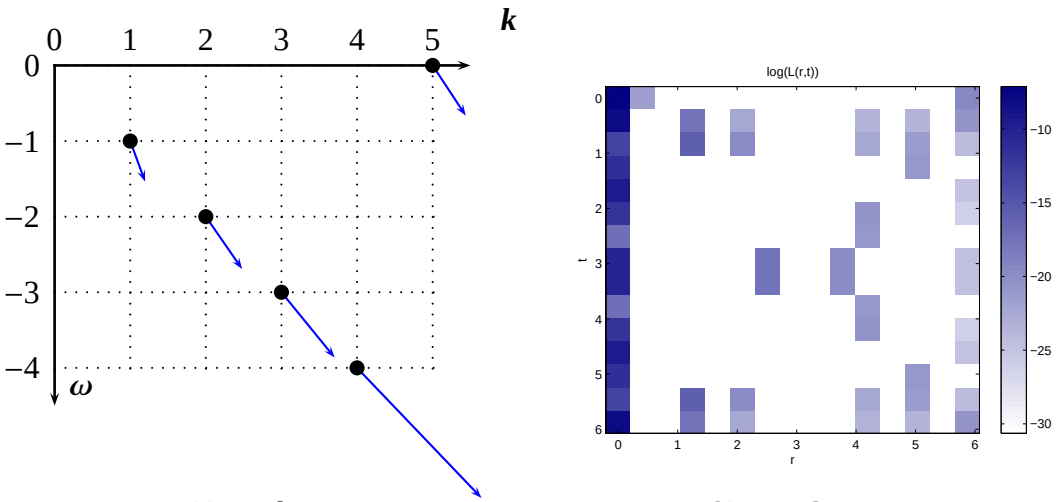
Local Search I did not alter the starting configuration very much as one can see by comparison of figures 8.9 and 8.10 one sees that only the state $k = 4$ alters at all. Comparing this with the absolute numerical minimiser for $n = 4$ this result is quite canonical and affirms the heuristics of the analysis with increasing n handled in this chapter.



(a) Configuration (b) Causal Structure

Figure 8.9: Start Configuration for $n = 5$, Local Search

Due to our definition the configuration is Dirac Sea like up to $k = 3$. Comparing the causal structure of the starting configuration (fig. 8.9) with the numerical minimiser of Local Search II (fig. 8.10) one sees that qualitatively the contributions in regions $r \neq 0$ are decreased. This holds especially near the origin. For $r > 3$ an approximate linear structure occurs, similar to a light cone.

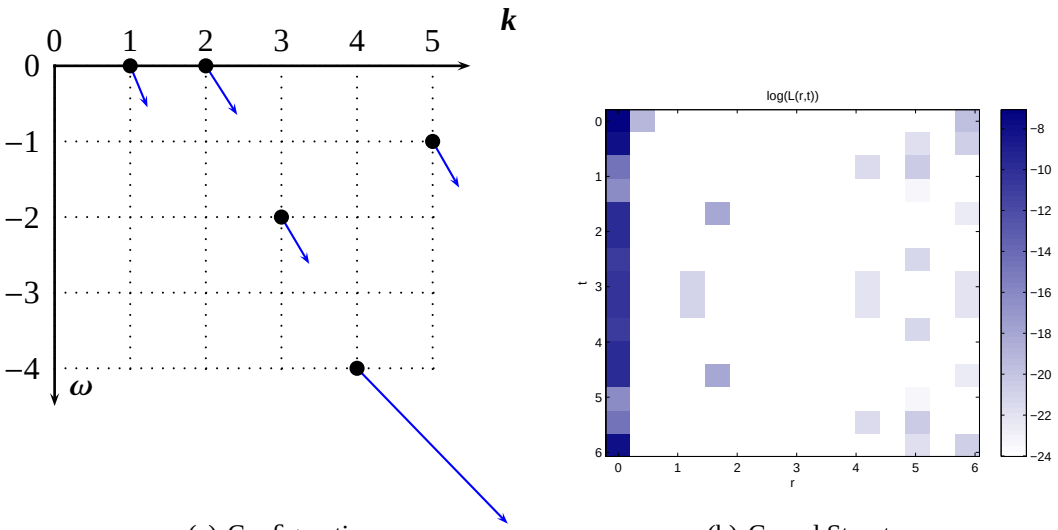


(a) Configuration (b) Causal Structure

Figure 8.10: Numerical Minimiser for $n = 5$, Local Search

Global Search

The Global Search led for $\nu = 1$ to one better solution then Local Search I. Nevertheless the numerical minimiser is in the formal notion Dirac-sea like up to $k = 4$.

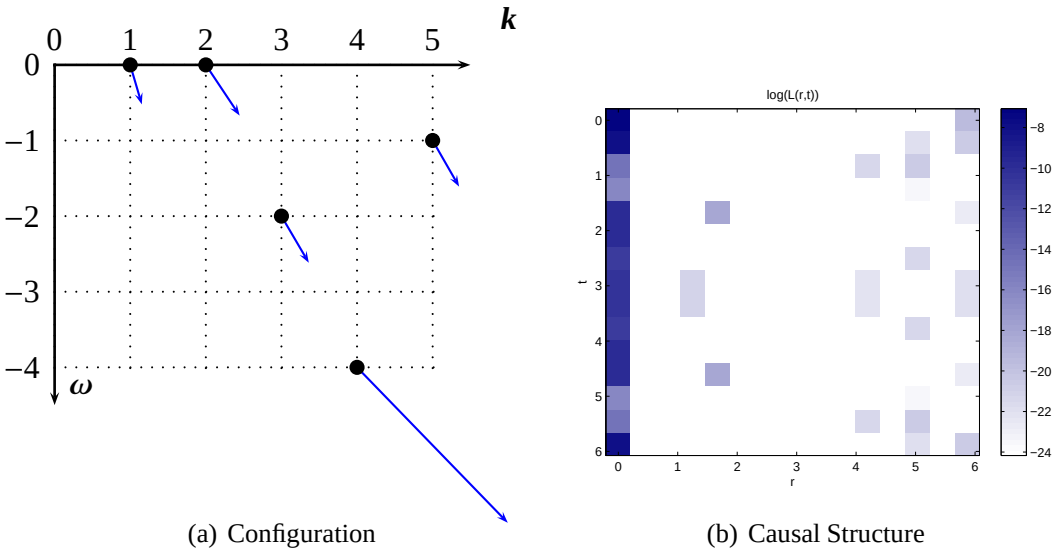


(a) Configuration (b) Causal Structure

Figure 8.11: Numerical Minimiser for $n = 5$, Global Search

Local Search II

We took the numerical minimiser of Global Search discussed in the last paragraph (see fig. 8.11) as the scattering centre of a second local search (Local Search II). The numerical minimiser found in this run did not alter the result of the last paragraph very much. The occupation did not change at all, the values τ_i changed just slightly and the variations in the causal structures are plotted in figure 8.13. Again the most significant variation occurs near the origin, especially for $r = 0$.



(a) Configuration (b) Causal Structure

Figure 8.12: Numerical Minimiser for $n = 5$, Local Search II

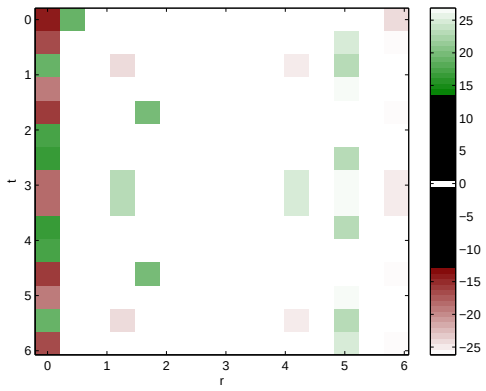


Figure 8.13: Comparison between the causal structures of minimiser 8.11 and 8.12. The plot shows the logarithm of the absolute of the difference between the causal structure of 8.11 and 8.12 ($\text{sign}(L_I - L_{II}) \cdot \log(|L_I - L_{II}|)$).

8.2.4 $n = 6$

We started the calculations for $n = 6$ with a local search around the scattering centre calculated from the numerical minimiser plotted in figure 8.10 in the usual manner.

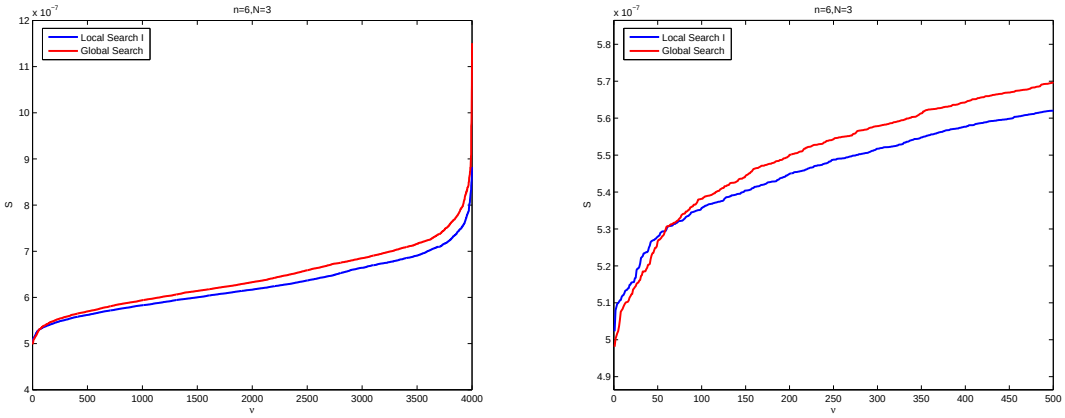
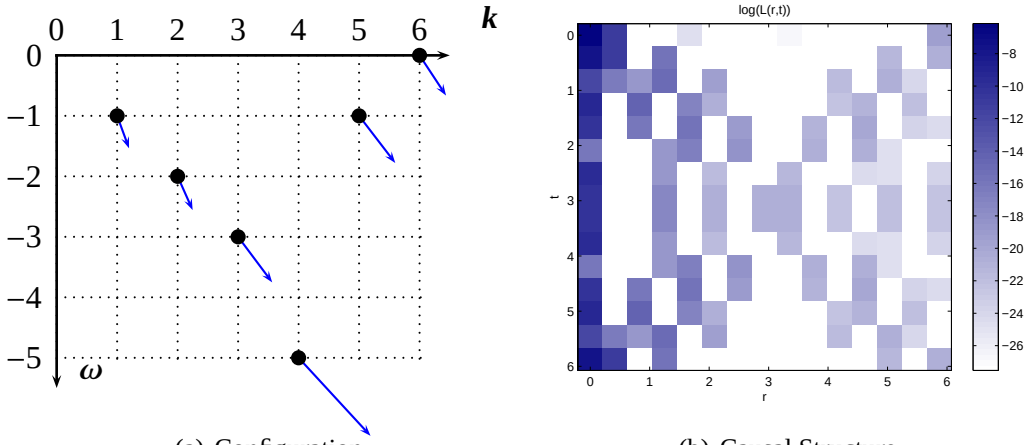


Figure 8.14: $S(\nu)$ for the local search Ia and the global search

A similar finding as for $n = 5$, but even more distinct, was archived. Again the global search delivered for small values of ν better results than the local search, whereas for most values of ν the local search is better.

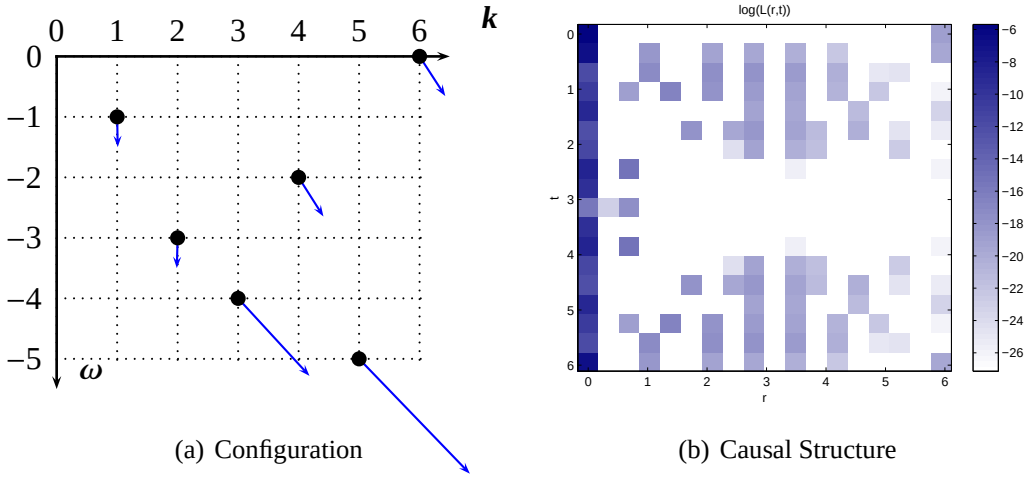
Local Search Ia

The numerical minimiser found in this local search is Dirac Sea like up to $k = 5$. The Lagrangian near the origin was clearly reduced by the optimisation.



(a) Configuration (b) Causal Structure

Figure 8.15: $n = 6$, Start Configuration Local Search I

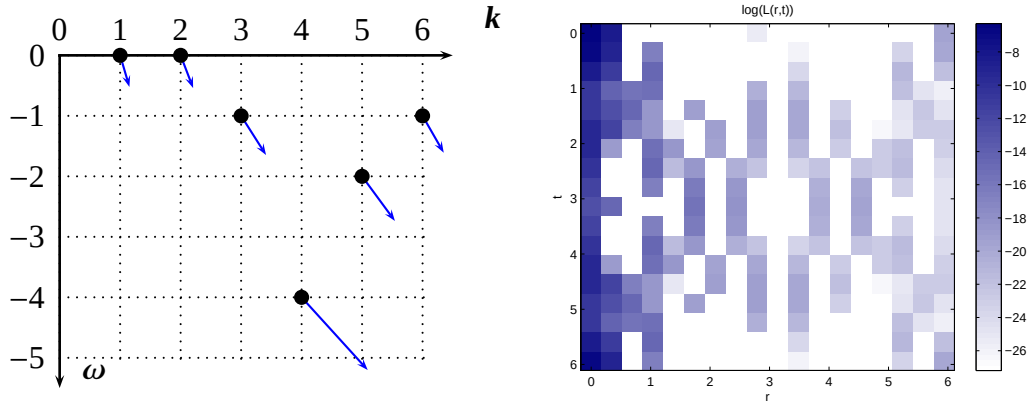


(a) Configuration (b) Causal Structure

Figure 8.16: Numerical Minimiser for $n = 6$, Local Search I

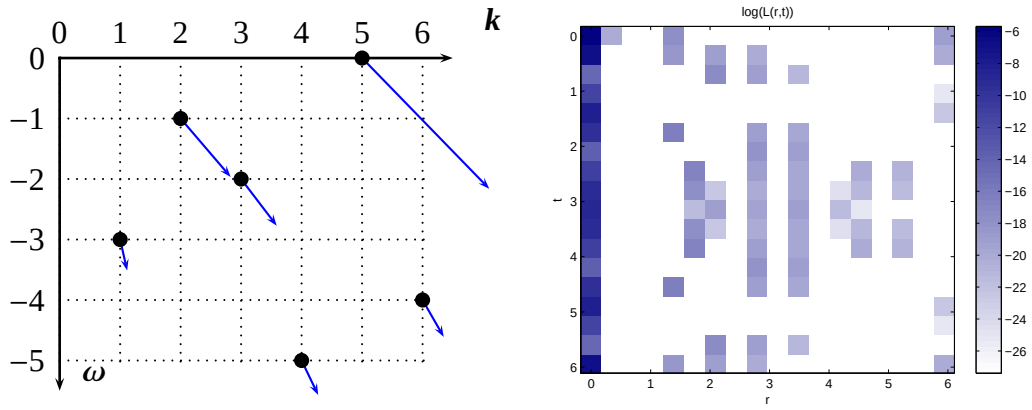
Local Search Ib

The numerical minimiser found in this local search is Dirac Sea like up to $k = 4$. Again the Lagrangian near the origin was clearly reduced, but beyond that the causal structure of the numerical minimiser has no distinguishable causal structure as exected from naive discretisations of Dirac Seas.



(b) Causal Structure

Figure 8.17: Start Configuration for $n = 6$, Local Search

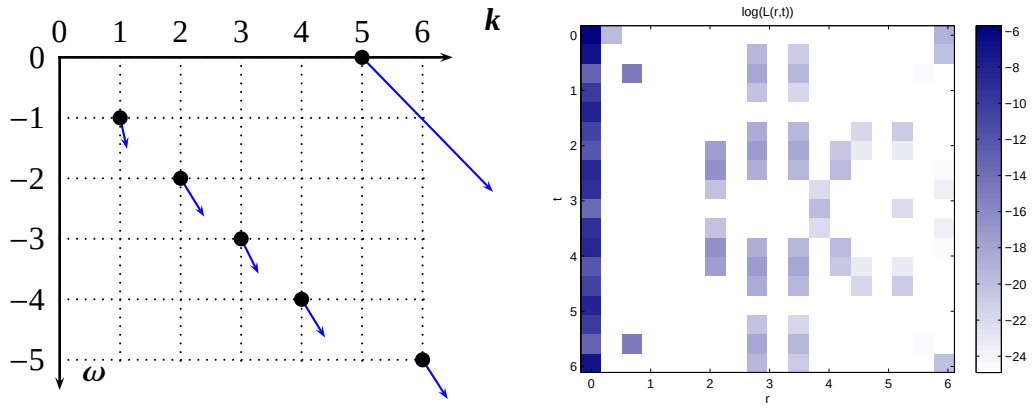


(a) Configuration (b) Causal Structure

Figure 8.18: Numerical Minimiser for $n = 6$, Local Search

Global Search

The numerical minimiser resulting from a global search is Dirac Sea like up to $k \leq 4$ and for $k = 6$. The causal structure contains the hint of a light cone. See figure 8.19.



