



Technische Universität München

Numerical Studies of Quantum Lattice Systems with the Tensor Renormalization Group

Robert Bamler
(Robert.Bamler@ph.tum.de)

Diploma Thesis in Physics

submitted to the

Physics Department of the Technical University of Munich

written at the

Arnold Sommerfeld Center for Theoretical Physics
of the Ludwig-Maximilians-University of Munich

supervised by

Prof. Dr. Wilhelm Zwerger
Prof. Dr. Ulrich Schollwöck

December 2011

Abstract

We study the ground state properties of both bosonic and fermionic systems with short-range interactions on the honeycomb lattice using the tensor entanglement renormalization group algorithm (TRG) [1] and its improvement, the second tensor renormalization group (SRG) [2]. We describe how local abelian symmetries can be exploited to significantly reduce computational costs. For fermionic systems, we describe a renormalization group method which naturally extends the Graßmann tensor entanglement renormalization group (GTERG) [3] to systems that are symmetric under an arbitrary symmetry group that has \mathbb{Z}_2 as a subgroup. The developed methods reproduce expectation values of a number of analytically solvable test systems to high accuracy, but exploiting local symmetries makes the method vulnerable to getting stuck in a trial wave function that does not correspond to the global but rather to some local energy minimum. Results of our method for the spin- $\frac{1}{2}$ Heisenberg model agree well with QMC results. Results for the Hubbard model suggest no existence of a topologically ordered phase but suffer from large numerical errors.

Introduction

The study of strongly correlated quantum mechanical systems in two dimensions has gained great interest in physics as a variety of exciting phases of matter have been observed in these systems. High-temperature superconductors are composed in stacks of two-dimensional layers that interact only weakly with each other [4,5]. Despite their potential applications, the nature of the interaction that leads to the superconducting phase is still not fully understood. As a further example, topologically ordered phases were observed in two-dimensional electron gases in the fractional quantum Hall effect [6]. Topologically ordered phases form a novel type of phases that go beyond the description by Landau theory of symmetry breaking [7].

The theoretical description of strongly correlated systems is often approached by the study of simplified model Hamiltonians that are believed to capture the key mechanisms of the investigated physical effect. Examples are the Heisenberg model of (anti)ferromagnetism [8] or the Hubbard model of interacting fermions [9], which is believed to give insight into mechanism behind high-temperature superconductivity. As a result of recent development in experiments with ultracold atomic gases in optical lattices, these model Hamiltonians can nowadays be realized experimentally [10,11] and one can tune the parameters, such as tunneling amplitudes or interaction strengths.

The numerical simulation of strongly correlated 2D systems remains a challenging problem. As the dimension of the Hilbert space grows exponentially with the system size, exact calculations are limited to relatively small systems (see, e.g., [12]). Due to strong quantum mechanical fluctuations, especially in frustrated systems, there is no obvious small parameter for perturbative expansions. Quantum Monte Carlo methods (QMC) provide accurate results for interacting spin or boson models without frustration [13,14], as well as in some special fermion models [15]. However, QMC suffers from the “minus-sign” problem in frustrated spin models and in most fermion systems [16].

In recent years, a multitude of algorithms were proposed to address these problems. The solution of the Kondo problem by Wilson with a numerical renormalization group method [17] sparked interest in the application of renormalization group ideas to numerical algorithms. The density matrix renormalization group (DMRG), introduced by White [18], is now a well-established method in the study of one-dimensional systems [19]. Efforts exist to extend DMRG to two-dimensional systems [20–22] and the method produces good results for narrow stripes or small cylinders [23–25]. However, the accuracy of DMRG decreases with growing width of the stripes.

New insight from quantum information theory explains the limitations of DMRG in two dimensions. In the ground states of gapped Hamiltonians with local interactions, the amount of entanglement between two parts of the system is expected to be asymptotically proportional to the size of the boundary between the subsystems [26,27]. It was pointed out by Östlund and Rommer [28,29] that DMRG produces wave functions in the form of so-called matrix product states (MPS). In these states, the entanglement between

two parts of the system is bounded by a constant. Therefore, MPS cannot approximate ground states of two-dimensional systems in the thermodynamic limit.

The DMRG is a variational method that uses MPS as ansatz wave functions to approximate the ground state of a Hamiltonian [30]. The generalization of this idea to higher spatial dimensions lead to a number of variational algorithms that use new classes of ansatz states which are tailored to capture the appropriate entanglement scaling. Prominent examples include algorithms based on projected entangled-pair states (PEPS) [31–33] and the multiscale entanglement renormalization ansatz (MERA) [34, 35].

In this thesis, we study the tensor entanglement renormalization group algorithm (TRG) proposed by Levin and Nave [1] (see also [3, 36–38]) and its improvement, the second tensor renormalization group (SRG) proposed by Xie et al. [2, 39]. The SRG algorithm uses PEPS to calculate ground state properties of Hamiltonians defined on the honeycomb (or hexagonal) lattice. In contrast to the algorithms mentioned above, the SRG is not a variational algorithm but uses a projective method to find an approximate ground state. This is expected to allow more free parameters in the ansatz states than what would be accessible with variational methods [36]. The honeycomb lattice is the computationally least difficult lattice for a study with PEPS since each site has only three neighbors. It is physically realized, e.g., in iridium oxides (Li_2IrO_3 and Na_2IrO_3), which are discussed as candidate systems for a topologically ordered phase [40–42].

We extend the SRG algorithm by exploiting local abelian symmetries. This has already been proposed in [39] but, to our knowledge, not yet been done. We analyze how the accuracy of numerical results depends on the parameters of the algorithm. Finally, we apply the algorithm to the spin- $\frac{1}{2}$ Heisenberg model and to the Hubbard model.

As a result, the SRG algorithm calculates accurate expectation values for a number of exactly solvable test systems. Exploiting local symmetries dramatically reduces computational costs and thereby extends the use cases of SRG to systems that would otherwise be inaccessible to this method. We present results for the ground state energy and the staggered magnetization of the spin- $\frac{1}{2}$ Heisenberg model. Our results for the Hubbard model suggest no existence of topological order but suffer from high numerical errors.

Outline

Chapter 1 gives a short introduction to the formalism of PEPS and motivates their choice for the approximation of ground states of local Hamiltonians. We present the SRG algorithm in its most basic form in chapter 2 and then describe three improvements to the algorithm in chapter 3: We significantly reduce computational costs by exploiting local symmetries (section 3.1) and we describe how the algorithm has been extended to handle fermionic systems (section 3.2, [3]) and to calculate Rényi entropies (section 3.3, [38, 43]). Chapter 4 analyses how the accuracy of numerical results depends on the parameters of the algorithm. In chapter 5, we use the SRG algorithm to study two physical systems, the spin- $\frac{1}{2}$ Heisenberg model and the Hubbard model, and compare numerical results of our simulations to results obtained with alternative methods. We conclude with a discussion of the developed methods and an outlook to future work.

Contents

Introduction	3
1. Preliminaries: Projected entangled-pair states	7
1.1. One-dimensional systems: matrix product states (MPS)	7
1.1.1. Definition and properties	7
1.1.2. Calculations with MPS	13
1.2. More than one dimension: PEPS	15
2. The basic algorithms	19
2.1. Finding the ground state of a model Hamiltonian	19
2.1.1. Imaginary time evolution	19
2.1.2. Implementation with PEPS	21
2.1.3. Summary of the parameters	25
2.2. Calculating physical observables	26
2.2.1. Tensor Renormalization Group (TRG)	27
2.2.2. Second Tensor Renormalization Group (SRG)	30
2.2.3. Summary of the parameters	33
2.3. Current state of the TRG method	34
3. Extending the algorithm	37
3.1. Exploiting local symmetries	37
3.1.1. From symmetric states to symmetric tensors	38
3.1.2. Symmetric tensors for abelian symmetries	42
3.1.3. Calculations with symmetric tensors & measured performance gain	44
3.2. Fermionic systems	47
3.2.1. Failure of naive PEPS for fermionic systems	49
3.2.2. Fermionic tensors	50
3.2.3. Calculations with fermionic tensors	54
3.3. Rényi entropy and topological order	58
3.3.1. An example system with topological order	59
3.3.2. Calculating Rényi entropies	61
4. Results I: characterization of the algorithms	63
4.1. Qualitative verification of TRG & SRG	63
4.1.1. Classical statistical system	63
4.1.2. String-net condensate and topological entropy	65
4.1.3. Fermion pairing states	67
4.2. Quantitative analysis of the parameter space	69
4.2.1. SRG: N , N^{env} , and D_{cut}	69

4.2.2. Imaginary time evolution: $ \Psi(\tau = 0)\rangle$, D , $\Delta\tau$, and τ	73
5. Results II: the Heisenberg model and the Hubbard model	77
5.1. The Heisenberg model	77
5.2. The Hubbard model	79
5.2.1. Quality of the simulations	79
5.2.2. Results	81
Conclusions and outlook	83
A. Rényi Entropy of the string-net condensate state	85
B. Analytic solution of the fermion pairing state	87
B.1. One-dimensional case	87
B.2. Two-dimensional case	89
C. Short manual to the developed programs	91
C.1. Usage of the programs	91
C.2. Organization of the source code	93

1. Preliminaries: Projected entangled-pair states

Before describing the algorithms we implemented for this thesis (chapter 2), we take a look at the physical systems that we want to describe. The aim of the algorithms is to find the ground state of a many-body Hamiltonian and to calculate quantum mechanical expectation values in this ground state. The main problem arises from the fact that the dimension of the Hilbert space of a many-body system grows exponentially with the system size. Finding the ground state of a given Hamiltonian may therefore seem to be computationally impossible for large system sizes. However, in case of a gapped local Hamiltonian, the ground state satisfies certain properties that effectively limit the size of the relevant part of the Hilbert space. These properties are referred to as the area laws of entanglement entropy [26, 27].

In this chapter, we introduce an efficient parametrization of many body wave functions that is tailored to allow to approximate the ground states of local Hamiltonians. We begin with a description for one-dimensional systems to motivate the approximations. This leads to the definition of the so called matrix product states (MPS). We limit the discussion of MPS to what is needed to motivate their usage and to understand basic operations on MPS. For a more in-depth discussion, see [30]. We then generalize MPS to higher spatial dimensions, which leads to projected entangled-pair states (PEPS). We conclude with an overview of alternative parametrizations of many body wave functions and algorithms that operate on those sets of states.

1.1. One-dimensional systems: matrix product states (MPS)

1.1.1. Definition and properties

We consider a chain of N sites labeled by $i = 1, \dots, N$. For each site i , we define a d -dimensional Hilbert space $\mathcal{H}_{\text{loc}}^{[i]}$ with orthonormal basis $\{|\sigma_i\rangle\}_{\sigma_i=1,\dots,d}$ that contains the local degrees of freedom. For concreteness, consider a chain of atoms that can each be in one of $d = 2$ different hyperfine structure states $|\uparrow_i\rangle$ and $|\downarrow_i\rangle$. An orthonormal basis of the Hilbert space of the complete chain is given by the product states $|\sigma_1, \dots, \sigma_N\rangle := |\sigma_1\rangle \otimes \dots \otimes |\sigma_N\rangle$ and we can express an arbitrary state $|\Psi\rangle$ in this basis,

$$|\Psi\rangle = \sum_{\sigma_1, \dots, \sigma_N=1}^d \psi_{\sigma_1, \dots, \sigma_N} |\sigma_1, \dots, \sigma_N\rangle, \quad (1.1)$$

with d^N coordinates $\psi_{\sigma_1, \dots, \sigma_N}$. To represent the state $|\Psi\rangle$ in a computer algorithm, one would have to store d^N complex numbers, which would be impossible for large system sizes (the current record for exact numerical diagonalization lies at $N = 42$ spins on the

1. Preliminaries: Projected entangled-pair states

honeycomb lattice [12]). We therefore introduce a reduced set of ansatz states that can be encoded with a small number of parameters but still allow to approximate ground states of physically relevant Hamiltonians.

Schmidt-decomposition and entanglement entropy. Let us partition the chain into two subsystems $A := \{1, \dots, \ell\}$ and $B := \{\ell + 1, \dots, N\}$ with $1 \leq \ell < N$. We rewrite (1.1) as

$$|\Psi\rangle = \sum_{\sigma_1, \dots, \sigma_\ell=1}^d \sum_{\sigma_{\ell+1}, \dots, \sigma_N=1}^d \psi_{(\sigma_1, \dots, \sigma_\ell), (\sigma_{\ell+1}, \dots, \sigma_N)} |\sigma_1, \dots, \sigma_\ell\rangle \otimes |\sigma_{\ell+1}, \dots, \sigma_N\rangle, \quad (1.2)$$

where we have grouped the indices of ψ . The states $|\sigma_1, \dots, \sigma_\ell\rangle$ and $|\sigma_{\ell+1}, \dots, \sigma_N\rangle$ form orthonormal bases of the Hilbert spaces for subsystems A and B , respectively. So far, (1.2) differs from (1.1) only in notation. We now recall that we can factorize any complex matrix A into a singular value decomposition $A = U\Lambda V^\dagger$ where Λ is a diagonal matrix with only non-negative real entries (the singular values of A), and the columns of U as well as the rows of V^\dagger each build an orthonormal set. The dimension of Λ is the smaller one of the row and the column dimensions of A . We interpret ψ in (1.2) as a matrix with row-index $(\sigma_1, \dots, \sigma_\ell)$ and column index $(\sigma_{\ell+1}, \dots, \sigma_N)$ and perform a singular value decomposition,

$$\psi_{(\sigma_1, \dots, \sigma_\ell), (\sigma_{\ell+1}, \dots, \sigma_N)} = \sum_{\alpha=1}^{\kappa} U_{(\sigma_1, \dots, \sigma_\ell), \alpha} \lambda_\alpha V_{\alpha, (\sigma_{\ell+1}, \dots, \sigma_N)}^\dagger, \quad (1.3)$$

where λ_α with $\alpha = 1, \dots, \kappa$ enumerate the non-vanishing singular values in decreasing order. We rewrite $|\Psi\rangle$ as

$$|\Psi\rangle = \sum_{\alpha=1}^{\kappa} \lambda_\alpha |\varphi_\alpha^A\rangle \otimes |\varphi_\alpha^B\rangle \quad (1.4)$$

with

$$|\varphi_\alpha^A\rangle := \sum_{\sigma_1, \dots, \sigma_\ell=1}^d U_{(\sigma_1, \dots, \sigma_\ell), \alpha} |\sigma_1, \dots, \sigma_\ell\rangle; \quad \text{and} \quad (1.5)$$

$$|\varphi_\alpha^B\rangle := \sum_{\sigma_{\ell+1}, \dots, \sigma_N=1}^d V_{\alpha, (\sigma_{\ell+1}, \dots, \sigma_N)}^\dagger |\sigma_{\ell+1}, \dots, \sigma_N\rangle. \quad (1.6)$$

Due to the unitarity of U and V , the states $|\varphi_\alpha^A\rangle$ and $|\varphi_\alpha^B\rangle$ each build an orthonormal set within the Hilbert space of the respective subsystem. For a normalized state, we therefore have $\langle\Psi|\Psi\rangle = \sum_\alpha \lambda_\alpha^2 = 1$. Equation (1.4) is known as the Schmidt-decomposition of $|\Psi\rangle$ and κ is the Schmidt-rank [44].