(a) Configuration (b) Causal Structure

Figure 8.19: Numerical Minimiser for $n = 6$, Global Search

We now want to have a closer look on the numerical minimiser of the global search to get a better understanding of the action. Appendix B.5 lists for $\nu = 1$ and $\nu = 3$ configurations with the same occupation and similar τ-values (naturally respecting the symmetry $-\tau \leftrightarrow \tau$). The question arises if $\nu = 3$ is a minimiser of its own or if it denotes the end point of a run which failed to reach a point more optimal and nearer to the configuration with $\nu = 1$.

To decide this question numerically we considered the function

$$\begin{array}{rcl} [-0.3, 0.3] \times [-0.2, 1.2] & \longrightarrow & \mathbb{R} \\ (t_1, t_2) & \longmapsto & S(\tau^{\nu=1} + t_1 \Delta\tau^{\perp} + t_2 \Delta\tau, \omega) \end{array} \tag{8.10}$$

with $\Delta\tau = \tau^{\nu=3} - \tau^{\nu=1}$ and

$$\Delta\tau^{\perp} = ((\Delta\tau)_2, -(\Delta\tau)_1, (\Delta\tau)_4, -(\Delta\tau)_3, (\Delta\tau)_6, -(\Delta\tau)_5), \tag{8.11}$$

which models a cut of the action along two dimensions in the τ-space. The heuristic idea is that the behaviour in any other direction orthogonal to $\Delta\tau$ around the origin is qualitatively just the same as that in direction of $\Delta\tau^{\perp}$.

Figure 8.20 shows the resulting graph. Taking into account that $\|\tau^{\nu=3} - \tau^{\nu=1}\|_2 \approx 0.4$ one sees that the end configurations for $\nu = 1$ and $\nu = 3$ are in the τ-space contain points which lay on the bottom of a sharp-edged valley with minimal slope. One sees further that $\nu = 1$ is supposable a real numerical minimiser whereas $\nu = 3$ labels an end point which has to be considered as a quite poor approximation of the same minimiser. Presumably the same behaviour can be observed for $\nu = 2$ and $\nu = 5$.

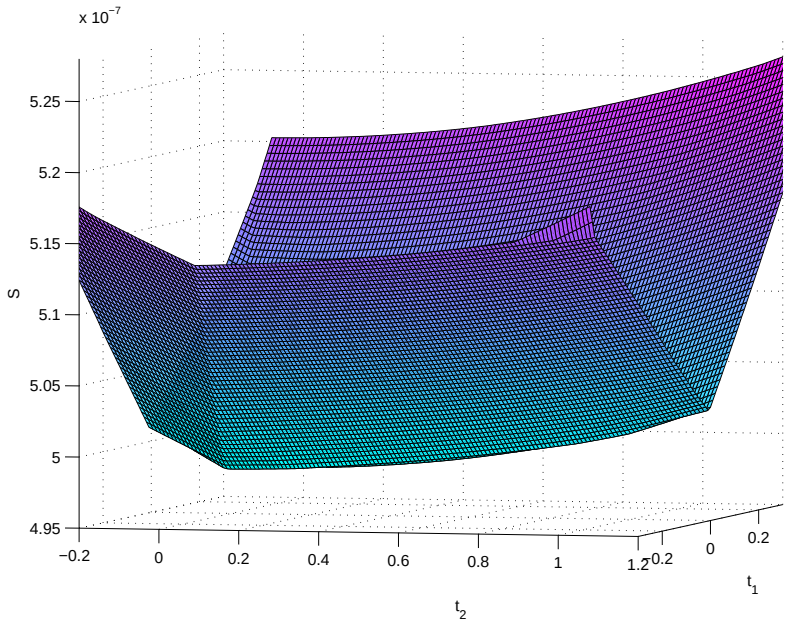


Figure 8.20: Cut of the action along two dimensions in the τ-space.

This result demonstrates how hard the optimisation in the τ-space might be, which has to be considered as a result of the lack of differentiability of the action. This situation gives rise to a new term, which should in some way gives a quantitative understanding of the quality of numerical minimal solutions.

Definition 8.1: *Two Solutions* $(\omega^{(1)}, \tau^{(1)})$ *and* $(\omega^{(2)}, \tau^{(2)})$ *are called to represent a numerical minimiser relatively qualified up to the ratio* μ_r*, if*

$$\omega^{(1)} = \omega^{(2)} \qquad \textit{and} \qquad \mu_r = \frac{||\tau^{(1)} - \tau^{(2)}||_2}{||\tau^{0(1)} - \tau^{0(2)}||_2}. \tag{8.12}$$

They are called to represent a numerical minimiser absolutely qualified up to the ratio μ_a*, if*

$$\omega^{(1)} = \omega^{(2)} \qquad \textit{and} \qquad \mu_a = \frac{||\tau^{(1)} - \tau^{(2)}||_2}{\sqrt{n}}. \tag{8.13}$$

In this notion the solutions $\nu = 1$ and $\nu = 3$ represent numerical minimisers up to $\mu_r = 9.8\%$ and $\mu_a = 0.16$.

As an addition to the last sections we list the values of μ_r and μ_a for $3 \leq n \leq 5$. as well (see fig. 8.21). Remarkably the numerical minimiser for $n = 3$ is much better qualified than that for $n \geq 4$. The results in the next subsection together with fig. 8.21 indicate

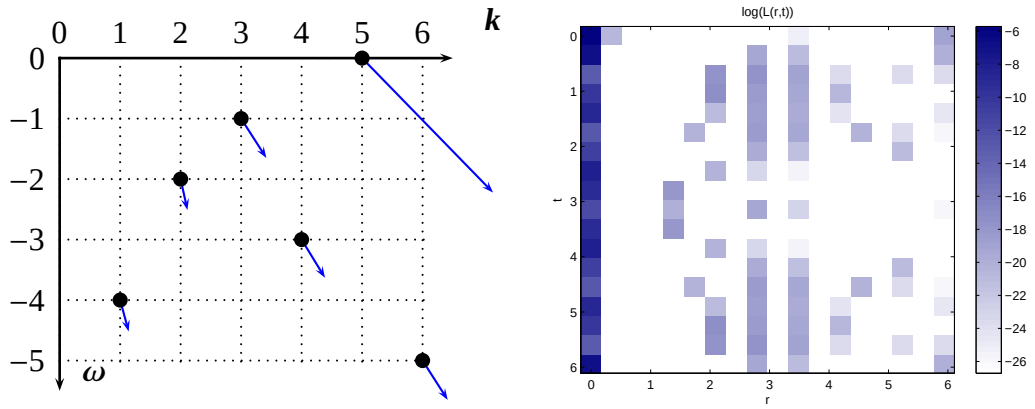
clearly that the quality of the minima attained by our optimisation algorithm gets rapidly worse with increasing n.

n	3	4	5
(ν_1, ν_2)	(1,2)	(1,2)	(1,2)
μ_r	0.61%	5.8%	3.5%
μ_a	0.0074	0.0568	0.014

Figure 8.21: μ_r and μ_a for the best numerical minimisers with $1 \leq n \leq 5$.

Local Search II

Since the numerical minimiser from the global search in the last section was the best minimiser achieved for $n = 6$ another local run was performed starting from this minimiser, which lead to more optimal solutions. But the result just illustrates the mess of our optimisation problem even for system size $n = 6$.



(a) Configuration (b) Causal Structure

Figure 8.22: Numerical Minimiser for $n = 6$, Local Search II

ν	τ	S
1	[-0.2788;-0.2229;-0.7604;-0.7118;-2.1722;-0.757]	4.9410e-07
2	[-0.1222;-0.1259;-1.1628;-0.711;-2.1314;-0.7184]	4.9434e-07
3	[-0.0966;-0.2287;-1.0407;-0.7118;-2.1627;-0.6757]	4.9581e-07
4	[-0.5008;-0.5775;-1.0666;-0.7118;-2.1045;-0.757]	4.9620e-07
5	[-0.517;-0.5731;-1.1609;-0.607;-2.1108;-0.7546]	4.9634e-07

Figure 8.23: First 5 Solutions, which all have the same value of ω.

Figure 8.22 shows the numerical minimiser and figure 8.23 lists additionally the following four solutions with the same occupation as $\nu = 1$. Judging the quality of the solutions we have a look at figure 8.24, which lists all possible combinations of pairs of ν according to 8.23 and the corresponding values of μ_r and μ_a.

ν_1	1	1	1	1	2	2	2	3	3	4
ν_2	2	3	4	5	3	4	5	4	5	5
μ_r	81 %	184 %	175%	27%	34%	90 %	28%	165%	27%	6.7 %
μ_a	0.18	0.14	0.21	0.24	0.069	0.24	0.25	0.22	0.23	0.058

Figure 8.24: μ_a and μ_r for all combinations $1 \leq \nu \leq 5$ according to figure 8.23

First the solutions $\nu = 4$ and $\nu = 5$ represent clearly the best qualified relative numerical minimiser, followed by the solutions with $\nu = 2$ and $\nu = 3$. But although a value of $\mu_a = 0.024$ might seem quite small it is nevertheless large compared to the breaking condition $\Delta\tau_i = 1e-05$ used throughout all calculations so far. That illustrates again how hard the optimisation is especially in the τ-space.

8.3 Considering the Runtime

To back further strategic decisions the final calculation in this chapter discussed here concerns the runtime of a single optimisation run. For this purpose the total runtime of the function `mads` – the core optimisation algorithm of the NOMADm software – was profiled for different system sizes n. As starting values the configuration

$$\tau_i = -0.1(i-1) \quad \text{and} \quad \omega_i = 0 \; \forall i \tag{8.14}$$

was chosen.

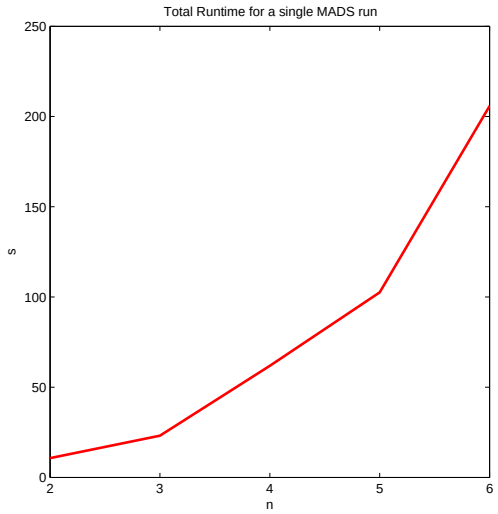


Figure 8.25: Total Runtime for a single MADS run, $2 \leq n \leq 6$.

Figure 8.25 shows the dramatic increase of the numerical cost depending on n. Even increasing the system by only one "particle" ($n = 5 \rightarrow n = 6$) doubles the numerical effort.

8.4 Conclusion

The calculations analysed in this chapter show how numerical minimisers can be calculated and with which restrictions this is attached. First of all the lack of global differentiability only allow to apply the GPS algorithm in a way which is in some kind on one level with heuristic search algorithms like Genetic Programming or Simulated Annealing. The wide range of problems capable for such algorithms is paid by the cost that these algorithms have in general a bad performance compared to methods, that make successful use of derivatives. Hence it is plausible that relaxing the problem in the ω-space does not give any advantage, as was confirmed numerically.

Second it turned out that for system sizes analysed in this chapter the optimisation in the τ-space is much harder than in the ω-space. This has to be seen as a result of the sharp-edged canyons of the action as illustrated in figure 8.20. Nevertheless all numerical minimisers found by the calculations discussed in this chapter are Dirac Sea like up to a certain value of k in the sense of our quite weak operationalisation of this term in definition 3.1. Beside this operationalisation we want also add some remarks according to our qualitative term of "Dirac Sea like". In general the system with $n = 3$ delivered a causal structure which

come closest to the expectations from the continuous theory. It is likely that the reason for this is the outstanding quality of the best numerical minimiser of this size compared to the systems with $n > 3$. Further every optimisation run reduced the Lagrangian near the origin. These results confirm the general hypothesis of this thesis.

On the other hand there was no systematic way found to archive highly qualified minima with the general system size n based on numerical minimisers for systems of size $n-1$. If global searches deliver better results or local searches based on heuristic scattering centres is not clearly determined. There might be other methods doing the "induction step" $n \to n+1$ of producing scattering centres, which deliver better results than archived so far. This might be a task for further research.

The consideration of the runtime in Section 8.25 showed that the runtime of our implementation of the GPS algorithm increases dramatically with the system size. This clearly limited the numerical analysis of systems in our model to quite small system sizes. It should also mentioned that for all systems analysed in this chapter the equivalence classes mentioned in chapter 4 were determined numerically. Thus each Configuration plotted in this section represents actually a class of Configurations due to the symmetry according to the mentioned equivalence classes as well as to the other symmetries of our model. It has to be made clear that the knowledge of the symmetry classes is essential for the analysis of the optimisation data, since it is necessary for the determination of the quality of minimisation solutions, at least in the case of global search. The numerical cost of the calculation of the equivalence classes increases even more rapidly as the optimisation itself, which also constitutes a hard barrier of numerical cost explosion. Since it is due that the transformation $n \to n+1$ does not preserve the discrete ω-symmetry there was selected manually one element of the equivalence class to perform this transformation. Naturally the selected occupation should match our qualitative term of "Dirac Sea like" as much as possible.

Chapter 9

Local Search with fast increasing n

In this chapter we modify the program of the last chapter in a way to treat bigger systems with $n \gtrsim 10$. As figured out in chapter 5 causal structures of naively discretised systems are hard to judge for small systems. Thus it is of interest to find a way to treat larger systems. The general idea is not to increase n by the transition $n \to n+1$ but by the transition $n \to 2n$. The heuristic concept is that systems of size n can be regarded as canonical simplifications or kind of approximations of systems of size $2n$.

9.1 Preparation

Explicitly we will do the transformation by setting

$$\hat{\tau}_k = \tau_n, \quad \hat{\omega}_k = 2\omega_n, \quad \text{for} \quad 2n-2 < k \leq 2n \tag{9.1}$$

with (ω, τ) the absolute numerical minimiser of a system with size n and $(\hat{\omega}, \hat{\tau})$ the scattering centre for a local search at system size $2n$. As one clearly sees from the performance check done in chapter 7 the number of optimisation runs $\nu_{\max}$ has to be decreased with raising n. This weakens the evidence or every archived result. Obviously in this programme the demand of finding absolute minima has to be given up. It has to be weakened for larger systems by the demand of finding local minima with Dirac Sea like configurations and quite possibly causal structures according to that.

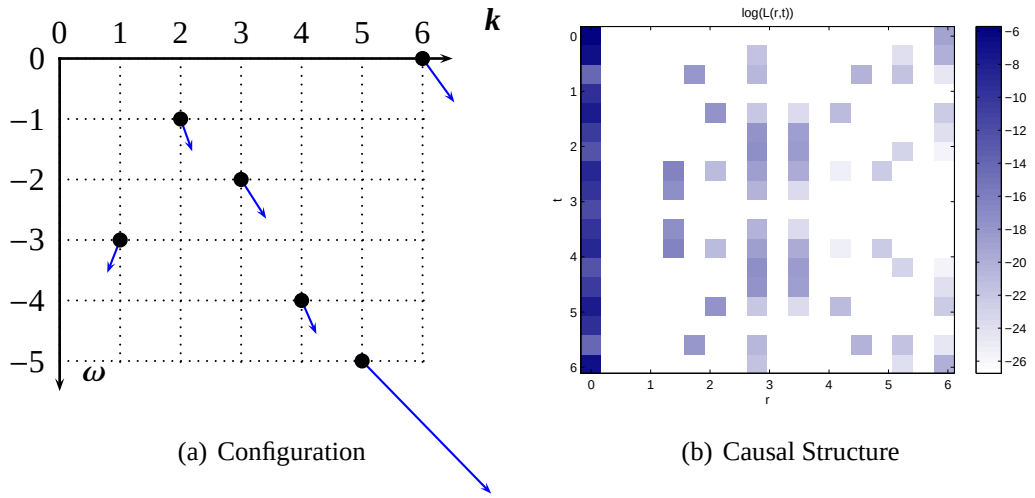
9.2 The calculations

The calculations done in this section aim on constructing start values and doing minimisation of systems with size $2n$ from numerical minimisers from systems with size n. First this is done by starting with the best numerical minimiser for $n = 3$ to deal with the case $n = 6$. Secondly we will take the result as well as all numerical minimisers with $n = 6$ from the last chapter to create start values for $n = 12$. Beside this we will also construct some heuristic start values based on the considerations made in chapter 5. All these calculations are done with $\nu_{\max} = 10$ and will lead to a number of numerical minimisers for $n = 12$. By comparing the action we then will take the best configuration as the basis for a more extensive search with $\nu_{\max} = 200$.

9.2.1 Extrapolated Start Values

n=6

First the numerical minimiser taken from subsection 8.2.1 for $n = 3$ was transformed to a start value for $n = 6$. The resulting numerical minimiser is plotted at figure 9.1.

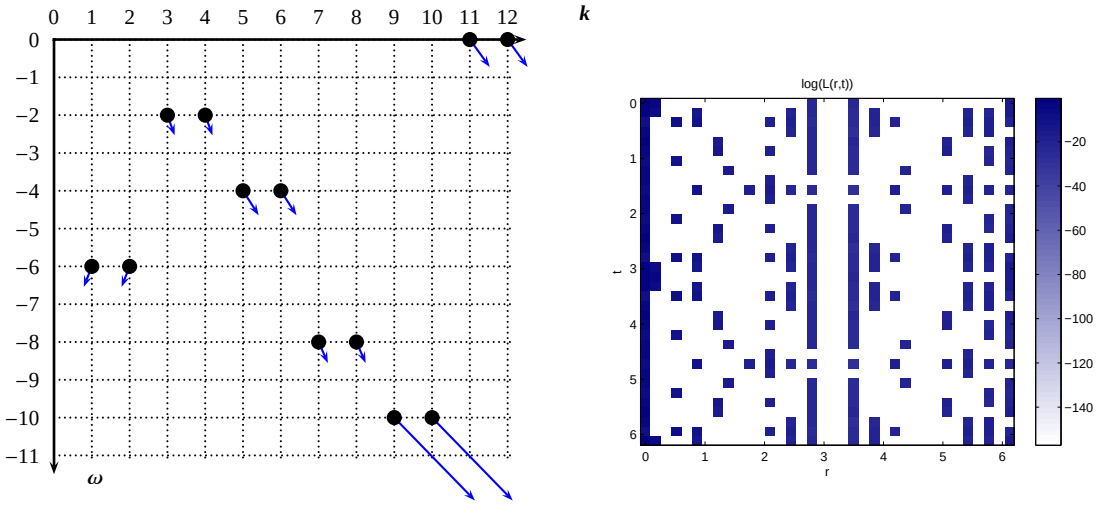


(a) Configuration (b) Causal Structure

Figure 9.1: $n = 6$, End configuration $n = 6$

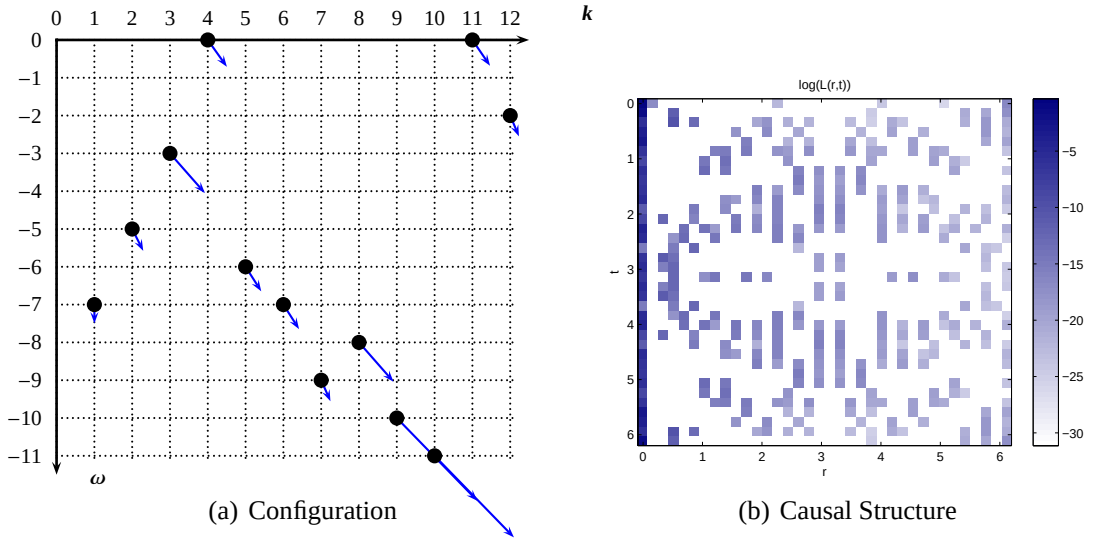
n=12 – 1

The numerical minimiser plotted in figure 9.1 was now extrapolated to a start value for $n = 12$ (see figure 9.2).



(a) Configuration (b) Causal Structure

Figure 9.2: $n = 12$, Starting point with Configuration I



(a) Configuration (b) Causal Structure

Figure 9.3: $n = 12$, Numerical minimiser with start configuration I

To judge these results we do not have to compare only the final value of the action with those of other numerical minimisers, but we have also to consider the variation in the ω-space. This run was done with $\alpha_\omega = 0.8$, so the difference between the start configuration figured in 9.2 and the end configuration 9.3 not only results from the variation in the ω-space done by the GPS algorithm but also from the initial scatterinc process described in Section 9.3.

To get a better term of these different variations in the ω-space, we consider the quantity

$$\widetilde{\Delta\omega}_{a,b} := \frac{\|\omega_a - \omega_b\|_1}{n}, \tag{9.2}$$

with a, b denoting arbitrary indices. If ω_0 denotes the scattering centre, ω_S the start occupation determined by the scattering and ω_e the end occupation of a run, we determined

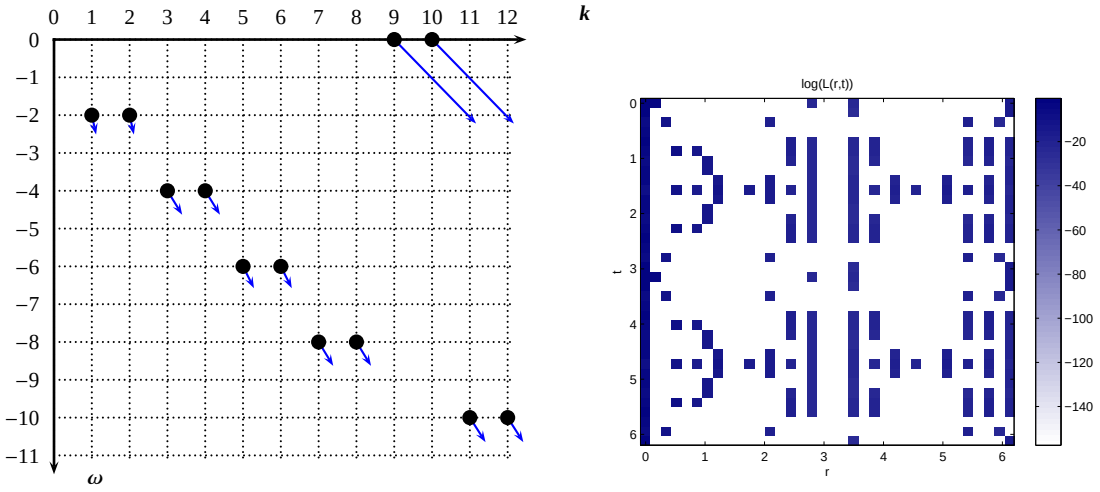
$$\frac{10}{12} \leq \widetilde{\Delta\omega}_{\mathrm{S,e}} \leq \frac{24}{12} \qquad \text{and} \qquad \frac{13}{12} \leq \widetilde{\Delta\omega}_{0,\mathrm{S}} \leq \frac{43}{12}. \tag{9.3}$$

This illustrates that in our specific case the variation in the ω-space caused by the scattering process is up to the factor about 2 greater than the variation caused by the GPS algorithm.

The numerical minimiser itself does not allow to draw far-reaching conclusions. The minimiser is not formally Dirac Sea like for $k \leq 6$ and thus does not confirm the general hypothesis of this thesis. The minimiser might be interpreted in such way that the occupied states tend to form configuration with states on more or less straight but not horizontal lines.

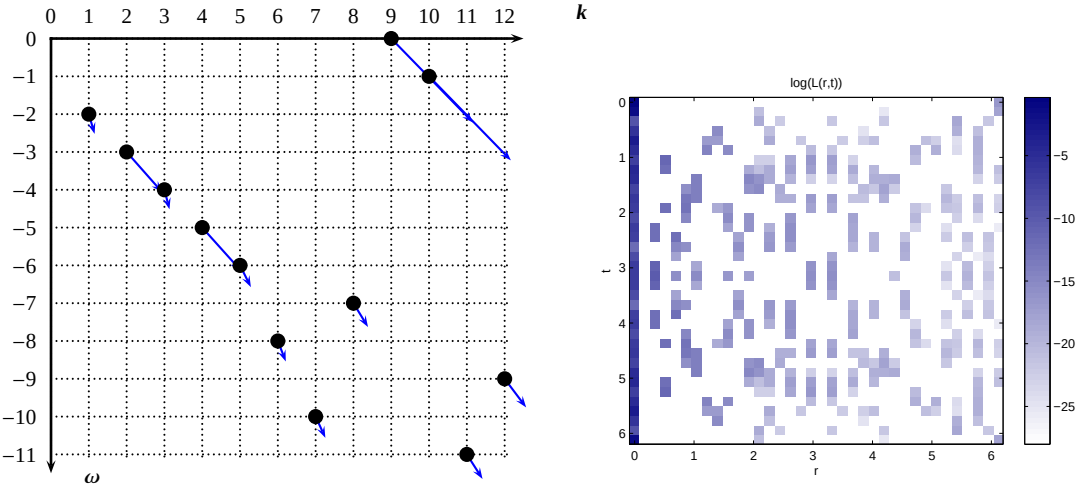
$n = 12 - 3$

The next scattering centre was extrapolated from configuration plotted at figure 8.19. Since the variation of ω in the last section where dominated by the scattering process, we now decrease α_ω and performed runs with $\alpha_\omega = 0.5$ and $\alpha_\omega = 0.2$. (A further run labeled by $n = 12 - 2$, which used a scattering centre extrapolated from the configuration plotted in figure 8.18 resulted in a less optimal numerical minimiser and is listed in Appendix B.6.) The runs with $\alpha_\omega = 0.2$ lead to a more optimal solution (labelled $n = 12 - 3$, see figure 9.4 and 9.5).



(a) Configuration (b) Causal Structure

Figure 9.4: $n = 12$



(a) Configuration (b) Causal Structure

Figure 9.5: $n = 12$, Numerical minimiser

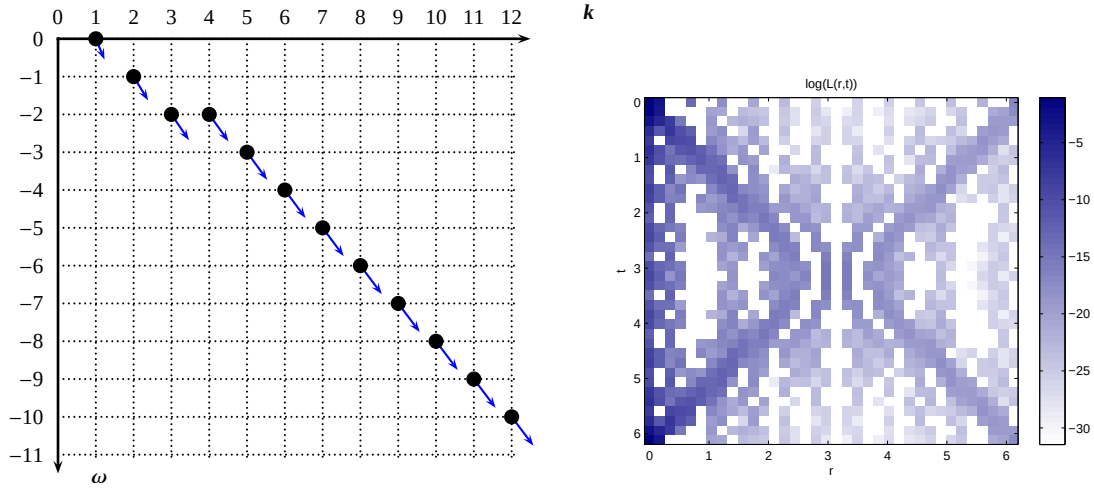
The numerical minimiser confirms convincingly the general hypothesis of this thesis. The configuration of the numerical minimiser is Dirac Sea like in our formal term up to $k = 8$ but moreover fits very clearly the heuristic expectation of Dirac Sea like configurations. Comparing the start configuration with the numerical minimiser one sees that from the raw approximation of a Dirac Sea like configuration resulted from the extrapolation process, the minimising process lead to a quite better approximation of a continuous Dirac Sea in the ω-space. In this note the action principle stabilises Dirac Sea like configurations

while increasing the system size. Further more the causal structure evidences some kind of causality expected from the naively discretised Dirac Seas.

9.2.2 Dirac Sea like Start Values

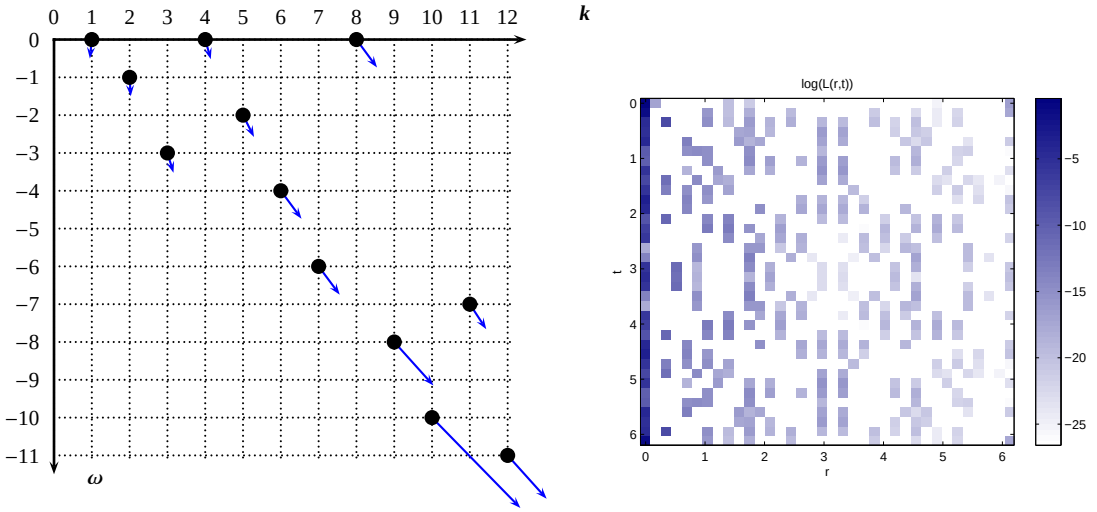
The calculations done in this subsection used a different heuristic in choosing the scattering centres of the minimisation runs. We now set up the naive Dirac Sea like configurations as analysed in chapter 5 as scattering centres for the optimisation runs. Therefore the mass parameters $m \in \{2, 3, 5\}$ where used. (Only the results for $m = 2$ are displayed here, while the others are listed in Appendix B.6.1.)

$m = 2$



(a) Configuration (b) Configuration

Figure 9.6: $n = 12$, Start configuration $m = 2$



(a) Configuration (b) Causal Structure

Figure 9.7: $n = 12$, End configuration $m = 2$

The result of the minimising series for $m = 2$ just show that the starting configuration is changed in an unsystematic way. There seems to be a vague tendency to increase ω_i and τ_i for small values of k and to decrease ω_i and τ_i for large values of k. The same qualitative result was archived for $m \in \{3, 5\}$.

9.3 Interim Result

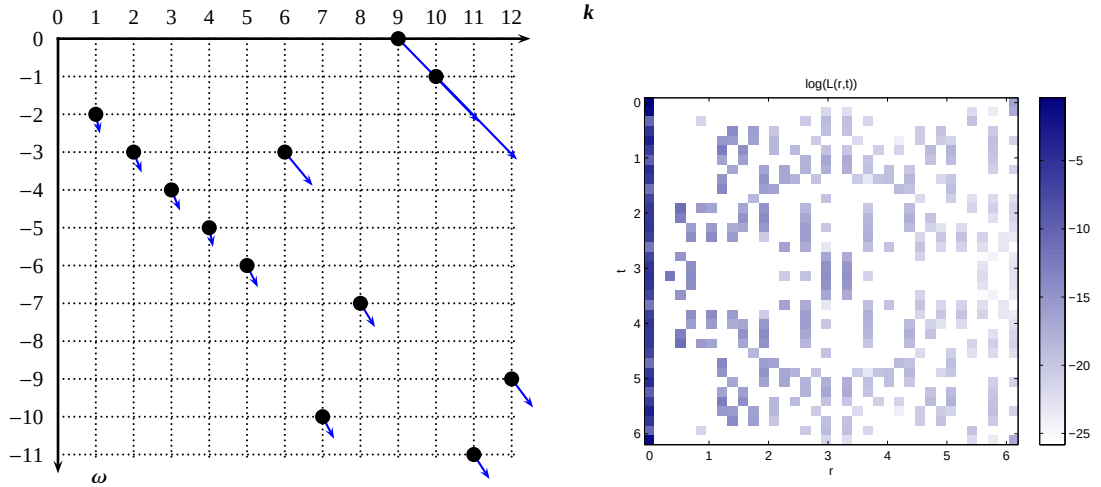
In this section we want to discuss the best calculations so far. Figure 9.8 gives an overview listing the minimisation series and the final value of the action.