To get a picture of the physical meaning of the singular values λ_α in (1.4), we write down the reduced density matrix of the subsystem A ,

$$\rho_A := \text{Tr}_B |\Psi\rangle\langle\Psi| = \sum_{\alpha=1}^{\kappa} \langle\varphi_\alpha^B|\Psi\rangle\langle\Psi|\varphi_\alpha^B\rangle = \sum_{\alpha=1}^{\kappa} \lambda_\alpha^2 |\varphi_\alpha^A\rangle\langle\varphi_\alpha^A|. \quad (1.7)$$

1.1. One-dimensional systems: matrix product states (MPS)

The reduced density matrix ρ_A contains all information one can obtain about $|\Psi\rangle$ by performing local measurements on the subsystem A . Equation (1.7) states that the subsystem A is in any of the states $|\varphi_\alpha^A\rangle$ with respective probability λ_α^2 . If the Schmidt-rank $\kappa = 1$ then the subsystem A is in a pure state, $\rho_A = |\varphi_1^A\rangle\langle\varphi_1^A|$, and measurements on the subsystem A have no influence on the state of the subsystem B . In contrast, if $\kappa > 1$, then the subsystem A may be in one of several states $|\varphi_\alpha^A\rangle$. If we carry out a measurement on subsystem A that distinguishes between these states, then ρ_A collapses to a pure state $|\varphi_{\alpha_0}^A\rangle\langle\varphi_{\alpha_0}^A|$. According to (1.4), subsystem B is then in the state $|\varphi_{\alpha_0}^B\rangle$, i.e. a measurement on the subsystem A influences the state of the subsystem B . We call a state $|\Psi\rangle$ with this property an *entangled* state. A measure of the amount of entanglement between two complementary subsystems A and B is the von Neumann entanglement entropy,

$$S_{A|B} := -\text{Tr}(\rho_A \log_2(\rho_A)) = -\sum_{\alpha=1}^{\kappa} \lambda_\alpha^2 \log_2(\lambda_\alpha^2). \quad (1.8)$$

Construction of MPS for arbitrary states. Writing the state $|\Psi\rangle$ in a Schmidt-decomposition (1.4) allows us to directly read off the entanglement entropy (1.8) for a bipartition into $A = \{1, \dots, \ell\}$ and $B = \{\ell + 1, \dots, N\}$. We now introduce a notation that allows us to read off the entanglement entropy for *any* bipartition. We follow [45] for the construction.

We first perform a Schmidt-decomposition of $|\Psi\rangle$ into the subsystems $A := \{1\}$ and $B := \{2, \dots, N\}$. Equation (1.3) then reads

$$\psi_{\sigma_1, (\sigma_2, \dots, \sigma_N)} = \sum_{\alpha_1=1}^{\kappa_1} U_{\sigma_1, \alpha_1}^{[1]} \lambda_{\alpha_1}^{[1]} V_{\alpha_1, (\sigma_2, \dots, \sigma_N)}^{[1] \dagger} \quad (1.9)$$

For each $\sigma_1 = 1, \dots, d$, we define a (row) vector $\Gamma_{\alpha_1}^{[1] \sigma_1}$ by $\Gamma_{\alpha_1}^{[1] \sigma_1} := U_{\sigma_1, \alpha_1}^{[1]}$. We then focus on the remaining part $\lambda_{\alpha_1}^{[1]} V_{\alpha_1, (\sigma_2, \dots, \sigma_N)}^{[1] \dagger}$ on the right hand side of (1.9), whose square is the reduced density matrix of the subsystem B . We perform a singular value decomposition

$$\lambda_{\alpha_1}^{[1]} V_{\alpha_1, (\sigma_2, \dots, \sigma_N)}^{[1] \dagger} = \sum_{\alpha_2=1}^{\kappa_2} U_{(\alpha_1, \sigma_2), \alpha_2}^{[2]} \lambda_{\alpha_2}^{[2]} V_{\alpha_2, (\sigma_3, \dots, \sigma_N)}^{[2] \dagger} \quad (1.10)$$

For each σ_2 , we define a matrix $\Gamma_{\alpha_1}^{[2] \sigma_2}$ such that $\lambda_{\alpha_1}^{[1]} \Gamma_{\alpha_1, \alpha_2}^{[2] \sigma_2} = U_{(\alpha_1, \sigma_2), \alpha_2}^{[2]}$. The coordinates of the wave function are then given by

$$\begin{aligned} \psi_{\sigma_1, \dots, \sigma_N} &= \sum_{\alpha_1=1}^{\kappa_1} \sum_{\alpha_2=1}^{\kappa_2} \Gamma_{\alpha_1}^{[1] \sigma_1} U_{(\alpha_1, \sigma_2), \alpha_2}^{[2]} \lambda_{\alpha_2}^{[2]} V_{\alpha_2, (\sigma_3, \dots, \sigma_N)}^{[2] \dagger} \\ &= \sum_{\alpha_1=1}^{\kappa_1} \sum_{\alpha_2=1}^{\kappa_2} \left(\Gamma_{\alpha_1}^{[1] \sigma_1} \lambda_{\alpha_1}^{[1]} \Gamma_{\alpha_1, \alpha_2}^{[2] \sigma_2} \right) \left(\lambda_{\alpha_2}^{[2]} V_{\alpha_2, (\sigma_3, \dots, \sigma_N)}^{[2] \dagger} \right) \quad (1.11) \end{aligned}$$

Next, we perform a singular value decomposition of $\lambda_{\alpha_2}^{[2]} V_{\alpha_2, (\sigma_3, \dots, \sigma_N)}^{[2] \dagger}$ and define the matrices $\Gamma_{\alpha_2}^{[3] \sigma_3}$ similar to the definition of $\Gamma_{\alpha_1}^{[2] \sigma_2}$. Continuing this construction eventually leads

1. Preliminaries: Projected entangled-pair states

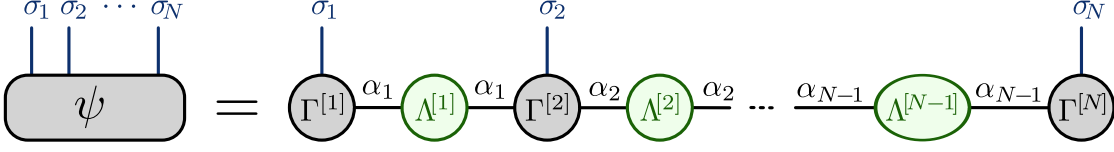


Figure 1.1.: Graphical representation of an MPS with open boundary conditions (1.12). A set of matrices $\Gamma^{[i]\sigma_i}$ is defined on each site i and a diagonal matrix $\Lambda^{[i]}$ on each link between sites i and $i+1$. The first (last) Γ matrix is actually a row (column) vector. The coordinates of the wave function are given by the product of the corresponding matrices.

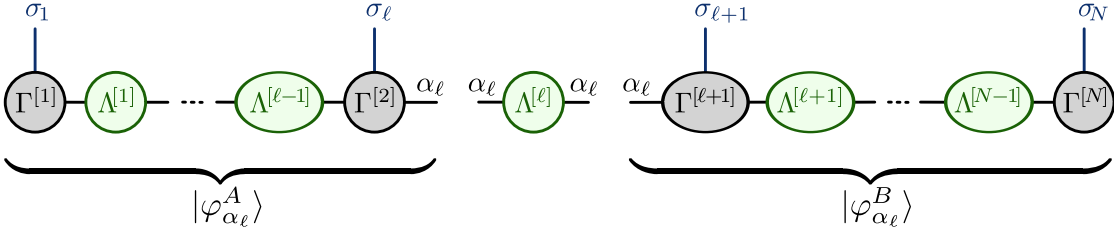


Figure 1.2.: Schmidt decomposition of an MPS (1.13). The states $|\varphi_{\alpha_\ell}^A\rangle$ and $|\varphi_{\alpha_\ell}^B\rangle$ both form orthonormal sets due to the normalization relations illustrated in figure 1.3.

to

$$\begin{aligned} \psi_{\sigma_1, \dots, \sigma_N} &= \sum_{\alpha_1, \dots, \alpha_{N-1}} \Gamma_{\alpha_1}^{[1]\sigma_1} \lambda_{\alpha_1}^{[1]} \Gamma_{\alpha_1, \alpha_2}^{[2]\sigma_2} \lambda_{\alpha_2}^{[2]} \Gamma_{\alpha_2, \alpha_3}^{[3]\sigma_3} \lambda_{\alpha_3}^{[3]} \dots \lambda_{\alpha_{N-1}}^{[N-1]} \Gamma_{\alpha_{N-1}}^{[N]\sigma_N} \\ &= \Gamma^{[1]\sigma_1} \Lambda^{[1]} \Gamma^{[2]\sigma_2} \Lambda^{[2]} \dots \Lambda^{[N-1]} \Gamma^{[N]\sigma_N} \end{aligned} \quad (1.12)$$

where the last line is meant as matrix multiplications where each $\Lambda^{[i]}$ is a diagonal matrix with entries $\lambda_{\alpha_i}^{[i]}$. The last matrix $\Gamma^{[N]\sigma_N}$ has only one column, i.e. it is a column vector. We illustrate (1.12) in figure 1.1. For each matrix $\Gamma^{[i]\sigma_i}$, we draw a circle with two horizontal and one vertical line, representing the indices α_{i-1} , α_i , and σ_i , respectively. Connected lines are to be summed over, subject to the weights $\lambda_{\alpha_i}^{[i]}$.

We call a state $|\Psi\rangle$ whose coordinates are given in the decomposition (1.12) a *matrix product state (MPS)* with open boundary conditions [28, 46]. It is easy to see that the Schmidt-decomposition (1.4) for a bipartition into subsystems $A = \{1, \dots, \ell\}$ and $B = \{\ell+1, \dots, N\}$ of an MPS reads (cf. figure 1.2)

$$|\Psi\rangle = \sum_{\alpha_\ell=1}^{\kappa_\ell} \lambda_{\alpha_\ell}^{[\ell]} |\varphi_{\alpha_\ell}^A\rangle \otimes |\varphi_{\alpha_\ell}^B\rangle \quad (1.13)$$

1.1. One-dimensional systems: matrix product states (MPS)

with states

$$|\varphi_{\alpha_\ell}^A\rangle := \sum_{\sigma_1, \dots, \sigma_\ell=1}^d \left(\Gamma^{[1]\sigma_1} \Lambda^{[1]} \dots \Lambda^{[\ell-1]} \Gamma^{[\ell]\sigma_\ell} \right)_{\alpha_\ell} |\sigma_1, \dots, \sigma_\ell\rangle; \quad \text{and} \quad (1.14)$$

$$|\varphi_{\alpha_\ell}^B\rangle := \sum_{\sigma_{\ell+1}, \dots, \sigma_N=1}^d \left(\Gamma^{[\ell+1]\sigma_{\ell+1}} \Lambda^{[\ell+1]} \dots \Lambda^{[N-1]} \Gamma^{[N]\sigma_N} \right)_{\alpha_\ell} |\sigma_{\ell+1}, \dots, \sigma_N\rangle, \quad (1.15)$$

which are orthonormal by construction. Hence, we can read off the entanglement entropy,

$$S_{A|B} = - \sum_{\alpha_\ell=1}^{\kappa_\ell} (\lambda_{\alpha_\ell}^{[\ell]})^2 \log_2((\lambda_{\alpha_\ell}^{[\ell]})^2) \quad . \quad (1.16)$$

Area laws and truncation of MPS. The above construction allows us to write any state $|\Psi\rangle$ as a matrix product state. The matrices $\Gamma^{[i]\sigma_i}$ have dimensions $\kappa_{i-1} \times \kappa_i$. In particular, the first (last) matrix is a row (column) vector. For an arbitrary state, the matrix dimension κ_i grows exponentially as $\kappa_i = d^{\min\{i, N-i\}}$ the further we go from the boundary towards the center of the system, unless some singular values happen to vanish. Therefore, we already need d^N parameters only to encode the matrices for the lattice site in the system center. We now motivate a truncation scheme for MPS that allows us to encode physically relevant states with a number of parameters that grows only linearly with the system size N .

The entanglement entropy (1.16) for a bipartition into subsystems $A := \{1, \dots, \ell\}$ and $B := \{\ell+1, \dots, N\}$ is maximal if the spectrum of the reduced density matrix ρ_A (1.7) is flat, i.e. if $(\lambda_{\alpha_\ell}^{[\ell]})^2 = \kappa_\ell^{-1}$. In this case, we have

$$S_{A|B}^{\max} = \log_2(\kappa_\ell) = \min\{\ell, N - \ell\} \log_2(d), \quad (1.17)$$

i.e. the maximal entanglement entropy is proportional to the size of the smaller one of the subsystems A and B . However, it turns out that states whose entanglement entropy scales as in (1.17) are uncommon in nature. We consider the case in which $|\Psi\rangle$ is the ground state of a Hamiltonian with only local interactions and a gap in the excitation spectrum.¹ The so called area laws of entanglement entropy [26, 27] then state that the entanglement entropy is asymptotically proportional to the size of the *boundary* of the bipartition, i.e. constant for one-dimensional systems. Therefore, the spectrum of ρ_A cannot be flat. In fact, it has been shown [48] for analytically solvable example systems that the spectrum of ρ_A decays asymptotically as

$$(\lambda_{\alpha_\ell}^{[\ell]})^2 \sim \exp(-\text{const} \times (\log_2(\alpha_\ell))^2) \quad . \quad (1.18)$$

The Schmidt decomposition (1.13) then contains terms with exponentially small weight $\lambda_{\alpha_\ell}^{[\ell]}$. We introduce a maximal matrix dimension $D \in \mathbb{N}$ and neglect all terms in (1.13) with $\alpha_\ell > D$. This truncation effectively reduces the Hilbert spaces of the subsystems

¹For critical one-dimensional systems, the entanglement is proportional to the logarithm of the subsystem size [47].

1. Preliminaries: Projected entangled-pair states

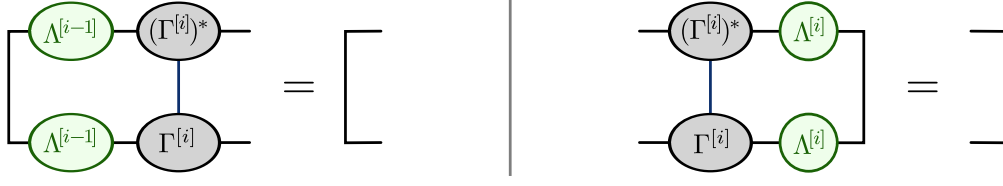


Figure 1.3.: Normalization relations (1.21)–(1.22) for the matrices that compose an MPS.

A and B to subspaces of dimension D each. The truncated wave function

$$|\tilde{\Psi}\rangle := \sum_{\alpha_\ell=1}^D \lambda_{\alpha_\ell}^{[\ell]} |\varphi_{\alpha_\ell}^A\rangle \otimes |\varphi_{\alpha_\ell}^B\rangle \quad (1.19)$$

is the best approximation (in terms of the two-norm) of $|\Psi\rangle$ for fixed dimensions of the Hilbert spaces of the subsystems A and B . This is due to the orthonormality of $\{|\varphi_{\alpha_\ell}^A\rangle\}$ and $\{|\varphi_{\alpha_\ell}^B\rangle\}$. At the same time, keeping the largest singular values λ_{α_ℓ} maximizes the retained entanglement entropy of the truncated state $|\tilde{\Psi}\rangle$ for the chosen bipartition.²

In (1.19), we effectively truncated the dimensions of the diagonal matrix $\Lambda^{[\ell]}$ as well as the column dimensions of $\Gamma^{[\ell]\sigma_\ell}$ and the row dimensions of $\Gamma^{[\ell+1]\sigma_{\ell+1}}$. Let us now truncate *all* matrices in (1.12) to the dimensions $D \times D$. The number of parameters required to specify the state is then of order $O(NdD^2)$, i.e. only proportional to the system size.

MPS with a fixed matrix dimension D can be used to approximate the ground states of gapped local Hamiltonians for one-dimensional systems with a modest number of parameters. This explains the success of the density matrix renormalization group algorithm (DMRG) [18, 19] for one-dimensional systems. The DMRG uses MPS with fixed D as a variational set of ansatz wave functions and minimizes the expectation value of a given Hamiltonian in order to find an approximate ground state [28, 29] (for a recent review of DMRG in the context of MPS, see [30]).

Normalization relations for the matrices. The truncation procedure described in the preceding paragraph is based on the fact that the states $|\varphi_{\alpha_\ell}^A\rangle$ and $|\varphi_{\alpha_\ell}^B\rangle$ defined in (1.14) and (1.15) each build an orthonormal set, for all bipartitions ℓ . The orthonormality of $|\varphi_{\alpha_\ell}^A\rangle$ and $|\varphi_{\alpha_\ell}^B\rangle$ manifests itself in so-called normalization relations of the matrices that define the MPS. By construction (1.10), the matrices $\Gamma^{[i]\sigma_i}$ in (1.12) satisfy the relation

$$\sum_{\sigma_i, \alpha_{i-1}} (\lambda_{\alpha_{i-1}}^{[i-1]})^2 (\Gamma_{\alpha_{i-1}, \alpha_i}^{[i]\sigma_i})^* \Gamma_{\alpha_{i-1}, \alpha_i'}^{[i]\sigma_i} = \delta_{\alpha_i, \alpha_i'} \quad , \quad (1.20)$$

in short,

$$\sum_{\sigma_i} (\Gamma^{[i]\sigma_i})^\dagger (\Lambda^{[i-1]})^2 \Gamma^{[i]\sigma_i} = \mathbb{1}. \quad (1.21)$$

²Note that $|\tilde{\Psi}\rangle$ is not normalized. In this discussion, we neglect corrections to the entanglement entropy due to the new normalization of $|\tilde{\Psi}\rangle$. This is a good approximation if the discarded singular values are small.

1.1. One-dimensional systems: matrix product states (MPS)

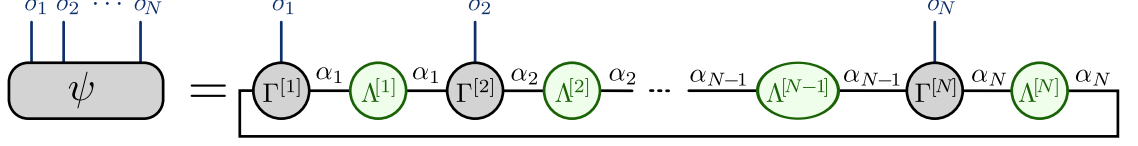


Figure 1.4.: An MPS with periodic boundary conditions (1.23). We introduce a new diagonal matrix $\Lambda^{[N]}$ and trace out the row index of the first matrix with the column index of the last matrix.

It is easy to see that also the relation

$$\sum_{\sigma_i} \Gamma^{[i] \sigma_i} (\Lambda^{[i]})^2 (\Gamma^{[i] \sigma_i})^\dagger = \mathbb{1} \quad (1.22)$$

holds. We illustrate relations (1.21) and (1.22) in figure 1.3. The normalization relations also hold for the first and the last site if we define dummy diagonal matrices $\Lambda^{[0]}$ and $\Lambda^{[N]}$ of dimensions 1×1 with matrix element 1. It is easy to see that equations (1.21)–(1.22) imply orthonormality of $|\varphi_{\alpha_\ell}^A\rangle$ and $|\varphi_{\alpha_\ell}^B\rangle$ for all ℓ .

It should be mentioned that an alternative notation for MPS is often used in the context of DMRG. One combines two successive matrices to left-normalized matrices $A^{[i] \sigma_i} := \Lambda^{[i-1]} \Gamma^{[i] \sigma_i}$ for $i = 1, \dots, \ell$ and to right-normalized matrices $B^{[i] \sigma_i} := \Gamma^{[i] \sigma_i} \Lambda^{[i]}$ for $i = \ell + 1, \dots, N$ (see [30]).

Translationally invariant MPS. In (1.12), the matrices $\Gamma^{[i] \sigma_i}$ and $\Lambda^{[i]}$ explicitly depend on the position i and the first (last) Γ matrix is actually a row (column) vector so that the right hand side of (1.12) evaluates to a scalar. When studying translationally invariant systems in the thermodynamic limit, it is often useful to express the state as an MPS with position-independent matrices. This means, in particular, that all matrices have the same dimensions $D \times D$, so that the right hand side of (1.12) evaluates to a matrix rather than a scalar. We overcome this difficulty by using periodic boundary conditions (PBC). With PBC, equation (1.12) changes to (cf. figure 1.4)

$$\psi_{\sigma_1, \dots, \sigma_N} = \text{Tr} \left\{ \Gamma^{[1] \sigma_1} \Lambda^{[1]} \Gamma^{[2] \sigma_2} \Lambda^{[2]} \dots \Gamma^{[N] \sigma_N} \Lambda^{[N]} \right\} \quad (1.23)$$

If the matrices in (1.23) do not depend on the position i , then the state is translationally invariant (w.r.t. periodic boundary conditions). The reverse is not necessarily true, i.e. one can easily write down an MPS with PBC that describes a translationally invariant state but whose matrices do depend on the position. But this MPS can formally be transformed into a PEPS with position independent matrices by replacing each matrix $\Gamma^{[i] \sigma_i}$ by the direct sum $\Gamma^{[i], \sigma_i} \oplus \Gamma^{[i+1], \sigma_{i+1}} \oplus \dots \oplus \Gamma^{[i-1], \sigma_{i-1}}$ with cyclic i and then averaging over all Γ matrices.

1.1.2. Calculations with MPS

Applying a bond operator to an MPS. Consider a Hamiltonian $H = \sum_i H_{i,i+1}$ where each bond operator $H_{i,i+1}$ acts only on the local Hilbert spaces of neighboring sites i

1. Preliminaries: Projected entangled-pair states

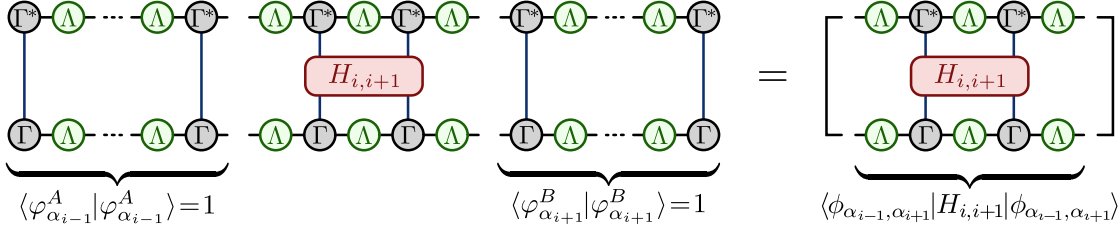


Figure 1.5.: Expectation value of a bond operator in an MPS (1.27). Due to the normalization relations (1.21)–(1.22) (figure 1.3), we can ignore the states left and right from the operator.

and $i+1$. An important operation in the algorithm we describe in section 2.1 is to apply a bond operator like $H_{i,i+1}$ to (a two-dimensional generalization of) MPS. We write the state whose coordinates are given in (1.12) as

$$|\Psi\rangle = \sum_{\alpha_{i-1}, \alpha_{i+1}} |\varphi_{\alpha_{i-1}}^A\rangle \otimes |\phi_{\alpha_{i-1}, \alpha_{i+1}}\rangle \otimes |\varphi_{\alpha_{i+1}}^B\rangle \quad (1.24)$$

where $|\varphi_{\alpha_{i-1}}^A\rangle$ and $|\varphi_{\alpha_{i+1}}^B\rangle$ are defined as in (1.14) and (1.15), respectively, and

$$|\phi_{\alpha_{i-1}, \alpha_{i+1}}\rangle := \sum_{\sigma_i, \sigma_{i+1}=1}^d \left(\Lambda^{[i-1]} \Gamma^{[i] \sigma_i} \Lambda^{[i]} \tilde{\Gamma}^{[i+1] \sigma_{i+1}} \Lambda^{[i+1]} \right)_{\alpha_{i-1}, \alpha_{i+1}} |\sigma_i, \sigma_{i+1}\rangle. \quad (1.25)$$

This is similar to the decomposition in figure 1.2, but with an additional two-site state $|\phi_{\alpha_{i-1}, \alpha_{i+1}}\rangle$ in the center. The bond operator $H_{i,i+1}$ acts only on $|\phi_{\alpha_{i-1}, \alpha_{i+1}}\rangle$. We apply $H_{i,i+1}$ to each of those states and express the result as

$$H_{i,i+1} |\phi_{\alpha_{i-1}, \alpha_{i+1}}\rangle = \sum_{\sigma_i, \sigma_{i+1}=1}^d \left(\Lambda^{[i-1]} \tilde{\Gamma}^{[i] \sigma_i} \tilde{\Lambda}^{[i]} \tilde{\Gamma}^{[i+1] \sigma_{i+1}} \Lambda^{[i+1]} \right)_{\alpha_{i-1}, \alpha_{i+1}} |\sigma_i, \sigma_{i+1}\rangle \quad (1.26)$$

with updated matrices $\tilde{\Gamma}^{[i] \sigma_i}$, $\tilde{\Lambda}^{[i]}$, and $\tilde{\Gamma}^{[i+1] \sigma_{i+1}}$. This can be done by a singular value decomposition similar to the construction of MPS, followed by multiplying the inverse of $\Lambda^{[i-1]}$ and $\Lambda^{[i+1]}$ from the left and from the right, respectively. The new state $|\Psi'\rangle := H_{i,i+1} |\Psi\rangle$ is then again an MPS. It differs from $|\Psi\rangle$ only in the matrices $\tilde{\Gamma}^{[i] \sigma_i}$, $\tilde{\Lambda}^{[i]}$, and $\tilde{\Gamma}^{[i+1] \sigma_{i+1}}$.

We emphasize that we include the diagonal matrices $\Lambda^{[i-1]}$ and $\Lambda^{[i+1]}$ in the definition (1.25) of $|\phi_{\alpha_{i-1}, \alpha_{i+1}}\rangle$. This complicates the factorization (1.26) of $H_{i,i+1} |\phi_{\alpha_{i-1}, \alpha_{i+1}}\rangle$ slightly since we have to multiply inverse diagonal matrices to restore the original form. However, this extra work ensures that the matrices of the new MPS satisfy again the normalization relations (1.21)–(1.22). If the matrices of the original MPS $|\Psi\rangle$ have dimensions $D \times D$, then the updated diagonal matrix $\tilde{\Lambda}^{[i]}$ has dimensions $dD \times dD$. Since all matrices of the new MPS $|\Psi'\rangle$ again satisfy orthonormality relations, we can approximate $|\Psi'\rangle$ with an MPS with matrix dimensions $D \times D$ by simple matrix truncation.