Start Value	S best Sol.
$n = 12, 1$	$1.2336e-05$
$n = 12, 2$	$1.2149e-05$
$n = 12, 3$	$1.1231e-05$
$m = 2$	$1.2742e-05$
$m = 3$	$1.3236e-05$
$m = 6$	$1.3068e-05$

Figure 9.8: Comparison of minimisation series of this chapter

Clearly the results got from the “extrapolation heuristics” are better than that from the “naive Dirac heuristics”. The latter one might be advanced by starting from the minimal

configuration resulting from varying m over the resulting naively discretised Dirac Sea configurations analysed in 5.2. But within this work we did not follow this path, since the "extrapolation heuristics" delivered quite satisfying results. Instead we did one more minimising series increasing $\nu_{\max}$ from 10 to 200. starting from the start value "$n = 12, 3$".



(a) Configuration (b) Causal Structure

Figure 9.9: Absolute numerical minimiser for $N = 3$, $n = 12$, $\nu_{\max} = 200$

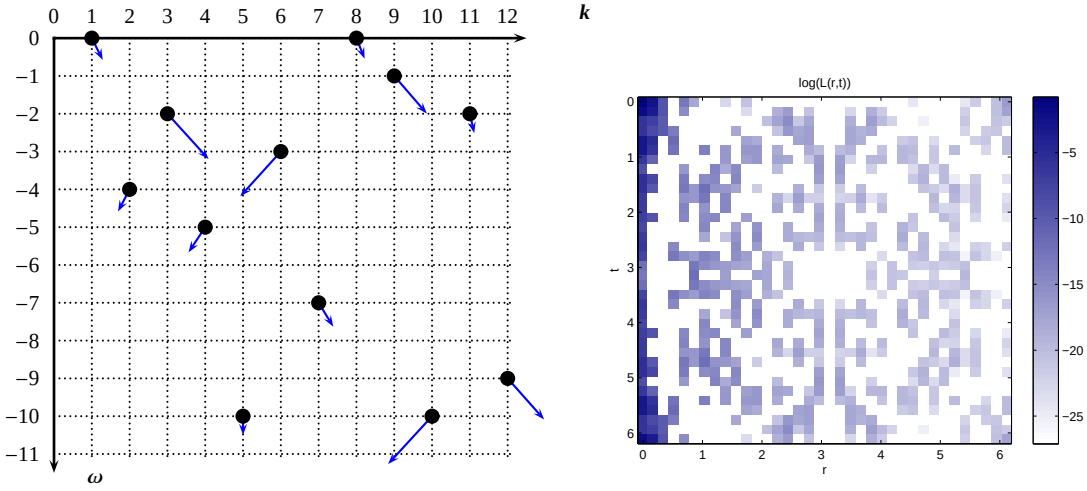
9.4 Using Advanced Search Steps

The calculations so far were done within the framework of the GPS alogrithm without making use of any SEARCH strategy. This is consistent with the GPS algorithm, because at this the SEARCH step can always be empty. In this case the GPS algorithm performs basically a local search upon the continuous variables. For systems with a systemsize about $n = 5$ this is of no relevance since the variation in the ω during the minimisation is great enough to ensure a satisfying variation.

9.4.1 Global Search for $n = 12$

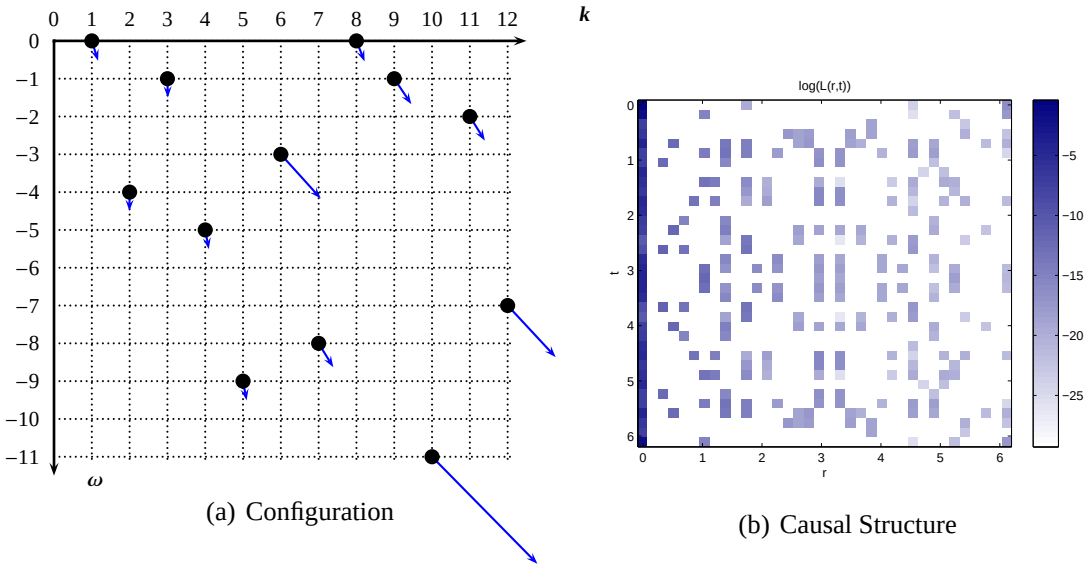
In this subsection we will discuss a global minimising series with $\nu_{\max} = 200$, which has a double purpose. First we want to see if there can be made any advantage concerning the variation in the ω-space. Since we have seen that the GPS algorithm with empty SEARCH step only provides a rather local search in the τ-space, it might be desirable to make methods of more global optimisation available. Originally it was planed to make a comparison between different SEARCH strategies. Unfortunately it turned out that in

the NOMADm software these features are quite buggy implemented. Hence it was only possible to execute a minimising series including a Generic Search strategy.[1]



(a) Configuration (b) Causal Structure

Figure 9.10: Starting point for $\nu = 1$, $n = 12$, $\nu_{\max} = 200$, Global Search



(a) Configuration (b) Causal Structure

Figure 9.11: Numerical Minimiser for $n = 12$, $\nu_{\max} = 200$, Global Search

As in subsection 9.2.1 we have to discuss the variation in the ω space. Since a global

[1] the exact settings where:
`Initial Search: 5-pt. Genetic Algorithm (2)`
`Search: 5-pt. Genetic Algorithm (2)`

search does not incorporate a scattering process around a scattering centre, we have only to discuss the quantity $\widetilde{\Delta\omega}_{0,e}$. We found

$$\frac{1}{12} \leq \widetilde{\Delta\omega}_{0,e} \leq \frac{26}{12} \tag{9.4}$$

and the average for $1 \leq \nu \leq 200$

$$\overline{\widetilde{\Delta\omega}}_{0,e} = \frac{10.67}{12}. \tag{9.5}$$

That means that the variation in the ω-space is still low, by average approximately 1 per dimension. A closer look to the run history shows that the variation in the ω-space takes place mainly at the beginning of the run. Hence there might be some "pinning effect", i.e. during the optimisation process τ reaches for a specific value of ω or even for a quite small class of ω-values a value that forecloses larger variations in the ω-space. In other words the algorithm would have to change ω considerably and simultaneously τ in a way heavily depending on the new ω-value. Since there is no unique term of "minimum" concerning discrete variables, this can also be regarded as a usual problem concerning optimisation, that is the existence of numerous local minima, which catch the run of the algorithm.

It has to be mentioned that the problem of low variation does not occur concerning the variation of τ. The numerical minimiser plotted in figure 9.11 shows very clearly that the qualitative result observed constantly throughout the minimisation series of this thesis still endures. The action principle forces the components of τ to have the same sign and the absolute value of τ_i tends to increase with raising k.

Although the calculations of this subsection seem not very successful in putting evidence to the main hypothesis of this thesis, there is one result, which does: The action of the numerical minimiser discussed here takes the value

$$S_{\nu=1}^{\text{end}} = 1.1681e - 05 \tag{9.6}$$

which is worse than the result above for the run "$n = 12 - 3$". This is quite striking, since the result at "$n = 12 - 3$" was archived with $\nu_{\max} = 10$, 20 times less than $\nu_{\max}$ for the global search considered here.

9.5 Conclusion

The calculations done in this chapter are mainly devoted to systems with $n = 12$. In consideration of the general technical framework of this work it seems not appropriate to step further to systems with $n = 24$. Systems of this size would be capable within a

more effective numerical architecture by making heavily use of parallelisation and highly performant computers.

Further more the best numerical minimisers resulting from the local search matches quite clearly the general hypothesis of this thesis (see fig. 9.5 and 9.9). Qualitatively as well as formally this numerical minimisers can be considered as Dirac Sea like configurations (formally up to $k = 8$ and $k = 5$ respectively with $k_{\max} = 12$). The numerical strategy leading to these configurations does not include any arbitrary restriction to the configuration space expect those figured out in chapter 4 and 6. On this note the numerical minimiser plotted in figure 9.9 can be regarded as the global numerical minimiser over all calculations done so far for $n = 12$. Notably this solution also holds the assumed characteristic that the Dirac Sea like behaviour only appears for small values of k.

We want to close this conclusion with a quite intuitive consideration. Heuristically there are three qualitative properties classifying a configuration to be Dirac Sea like, if they show for small k the following characteristics

(a) ω should take a value that form a kind of mass hyperbola

(b) The components of τ have the same sign and the absolute value increases approximately monotone.

(c) The causal structure is in some way similar to that obtained from naively discretised Dirac Seas, i.e there should be some kind of "light cone". At least near the origin the Lagrangian of space like points of the space time should vanish.

The general experience during all the minimisation series was that it is hard to fullfill all three requirements in the same way. Often a local search leads to a better match of one requirement by loosing some quality in the term of another requirement. For instance comparing the numerical minimiser plotted in figure 9.5 and 9.9 one sees that the minimiser 9.5 is better according to requirement (a) and 9.9 is better according to requirement (b).

Chapter 10

Discussion and Conclusion

10.1 Discussion of the main Assumption of this Thesis

The numerous calculations presented in this work produce formidable evidence, that the principle of the Fermionic Projector describes systems of physical relevance. They also illustrate that there is no straight forward way to find the relevant numerical minimisers. But when they are found, they have likely quite interesting properties:

(a) The widest numerical basis was found for the tendency caused by the action principle first to level the sign of the components τ_i and second to make $|\tau_i|$ to be monotone increasing. This very solid result opposites the result that in the τ-space the optimisation near a promising candidate of a global numerical minimiser is very hard to perform. This has to be seen as a direct result of the lack of differentiability of the considered action.

(b) Concerning the occupation ω various symmetries where found. These symmetries are classes of $(\omega^i)_{i\in\Lambda}$ with $S(\omega^i, \tau) = S(\omega^j, \tau)$ for all $\tau \in \mathbb{R}^n$. Some of these symmetries are well understood theoretically, some not.

(c) There are Dirac Sea like minimisers, which are "stable" under extrapolation processes. That is a Dirac Sea like minimiser can be extrapolated to a system of larger size, taken as the scattering centre of a local search, which results in a numerical minimiser, which is Dirac Sea like as well. Especially the minimising process enhances the Dirac Sea like properties in some way.

(d) Heuristics as mentioned in (c) which aim on step forward from small systems to larger ones are especially successfull in the case of the transformation $n = 6 \rightarrow n =$

12. Although a extensive global search was performed ($\nu_{\max} = 200$) the far less extensive local search led to a more optimal solution.

Beside this results, which confirm our main hypothesis of this thesis, we want to mention the aspects of our analysis, which turned out to be problematic:

(a) Generally it has turned out that it does not lead to any satisfying result, if large systems are analysed without a solid understanding of the corresponding systems of smaler size. The reason why we increase the system size n very carefully throughout this work is, that we made quite a lot of bad experiences in stepping too fast forward to relatively large systems. These calculations are not discussed in this thesis.

(b) The lack of differentiability of the action is not a fact a priori given to our problem. The occurrence of absolute values in (1.26) reasons the possibility that the action is not differentiable, but does not prove it. Hence the action was assumed not to be differentiable due to the failure of gradient oriented optimisation algorithms.

(c) The analysis of small systems is complicated by the fact that for such systems the evaluation of the Lagrangian $\mathcal{L}(t, r)$ is done on a relative coarse lattice. By the reason of the special r-dependency of $\mathcal{L}(t, r)$ and the resulting domination of the action by the contribution $\mathcal{L}(0, 0)$ sufficiently small systems only have trivial minima. This difficulty was overcome by considering larger lattices and configurations which are for higher momentum empty. This correspondends formally with the introduction of the lattice factor N.

(d) Another challenge comes along with the exponential raise of combinatorial complexity concerning ω and the discrete symmetries attached to ω. The very problem, that these symmetries could it make impossible to decide, whether a occupation ω represents a Dirac Sea like occupation, can be handled for small systems (in our work $n \leq 6$) by a numerical determination of the according equivalence classes. To keep the numerical effort in a manageable scale, the restriction $\omega_{\max} = k - 1$ was met.

10.2 Further Research

10.2.1 Ideas for future Research Programs

During the finishing of this thesis thesis several ideas emerged, which lead to research programs beyond the scope of this work. Many of these ideas concern numerical tasks

which should be done *before* putting the problem towards extensive methods of high performance computing.[1] The reason is that huge numerical jobs usually take quite a lot of time, hence in the forefront of this it should be very clear which methods are most promising. In this section we just want to list these ideas without a detailed discussion on how sustainable these ideas are. They are just mentioned to give possibly further researchers some inspiration on following our work.

(a) The programming done for this thesis still offers potential for enhancing the numerics. Beside the technical issues discussed in subsection 10.2.2 it is recommendable to use the symmetry of the causal structure discussed in chapter 7 to speed up the numerics. The profiling done in 5 showed that the function performing the GPS algorithm (`mads`) takes about $\frac{2}{3}$ of the calculation time, whereas the function evaluation only takes about $\frac{1}{3}$. Hence unfortunately the effect of speeding up the function evaluation is weakened by the algorithmic framework of the optimisation routine.

(b) It has turned out in this thesis that it is no trivial task to find non trivial minima for fermionic systems described in the model developed in chapter 1. To archive satisfying results it was essentially first to introduce the *lattice factor* N and the boundary $\omega_{\max}$ and second to make a proper choice of this quantities. Throughout this thesis there was mainly used the settings $N = 3$ and $\omega_{\max} = k_{\max} - 1$. These settings are motivated phenomenological. This has to be considered in further computations. For instance large systems should be also analysed with $N = 2$ or even $N = 1$, since the last setting is equal to the original setting of our model. Further lower values of N should result in a better computational performance.

(c) The Configuration of Dirac Sea like numerical minimisers suggest to analyse settings with $k_{\max} - 1 < \omega_{\max} < 2k_{\max} - 1$. But it has to be respected the fact that increasing $\omega_{\max}$ always comes along with a rapid increasing of numerical complexity in the ω-space. This was the reason why systems with $\omega_{\max} = 2k_{\max} - 1$ were left beside during the numerical analysis of this thesis.

(d) The lack of differentiability turned out to be the worst handicap for the optimisation. Since the action is differentiable in the largest part of the configuration space, it might be useful to modify the action in a way that it becomes differentiable everywhere. In contrast that in the analysis of this thesis relaxation was not adequate it would then make sense to relax the action and make use of minimising methods using derivatives. This should lead to a considerable acceleration of the optimisation process.

(e) Since there exist three generations of elementary particles, one should generally aim on the analysis of systems with three Dirac Seas instead of one as in this work. Clearly this could only be managed reasonable in larger systems, supposably $n \gtrsim$

[1] Parallelisation, profiling and using high performance computer clusters.

20. At the time systems of such size are not under control numerically. Hence handling multiple Dirac Seas in one system has to be considered as a long term aim. Until this is reachable one has to enlarge the size n of systems, which allow a certain way in finding qualified minima.

(f) For larger systems the framework of the GPS algorithm turned out to deliver quite local searches in the ω-space, even in the case where the SEARCH step was non empty but done by a Genetic Algorithm. Before considering larger systems one should keep an eye on further heuristic optimisation algorithms as discussed in chapter 2. For instance one should ask if a global search should be started by Simulated Annealing and finished by GPS. Since Simulated Annealing should escape local minima especially far away from any promising candidate of a global minimum, it might make the minimising series more global. Further since Simulated Annealing usually performes very badly near a promising candidate of a global minimum, GPS could compensate this disadvantage. To overcome the mess of optimisation in the ω-space concerning the "sharp-edged canyons" one should reflect on using some kind of "tabu search". That is that the search direction which lead to a better solution is used in the next step either to "forbid" searches in the opposite direction or to determine a preferred search direction.

(g) The GPS implementation NOMADm offers the opportunity to use surrogate functions to enhance the minimisation performance. Surrogate functions are functions depending on a set $\{(x_i, S(x_i)) \mid 1 \leq i \leq i_{\max}\}$, with (x_i) a series of points in the configuration space and (S_i) a corresponding series of acion values. They have to be considered in some way as "replacements" of the original function, which can be calculated much faster and minimised by standard methods easily. Since the number $i_{\max}$ increases during the optimisation process the algorithm defines a series of surrogates. These series are not intended to converge pointwise against the original action function but to do so for one minimiser, modulo a constant offset. The general process to archive this (in a numerical sense) is to alter the steps of updating the surrogate with points $(x_i, S(x_i))$ calculated from the original action function and minimising the actual surrogate. There is numerical experience that the combination of using surrogates with the GPS algorithm is quite successfull.
Unfortunately the use of surrogates is mainly resticted to pure continuous problems or problems, which have only a very limited dependence on discrete variables (see [SRI]). The reason is that surrogates usually are calculated for each combination of discrete variables separately. Here is the point where in contrast to the numerical analysis of this work the use of relaxation could make a real advantage. Since NOMADm offers several methods of using derivatives a series of minimisation runs should performed testing these methods and comparing the archived optimality as well as the performance.