Calculating expectation values of local operators. Consider the bond operators $H_{i,i+1}$ from the preceding paragraph. We can easily calculate the expectation value $\langle \Psi | H_{i,i+1} | \Psi \rangle$

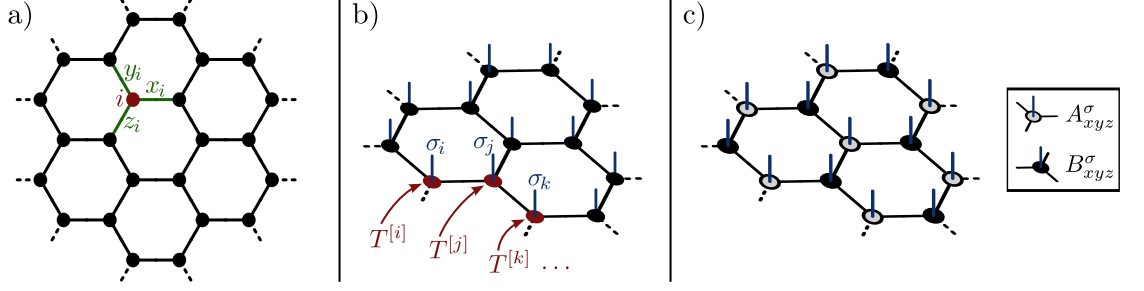


Figure 1.6.: a) The honeycomb lattice with vertex i and links x_i , y_i , and z_i . b) A PEPS on the honeycomb lattice (1.29). On each site i , we define a tensor $T^{[i]}$ (small circles) where each tensor index corresponds to a line that sticks out of the small circle. We sum over indices that correspond to connected lines. c) A translationally invariant PEPS is defined by two tensors A and B since the honeycomb lattice has a two-atomic basis (1.34).

for an MPS $|\Psi\rangle$. We factorize $|\Psi\rangle$ as in (1.24). Since $\{|\varphi_{\alpha_{i-1}}^A\rangle\}_{\alpha_{i-1}}$ and $\{|\varphi_{\alpha_{i+1}}^B\rangle\}_{\alpha_{i+1}}$ are sets of orthonormal states, the expectation value is given by

$$\langle\Psi|H_{i,i+1}|\Psi\rangle = \sum_{\alpha_{i-1}, \alpha_{i+1}} \langle\phi_{\alpha_{i-1}, \alpha_{i+1}}|H_{i,i+1}|\phi_{\alpha_{i-1}, \alpha_{i+1}}\rangle \quad (1.27)$$

where the right hand side can be evaluated with moderate computational costs since $|\phi_{\alpha_{i-1}, \alpha_{i+1}}\rangle$ are only defined on two sites (cf. figure 1.5).

1.2. More than one dimension: PEPS

MPS are most applicable to approximate ground states of one-dimensional Hamiltonians with local interactions since they satisfy the area law of entanglement entropy for one spatial dimension. We now generalize the idea behind the construction of MPS to two (and more) spatial dimensions.

Definition of PEPS. An MPS is defined by the matrices $\Gamma_{\alpha_{i-1}, \alpha_i}^{[i] \sigma_i}$ and $\Lambda^{[i]}$ in (1.12). Our notation of MPS treats the physical index σ_i of the set of matrices $\{\Gamma^{[i] \sigma_i}\}_{\sigma_i=1, \dots, d}$ different than the row and column indices so that we can use matrix multiplication to simplify the notation. For the generalization to higher spatial dimensions, we interpret $\Gamma^{[i]}$ as a single tensor of order three with indices σ_i , α_{i-1} , and α_i . The indices α_{i-1} and α_i connect site i with its left and right neighbor on the one-dimensional chain, respectively.

In more than one spatial dimension, each site has by definition more than two neighbors. We consider a system \mathcal{S} where the local degrees of freedom are located on the vertices of a honeycomb lattice (figure 1.6a). We label the vertices by $i \in \mathbb{N}$ and the links that connect vertex i with its three neighboring vertices by x_i , y_i , and z_i . In analogy to the definition of MPS, we define an order four tensor $T_{x_i y_i z_i}^{[i] \sigma_i}$ on each vertex, where σ_i denotes the local state on vertex i and x_i , y_i , and z_i each run from one to some

1. Preliminaries: Projected entangled-pair states

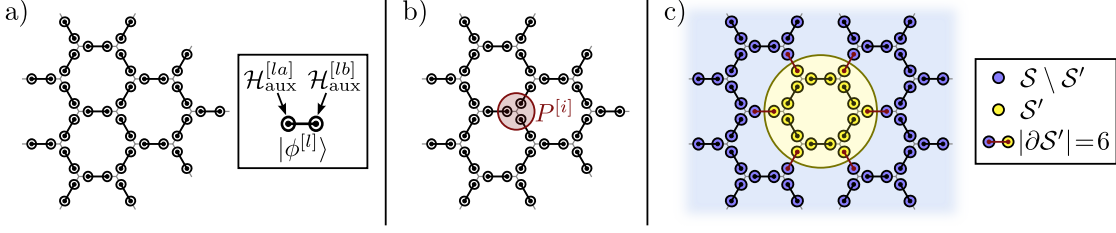


Figure 1.7.: Projective construction of a PEPS. a) Two auxiliary Hilbert spaces $\mathcal{H}_{\text{aux}}^{[la]}$ and $\mathcal{H}_{\text{aux}}^{[lb]}$ occupy each link l of the lattice and we define a maximally entangled state $|\phi^{[l]}\rangle$ on each link (1.30). b) We group the three auxiliary Hilbert spaces around each vertex i together and apply a map $P^{[i]}$ from the auxiliary spaces to the local physical Hilbert space (1.31). c) A bipartition into subsystems S' and $S \setminus S'$ cuts through the auxiliary maximally entangled states on the boundary. This bounds the entanglement to a value proportional to the circumference $|\partial S'|$ (1.33).

bond dimension D . The tensors $T^{[i]}$ define a many-body state $|\Psi\rangle$ with coordinates

$$\psi_{\sigma_1, \dots, \sigma_N} = \text{Tr}_{\{x_i, y_i, z_i\}} \left\{ \prod_i T_{x_i y_i z_i}^{[i] \sigma_i} \right\}, \quad (1.28)$$

where the trace is to sum over all lower indices with the restriction that tensors on neighboring vertices take the same value for the index corresponding to the connecting link. For the state $|\Psi\rangle$, we introduce the notation

$$|\Psi\rangle = \text{Tr} \left\{ \prod_i T_{x_i y_i z_i}^{[i] \sigma_i} |\sigma_i\rangle \right\}, \quad (1.29)$$

where the trace is to sum over lower indices as in (1.28) as well as over the physical configurations σ_i . We call a state that is given in the form (1.29) a *projected entangled-pair state* (PEPS) [32], where the name will become clear in the next paragraph. We illustrate (1.29) in figure 1.6b. Note that, in contrast to our notation for MPS, we did not set any diagonal matrices on the links between two vertices, because, in general, normalization relations like (1.21)–(1.21) do not exist for PEPS. (We will, however, see that the specific algorithm we describe in section 2.1 produces PEPS whose tensors satisfy some *approximate* normalization relations.)

Projective construction and area law. Unlike MPS, PEPS do not allow to directly read off the entanglement entropy for any bipartition. However, we can gain some understanding of the scaling of the entanglement entropy for PEPS with the following consideration [32]. Consider a system where a pair of auxiliary Hilbert spaces $\mathcal{H}_{\text{aux}}^{[la]}$ and $\mathcal{H}_{\text{aux}}^{[lb]}$ is defined on each link l of the honeycomb lattice (figure 1.7a). The auxiliary Hilbert spaces are D -dimensional with basis $\{|\tau_\alpha\rangle\}_{\alpha=1, \dots, D}$. On each link, we define a

state that is maximally entangled between $\mathcal{H}_{\text{aux}}^{[la]}$ and $\mathcal{H}_{\text{aux}}^{[lb]}$,

$$|\phi^{[l]}\rangle := \sum_{\alpha=1}^D |\tau_{\alpha}\rangle \otimes |\tau_{\alpha}\rangle. \quad (1.30)$$

We now associate $\mathcal{H}_{\text{aux}}^{[la]}$ to one of the vertices connected by the link l and $\mathcal{H}_{\text{aux}}^{[lb]}$ to the other one. This attributes three auxiliary Hilbert spaces to each vertex i (figure 1.7b). We define a map $P^{[i]}$ from the product space of these three auxiliary Hilbert spaces to the local Hilbert space,

$$\begin{aligned} P^{[i]} : \mathcal{H}_{\text{aux}}^{[x_i]} \otimes \mathcal{H}_{\text{aux}}^{[y_i]} \otimes \mathcal{H}_{\text{aux}}^{[z_i]} &\longrightarrow \mathcal{H}_{\text{loc}}^{[i]} \\ P^{[i]} = \sum_{\sigma_i=1}^d \sum_{x_i, y_i, z_i=1}^D &T_{x_i y_i z_i}^{[i] \sigma_i} |\sigma_i\rangle (\langle \tau_{x_i}| \otimes \langle \tau_{y_i}| \otimes \langle \tau_{z_i}|) \end{aligned} \quad (1.31)$$

The state $|\Psi\rangle$ from (1.29) can then be written as

$$|\Psi\rangle = \left(\prod_i P^{[i]} \right) \left(\prod_l |\phi^{[l]}\rangle \right), \quad (1.32)$$

i.e., we constructed $|\Psi\rangle$ by projecting an auxiliary state with entangled pairs $|\phi^{[l]}\rangle$ on each link to the physical Hilbert space.

Consider now a bipartition of the system \mathcal{S} into a simply connected subsystem $\mathcal{S}' \subset \mathcal{S}$ and its complement $\mathcal{S} \setminus \mathcal{S}'$ (figure 1.7c). For this bipartition, the entanglement entropy $S_{\mathcal{S}'}^{[\text{aux}]}$ of the auxiliary state $\prod_l |\phi^{[l]}\rangle$ is proportional to the number $|\partial\mathcal{S}'|$ of entangled auxiliary states that connect \mathcal{S}' with $\mathcal{S} \setminus \mathcal{S}'$. The projectors $P^{[i]}$ are local operators, hence they cannot increase the entanglement. Therefore, the entanglement entropy of the state $|\Psi\rangle$ is bounded by

$$S_{\mathcal{S}'} \leq S_{\mathcal{S}'}^{[\text{aux}]} = |\partial\mathcal{S}'| \log_2(D). \quad (1.33)$$

The area law of entanglement entropy states that $S_{\mathcal{S}'}$ scales proportionally to the size $|\partial\mathcal{S}'|$ of the boundary for ground states of two-dimensional gapped local Hamiltonians. Therefore, (1.33) motivates the use of PEPS to approximate such states.

Translationally invariant PEPS. We use PEPS whose tensors do not depend on the position in the lattice to describe translationally invariant states. As the honeycomb lattice has a two-atomic basis, we still need two tensors A and B (figure 1.6c). We introduce the notation

$$|\Psi\rangle = \text{Tr} \left\{ \prod_{i \in \mathcal{A}, j \in \mathcal{B}} A_{x_i y_i z_i}^{\sigma_i} B_{x_j y_j z_j}^{\sigma_j} |\sigma_i \sigma_j\rangle \right\} \quad (1.34)$$

to describe a translationally invariant PEPS on the honeycomb lattice.

1. Preliminaries: Projected entangled-pair states

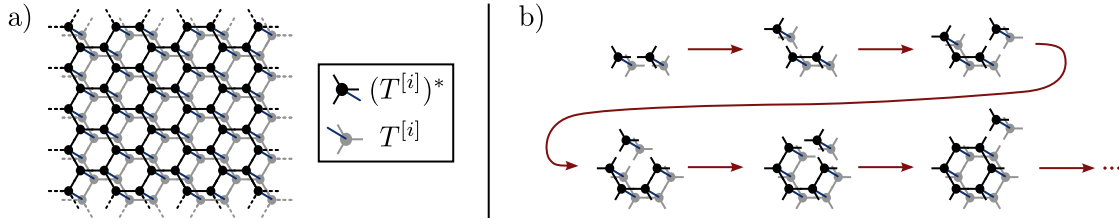


Figure 1.8.: a) To calculate the norm of a PEPS, one has to trace out the illustrated tensor network. (For the calculation of an expectation value, there would additionally be an operator in between the two layers of tensors). b) Multiplying out the tensor network in (a) in the naive way would lead to tensors of ever growing order.

Calculations with PEPS. We can apply a bond operator to a PEPS in a similar way as described for MPS in section 1.1.2. One has to be more careful with the truncation of the updated tensors due to the lack of normalization relations. We discuss this when needed in section 2.1.2.

In contrast to MPS, calculating expectation values of local operators is a computationally challenging problem for PEPS. In the one-dimensional case, cutting out two neighboring sites of the chain partitioned the remaining system into a left remainder A and a right remainder B (see equation (1.24) and figure 1.5) and we were able to define an orthonormal basis in both remainders due to the normalization relations (1.21) and (1.21). This is not the case in two dimensions. One therefore has to explicitly evaluate the product of tensors illustrated in figure 1.8a. Multiplying one tensor to the next, step by step, leads to tensors of growing order as the circumference of the traced-out region grows (figure 1.8b). This evaluation becomes computationally impossible for large system sizes because the computational costs for tensor multiplications scale exponentially with the order of the tensor. We present an efficient algorithm to approximately trace out the tensor network in figure 1.8a in section 2.2.

2. The basic algorithms

In this chapter, we describe an algorithm proposed by Xie et al. [2, 39] to calculate local expectation values in the ground states of two-dimensional lattice systems with local Hamiltonians. The algorithm consists of two parts. In the first part, one calculates a PEPS that approximates the ground state of a given Hamiltonian. This is done by a projective method that starts from a random wave function and iteratively suppresses the contribution of excited states (section 2.1). In the second part, one calculates expectation values of local operators. For this purpose, one expresses the expectation value as a product of tensors, which one evaluates approximately by performing a coarse-graining of the lattice (section 2.2).

2.1. Finding the ground state of a model Hamiltonian

2.1.1. Imaginary time evolution

We follow [39] for the description of the imaginary time evolution algorithm. The method is similar to the infinite time-evolving block decimation used in [49]. We consider a quantum mechanical system defined on the honeycomb lattice. The honeycomb lattice is bipartite, i.e. it can be divided into two sublattices \mathcal{A} and \mathcal{B} such that nearest neighbors of any site $i \in \mathcal{A}$ all belong to the sublattice \mathcal{B} and vice versa (figure 2.1a). We restrict our description to the case in which the Hamiltonian H of the system is translationally invariant and contains only nearest-neighbor interactions. As an example, consider the antiferromagnetic spin- $\frac{1}{2}$ Heisenberg model. In this model, the local Hilbert space $\mathcal{H}_{\text{loc}}^{[i]}$ on site i is $(d = 2)$ -dimensional with basis $\{|\uparrow_i\rangle, |\downarrow_i\rangle\}$ and the Hamiltonian is given by

$$H = \sum_{\langle i,j \rangle} H_{ij} = J \sum_{\langle i,j \rangle} \mathbf{S}_i \cdot \mathbf{S}_j, \quad (2.1)$$

where $J > 0$ is the interaction strength that sets the energy scale, the sum runs over all pairs of nearest neighboring lattice sites and \mathbf{S}_i is the three-component vector of spin operators acting on site i . Note that each bond Hamiltonian H_{ij} acts only on $\mathcal{H}_i \otimes \mathcal{H}_j$.

To find the ground state of the Hamiltonian (2.1), we start with a random wave function $|\Psi(\tau = 0)\rangle$ and project it to the ground state subspace by imaginary time evolution. In our implementation (see next section), $|\Psi(\tau = 0)\rangle$ will be a translationally invariant PEPS. We define the state at imaginary time $\tau \geq 0$,

$$|\Psi(\tau)\rangle := e^{-\tau H} |\Psi(\tau = 0)\rangle. \quad (2.2)$$

The operator $e^{-\tau H}$ suppresses contributions of high-energy eigenstates to the wave function. For τ much larger than the inverse of the typical energy scale, $|\Psi(\tau)\rangle$ converges to the ground state,

$$\lim_{\tau \rightarrow \infty} |\Psi(\tau)\rangle \propto |\Psi_0\rangle, \quad (2.3)$$

2. The basic algorithms

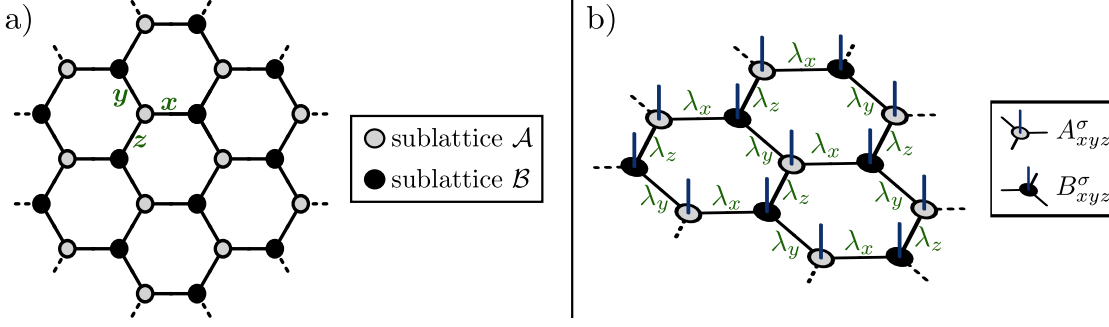


Figure 2.1.: a) The honeycomb lattice is bipartite. All sites in sublattice \mathcal{A} have only neighbors in sublattice \mathcal{B} and vice versa. We label the three bond directions by x , y , and z . b) A translationally invariant PEPS on the honeycomb lattice is defined by two tensors A_{xyz}^σ and B_{xyz}^σ because the unit cell contains two sites. We also define bond vectors λ_x , λ_y , and λ_z for each bond direction (cf. (2.12)).

where we assumed that the start wave function $|\Psi(\tau=0)\rangle$ is not orthogonal to the ground state $|\Psi_0\rangle$, since $|\Psi(\tau=0)\rangle$ was chosen randomly and the subspace of states orthogonal to $|\Psi_0\rangle$ has measure zero in the complete Hilbert space.

The bond Hamiltonians H_{ij} in (2.1) act on two neighboring sites only. They could therefore be applied to a PEPS as described in section 1.1.2. However, the imaginary time evolution operator $e^{-\tau H}$ in (2.2) contains terms of higher orders in H_{ij} , e.g. for the Heisenberg model,

$$e^{-\tau H} = 1 - \tau J \sum_{\langle i,j \rangle} \mathbf{S}_i \cdot \mathbf{S}_j + \tau^2 J^2 \sum_{\langle i,j \rangle} \sum_{\langle i',j' \rangle} (\mathbf{S}_i \cdot \mathbf{S}_j)(\mathbf{S}_{i'} \cdot \mathbf{S}_{j'}) + O(\tau^3), \quad (2.4)$$

where the term proportional to τ^2 is a sum of long-range operators, for which we do not have a method to apply them to a PEPS.

To evaluate (2.2) numerically, we split the imaginary time τ into n steps such that the step size $\Delta\tau := \tau/n$ is much smaller than the inverse of the typical energy scale. We rewrite (2.2) as

$$|\Psi(\tau)\rangle = (e^{-\Delta\tau H})^n |\Psi(\tau=0)\rangle. \quad (2.5)$$

We group the terms in the Hamiltonian (2.1) into three operators,

$$H = H_x + H_y + H_z, \quad (2.6)$$

where H_α is the sum of all bond Hamiltonians that act on bonds in the α direction (figure 2.1a),

$$H_\alpha := \sum_{i \in \mathcal{A}} H_{i, i+\hat{\alpha}} \quad \text{for } \alpha = x, y, z. \quad (2.7)$$

For small $\Delta\tau$, we can approximate the imaginary time step operator that appears on the right hand side of (2.5)

$$e^{-\Delta\tau H} = e^{-\Delta\tau(H_x+H_y+H_z)} \approx e^{-\Delta\tau H_x} e^{-\Delta\tau H_y} e^{-\Delta\tau H_z} + O(\Delta\tau^2). \quad (2.8)$$

2.1. Finding the ground state of a model Hamiltonian

Equation (2.8) is known as Trotter-Suzuki decomposition of first order. It is an approximate expression because the operators H_α in general do not commute with each other. We refer to the loss of precision due to the approximation (2.8) as Trotter error, to distinguish it from truncation errors which we will discuss in the next section.

For fixed α , the bond Hamiltonians $H_{i,i+\hat{\alpha}}$ on the right hand side of (2.7) all act on disjoint subsets of the honeycomb lattice (cf. figure 2.3a). Therefore, all terms in the sum in (2.7) commute with each other and we can decompose each factor on the right hand side of (2.8) exactly

$$e^{-\Delta\tau H_\alpha} = \exp\left(-\Delta\tau \sum_{i \in \mathcal{A}} H_{i,i+\hat{\alpha}}\right) = \prod_{i \in \mathcal{A}} \exp(-\Delta\tau H_{i,i+\hat{\alpha}}) \quad \text{for } \alpha = x, y, z. \quad (2.9)$$

The operators $e^{-\Delta\tau H_{i,i+\hat{\alpha}}}$ on the right hand side of (2.9) act on the two neighboring sites i and $i + \hat{\alpha}$ only. They can therefore be applied to a PEPS as described in section 1.1.2.

Putting together (2.9), (2.8), and (2.5), we conclude

$$|\Psi(\tau)\rangle = \left(\prod_{\alpha=x,y,z} \prod_{i \in \mathcal{A}} e^{-\Delta\tau H_{i,i+\hat{\alpha}}} \right)^n |\Psi(\tau=0)\rangle + O(n \Delta\tau^2) \quad \text{with } \Delta\tau = \tau/n, \quad (2.10)$$

where we have assumed the worst-case propagation of errors. At fixed τ , the Trotter error $O(n \Delta\tau^2) = O(\tau \Delta\tau)$ can be reduced by using sufficiently small imaginary time steps $\Delta\tau$. In the limit $\tau \rightarrow \infty$, errors of successive steps do not accumulate since finding the ground state by imaginary time evolution is a projective method. Each successive imaginary time step suppresses non-ground state contributions to the wave function and therefore corrects errors introduced in previous steps. The overall Trotter error for ground-state calculations is therefore only of order $O(\Delta\tau^2)$. In our implementation, we further reduce the Trotter error by using second order Trotter-Suzuki decomposition. Equation (2.8) then reads

$$e^{-\Delta\tau(H_x+H_y+H_z)} \approx e^{\gamma H_x} e^{\gamma H_y} e^{\gamma H_z} e^{\gamma^* H_x} e^{\gamma^* H_y} e^{\gamma^* H_z} + O(\Delta\tau^3), \quad (2.11)$$

with $\gamma = -\frac{1+i}{2}\Delta\tau$.

2.1.2. Implementation with PEPS

We describe an algorithm that approximates the ground state $|\Psi_0\rangle$ of the Hamiltonian (2.1) by evaluating (2.10) numerically for large τ . We use PEPS to efficiently encode the ground state wave function.

Since the Hamiltonian reflects the translational symmetry of the lattice, there exists a translationally invariant ground state. We use the ansatz of a translationally invariant PEPS for the starting wave function $|\Psi(\tau=0)\rangle$ in (2.10). We point out that, even if the ground state spontaneously breaks the translational symmetry of the Hamiltonian there always exists a translationally invariant state among the degenerate set of ground states. This translationally invariant ground state can formally be written as $|\Psi_0^{\text{TI}}\rangle = \sum_{i \in \mathcal{A}} T_i |\Psi_0\rangle$ where T_i is the operator that translates the system by the position vector of site i and $|\Psi_0\rangle$ is any ground state. This shows that our algorithm is formally valid

2. The basic algorithms

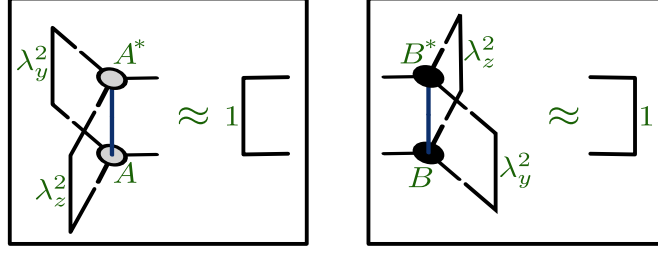


Figure 2.2.: The imaginary time evolution algorithm produces translationally invariant PEPS whose tensors are approximately canonical in every bond direction. We illustrate the canonical form (2.13) for x -bonds.

even in case of spontaneous breaking of translational symmetry. However, the accuracy of any algorithm based on translationally invariant PEPS decreases significantly in the case of spontaneous breaking of translational symmetry since encoding the symmetrized state $|\Psi_0^{\text{TI}}\rangle$ as a PEPS requires high bond dimensions.

Motivated by the normalization relations (1.21) and (1.22) for MPS, we use the following ansatz for the random start wave function.

$$|\Psi(\tau = 0)\rangle := \text{Tr} \left\{ \prod_{i \in \mathcal{A}, j \in \mathcal{B}} \lambda_{x_i} \lambda_{y_i} \lambda_{z_i} A_{x_i y_i z_i}^{\sigma_i} B_{x_j y_j z_j}^{\sigma_j} |\sigma_i \sigma_j\rangle \right\}. \quad (2.12)$$

Here, $\alpha = x, y, z$ labels the three bond directions of the honeycomb lattice, $A_{x_i y_i z_i}^{\sigma_i}$ and $B_{x_j y_j z_j}^{\sigma_j}$ are tensors defined on the two sublattices \mathcal{A} and \mathcal{B} , and λ_{α_i} is a non-negative bond vector assigned to the bond emitted from site i in direction α (figure 2.1b). The physical indices σ run from 1 to the dimension d of the local physical Hilbert space while the bond indices x, y, z run from 1 to an ansatz bond dimension D . The trace is to contract the product of tensors and bond vectors according to the lattice topology and also to sum over all configurations of the local physical basis states $|\sigma_i\rangle$.

It should be mentioned that, in contrast to the one-dimensional case with MPS, strict normalization relations like (1.21) and (1.22) do not exist for two-dimensional PEPS. However, our calculations showed that the algorithm we describe below leads to a PEPS that is approximately canonical in the sense that

$$\sum_{\sigma yz} (A_{xyz}^{\sigma})^* A_{x'yz}^{\sigma} \lambda_y^2 \lambda_z^2 \approx \delta_{x,x'} \quad (2.13)$$

and analogous relations for B_{xyz}^{σ} and for permutations of x, y, z (figure 2.2). The deviations from an exact canonical form vanish as $\Delta\tau \rightarrow 0$. This allows us to give an interpretation to the bond vectors λ_{α_i} . They are a measure of the relative weight of different bond configurations on the mean-field level. As the bare tensors A and B are approximately canonical for each bond direction, the additional factors of λ_{α_i} in (2.12) attribute a weight to each bond configuration. Configurations with high bond-weights contribute the most to the state.

To perform the imaginary time evolution (2.10), we have to iteratively apply projection operators of the form $\prod_{i \in \mathcal{A}} e^{-\Delta\tau H_{i,i+\hat{\alpha}}}$ to the wave function. For a fixed bond direction

2.1. Finding the ground state of a model Hamiltonian

α , we group tensors defined on sites i and $i + \hat{\alpha}$ into pairs and apply $e^{-\Delta\tau H_{i,i+\hat{\alpha}}}$ to each of the pairs. Since both the original wave function and the operators for the imaginary time step are translationally invariant, the result will again be translationally invariant. Finally, we approximate the new state by a translationally invariant PEPS.