10.2.2 Technical issues

This section is of no theoretical interest. It summarises some important technical issues, which might be important in the case that this work will be continued by others. Doing numerics generally offers two important tasks. In military terms this topics can be called the topic of tactic and the topic of strategy. The strategical topic is to find *one* way to solve the numerical problem. The tactical topic is to find the *best* way. In this terminology this work deals mainly with strategical problems. The numerical problem handled in this work was not analysed before so we had to solve the general problem to find a viable strategy to find solutions at all, not primarily to find these solutions in the best way.

In this spirit it was consequent to use an interpreted programming language as MATLAB, since the programming and debugging process is much more transparent and faster than using compiled languages as C or FORTRAN. This advantage is paid by the the price of lower performance. But for small systems the runtime of the calculations is not the critical factor. One major disadvantage of MATLAB is the fact that it is a proprietary product. This brings along some annoying handicaps. The most fatal one is the lack of good parallelisation options. Though there exist some parallelisation solution from third-party developers, they all require multiple MATLAB licenses, which either raises the financial costs dramatically or offers only a moderate advantage of calculating capacity. If the tasks of this work will be handled from the tactical point of view, this restriction of MATLAB should be taken into account. Maybe further versions of MATLAB will be more capable for parallelisation tasks, but there are commercial reasons not to implement these features very well.

But the choice of the programming language not only depends on features of the languages but also on the availability of optimisation programming packages. MINLP problems are quite hard to handle numerically so it is not to be recommended to start implementing methods from scratch. MINLP problems are in practice usually treated with advanced programm packages and usually these packages like GAMS are not free. By the time of general design decision the MATLAB tool NOMADm was the only free software package capable of MVP and MINLP problems, so that in practice there was no alternative to MATLAB, since NOMADm is written in this language.

Apart from this it would be worth to be considered to continue studying the problem of this work within another programming framework. For instance using C for the function evaluation and PYTHON for the analysing overhead would combine the advantages of using an interpreted language at higher level and a fast language for the basic numerical work (see [PY3]). There are a lot of free scientific and graphic libraries written in PYTHON (see [PY2]), which should be able to replace the functionality of MATLAB. This strategy would be consistent with our experience that during the further progress of our work the function evaluation was nearly not changed at all, so it can be easily put into

a “black box” of a C function which can be executed by PYTHON scripts. And in the end: PYTHON offers full parallelisation options (see [PY1]).

Appendix A

Calculating the Formal Gradient in the τ-subspace

To apply any derivative method to the minimisation problem one has to calculate the gradient. Allthough the action has to be considered as not differentiable we note here the deduction of a formal gradient, which might be usefull for some topics of further research as mentioned in subsection 10.2.1. We first will glance to the more general case following the deductions given in Chapter 5.2 of [FIN2]. Then we will come to our special problem and deduce the formulas used in our programming.

The two point action in the general case is given by

$$S = \sum_{x,y \in M} \mathcal{L}[P(x,y)P(y,x) \tag{A.1}$$

With $A_{xy} = P(x,y)P(y,x)$ as in (1.5) we define the gradient $\mathcal{M}$ of A as

$$\mathcal{M}[A] = \left(\mathcal{M}[A_{xy}]^{\alpha}_{\beta}\right)^{4}_{\alpha,\beta=1} = \left(\frac{\partial\, \mathcal{L}[A]}{\partial\, A^{\beta}_{\alpha}}\right)^{4}_{\alpha,\beta=1} \tag{A.2}$$

and obtain for the variation of $\mathcal{L}$ (with "Tr" denoting the trace of 4×4 matrices)

$$\delta\mathcal{L}[A] = \sum_{\alpha,\beta=1}^{4} \mathcal{M}[A^{\alpha}_{\beta}]\ \delta\mathcal{A}^{\beta}_{\alpha} = \mathrm{Tr}(\mathcal{M}\ \delta A_{xy}) \tag{A.3}$$

Summing up over x and y leads to the variation of the action

$$\delta S = \sum_{x,y \in M} \delta\mathcal{L}[A_{xy}] = \sum_{x,y \in M} \mathrm{Tr}(\mathcal{M}\ \delta A_{xy}) \tag{A.4}$$

We now substitute δA by

$$\delta A_{xy} = \delta P(x,y)\ P(y,x) + P(x,y)\ \delta P(y,x) \tag{A.5}$$

and use the symmetry of $x \leftrightarrow y$ as well as the fact that the trace is cyclic to obtain

$$\delta S\ = 4 \sum_{x,y\in M} \mathrm{Tr}(Q(x,y)\ \delta P(y,x) \tag{A.6}$$

with

$$Q(x,y) = \frac{1}{4}\left(\mathcal{M}[A_{xy}]P(x,y) + P(x,y)\mathcal{M}[A_{yx}]\right). \tag{A.7}$$

It is proofed in 5.2 of [FIN2] that (A.7) can be simplified to

$$Q(x,y) = \frac{1}{2}\left(\mathcal{M}[A_{xy}]P(x,y)\right). \tag{A.8}$$

From this the components of the gradient can be obtained via

$$\delta P(y,x) = \frac{\partial\, P(y,x)}{\partial\, \tau_i}\ \delta\tau_i \tag{A.9}$$

to be

$$(\nabla\mathcal{L})_i = 4 \sum_{x,y\in M} \mathrm{Tr}\left(Q(x,y)\ \frac{\partial P(x,y)}{\partial\tau_i}\right) \tag{A.10}$$

The partial derivative with respect to τ_i can be calculated similar as the (transformed) scalar components of $P(x,y)$ ϕ, v_k, v_ω. But numerically one gets this calculation much cheaper, since all summands of the numerical integration, which do not depend on τ_i, vanish. For the one occupied state which depend on τ_i there has just to be carried out an replacement $\sinh(\tau_i) \leftrightarrow \cosh(\tau_i)$.

The straight forward way to calculate the matrix $\mathcal{M}[A_{xy}]$ would be to express the action in terms of the matrix elements $(a_{ij}) = A$. For this it is necessary to choose a basis in the Algebra of the Dirac-Matrices. By taking the standard choice, it can be proofed for instance that the Eigenvalues λ_+ and λ_- can be expressed in the way

$$\lambda_\pm = \frac{1}{2}(a_{11}+a_{33}) \pm \sqrt{\frac{1}{4}(a_{11}-a_{33})^2 + a_{32}a_{23} - a_{24}a_{31}} \tag{A.11}$$

Since $\mathcal{L}[A]$ is given by the eigenvalues, it is now possible to determine some of it's partial derivatives. But one problem in this account is that the other partial derivatives do not

necessarily vanish, because the representation (A.11) is not unique. Thus one has to figure out relations between all coefficients (a_{ij}) and this is quite pedestrian.

In general there is a much more elegant account to this problem. Since the eigenvalues $\lambda_\pm$ occur either in complex conjughated or in real but positive pairs, the Lagrange-density $\mathcal{L}(\hat{x}) = (|\lambda_+| - |\lambda_-|)^2$ vanishes in the complex conjugated case. Otherwise it's just the sum of the eigenvalues and thus identically to the trace. So we can express the Lagrange density in the proper region by

$$\mathcal{L}(\hat{x}) = \mathrm{Tr}\left(A^2\right) - \frac{1}{4}\mathrm{Tr}(A)\,\mathbb{1}. \tag{A.12}$$

From this we get

$$\delta\mathcal{L}(\hat{x}) = 2\,\mathrm{Tr}(A\,\delta A) - \frac{1}{2}\,\mathrm{Tr}(A)\,\mathrm{Tr}(\delta A) \tag{A.13}$$

$$= 2\,\mathrm{Tr}\left(\left(A - \frac{1}{4}\mathrm{Tr}(A)\,\mathbb{1}\right)\delta A\right). \tag{A.14}$$

Comparing this with (A.3) we conclude that

$$\mathcal{M}[A] = \left(A - \frac{1}{4}\mathrm{Tr}(A)\right) \tag{A.15}$$

and in our special case of the Lagrangian (1.26) with (1.25)

$$\mathcal{M}[A] = 2\Re\left(\phi\bar{v}_\omega\right)\gamma^0 + 2\Re\left(\phi\bar{V}_k\right)\gamma^k + 2\Im\left(\bar{v}_\omega\bar{V}_k\right)\gamma^0\gamma^k. \tag{A.16}$$

But if we want to use the symmetries to eliminate redundant calculations by integrating just over positive values of t, we cannot use formula (A.8), since it is not clear how the substitution (A.5) interchanges an appropriate constriction of our set M. If we put aside the general considerations and just focus on the formula (1.26) and using the properties of the eigenvalues we conclude that in the proper region (of not vanishing lagrangian) we have

$$\begin{aligned}\mathcal{L} &= (|\lambda_+| - |\lambda_-|)^2 = 16\left[\Re^2\left(\phi\bar{v}_\omega\right) - \Re^2\left(\phi\bar{v}_k\right) - \Im^2\left(v_\omega\bar{v}_\omega\right)\right] \\ &= 16\Big[\left(\Re(\phi)\Re(v_\omega) + \Im(\phi)\Im(v_\omega)\right)^2 - \left(\Re(\phi)\Re(v_k) + \Im(\phi)\Im(v_k)\right)^2 - \\ &\quad \left(\Im(v_\omega)\Re(v_k) - \Re(v_\omega)\Im(v_k)\right)^2\Big].\end{aligned} \tag{A.17}$$

We consider now by using the fact that ϕ=const that

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\tau_i} = \frac{\partial\mathcal{L}}{\partial\Re(v_k)}\frac{\partial\Re(v_k)}{\partial\tau_i} + \frac{\partial\mathcal{L}}{\partial\Im(v_k)}\frac{\partial\Im(v_k)}{\partial\tau_i} + \frac{\partial\mathcal{L}}{\partial\Re(v_\omega)}\frac{\partial\Re(v_\omega)}{\partial\tau_i} + \frac{\partial\mathcal{L}}{\partial\Im(v_\omega)}\frac{\partial\Im(v_\omega)}{\partial\tau_i}, \tag{A.18}$$

and thus after a short calculation

$$\begin{aligned}
\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\tau_i} = 16 \Big[&- 2\,\Re(\phi\overline{v_k})\,\Re(\phi)\,\frac{\partial\Re(v_k)}{\partial\tau_i} - 2\,\Im(v_\omega\overline{v_k})\,\Im(v_\omega)\,\frac{\partial\Re(v_k)}{\partial\tau_i} \\
&- 2\,\Re(\phi\overline{v_k})\,\Im(\phi)\,\frac{\partial\Im(v_k)}{\partial\tau_i} + 2\,\Im(v_\omega\overline{v_k})\,\Re(v_\omega)\,\frac{\partial\Im(v_k)}{\partial\tau_i} \\
&+ 2\,\Re(\phi\overline{v_\omega})\,\Re(\phi)\,\frac{\partial\Re(v_\omega)}{\partial\tau_i} + 2\,\Im(v_\omega\overline{v_k})\,\Im(v_k)\,\frac{\partial\Re(v_\omega)}{\partial\tau_i} \\
&+ 2\,\Re(\phi\overline{v_\omega})\,\Im(\phi)\,\frac{\partial\Im(v_\omega)}{\partial\tau_i} - 2\,\Im(v_\omega\overline{v_k})\,\Re(v_k)\,\frac{\partial\Im(v_\omega)}{\partial\tau_i} \Big] = \\
= 32 \Big[&- \Re(\phi\overline{v_k})\,\Re\Big(\phi\,\frac{\partial\overline{v_k}}{\partial\tau_i}\Big) - \Im(v_\omega\overline{v_k})\,\Im\Big(v_\omega\,\frac{\partial\overline{v_k}}{\partial\tau_i}\Big) \\
&+ \Re\Big(\phi\overline{v_\omega}\Big)\Re(\phi\,\frac{\partial\overline{v_\omega}}{\partial\tau_i}) + \Im(v_\omega\overline{v_k})\,\Im\Big(V_k\,\frac{\partial(v_k)}{\partial\tau_i}\Big) \Big]. \qquad \text{(A.19)}
\end{aligned}$$

In the end, we get with the use of (A.19), (1.29), (1.27) and (1.28)

$$(\nabla S)_i = \sum_{t,r} v(r)\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\tau_i} \qquad \text{(A.20)}$$

As mentioned above, the terms $\frac{\partial\overline{v_k}}{\partial\tau_i}$, $\frac{\partial\overline{V_k}}{\partial\tau_i}$, etcetera has to be calculated analogous to ϕ, v_k and v_ω with eliminating the vanishing summands of the numerical integration and taking out the interchange $\sinh(\tau_i) \leftrightarrow \cosh(\tau_i)$.

For making use of relaxation methods it is necessary to calculate not only the gradient in the direction of the τ_is but also in the direction of the lattice occupations. Relaxation in this context means that the lattice occupations modelled as natural numbers are interpreted as continuous variables. Since formula (1.29) is quite well defined, if we take ω not as a function of natural numbers $w(k)$ with $k \in \{1, ..., n_k\}$ but as a function with $k \in \mathbb{R}$. A quite analogous calculation to (A.19) we get

$$\begin{aligned}
\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\omega_i} = 32 \Big[&- \Re(\phi\overline{v_k})\,\Re\Big(\phi\,\frac{\partial\overline{v_k}}{\partial\omega_i}\Big) - \Im(v_\omega\overline{v_k})\,\Im\Big(v_\omega\,\frac{\partial\overline{v_k}}{\partial\omega_i}\Big) \\
&+ \Re\Big(\phi\overline{v_\omega}\Big)\Re(\phi\,\frac{\partial\overline{v_\omega}}{\partial\omega_i}) + \Im(v_\omega\overline{v_k})\,\Im\Big(V_k\,\frac{\partial(v_k)}{\partial\omega_i}\Big) \Big]. \qquad \text{(A.21)}
\end{aligned}$$

the terms Again $\frac{\partial\overline{v_k}}{\partial\omega_i}$, $\frac{\partial\overline{V_k}}{\partial\omega_i}$, etcetera has to be calculated analogous to ϕ, v_k and v_ω with eliminating the vanishing summands of the numerical integration. In the end the calculation is done with a calculation analogous to (A.20).

Appendix B

Analysis Data

B.1 Data belonging to Chapter 4.2.2

B.1.1 $n = 3$

Occupation	τ_1	τ_2	τ_3	Action
1 0 2	-2.05126953e+00	-7.03875679e-01	-1.75000000e+00	2.99858771e-08
1 2 0	-1.86271481e+00	-3.14453125e-01	-1.85372794e+00	3.14710863e-08
1 2 0	-1.81077549e+00	-3.23530835e-01	-1.85937500e+00	3.15334731e-08
1 0 2	-1.74150096e+00	-3.30078125e-01	-1.86718750e+00	3.15910306e-08
0 1 2	-1.57617188e+00	-9.66768322e-01	-1.75846185e+00	3.16436564e-08
2 1 0	-1.47524941e+00	-1.43261719e+00	-1.61422418e+00	3.16822109e-08
2 1 0	-1.43989523e+00	-1.46875000e+00	-1.60250788e+00	3.17697378e-08
0 1 2	-1.15711691e+00	-1.69813205e+00	-1.51308790e+00	3.17800152e-08
0 1 2	-1.52633544e+00	-4.28314244e-01	-1.87500000e+00	3.37319911e-08
0 2 1	-1.78048211e+00	-2.50976562e-01	-1.87500000e+00	3.66858466e-08

Figure B.1: $N = 4$, Global Search

Occupation	τ_1	τ_2	τ_3	**Action**
0 1 2	-2.90945062e-01	-1.54359863e+00	-1.12597656e+00	2.85167939e-08
2 1 0	-2.90039062e-01	-1.54386213e+00	-1.12597656e+00	2.85169484e-08
0 1 2	-2.89479341e-01	-1.54391840e+00	-1.12597656e+00	2.85171221e-08
0 1 2	-2.90527344e-01	-1.54423606e+00	-1.12548828e+00	2.85181493e-08
2 1 0	-2.89537034e-01	-1.54232463e+00	-1.12731419e+00	2.85232749e-08
2 1 0	-2.85553466e-01	-1.54311654e+00	-1.12705459e+00	2.85246823e-08
0 1 2	-2.88574219e-01	-1.54101562e+00	-1.12850400e+00	2.85292135e-08
1 0 2	-4.91426711e-01	-1.19525457e+00	-1.39335164e+00	2.94054545e-08
1 0 2	-1.27043928e-01	-1.89177683e+00	-7.50000000e-01	3.00083360e-08
1 2 0	0.00000000e+00	0.00000000e+00	0.00000000e+00	3.20569054e-08