We demonstrate the calculation for horizontal bonds (cf. figure 2.3b).

$$\begin{aligned}
& \prod_{i \in \mathcal{A}} e^{-\Delta\tau H_{i,i+\hat{x}}} |\Psi(\tau = 0)\rangle = \\
& = \text{Tr} \left\{ \prod_{i \in \mathcal{A}, j=i+\hat{x}} e^{-\Delta\tau H_{ij}} \lambda_{x_i} \lambda_{y_i} \lambda_{z_i} A_{x_i y_i z_i}^{\sigma_i} B_{x_j y_j z_j}^{\sigma_j} |\sigma_i \sigma_j\rangle \right\} \\
& = \text{Tr} \left\{ \prod_{i \in \mathcal{A}, j=i+\hat{x}} \sum_{\sigma'_i, \sigma'_j} \underbrace{\langle \sigma_i \sigma_j | e^{-\Delta\tau H_{ij}} | \sigma'_i \sigma'_j \rangle}_{=: P^{\sigma_i \sigma_j}_{\sigma'_i \sigma'_j}} \lambda_{y_i} \lambda_{z_i} A_{x_i y_i z_i}^{\sigma'_i} \lambda_{x_i} B_{x_j y_j z_j}^{\sigma'_j} |\sigma_i \sigma_j\rangle \right\} \\
& = \text{Tr} \left\{ \prod_{i \in \mathcal{A}, j=i+\hat{x}} \sum_{\sigma'_i, \sigma'_j} P^{\sigma_i \sigma_j}_{\sigma'_i \sigma'_j} \lambda_{y_i}^{1/2} \lambda_{z_i}^{1/2} A_{x_i y_i z_i}^{\sigma'_i} \lambda_{x_i} B_{x_j y_j z_j}^{\sigma'_j} \lambda_{y_j}^{1/2} \lambda_{z_j}^{1/2} |\sigma_i \sigma_j\rangle \right\}, \quad (2.14)
\end{aligned}$$

where, in the last step, we have rearranged the bond vectors by associating a factor of $\lambda_{\alpha_i}^{1/2}$ to each site connected by the corresponding bond. The local projection operator P mixes tensors linked by a horizontal bond. We explicitly apply P to the two tensors defined on sites i and $j = i + \hat{x}$,

$$\begin{aligned}
& \sum_{\sigma'_i, \sigma'_j} P^{\sigma_i \sigma_j}_{\sigma'_i \sigma'_j} \lambda_{y_i}^{1/2} \lambda_{z_i}^{1/2} A_{x_i y_i z_i}^{\sigma'_i} \lambda_{x_i} B_{x_j y_j z_j}^{\sigma'_j} \lambda_{y_j}^{1/2} \lambda_{z_j}^{1/2} = \\
& = \lambda_{y_i}^{-1/2} \lambda_{z_i}^{-1/2} \left(\sum_{\sigma'_i, \sigma'_j} P^{\sigma_i \sigma_j}_{\sigma'_i \sigma'_j} \lambda_{y_i} \lambda_{z_i} A_{x_i y_i z_i}^{\sigma'_i} \lambda_{x_i} B_{x_j y_j z_j}^{\sigma'_j} \lambda_{y_j} \lambda_{z_j} \right) \lambda_{y_j}^{-1/2} \lambda_{z_j}^{-1/2} \\
& =: \lambda_{y_i}^{-1/2} \lambda_{z_i}^{-1/2} S_{y_i z_i y_j z_j}^{\sigma_i \sigma_j} \lambda_{y_j}^{-1/2} \lambda_{z_j}^{-1/2}, \quad (2.15)
\end{aligned}$$

where we have introduced the two-site tensor S that includes full factors of λ_{α_i} rather than only the square root. These additional bond vectors weigh the possible bond configurations according to the amount each configuration contributes to the total wave function, as estimated on a mean-field level.

To bring the wave function back to the form of a PEPS, we perform a singular value decomposition of the tensor S ,

$$S_{y_i z_i y_j z_j}^{\sigma_i \sigma_j} \equiv S_{(\sigma_i y_i z_i), (\sigma_j y_j z_j)} = \sum_{\tilde{x}_i=1}^{dD^2} U_{(\sigma_i y_i z_i), \tilde{x}_i} \tilde{\Lambda}_{\tilde{x}_i} (V^\dagger)_{\tilde{x}_i, (\sigma_j y_j z_j)}. \quad (2.16)$$

Since the combined bond index $(\sigma y z)$ in (2.16) runs from 1 to dD^2 , so does the index \tilde{x}_i for the singular values. To set an upper bound to the memory requirements of the algorithm, we truncate \tilde{x}_i by keeping only the highest D singular values. We sort the $\tilde{\Lambda}_{\tilde{x}_i}$ in descending order and define

$$\tilde{\lambda}_{x_i} := \tilde{\Lambda}_{x_i} \quad \text{for } x_i = 1, \dots, D, \quad (2.17)$$

2. The basic algorithms

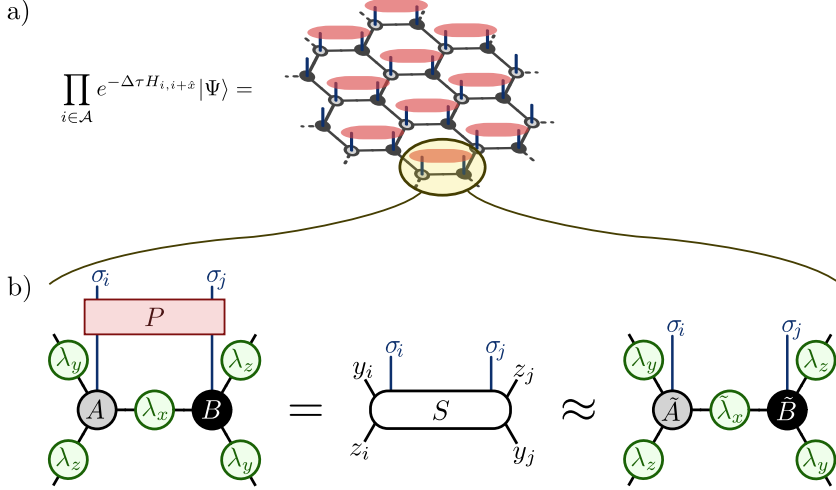


Figure 2.3.: Illustration of an imaginary time step on horizontal bonds. a) The wave function is translationally invariant, so we only focus on one unit cell (2.14). b) We apply the local evolution operator P to two sites and multiply full bond vectors to all open bonds (2.15). Then we split up the result S into two tensors, truncate the new bond vector $\tilde{\lambda}_x$ and multiply inverse bond vectors on the y and z bonds to restore the original form of the PEPS (2.17)–(2.19). We continue with the same operation on y and then on z bonds.

where we have introduced the truncated updated bond index x_i . We further define the truncated updated tensors

$$\tilde{A}_{x_i y_i z_i}^{\sigma_i} := \lambda_{y_i}^{-1} \lambda_{z_i}^{-1} U_{(y_i z_i), x_i}^{\sigma_i}; \quad \tilde{B}_{x_j y_j z_j}^{\sigma_j} := (V^\dagger)_{x_i, (y_j z_j)}^{\sigma_j} \lambda_{y_j}^{-1} \lambda_{z_j}^{-1}. \quad (2.18)$$

Putting together equations (2.14)–(2.18), we approximate the updated wave function by a PEPS,

$$\prod_{i \in \mathcal{A}} e^{-\Delta\tau H_{i,i+\hat{x}}} |\Psi(\tau=0)\rangle \approx \text{Tr} \left\{ \prod_{i \in \mathcal{A}, j \in \mathcal{B}} \tilde{\lambda}_{x_i} \lambda_{y_i} \lambda_{z_i} \tilde{A}_{x_i y_i z_i}^{\sigma_i} \tilde{B}_{x_j y_j z_j}^{\sigma_j} |\sigma_i \sigma_j\rangle \right\}. \quad (2.19)$$

The PEPS representation (2.19) of the imaginary time evolved state is an approximate expression due to the truncation (2.17). The decomposition $U\lambda V^\dagger$ is the best approximation (w.r.t. operator norm) of the two-site tensor S by a product of tensors with bond dimensions D . We included additional factors of $\lambda_{\alpha_i}^{1/2}$ in the definition of S (2.15). Therefore, by choosing the best approximation of S , we approximated the updated wave function taking into account the effects of the environment of the two sites i and j on the mean-field level.

Equation (2.19) concludes the calculation of one imaginary time step on horizontal bonds. The calculation for bonds in the y and z direction is analogous and updates (in addition to A and B) the bond vectors λ_{y_i} and λ_{z_i} , respectively. To find an approximate ground state of the Hamiltonian (2.1), we evaluate (2.10) for large τ , i.e. we iteratively

perform imaginary time steps for different bond directions, thus updating the current wave function from step to step until it converges.

It is now easy to see why the approximate normalization relation (2.13) holds. For each imaginary time step, we apply the operators $e^{-\Delta\tau H_x}$, $e^{-\Delta\tau H_y}$, and $e^{-\Delta\tau H_z}$ to the wave function, one after the other. Let us first apply $e^{-\Delta\tau H_x}$ to a PEPS. The singular value decomposition (2.16) is defined such that the updated tensors \tilde{A} and \tilde{B} (2.18) satisfy the normalization relation (2.13) *exactly*. Next, we apply $e^{-\Delta\tau H_y}$ to the updated PEPS. This defines new tensors which satisfy the corresponding normalization relation for y -bonds *exactly*, but it destroys the normalization relation for x -bonds. However, since $\Delta\tau$ is small, $e^{-\Delta\tau H_y}$ is approximately the identity operator and therefore destroys the normalization relation for x -bonds only weakly. The same holds for the application of $e^{-\Delta\tau H_z}$. At any moment during the imaginary time evolution, the normalization relation is satisfied exactly for one bond direction and approximately for the other two.

The error due to the truncation (2.17) adds to the Trotter error discussed in the previous section. The truncation error is controlled by the bond dimension D rather than $\Delta\tau$ as in the case of Trotter error. As argued for the Trotter error, truncation errors of successive imaginary time steps do not accumulate in the limit $\tau \rightarrow \infty$ either.

2.1.3. Summary of the parameters

In the above description of the imaginary time evolution algorithm, we have introduced the following parameters.

- The randomly chosen start wave function $|\Psi(\tau=0)\rangle$. More explicitly, the randomly chosen entries of the tensors A and B in (2.12).
- The bond dimension D , which also serves as a cut-off bond dimension for the truncation (2.17).
- The imaginary time τ of the final wave function. Since we are interested in the limit $\tau \rightarrow \infty$, we calculate $|\Psi(\tau)\rangle$ up to a value of τ much larger than the inverse energy scale ($\tau \gg 1/J$ in the case of the Heisenberg model (2.1)).
- The size $\Delta\tau$ of imaginary time steps. To minimize the Trotter error, we set $\Delta\tau$ to a value much smaller than the inverse energy scale ($\Delta\tau \ll 1/J$).

We investigated how the accuracy of the algorithm depends on the choice of the above parameters and present results in section 4.2.2.

To calculate $|\Psi(\tau)\rangle$, one has to perform $n = \tau/\Delta\tau$ imaginary time steps. The computationally most expensive part of each imaginary time step is the singular value decomposition (2.16) of the matrix $S_{(\sigma_i y_i z_i), (\sigma_j y_j z_j)} \in \mathbb{C}^{dD^2 \times dD^2}$. The CPU time required for an SVD scales with the third power of the matrix dimension, i.e. $O(d^3 D^6)$. Therefore, the overall CPU time required for the imaginary time evolution (ITE) is of order

$$T_{\text{ITE}} \in O(d^3 D^6 \cdot \tau / \Delta\tau). \quad (2.20)$$

The tensor S defined in (2.15) is the most memory consuming object that appears in the algorithm. The memory requirements therefore scale as

$$M_{\text{ITE}} \in O(d^2 D^4). \quad (2.21)$$

2. The basic algorithms

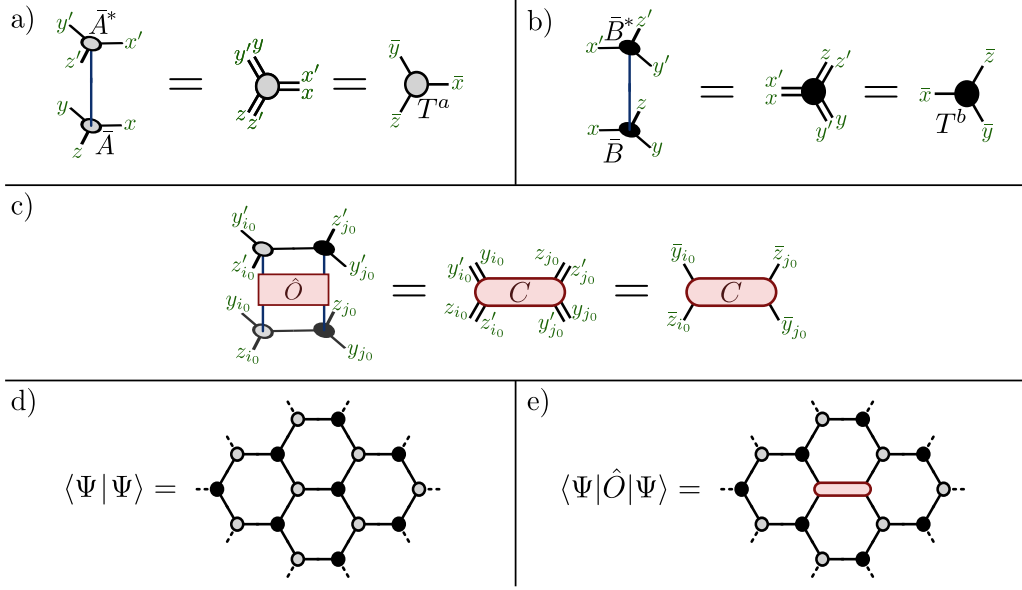


Figure 2.4.: a)-c) Definition of the tensors T^a , T^b , and C ((2.26) and (2.28)). d)-e) Illustration of the tensor contraction scheme to express the norm of the wave function and the expectation value of a local operator \hat{O} as a product of tensors ((2.25) and (2.27)).