Figure B.2: $N = 3$, Global Search

Occupation	τ_1	τ_2	τ_3	**Action**
1 2 0	0.00000000e+00	0.00000000e+00	0.00000000e+00	1.78430754e-08
1 2 0	0.00000000e+00	0.00000000e+00	0.00000000e+00	1.78430754e-08
1 2 0	-7.21325776e-05	0.00000000e+00	0.00000000e+00	1.78430754e-08
1 0 2	0.00000000e+00	0.00000000e+00	-1.30109626e-04	1.78430754e-08
1 2 0	-1.89436267e-04	0.00000000e+00	0.00000000e+00	1.78430755e-08
1 0 2	0.00000000e+00	-1.05442839e-04	-1.18802023e-04	1.78430755e-08
1 0 2	-2.37836817e-04	0.00000000e+00	0.00000000e+00	1.78430755e-08
1 2 0	-4.27960202e-04	0.00000000e+00	0.00000000e+00	1.78430759e-08
1 2 0	-4.64212668e-04	0.00000000e+00	0.00000000e+00	1.78430760e-08
1 2 0	-2.97099176e-04	-2.99528954e-04	-3.13296914e-04	1.78430766e-08

Figure B.3: $N = 2$, Global Search

Occupation	τ_1	τ_2	τ_3	**Action**
0 2 1	0.00000000e+00	0.00000000e+00	0.00000000e+00	5.59701446e-08
0 1 2	0.00000000e+00	0.00000000e+00	0.00000000e+00	5.59701446e-08
0 2 1	0.00000000e+00	0.00000000e+00	0.00000000e+00	5.59701446e-08
2 0 1	0.00000000e+00	0.00000000e+00	0.00000000e+00	5.59701446e-08
2 0 1	0.00000000e+00	0.00000000e+00	0.00000000e+00	5.59701446e-08
1 0 2	0.00000000e+00	0.00000000e+00	0.00000000e+00	5.59701446e-08
0 2 1	0.00000000e+00	-1.76662661e-05	0.00000000e+00	5.59701447e-08
0 2 1	-4.78104642e-05	0.00000000e+00	0.00000000e+00	5.59701447e-08
1 2 0	-5.47636941e-05	0.00000000e+00	0.00000000e+00	5.59701447e-08
0 2 1	0.00000000e+00	-4.81435456e-05	0.00000000e+00	5.59701447e-08

Figure B.4: $N = 1$, Global Search

Occupation	τ_1	τ_2	τ_3	**Action**
0 1 2	-7.71666688e-01	-1.93003685e+00	-1.38187341e+00	3.10543976e-08
0 1 2	-7.41803794e-01	-1.95924036e+00	-1.35893867e+00	3.12791755e-08
2 1 0	-8.10867383e-01	-1.99995273e+00	-1.31476749e+00	3.15026397e-08
0 1 2	-1.44005736e+00	-1.77095818e+00	-1.43143598e+00	3.25576657e-08
0 2 1	-4.42345707e-04	-2.44029727e+00	-7.26305614e-01	3.37750036e-08
0 2 1	-2.24854233e-04	-2.43478254e+00	-7.39009831e-01	3.37966480e-08
2 0 1	-9.23045145e-05	-2.46845495e+00	-6.58452228e-01	3.38364880e-08
2 0 1	-1.50666865e-04	-2.47397189e+00	-6.44599185e-01	3.39408207e-08
2 0 1	-1.51242017e-02	-2.48448680e+00	-6.15686632e-01	3.41513176e-08
0 2 1	-2.72071412e-02	-2.50319618e+00	-5.64604426e-01	3.45803679e-08

Figure B.5: $N = 4$, Local Search ($\alpha = 0.2$)

Occupation	τ_1	τ_2	τ_3	**Action**
0 1 2	-2.90795762e-01	-1.54387857e+00	-1.12575106e+00	2.85160008e-08
2 1 0	-2.90578070e-01	-1.54367291e+00	-1.12605645e+00	2.85166088e-08
0 1 2	-2.89932550e-01	-1.54403965e+00	-1.12581128e+00	2.85167504e-08
0 1 2	-2.90614574e-01	-1.54372854e+00	-1.12568177e+00	2.85175446e-08
0 1 2	-2.89434276e-01	-1.54409685e+00	-1.12582293e+00	2.85176214e-08
2 1 0	-2.87891255e-01	-1.54396337e+00	-1.12611430e+00	2.85190388e-08
2 1 0	-2.90623994e-01	-1.54422045e+00	-1.12534850e+00	2.85194822e-08
0 1 2	-2.91626081e-01	-1.54240963e+00	-1.12686142e+00	2.85205545e-08
2 1 0	-2.92072220e-01	-1.54194528e+00	-1.12735283e+00	2.85214270e-08
0 1 2	-1.92642710e-01	-1.54520745e+00	-1.13476528e+00	2.86440660e-08

Figure B.6: $N = 3$, Local Search ($\alpha = 0.2$)

Occupation	τ_1	τ_2	τ_3	**Action**
1 0 2	-2.20219711e+00	-1.07176486e+00	-1.62113637e+00	2.86446349e-08
1 2 0	-2.20915292e+00	-1.09846412e+00	-1.61074673e+00	2.86614969e-08
1 0 2	-2.21239116e+00	-1.12501358e+00	-1.60118515e+00	2.86628024e-08
1 2 0	-2.18664522e+00	-9.08148613e-01	-1.67264576e+00	2.88586878e-08
1 2 0	-2.13547086e+00	-7.94835699e-01	-1.71262548e+00	2.93771091e-08
1 2 0	-2.15785481e+00	-7.48759026e-01	-1.71871901e+00	2.94072387e-08
0 1 2	-1.52398435e+00	-1.08547418e+00	-1.73202167e+00	3.13320601e-08
0 1 2	-1.53381133e+00	-1.05716653e+00	-1.73900517e+00	3.13847427e-08
2 1 0	-1.53549261e+00	-1.03517350e+00	-1.74495436e+00	3.14550904e-08
1 2 0	-2.64579916e+00	-7.49887691e-01	-1.57626106e+00	3.20184184e-08

Figure B.7: $N = 4$, Local Search ($\alpha = 0.2$), $\tau^0 = (-2.05126953e+00, -7.03875679e-01, -1.75000000e+00)$

B.2 $n = 4$

N	Occupation	τ_1	τ_2	τ_3	τ_4	Action
4	(2, 3, 0, 1)	−0.00036796875	−2.55735	−1.129852864583333	−1.4044	$1.05268029e-07$
3	(1, 3, 2, 0)	−0.3299578125	−0.0886	−1.54538020833333	−1.4044	$9.8445286e-08$
2	(2, 1, 3, 0)	−0.00036796875	−0.00022109375	−0.000458333333333	−0.000103125	$5.740103e-08$
1	(0, 1, 2, 3)	−0.00036796875	−0.00022109375	−0.000458333333333	−0.000103125	$1.83075566e-07$

Figure B.8: lala

N	Occupation	Action	Action for $\tau = 0$	% of not trivial Minima
4	()	0	$1.5973325e-07$	100
3	()	0	$1.17364076e-07$	100
2	()	0	$5.7401026e-08$	0
1	()	0	$1.8307555e-07$	0

Figure B.9: lala

Occupation	τ_1	τ_2	τ_3	τ_4	Action
2 1 3 0	0.00000000e+00	-1.03466797e+00	-2.25000000e+00	-1.13916027e+00	9.51633278e-08
1 2 0 3	0.00000000e+00	-1.03152220e+00	-2.25000000e+00	-1.13997903e+00	9.51901615e-08
2 0 1 3	-2.16295419e+00	-1.08356994e+00	-5.55664062e-01	-2.00000000e+00	9.90764446e-08
2 2 3 0	-8.75000000e-01	-1.02429505e+00	-2.35575467e+00	-8.75000000e-01	9.91978049e-08
1 3 2 0	-2.16331807e+00	-1.09375000e+00	-5.48339844e-01	-2.00000000e+00	9.92723547e-08
1 3 2 0	-2.16993171e+00	-1.09375000e+00	-5.45285912e-01	-2.00000000e+00	9.93622272e-08
3 3 1 0	-1.98773970e-01	-1.06817405e+00	-2.34086353e+00	-9.38311694e-01	1.01367632e-07
0 0 1 3	-1.09375000e-01	-8.31143520e-01	-2.35285012e+00	-9.87931287e-01	1.02494879e-07
0 0 1 3	-1.62723277e-01	-1.11621107e+00	-2.47344893e+00	-5.62500000e-01	1.04231257e-07
0 0 1 3	-1.62723277e-01	-1.11621107e+00	-2.47344893e+00	-5.62500000e-01	1.04231257e-07

Figure B.10: $N = 4$, Global Search

Occupation	τ_1	τ_2	τ_3	τ_4	Action
1 2 3 0	-1.85668713e+00	-5.97757089e-01	-1.89623584e+00	-6.85651377e-01	9.22382547e-08
1 0 3 2	-1.35411085e+00	-2.00000000e+00	-1.10108078e+00	-9.76606063e-01	9.49119431e-08
2 1 3 0	-1.87386317e+00	-1.44464877e-01	-2.00000000e+00	-6.10586071e-01	9.90708960e-08
2 1 3 0	-1.94682850e+00	-4.21729577e-01	-2.00000000e+00	-5.00000000e-01	9.97933974e-08
2 1 0 3	-1.59375000e+00	-1.32193188e+00	-2.00000000e+00	-1.56671675e-01	1.00033456e-07
1 3 2 0	0.00000000e+00	-1.93061148e-01	-1.35228251e+00	-1.53441055e+00	1.00148833e-07
2 2 0 3	-1.71572252e-01	-9.93847364e-01	-1.90173194e+00	-7.50000000e-01	1.00406205e-07
2 2 0 3	-1.33191992e+00	-6.52832031e-01	-1.92868941e+00	-7.08307813e-01	1.00722232e-07
1 3 2 0	-3.27150189e-01	-7.07068699e-01	-1.81510647e+00	-9.81943592e-01	1.01958074e-07
1 3 0 2	-1.93793931e-01	-5.72265625e-01	-1.62500000e+00	-1.25000000e+00	1.01978221e-07

Figure B.11: $N = 3$, Global Search

Occupation	τ_1	τ_2	τ_3	τ_4	Action
1 2 0 3	0.00000000e+00	0.00000000e+00	0.00000000e+00	0.00000000e+00	5.74010255e-08
1 2 0 3	0.00000000e+00	0.00000000e+00	0.00000000e+00	0.00000000e+00	5.74010255e-08
1 2 0 3	-1.77889183e-05	0.00000000e+00	0.00000000e+00	0.00000000e+00	5.74010255e-08
1 2 0 3	-3.33272113e-05	0.00000000e+00	0.00000000e+00	0.00000000e+00	5.74010255e-08
1 2 0 3	0.00000000e+00	-1.29568627e-04	-1.68054509e-05	0.00000000e+00	5.74010257e-08
1 2 0 3	-1.34331882e-04	0.00000000e+00	-1.72593709e-04	0.00000000e+00	5.74010260e-08
2 1 3 0	-1.29723739e-04	0.00000000e+00	-2.98892701e-05	-2.44851094e-04	5.74010264e-08
1 2 0 3	-2.39601767e-04	0.00000000e+00	-2.81762663e-04	0.00000000e+00	5.74010269e-08
2 1 3 0	0.00000000e+00	0.00000000e+00	0.00000000e+00	-3.53472006e-04	5.74010270e-08
1 2 0 3	-1.66516532e-04	-4.33395839e-04	0.00000000e+00	0.00000000e+00	5.74010277e-08

Figure B.12: $N = 2$, Global Search

Occupation	τ_1	τ_2	τ_3	τ_4	Action
3 2 1 0	0.00000000e+00	0.00000000e+00	0.00000000e+00	0.00000000e+00	1.83075550e-07
0 1 2 3	0.00000000e+00	0.00000000e+00	0.00000000e+00	0.00000000e+00	1.83075550e-07
0 1 2 3	0.00000000e+00	0.00000000e+00	0.00000000e+00	0.00000000e+00	1.83075550e-07
1 2 3 0	0.00000000e+00	0.00000000e+00	0.00000000e+00	0.00000000e+00	1.83075550e-07
3 2 1 0	0.00000000e+00	0.00000000e+00	0.00000000e+00	0.00000000e+00	1.83075550e-07
2 3 0 1	0.00000000e+00	0.00000000e+00	-1.83337856e-05	0.00000000e+00	1.83075550e-07
0 3 2 1	0.00000000e+00	-6.50399912e-05	0.00000000e+00	0.00000000e+00	1.83075550e-07
2 3 0 1	0.00000000e+00	0.00000000e+00	-4.87218060e-05	0.00000000e+00	1.83075550e-07
1 2 3 0	0.00000000e+00	-7.09015711e-05	0.00000000e+00	0.00000000e+00	1.83075550e-07
1 0 3 2	-3.77774676e-05	0.00000000e+00	0.00000000e+00	-6.60149269e-05	1.83075550e-07

Figure B.13: $N = 1$, Global Search

Occupation	τ_1	τ_2	τ_3	τ_4	Action
1 0 3 2	-1.37871473e+00	-2.24471433e-01	-2.43627320e+00	-8.38502842e-01	1.05826034e-07
1 0 3 2	-1.04705277e+00	-9.20580211e-01	-2.36183148e+00	-8.79692633e-01	1.06306910e-07
0 1 3 2	-1.55505047e-01	-9.08972372e-01	-2.09404654e+00	-1.39091767e+00	1.06886651e-07
2 3 0 1	-1.23355268e+00	-8.31760201e-01	-2.36375585e+00	-8.83848295e-01	1.07100917e-07
1 0 3 2	-1.57943965e+00	-5.55958934e-01	-2.39133357e+00	-8.48594317e-01	1.07748599e-07
3 1 0 3	-1.62610964e-04	-1.73186781e+00	-2.22423523e+00	-8.96530108e-01	1.10588391e-07
1 0 3 2	-1.18933938e+00	-9.17505688e-01	-2.45814624e+00	-6.01765452e-01	1.12988950e-07
2 0 3 1	-4.08099181e-02	-2.29067470e+00	-1.54941131e+00	-1.36418838e+00	1.13246722e-07
1 0 3 2	-6.09246664e-01	-1.52602744e+00	-2.45854200e+00	-3.51208388e-01	1.14916868e-07
0 1 2 3	-2.74699490e-05	-2.17861731e+00	-1.64107559e+00	-1.37160400e+00	1.15068725e-07

Figure B.14: $N = 4$, Local Search ($\alpha = 0.2$)

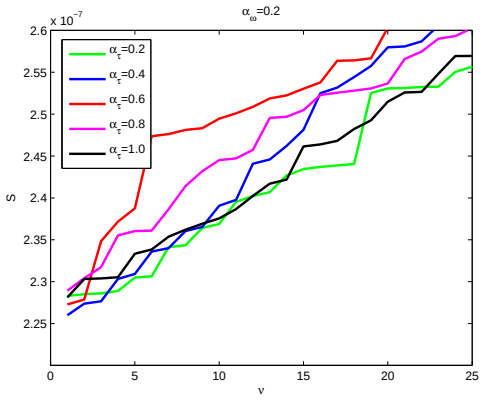
Occupation	τ_1	τ_2	τ_3	τ_4	Action
2 0 1 3	-1.55090109e-01	-3.79911948e-01	-1.54599590e+00	-1.36142686e+00	9.80137634e-08
2 0 1 3	-1.13573276e-01	-3.82732984e-01	-1.60436531e+00	-1.31254164e+00	9.80472749e-08
1 3 2 0	-1.40737250e-01	-3.99371328e-01	-1.57634871e+00	-1.33283160e+00	9.80476899e-08
2 0 1 3	-1.40455692e-01	-3.86564943e-01	-1.50861120e+00	-1.39073638e+00	9.80486543e-08
1 3 2 0	-1.47541247e-01	-4.01489863e-01	-1.53793035e+00	-1.36439370e+00	9.80501049e-08
1 3 2 0	-1.57039241e-01	-4.19029677e-01	-1.57226416e+00	-1.33181565e+00	9.80942631e-08
2 0 1 3	-1.53788312e-01	-4.31031555e-01	-1.53996690e+00	-1.35687770e+00	9.81182195e-08
2 0 1 3	-1.62701721e-01	-4.33958247e-01	-1.54567881e+00	-1.35118590e+00	9.81300759e-08
1 3 2 0	-1.49011430e-01	-5.50735141e-01	-1.65692975e+00	-1.22527242e+00	9.88200564e-08
1 3 2 0	-2.75038351e-01	-6.60449871e-01	-1.64252260e+00	-1.20764968e+00	9.93513446e-08

Figure B.15: $N = 3$, Local Search ($\alpha = 0.2$)