Both CPU time and memory requirements can be reduced significantly if the Hamiltonian is invariant under a local symmetry transformation of the Hilbert space (see section 3.1).

2.2. Calculating physical observables

The imaginary time evolution algorithm described in section 2.1 generates a PEPS $|\Psi\rangle := |\Psi(\tau \rightarrow \infty)\rangle$ that approximates the ground state of the Hamiltonian (2.1). In this section, we describe an algorithm to approximately calculate expectation values in the state $|\Psi\rangle$. We start from the notation in (2.12) to describe $|\Psi\rangle$ and define new tensors \bar{A} and \bar{B} that absorb the bond vectors λ ,

$$\bar{A}_{x_i y_i z_i}^{\sigma_i} := \sqrt{\lambda_{x_i} \lambda_{y_i} \lambda_{z_i}} A_{x_i y_i z_i}^{\sigma_i}; \quad \bar{B}_{x_j y_j z_j}^{\sigma_j} := \sqrt{\lambda_{x_j} \lambda_{y_j} \lambda_{z_j}} B_{x_j y_j z_j}^{\sigma_j}, \quad (2.22)$$

so that the PEPS description of $|\Psi\rangle$ simplifies to

$$|\Psi\rangle = \text{Tr} \left\{ \prod_{i \in \mathcal{A}, j \in \mathcal{B}} \bar{A}_{x_i y_i z_i}^{\sigma_i} \bar{B}_{x_j y_j z_j}^{\sigma_j} |\sigma_i \sigma_j\rangle \right\}. \quad (2.23)$$

We restrict our discussion to local (bond-)operators $\hat{O}_{i_0 j_0}$, i.e. operators that act on two neighboring sites $i_0 \in \mathcal{A}$ and $j_0 \in \mathcal{B}$ only. As an example, consider the bond Hamiltonian H_{ij} in (2.1). For concreteness, we let the neighbors i_0 and j_0 be connected by a horizontal bond.

The expectation value of $\hat{O}_{i_0 j_0}$ is defined by

$$\langle \hat{O}_{i_0 j_0} \rangle := \frac{\langle \Psi | \hat{O}_{i_0 j_0} | \Psi \rangle}{\langle \Psi | \Psi \rangle}. \quad (2.24)$$

We write the denominator as a product of tensors,

$$\langle \Psi | \Psi \rangle = \text{Tr} \prod_{\substack{i \in \mathcal{A} \\ j \in \mathcal{B}}} \sum_{\sigma_i \sigma_j} (\bar{A}_{x'_i y'_i z'_i}^{\sigma_i})^* (\bar{B}_{x'_j y'_j z'_j}^{\sigma_j})^* \bar{A}_{x_i y_i z_i}^{\sigma_i} \bar{B}_{x_j y_j z_j}^{\sigma_j} = \text{Tr} \prod_{\substack{i \in \mathcal{A} \\ j \in \mathcal{B}}} T_{\bar{x}_i \bar{y}_i \bar{z}_i}^a T_{\bar{x}_j \bar{y}_j \bar{z}_j}^b \quad (2.25)$$

where the trace is to contract all bond indices according to figure 2.4d and the tensors T^a and T^b are defined as (cf. figure 2.4a,b)

$$T_{\bar{x}\bar{y}\bar{z}}^a \equiv T_{xx', yy', zz'}^a := \sum_{\sigma} (\bar{A}_{x'y'z'}^{\sigma})^* \bar{A}_{xyz}^{\sigma} \quad (2.26)$$

and T^b accordingly. In equation (2.26), we have introduced the combined bond index $\bar{\alpha} := (\alpha\alpha') = 1, \dots, D^2$ for $\alpha = x, y, z$. We emphasize that the tensors T^a and T^b do not depend on the unit cell, due to translational invariance of $|\Psi\rangle$.

For the enumerator in (2.24), the tensor product formulation is more elaborate (cf. figure 2.4e),

$$\langle \Psi | \hat{O}_{i_0 j_0} | \Psi \rangle = \text{Tr} \left\{ C_{\bar{y}_{i_0} \bar{z}_{i_0} \bar{y}_{j_0} \bar{z}_{j_0}} \prod_{\substack{i \in \mathcal{A} \setminus \{i_0\} \\ j \in \mathcal{B} \setminus \{j_0\}}} T_{\bar{x}_i \bar{y}_i \bar{z}_i}^a T_{\bar{x}_j \bar{y}_j \bar{z}_j}^b \right\} \quad (2.27)$$

where the product runs over all sites i, j except i_0 and j_0 and the central tensor C is defined as (cf. figure 2.4c)

$$C_{\bar{y}_{i_0} \bar{z}_{i_0} \bar{y}_{j_0} \bar{z}_{j_0}} := \sum_{\substack{\sigma_{i_0} \sigma_{j_0} \\ \sigma'_{i_0} \sigma'_{j_0}}} \sum_{x_{i_0} x'_{i_0}} \langle \sigma'_{i_0} \sigma'_{j_0} | \hat{O}_{i_0 j_0} | \sigma_{i_0} \sigma_{j_0} \rangle (\bar{A}_{x'_{i_0} y'_{i_0} z'_{i_0}}^{\sigma'_{i_0}})^* (\bar{B}_{x'_{j_0} y'_{j_0} z'_{j_0}}^{\sigma'_{j_0}})^* \bar{A}_{x_{i_0} y_{i_0} z_{i_0}}^{\sigma_{i_0}} \bar{B}_{x_{j_0} y_{j_0} z_{j_0}}^{\sigma_{j_0}}. \quad (2.28)$$

Note that, if $\hat{O}_{i_0 j_0}$ is the identity operator, we have $C = T^a T^b$ and (2.27) simplifies to (2.25). Thus, it suffices to develop a method to calculate tensor products of the form (2.27).

2.2.1. Tensor Renormalization Group (TRG)

As discussed in section 1.2, the exact evaluation of tensor products like (2.27) is an exponentially hard problem in more than one spatial dimension. However, an efficient method to approximate the right hand side of (2.27) exists under the name tensor renormalization group (TRG).

The TRG method was first proposed by Levin and Nave [1] as a method to calculate the partition function of two-dimensional classical statistical lattice models. Its application has then been extended to the calculation of local expectation values of quantum

2. The basic algorithms

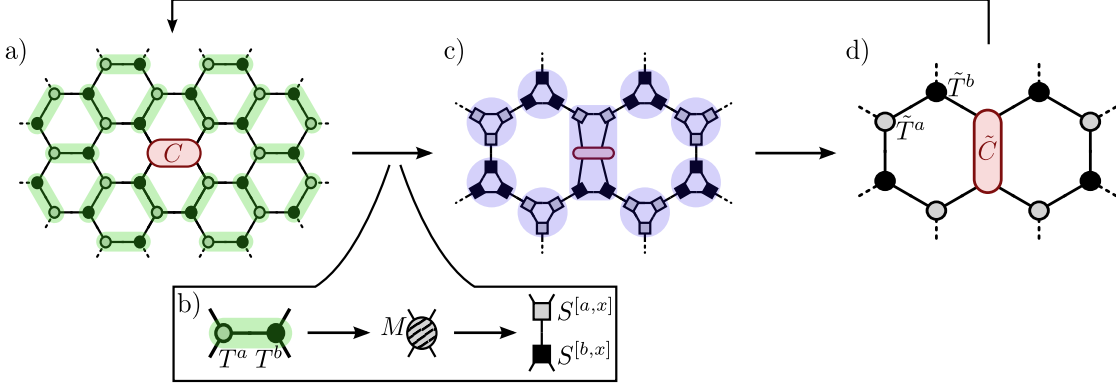


Figure 2.5.: Illustration of one TRG renormalization step. We group neighboring tensors into pairs (a), rewire each pair (b), and trace out short-range degrees of freedom (c) to arrive at a network (d) that is topologically equivalent to the original network but encodes three times as many original lattice sites per tensor.

mechanical lattice models in parallel efforts by Jiang et al. [36] and by Gu et al. [37]. To avoid confusion, we point out that the term "renormalization" in the context of TRG refers to a renormalization of the tensor network (2.27) rather than the renormalization of the Hamiltonian of a system.¹

The TRG method is an iterative algorithm. In each renormalization step, we trace out short-range effects, limit the remaining degrees of freedom to the most important ones, and reset the length scale. This procedure replaces the tensors in (2.27) by updated tensors \tilde{T}^a , \tilde{T}^b and \tilde{C} . At the same time, it transforms the graphical representation of the tensor network (figure 2.5a) into a topologically equivalent but scaled network (figure 2.5d). Each tensor in the new network encodes information from three tensors of the original network. Therefore, after N steps of this coarse-graining procedure, the updated central Tensor \tilde{C} effectively describes $2 \cdot 3^N$ original lattice sites and we can read off the expectation value in a system with $2 \cdot 3^N$ sites by imposing periodic boundary conditions onto \tilde{C} . Due to the exponential growth of the effective system size, we expect the result to converge quickly to the thermodynamic limit.

We illustrate the coarse-graining procedure in figure 2.5. We start with the tensor product on the right hand side of (2.27) where the tensors are to be contracted according to the network in figure 2.4e. In a first step, we group neighboring tensors T^a and T^b into pairs according to figure 2.5a. Three different kinds of pairs appear, according to the three different bond directions $\alpha = x, y, z$ of the honeycomb lattice. For each pair, we perform the operations illustrated in figure 2.5b. We demonstrate this step for horizontal bonds ($\alpha = x$). First, we multiply T^a and T^b to an order-four tensor,

$$M_{\bar{y}_i \bar{z}_i \bar{y}_j \bar{z}_j} := \sum_{\bar{x}} T_{\bar{x} \bar{y}_i \bar{z}_i}^a T_{\bar{x} \bar{y}_j \bar{z}_j}^b. \quad (2.29)$$

¹In the original application of TRG [1], i.e. the calculation the partition function of two-dimensional classical statistical systems, the renormalization indeed defines a flow of the Hamiltonian from the original to the coarse-grained lattice.

We regroup the indices of M and perform a singular value decomposition,

$$M_{\bar{y}_i \bar{z}_i \bar{y}_j \bar{z}_j} \equiv M_{(\bar{z}_j \bar{y}_i), (\bar{z}_i \bar{y}_j)} = \sum_{n_x=1}^{D^4} U_{(\bar{z}_j \bar{y}_i), n_x} \Lambda_{n_x} V_{n_x, (\bar{z}_i \bar{y}_j)}^\dagger, \quad (2.30)$$

where the index n_x for the singular values runs from 1 to D^4 since the original indices \bar{y} and \bar{z} each run from 1 to D^2 . To set an upper bound for the memory requirements, we introduce a cut-off bond dimension D_{cut} and keep only the largest D_{cut} singular values. We define

$$\left. \begin{aligned} S_{n_x \bar{z}_j \bar{y}_i}^{[a,x]} &:= U_{(\bar{z}_j \bar{y}_i), n_x} \sqrt{\Lambda_{n_x}} \\ S_{n_x \bar{z}_i \bar{y}_j}^{[b,x]} &:= V_{n_x, (\bar{z}_i \bar{y}_j)}^\dagger \sqrt{\Lambda_{n_x}} \end{aligned} \right\} \text{ for } n_x = 1, \dots, D_{\text{cut}} \quad . \quad (2.31)$$

The truncation (2.31) of the new bond index n_x ensures that the product $S^a S^b$ optimally approximates M for fixed D_{cut} w.r.t. operator norm.

The rewiring of pairs of T^a and T^b in the y - and z -direction in figure 2.5a is analogous to equations (2.29)–(2.31) and leads to new tensors $S^{[a/b,y]}$ and $S^{[a/b,z]}$. At this point, we can approximate the right hand side of (2.27) by a product of S -tensors subject to the contraction scheme in figure 2.5c. In the last step of the coarse graining procedure, we trace out the blue shaded regions and define the updated tensors,

$$\tilde{T}_{n_x n_y n_z}^a := \sum_{\bar{x} \bar{y} \bar{z}} S_{n_x \bar{z} \bar{y}}^{[a,x]} S_{n_y \bar{x} \bar{z}}^{[a,y]} S_{n_z \bar{y} \bar{x}}^{[a,z]}, \quad (2.32a)$$

$$\tilde{T}_{n_x n_y n_z}^b := \sum_{\bar{x} \bar{y} \bar{z}} S_{n_x \bar{z} \bar{y}}^{[b,x]} S_{n_y \bar{x} \bar{z}}^{[b,y]} S_{n_z \bar{y} \bar{x}}^{[b,z]}, \quad (2.32b)$$

$$\tilde{C}_{n_y n_z n'_y n'_z} := \sum_{\substack{\bar{y} \bar{z} \bar{y}' \bar{z}' \\ \bar{x}'' \bar{x}'''}} S_{n_y \bar{x}'' \bar{z}'}^{[a,y]} S_{n_z \bar{y} \bar{x}''}^{[a,z]} S_{n'_y \bar{x}''' \bar{z}}^{[b,y]} S_{n'_z \bar{y}' \bar{x}'''}^{[b,z]} C_{\bar{y} \bar{z} \bar{y}' \bar{z}'}. \quad (2.32c)$$