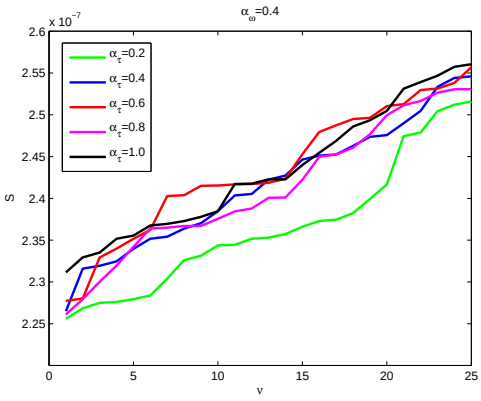
Occupation	τ_1	τ_2	τ_3	τ_4	Action
1 2 3 0	-4.00494970e-01	-1.27863036e+00	-2.16276978e+00	-1.19501199e+00	9.44081354e-08
1 2 3 0	-2.68581840e-01	-1.11510602e+00	-2.16320995e+00	-1.27944048e+00	9.56055314e-08
2 1 0 3	-4.18152754e-01	-1.02797485e+00	-2.17726125e+00	-1.27399687e+00	9.63719610e-08
1 1 3 0	-5.49679227e-01	-1.06375390e+00	-2.45055994e+00	-6.28971280e-01	1.03032255e-07
1 1 3 0	-5.91055163e-01	-1.19472269e+00	-2.44916409e+00	-5.72775425e-01	1.03327895e-07
0 0 1 3	-5.31455901e-02	-3.31743458e-01	-2.21760322e+00	-1.33318374e+00	1.03373985e-07
2 0 3 1	-1.80376435e+00	-1.80573725e+00	-1.45258072e+00	-1.58102127e+00	1.03607342e-07
0 0 3 1	-3.44980984e-01	-1.05918686e+00	-2.46510009e+00	-6.00763012e-01	1.03650169e-07
0 0 1 3	-9.04006255e-02	-3.80735074e-01	-2.16086840e+00	-1.40158787e+00	1.04006996e-07
1 3 2 0	-2.52101927e+00	-1.35143773e+00	-1.23625074e+00	-1.71955320e+00	1.04495979e-07

Figure B.16: $N = 4$, Local Search ($\alpha = 0.2$) $\tau^0 = (-2.05126953e{+}00, -7.03875679e{-}01, -1.75000000e{+}00)$ [interpolated]

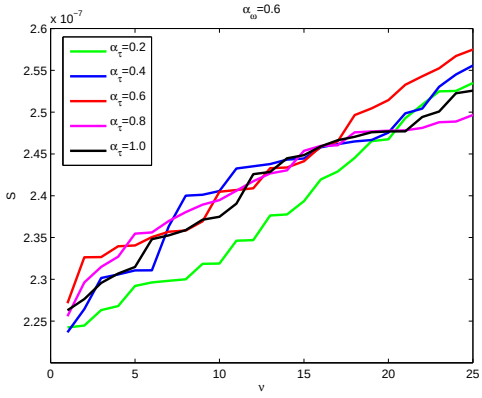
B.3 Data belonging to Subsection 8.1.3



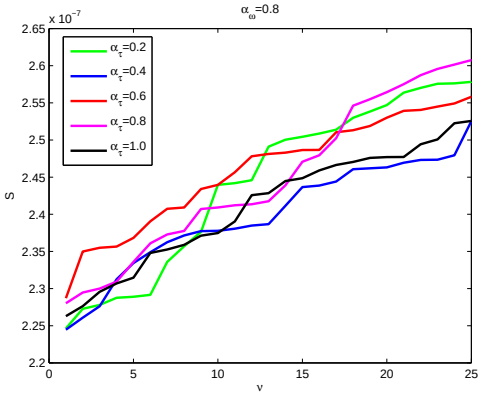
(a) $\alpha_\omega = 0.2$



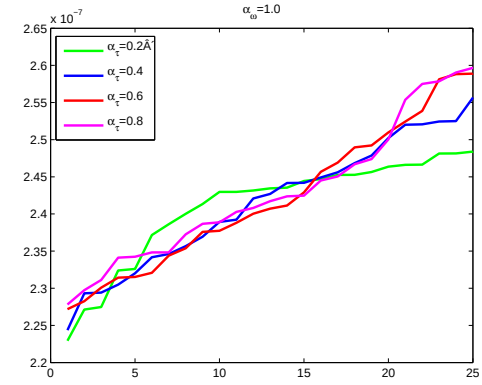
(b) $\alpha_\omega = 0.4$



(c) $\alpha_\omega = 0.6$



(d) $\alpha_\omega = 0.8$



(e) $\alpha_\omega = 1.0$

Figure B.17: $S(\nu)$ graphs for different values of α_ω and α_τ

B.4 Example for Discrete Symmetries

In this section we note all discrete symmetry classes for a system with three occupied states, $n = 3$, $\sharp\omega_i = 6$ and $N = 3$.

$$
\begin{aligned}
[0,0,0] &= \{\ (k,k,k)\ \},\\
[0,0,1] &= \{\ (k,k,l)\ \big|\ |k-l| \neq 0,\ |k-l| \neq 3\ \},\\
[0,0,3] &= \{\ (k,k,l)\ \big|\ |k-l| = 3\ \},\\
[0,1,0] &= \{\ (k,l,k)\ \big|\ |k-l| \neq 0,\ |k-l| \neq 3\ \},\\
[0,1,1] &= \{\ (k,l,l)\ \big|\ |k-l| \neq 0,\ |k-l| \neq 3\ \},
\end{aligned}
$$

$$
\begin{aligned}
[0,1,2] = \{\ & (0,1,2), && (0,2,4), && (0,5,1),\\
& (1,2,3), && (1,3,5), && (1,5,0),\\
& (2,1,0), && (2,3,4), &&\\
& (3,2,1), && (3,4,5), &&\\
& (4,3,2), && (4,0,5), && (4,2,0),\\
& (5,0,4), && (5,3,1), && (5,4,3)\ \},
\end{aligned}
$$

$$
\begin{aligned}
[0,1,3] = \{\ & (0,1,3), && (0,4,3),\\
& (1,2,4), && (1,5,4),\\
& (2,0,5), && (2,3,5),\\
& (3,2,0), && (3,5,0),\\
& (4,0,1), && (4,3,1),\\
& (5,1,2), && (5,4,2),\ \},
\end{aligned}
\qquad
\begin{aligned}
[0,1,4] = \{\ & (0,1,4), && (0,5,2),\\
& (1,2,5), && (1,3,0),\\
& (2,0,3), && (2,4,1),\\
& (3,1,4), && (3,5,2),\\
& (4,2,5), && (4,3,0),\\
& (5,0,3), && (5,4,1),\ \},
\end{aligned}
$$

$$
\begin{aligned}
[0,1,5] = \{\ & (0,1,5), && (0,2,1), && (0,4,2),\\
& (1,0,5), && (1,3,2), && (1,5,3),\\
& (2,0,1), && (2,4,3), &&\\
& (3,1,2), && (3,5,4), &&\\
& (4,0,2), && (4,2,3), && (4,5,0),\\
& (5,1,3), && (5,3,4), && (5,4,0)\ \},
\end{aligned}
$$

$$[0,2,3] = \{ (0,2,3), (0,5,3), (1,0,4), (1,3,4), (2,1,5), (2,4,5), (3,0,2), (3,0,0), (4,2,1), (4,5,1), (5,0,2), (5,3,2), \},$$

$$[0,2,5] = \{ (0,2,5), (0,4,1), (1,0,3), (1,5,2), (2,1,4), (2,3,0), (3,2,5), (3,4,1), (4,0,3), (4,5,2), (5,1,4), (5,3,0), \},$$

$$[0,3,0] = \{ (k,l,k) \mid |k-l| = 3 \},$$

$$[0,3,1] = \{ (0,3,1), (0,3,4), (1,4,2), (1,4,5), (2,5,0), (2,5,3), (3,0,2), (3,2,5), (4,1,0), (4,1,3), (5,2,1), (5,2,4), \},$$

$$[0,3,2] = \{ (0,3,2), (0,3,5), (1,4,0), (1,4,3), (2,5,1), (2,5,4), (3,0,1), (3,0,4), (4,1,2), (4,1,5), (5,2,0), (5,2,3), \},$$

$$[0,3,3] = \{ (k,l,k) \mid |k-l| = 3 \},$$

$$[0,4,5] = \{ (0,4,5), (0,5,4), (1,0,2), (1,2,0), (2,0,4), (2,1,3), (2,3,1), (2,4,0), (3,1,5), (3,2,4), (3,4,2), (3,5,1), (4,3,5), (4,5,3), (5,0,1), (5,1,0), \}$$

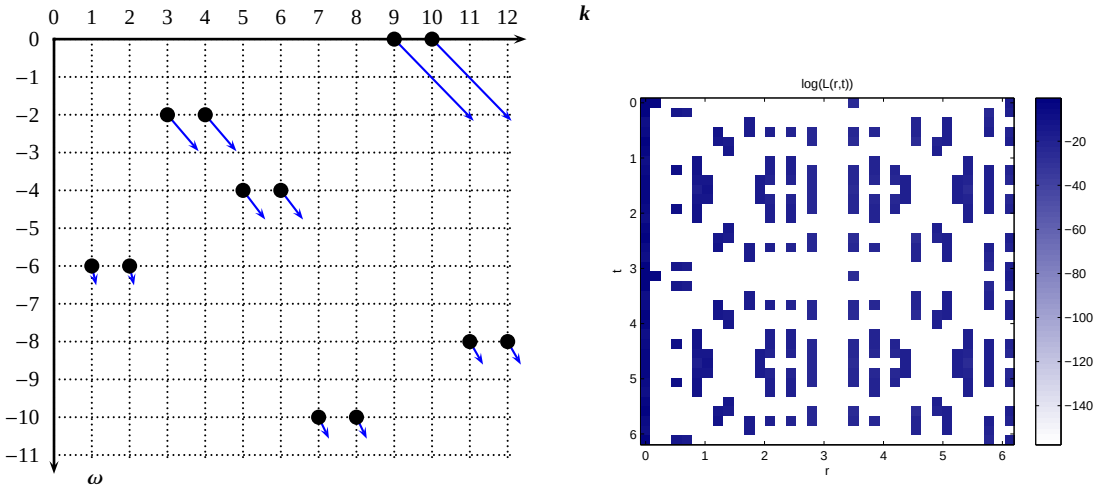
B.5 Increasing n, $n = 6$

B.5.1 Best 20 Solutions, Global Search

ν	$\sharp c$	τ_{end}	S
1	9738	(0.2245, 0.7181, 0.551, 0.7118, 2.1722, 0.757)	4.9812e-07
2	10483	(0.2249, 0.3899, 0.5859, 0.5677, 2.2134, 0.7577)	5.0024e-07
3	9738	(-0.0247, -0.5489, -0.2692, -0.7523, -2.1918, -0.7966)	5.0104e-07
4	13108	(0.0235, -0.5154, -0.9321, -0.7252, -2.0601, -0.9356)	5.0175e-07
5	10483	(0.1005, 0.0964, 1.0508, 0.7002, 2.1881, 0.6211)	5.0231e-07
6	14169	(0.3895, -0.2969, -0.5624, -0.648, -2.1265, -0.9783)	5.0361e-07
7	10483	(-0.1778, -0.3717, -0.7658, -0.7522, -2.2138, -0.5896)	5.0532e-07
8	9738	(0.1483, -0.7398, -0.2613, -0.5105, -2.2387, -0.7706)	5.0757e-07
9	13103	(0.0542, -0.5156, -0.616, -0.8256, -2.1752, -0.701)	5.0798e-07
10	14169	(-0.3382, 0.4395, 0.3556, 0.6675, 2.1801, 0.8764)	5.0832e-07
11	9738	(0.2316, -0.3231, -0.1526, -0.7476, 2.2475, -0.7118)	5.0892e-07
12	9738	(-0.1262, 0.4492, 0.0505, 0.5257, 2.2176, 0.9182)	5.0945e-07
13	13237	(-1.8501, -1.1055, -0.99, -0.8053, -2.0516, -0.6848)	5.0989e-07
14	13207	(-0.0785, -0.5131, -0.7492, -1.9505, -0.7321, -1.4838)	5.1014e-07
15	9838	(-0.082, 0.5556, 0.5285, 0.7974, 2.1561, 0.7939)	5.1019e-07
16	14113	(0.0753, -0.8269, -1.1228, -0.7675, -2.1209, -0.6311)	5.1026e-07
17	10483	(0.2075, 0.0569, 1.3785, 0.9873, 2.0874, 0.54)	5.1067e-07
18	10483	(-0.3842, -0.6044, -1.2623, -0.5574, -2.1811, -0.5252)	5.1140e-07
19	9838	(0.3505, -0.4785, -0.6534, -0.6028, -2.235, -0.6582)	5.1179e-07
20	10483	(-0.267, -0.5486, -0.6904, -0.4363, -2.2604, -0.6208)	5.1232e-07

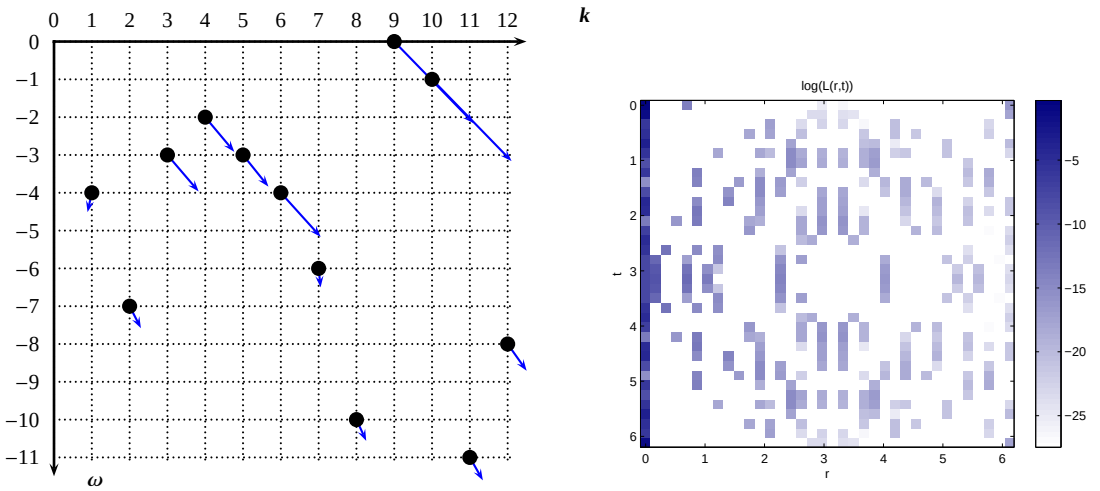
B.6 Data belonging to Chapter 9

n=12 – 2



(a) Configuration (b) Causal Structure

Figure B.18: $n = 12$, Starting point

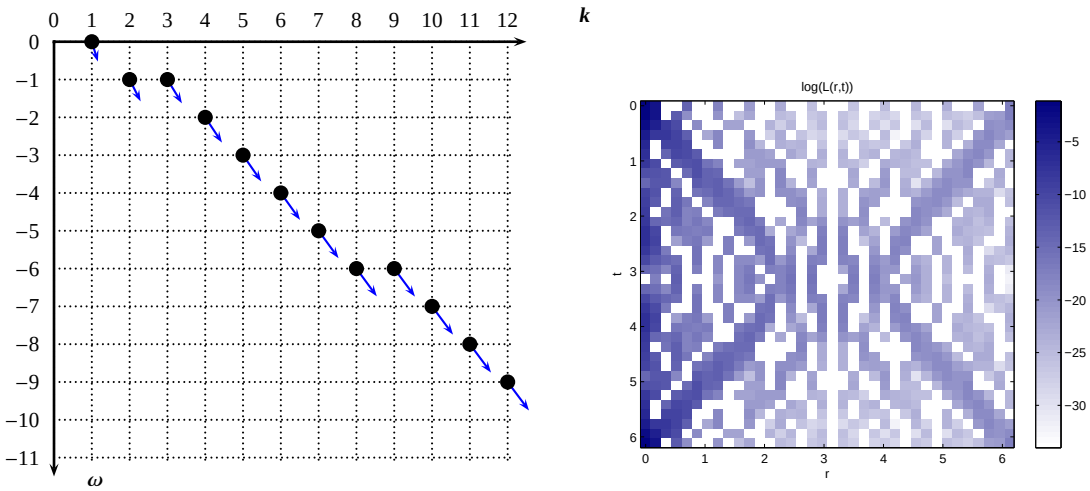


(b) Causal Structure

Figure B.19: $n = 12$, Numerical minimiser

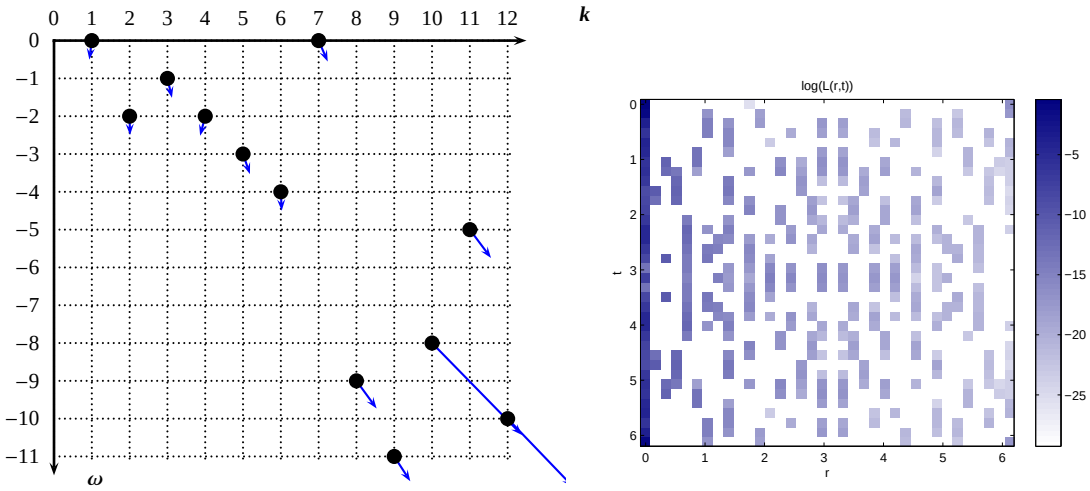
B.6.1 Dirac Sea like Starting Values

$m = 3$



(a) Configuration (b) Causal Structure

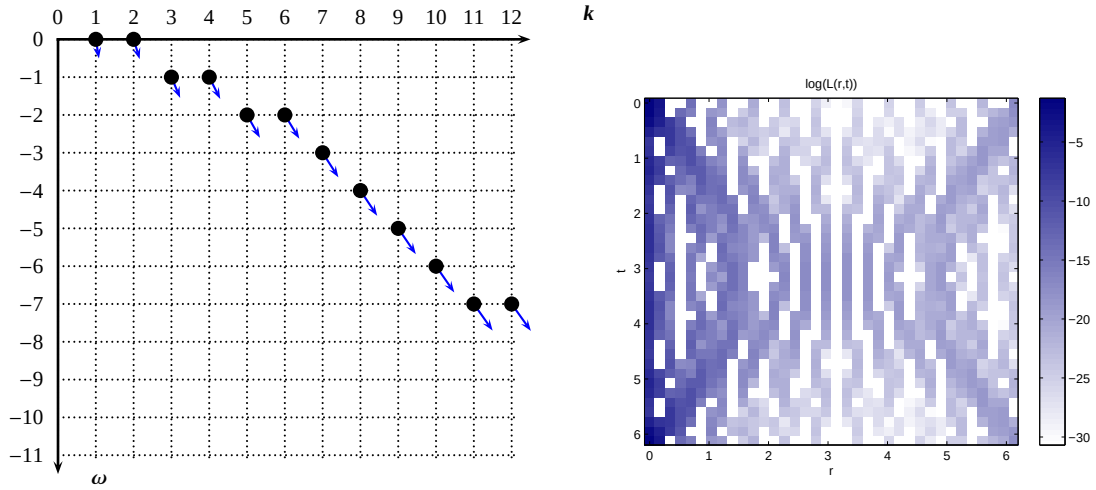
Figure B.20: $n = 12$, Start configuration $m = 3$



(a) Configuration (b) Causal Structure

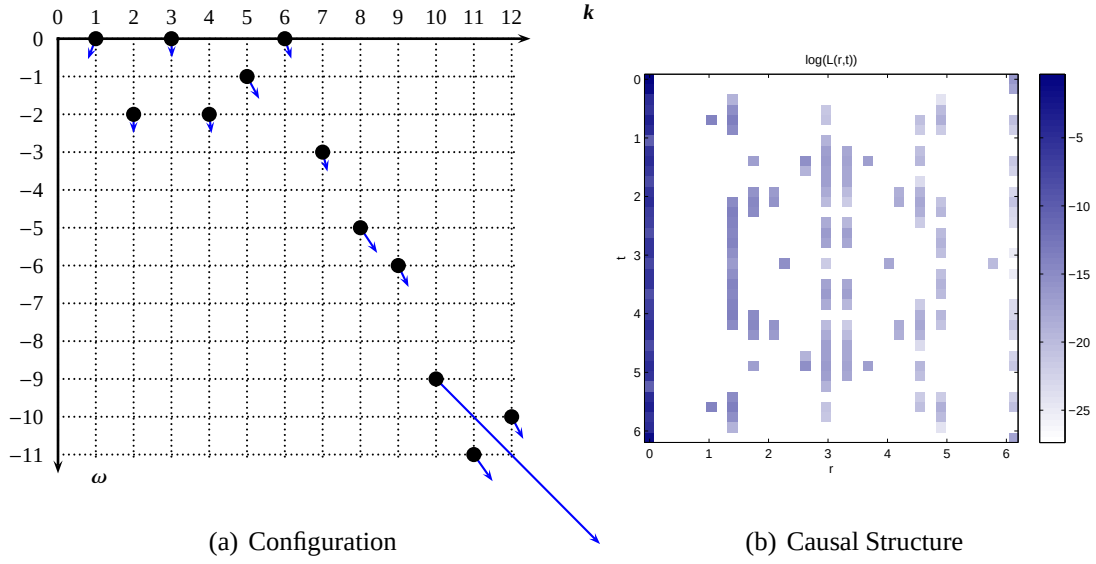
Figure B.21: $n = 12$, End configuration $m = 3$

$m = 6$



(a) Configuration (b) Causal Structure

Figure B.22: $n = 12$, Start configuration $m = 6$



(a) Configuration (b) Causal Structure

Figure B.23: $n = 12$, End configuration $m = 6$

Appendix C

Standard Settings of the NOMADm interface

NOMADM-Variable	Setting
Scaling of Mesh Direction	2.
Filter for Nonlinear Constraints	1.
Run a Sensor Placement Problem	0.
Scheme for Handling Delegate Linear Constraints	sequential
Discard Redundant Linear Constraints	1.
Run as a stochastic Optimisation Problem	0.
Accelerate Convergence	0.
Use Relative Termination Tolerance	0.
Print Debugging Messages on Screen	0.
Save History to Text File	0.
Plot History	1.
Plot Filter (Real time)	1.
R&S Initial Sample Size	5.
R&S Initial Alpha Parameter	0.8
R&S Initial Indifference Zone Parameter	100.
R&S Alpha Decay Factor	0.95
R&S Indifference Zone Decay Factor	0.95
R&S Termination Alpha Parameter FLAG	0.
R&S Termination Alpha Parameter	0.
R&S Terminal Indifference Zone Noise Thr... FLAG	0.
R&S Terminal Indifference Zone Noise Thr...	inf

Figure C.1: NOMADm-Options

NOMADM-Variable	**Setting**
Poll Direction	Standard_2n
Poll Order	Consecutive
Poll Center	0.
Convergence Tolerance (Mesh Size) FLAG	1.
Convergence Tolerance (Mesh Size)	0.0001
Maximum Number of Iterations FLAG	0.
Maximum Number of Iterations	inf
Maximum Number of Function Calls FLAG	1.
Maximum Number of Function Calls	5.0e+04
Maximum CPU Time FLAG	0.
Maximum CPU Time	inf
Maximum Number of consecutive Poll Falls FLAG	0.
Maximum Number of consecutive Poll Falls	inf
Initial Mesh Size	1.
Maximum Mesh Size	inf
Mesh Refining Factor	0.5
Mesh Coarsening Factor	1.
Cache Tolerance	0.0001
Minimum Filter Constraint Violating	0.
Maximum Filter Constraint Violating	1.
MVP Objective Poll Trigger	0.01
MVP Constraints Extended Poll Trigger	0.05

Figure C.2: MADS-Parameter-Settings

List of Figures

Bibliography

[ABR1] Abramson, Mark Aaron: Pattern Search Algorithms for Mixed Variable General Constrainted Optimization Problems; Doctoral Thesis; Rice University, Houston, Texas; 2002; http://www.caam.rice.edu/caam/trs/tr02.html#TR02-11.

[ABR2] Abramson, Mark Aaron: NOMADm version 3.85 User's Guide; 2005.

[AFIT] Air Force Institute of Technology, 2950 Hobson Way, Wright-Patterson AFB, Ohio, 45433-7765 USA, TEL: 937-255-6565.

[ALT] Alt, Werner: Nichtlineare Optimierung; Friedr. Vierweg & Sohn Verlagsgesellschaft; 2002.

[AD1] Audet, C; Dennis, J.E. Jr.: Analysis of generalized pattern searches; Analysis of Generalized Pattern Searches; SIAM Journal on Optimization archive, Volume 13, Issue 3; 2002.

[BAZ/SHE] Bazaraa, M. S.; C. M. Shetty: Nonlinear Programming, Theory and Algorithms; New York; 1979.

[BEK] Beker, S., Puech, N. and Friderikos, V.: A Tabu Search Heuristic for the Offline MPLS Reduced Complexity Layout Design Problem; Lecture Notes in Computer Science; 3042/2004; pp. 514-525; 2004.

[BER] Dimitri P. Bertsekas: Nonlinear Programming; Athena Scientific; 1999.

[BJO] Relativistic Quantum Mechanics; J. D, Bjorken, S. D. Drell; McGraw-Hill Science/Engineering/Math; 1998.

[BOR] Borchers, B. and J.E. Mitchell: An Improved Branch and Bound Algorithm for Mixed Integer Nonlinear Programming; Computers and Operations Research, 21; p. 359-367; 1994.

[BRO] Broyden, C.G.: The Convergence of a Class of Double-Rank Minimization Algorithms; Journal Inst. Math. Applic., Vol. 6, pp. 76-90; 1970.

[CAL] Callender, Craig (Hsg.), Huggett, Nick (Hsg.): Physics Meets Philosophy at the Planck Scale – Contemporary Theories in Quantum Gravity; Cambridge University Press; 2001.

[CLA] Clarke, F. H.:Optimization and Nonsmooth Analysis; SIAM Classics in Applied Mathematics, 5; 1990.

[DAK] Dakin, R.J.: A Tree Search Algorithm for Mixed-Integer Programming Problems; Computer Journal, 8; p. 250-255; 1965.

[DAV] Davis, C: Theory of positive linear dependence; American Journal of Mathe- matics; 76(4):733-746; 1954.

[DUR] Duran, M.A.; Grossmann, I. E.: An outer-approximation algorithm for a class of mixed-integer nonlinear programs; Mathematical Programming, Volume 36, Issue 3 p. 307 - 339 ; 1986.

[FIN1] F. Finster: A Variational Principle in Discrete Space-Time – Existence of Minimisers, math-ph/0503069v1.

[FIN2] F. Finster: The Principle of the Fermionic Projector; International Press; 2006.

[FIN3] F. Finster: Fermion systems in discrete space-time – outer symmetries and spontanuous symmetry breaking; math-ph/0601039; Adv. Math. Phys. 11; 91-146; 2007.

[FIN4] F. Finster: Fermion systems in discrete space-time; hep-th/0601140; J. Phys., Conf. Ser. 67; 012048; 2007.

[FIN5] F. Finster: On the regularised Fermionic Projector of the vacuum; math-ph/0612003v2; 2007.

[FIN/PLA] F. Finster, W. Plaum: A Lattice Model for the Fermionic Projector in a Static and Isotropic Space-Time; Journal-ref: Math. Nachr. 281, No. 6, 803-816; 2008.

[FH] F. Finster, S. Hoch: An action principle for the masses of Dirac particles; arXiv:0712.0678 [math-ph]; 2007.

[FSD] F. Finster, D. Schiefeneder, A. Diethert: Fermion systems in discrete space-time exemplifying the spontaneous generation of a causal structure; Int.J.Mod.Phys.A23:4579-4620; 2008.

[FLE] Fletcher, R., Leyffer, S.: Solving mixed integer nonlinear programs by outer approximation; Journal Mathematical Programming; Volume 66, Numbers 1-3; 1994.

[FLE2] Fletcher, R.: A New Approach to Variable Metric Algorithms," Computer Journal; Vol. 13, pp. 317-322; 1970.

[GEO] Geoffrion, A. M.: Generalized Benders Decomposition; Journal of Optimization Theory and Applications; 10(4); 237-260; 1972.

[GLO1] Glover, F.: Tabu Search – Part I; ORSA Journal on Computing: 1: 3, 190-206; 1989.

[GLO2] Glover, F.: Tabu Search – Part II; ORSA Journal on Computing; 2: 1, 4-32; 1990.

[GLO3] Glover, F. and M. Laguna: Tabu Search; Kluwer Academic Publishers; 1997.

[GOL] Goldfarb, D.: A Family of Variable Metric Updates Derived by Variational Means," Mathematics of Computing; Vol. 24, pp. 23-26; 1970.

[GRO] Grossmann, I.E.: Mixed-integer nonlinear programming techniques for the synthesis of engineering; Research in Engineering Design; Volume 1, Numbers 3-4 / September 1990; p. 205-228; 1990.

[GUP] Gupta, O. K. and Ravindran, V.: Branch and Bound Experiments in Convex Nonlinear Integer Programming; Management Science, 31(12); pp. 1533-1546; 1985.

[HAR] Hartmann, A. K. and H. Rieger: Optimization Algorithms in Physics; Wiley-VCH; 2002.

[HEM] Hemaspaandra, Lane A.: SIGACT News Complexity Theory Column 36; http://www.cs.umd.edu/ gasarch/papers/poll.ps.

[HUK] Hukushima, K. and K. Nemoto; J.Phys. Soc. Jpn 65; 1604; 1996.

[IMA] Inan, R. L.; Davenport, J. M.; Zeigler, D. K.: Latin hypercube sampling (Program user's guide); SANDIA Laboratory report SAND79-1473; 1980.

[KAL] Kallrath, Josef: Gemischt-ganzzahlige Optimierung: Modellierung in der Praxis, Vieweg, 2002.

[KAR] Karush, W.: Minima of functions of several variables with inequalities as side conditions; Master's thesis; Department of Mathematics; University of Chicago; 1939.

[KEL] Kelley Jr., J.E.: The Cutting-Plane Method for Solving Convex Programs; Journal of SIAM 8, pp. 703-712; 1960.

[KUH] Kuhn, H. W.; Tucker, A. W.: Nonlinear programming. In: Neyman, J. ()editor): Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability; Berkeley, CA; 1951.

[LEY] Leyffer, S.: Integrating SQP and Branch and Bound for Mixed Integer Nonlinear Programming; Optimization and Engineering, Volume 3, Number 3; pp. 227-252; September 2002.

[MAJ] S. Majid, Shahn.: Meaning of noncommutative geometry and the Planck scale quantum group, in Towards Quantum Gravity, Springer Lect. Notes Phys. 541; 227-276; 2000. (Also avaiable at hep-th/0006166.)

[MET] N. Metropolis, A. N. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller: Equation of state calculation by fast computing machines; Journal of Chemical Physics, 21(6):1087-1092; 1953.

[MCK] McKay, M. D., W. J. Conover and R. J. Beckman: A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code; Technometrics 21: 239-245; . 1979.

[NAB] Nabar, S. and Schrage, L.: Modeling and Solving Nonlinear Integer Programming Problems; Presented at Annual AIChE Meeting; Chicago, 1991.

[NOMADm] Abramson, Mark, A.: NOMADm Optimization Software; http://www.gerad.ca/NOMAD/nomadm.html; 2008.

[PAR] Pardalos, P. M. (Hsg.); Resende, Mauricio G. C. (Hsg.); Pardalos, Panos M. (Hsg.): Handbook of Applied Optimization; Oxford University Press; 2002.

[POER] Pörn, Ray: Mixed integer non-linear programming; Turku, Åbo; Akad. Univ.; PhD. thesis.; 2000.

[PLB] Plaum, Burkhard: Optimierung von überdimensionalen Hohlleiterkomponenten; Dissertation am Institut für Plasmaforschung an der Universität Stutttgart; 2001.

[PY1] Parallel Python: http://www.parallelpython.com/ (23.9.2008).

[PY2] Python for Numerics and Science: http://wiki.python.org/moin/NumericAndScientific (23.9.2008).

[PY3] Some Propaganda: Python instead of Matlab for plotting?: http://www.ii.uib.no/ avle/python.html(23.9.2008).

[QUA] Quarteroni, Alfio; Sacco, Riccardo; Saleri, Fausto: Numerische Mathematik; Springer; 2002.

[REB] Rebhan, Eckhard: Theoretische Physik, Bd.2, Quanten-, Quantenfeld-, Elementarteilchen-, WÃ¯Â¿Â½metheorie; Spektrum Akademischer Verlag; 2004.

[REC] Press, William H.; Flannery, Brian P.; Teukolsky, Saul A.; Vetterling, William T.: Numerical Recipes in C; Cambridge University Press; 1992.

[ROV] Rovelli, Carlo: Quantum Gravity; Cambridge University Press; 2004.

[RUS] Russo, M. ans Lakhmi C. Jain: An introduction to evolutionary computing; The Crc Press International Series On Computational Intelligence archive, Evolution of engineering and information systems and their applications; pp. 3 - 29; 1999.

[SHA] Shanno, D.F.: Conditioning of Quasi-Newton Methods for Function Minimization; Mathematics of Computing; Vol. 24, pp. 647-656; 1970.

[SRI] T. A. Sriver and J. W. Chrissis: A Framework for Mixed-Variable Optimization Under Uncertainty Using Surrogates and Statistical Selection, 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 30 August-1 September 2004, Albany, NY; 2004.

[SAN] Java applett demonstrating simulated annealing for the travellling salesman problem (23.10.2006) http://www.heatonresearch.com/articles/64/page1.html

[SRI] Sriver, Todd A.; Chrissis, James W.: A Framework for Mixed-Variable Optimization Under Uncertainty Using Surrogates and Statistical Selection; 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, New York; 30 August - 1 September, 2004.

[STU] Stubbs, R.A. and S.Mehrotra: A Branch and Cut Method for 0-1 Mixed Convex Programming; presented at INFORMS Meeting; Washington, 1996.

[TAB] Homepage of Tabu Search: http://www.tabusearch.net/; 25.10.2006.

[TOR] Torczon, V.: On the Convergence of Pattern Search Algorithms; SIAM Journal on Optimization; Vol. 7, No.1; p.1-25; 1997.

[WES] Westerlund, T. and F. Pettersson: A Cutting Plane Method for Solving Convex MINLP Problems; Computers and Chemical Engineering, 19, S131-S136; 1995.

[WOL] Wolsey, Laurence A.: Integer Programming; Wiley-Interscience; 1998.