The definitions (2.32) conclude one coarse-graining step. The updated tensors have bonddimension D_{cut} on all bonds. We approximate the original tensor product (2.27) by a product of the updated tensors,

$$\langle \Psi | \hat{O}_{i_0 j_0} | \Psi \rangle \approx \text{Tr} \left\{ \tilde{C}_{n_y n_z n'_y n'_z} \prod_{\substack{i \in \mathcal{A} \setminus i_0 \\ j \in \mathcal{B} \setminus j_0}} \tilde{T}_{n_{x_i} n_{y_i} n_{z_i}}^a \tilde{T}_{n_{x_j} n_{y_j} n_{z_j}}^b \right\}, \quad (2.33)$$

subject to the updated contraction scheme illustrated in figure 2.5d, which is equivalent to the original network (figure 2.5a) up to rotation and scaling. Equation (2.33) therefore has the same form as the starting point (2.27) and we can continue to perform coarse-graining operations. In the definitions (2.32a) and (2.32b), three original tensors are multiplied together to form each one of the updated tensors \tilde{T}^a and \tilde{T}^b . Therefore, the network grows by a factor of three in each coarse-graining step. After N cycles, the updated central tensor \tilde{C} effectively describes $2 \cdot 3^N$ original lattice sites. We then approximate the expectation value by imposing, periodic boundary conditions,

$$\langle \hat{O}_{i_0 j_0} \rangle = \frac{\langle \Psi | \hat{O}_{i_0 j_0} | \Psi \rangle}{\langle \Psi | \Psi \rangle} \approx \frac{\sum_{n_y n_z} \tilde{C}_{n_y n_z n_y n_z}}{\sum_{n_x n_y n_z} \tilde{T}_{n_x n_y n_z}^a \tilde{T}_{n_x n_y n_z}^b}. \quad (2.34)$$

2. The basic algorithms

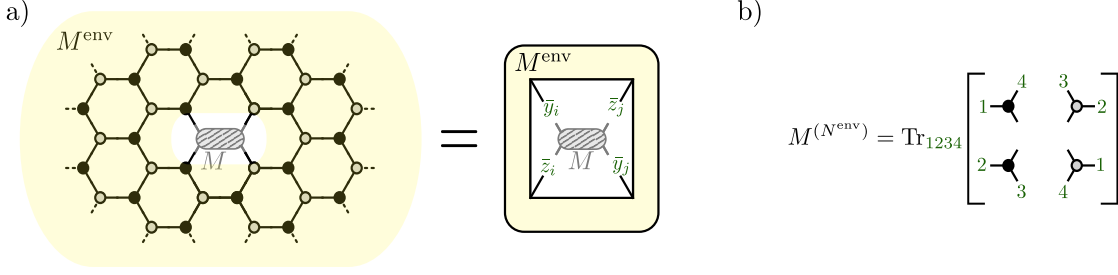


Figure 2.6.: a) The matrix M from figure 2.5b is embedded in an environment M^{env} . Rather than finding the optimal approximation of M , the SRG algorithm finds the optimal approximation of $\text{Tr}[MM^{\text{env}}]$. b) After transforming the environment lattice with N^{env} TRG steps, we start with an environment matrix calculated for four (renormalized) sites (cf. (2.42)).

In practice, we evaluate (2.34) after each step and continue the renormalization until the expectation values converge at some step N (usually $N \lesssim 10$ suffices, see section 4.2.1).

2.2.2. Second Tensor Renormalization Group (SRG)

The tensor renormalization group outlined in the previous section is an approximate method to calculate tensor products of the form (2.27) for large systems. Deviations from the exact result are due to the truncation (2.31), where we defined tensors S^a and S^b with bond dimensions D_{cut} such that their product $S^a S^b$ optimally approximates the matrix M from (2.30). In this approximation, we implicitly made the assumption that each entry of $M_{(\bar{z}_j \bar{y}_i), (\bar{z}_i \bar{y}_j)}$ (i.e. each combination of indices $\bar{z}_j, \bar{y}_i, \bar{z}_i, \bar{y}_j$) is equally important and we minimized the sum of the squares of the errors of all entries. We did not take into account the fact that M is embedded in an environment, which can be seen as a multiplicative weight associated to each combination of indices.

Xie et al. proposed an improved algorithm, called “Second Tensor Renormalization Group” (SRG) [2, 39], which differs from TRG in the definition of the truncated tensors S^a and S^b (2.31). Rather than optimizing S^a and S^b locally, the SRG globally optimizes the result of the complete product of tensors (2.27). To explain the algorithm, we rewrite the expectation value as

$$\langle \Psi | \hat{O}_{i_0 j_0} | \Psi \rangle = \text{Tr} \left\{ M_{(\bar{z}_j \bar{y}_i), (\bar{z}_i \bar{y}_j)} M_{(\bar{z}_i \bar{y}_j), (\bar{z}_j \bar{y}_i)}^{\text{env}} \right\}, \quad (2.35)$$

where M is defined in (2.30) and the matrix M^{env} is the result of tracing out the indices corresponding to connected bonds in the environment lattice shown in figure 2.6a.

Global optimization of S^a and S^b . We will describe how to calculate M^{env} at the end of this section. For now, suppose M^{env} is given. We regard the components of M^{env} in (2.35) as relative weights for the corresponding components of M and find tensors S^a and S^b with bond dimensions D_{cut} such that their product $S^a S^b$ optimally approximates M with respect to the weights M^{env} . To treat row and column indices symmetrically,

we first perform a singular value decomposition of M^{env} ,

$$M_{(\bar{z}_i \bar{y}_j), (\bar{z}_j \bar{y}_i)}^{\text{env}} = U_{(\bar{z}_i \bar{y}_j), s}^{\text{env}} \Lambda_s^{\text{env}} V_{s, (\bar{z}_j \bar{y}_i)}^{\text{env} \dagger}. \quad (2.36)$$

We rewrite (2.35) using a cyclic permutation of the matrices inside the trace,

$$\langle \Psi | \hat{O}_{i_0 j_0} | \Psi \rangle = \text{Tr} \left\{ \underbrace{\sqrt{\Lambda_s^{\text{env}}} V_{s, (\bar{z}_j \bar{y}_i)}^{\text{env} \dagger} M_{(\bar{z}_j \bar{y}_i), (\bar{z}_i \bar{y}_j)} U_{(\bar{z}_i \bar{y}_j), t}^{\text{env}} \sqrt{\Lambda_t^{\text{env}}}}_{=: \tilde{M}_{st}} \right\} = \text{Tr} \tilde{M}_{st}. \quad (2.37)$$

Thus, to minimize the error in the expectation value, one has to minimize the truncation error in \tilde{M} . To do this, we perform a singular value decomposition of \tilde{M} ,

$$\tilde{M}_{st} = \tilde{U}_{sn_x} \tilde{\Lambda}_{n_x} \tilde{V}_{n_x t}^\dagger. \quad (2.38)$$

Then we truncate the bond space by keeping only the D_{cut} largest singular values $\tilde{\Lambda}_{n_x}$ and define

$$\left. \begin{aligned} S_{n_x \bar{z}_j \bar{y}_i}^{[a,x]} &:= V_{(\bar{z}_j \bar{y}_i), s}^{\text{env}} (\Lambda_s^{\text{env}})^{-1/2} \tilde{U}_{sn_x} (\tilde{\Lambda}_{n_x})^{1/2} \\ S_{n_x \bar{z}_i \bar{y}_j}^{[b,x]} &:= (\tilde{\Lambda}_{n_x})^{1/2} \tilde{V}_{n_x t}^\dagger (\Lambda_t^{\text{env}})^{-1/2} U_{t, (\bar{z}_i \bar{y}_j)}^{\text{env} \dagger} \end{aligned} \right\} \text{ for } n_x = 1, \dots, D_{\text{cut}}, \quad (2.39)$$

such that

$$M_{(\bar{z}_j \bar{y}_i), (\bar{z}_i \bar{y}_j)} \approx \sum_{n_x=1}^{D_{\text{cut}}} S_{n_x \bar{z}_j \bar{y}_i}^{[a,x]} S_{n_x \bar{z}_i \bar{y}_j}^{[b,x]}. \quad (2.40)$$

To conclude the renormalization step, it remains to trace out the short-distance degrees of freedom as in (2.32).

Determination of the environment matrix. The key ingredient for the improved truncation step in the SRG algorithm is the environment matrix M^{env} . It is not necessary to know M^{env} to high accuracy since it cancels out, up to truncation errors, in (2.39)–(2.40). We use TRG to approximate M^{env} .

We begin with the configuration of the environment lattice illustrated in figure 2.7a, where we neglected the central tensor C from (2.27), i.e. we optimize the approximation for M being far away from the central tensor C . We coarse grain the environment lattice in a sequence of TRG iterations. In each iteration, we first obtain a rewired lattice (figure 2.7b) by the singular value decomposition (2.30)–(2.31). Multiplying out the blue shaded regions as in (2.32) leads to the tensor network illustrated in figure 2.7c. Note the two incomplete triangles with bonds connecting to M that cannot be multiplied out yet, leading to four surviving S tensors in the coarse-grained lattice. We refer to these tensors as $S^{[a,\alpha,n]}$ and $S^{[b,\alpha,n]}$, where the superscript n denotes the n th TRG iteration in the calculation of the environment matrix.

The remaining coarse grained environment lattice (without the extra S tensors, see figure 2.7d) is topologically equivalent to the original environment lattice and we can continue with the next TRG iteration. We denote the matrix encoded by the environment lattice at the beginning of the n th TRG iteration (figure 2.7a) by $M^{(n-1)}$. This means that $M^{\text{env}} = M^{(0)}$. Figure 2.7d translates into the recursion formula

$$M_{\bar{z}_i \bar{y}_j \bar{z}_j \bar{y}_i}^{(n-1)} = \sum_{\substack{\bar{z}'_i \bar{y}'_j \bar{z}'_j \bar{y}'_i \\ \bar{x}_1 \bar{x}_2}} M_{\bar{z}'_i \bar{y}'_j \bar{z}'_j \bar{y}'_i}^{(n)} S_{\bar{y}'_i \bar{x}_1 \bar{z}_j}^{[a,y,n]} S_{\bar{z}'_i \bar{y}_i \bar{x}_1}^{[a,z,n]} S_{\bar{y}'_j \bar{x}_2 \bar{z}_i}^{[b,y,n]} S_{\bar{z}'_j \bar{y}_j \bar{x}_2}^{[b,z,n]}. \quad (2.41)$$

2. The basic algorithms

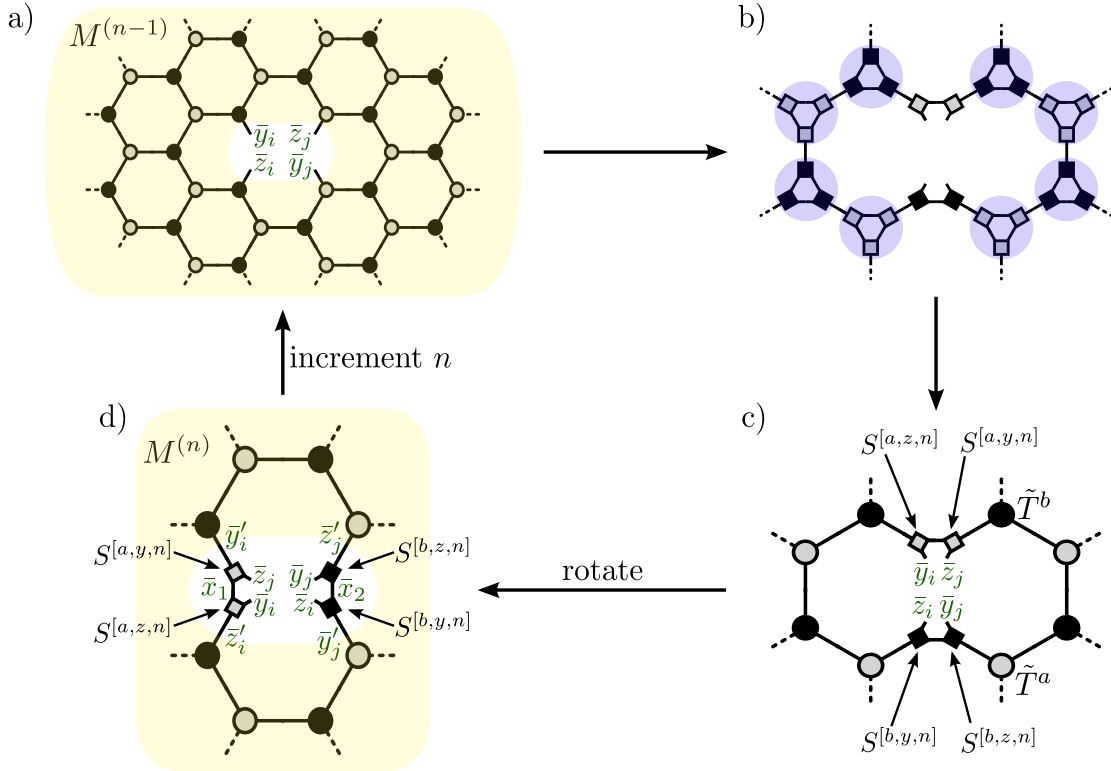


Figure 2.7.: Calculation of the environment matrix. a) We start with the configuration from figure 2.6a. b), c) Rewiring the lattice with a TRG step leaves four unmatched S tensors. d) We store the four unmatched S tensors in memory and continue with the next TRG step for the rest of the environment lattice.

In each TRG step, the effective number of sites encoded by each tensor in the environment lattice grows by a factor of three. We repeat the renormalization of the environment lattice for N^{env} iterations and then approximate $M^{(N^{\text{env}})}$ by only four tensors with periodic boundary conditions (cf. figure 2.6b)

$$M_{\bar{z}_i \bar{y}_j \bar{z}_j \bar{y}_i}^{(N^{\text{env}})} = \sum_{\substack{\bar{x}', \bar{y}', \\ \bar{x}'', \bar{z}''}} \tilde{T}_{\bar{x}' \bar{y}' \bar{z}_i}^a \tilde{T}_{\bar{x}' \bar{y}' \bar{z}_j}^b \tilde{T}_{\bar{x}'' \bar{y}_i \bar{z}''}^a \tilde{T}_{\bar{x}'' \bar{y}_j \bar{z}''}^b \quad . \quad (2.42)$$

Starting from $M^{(N^{\text{env}})}$, we find the desired environment matrix $M^{\text{env}} = M^{(0)}$ by repeated application of the recursion formula (2.41).

In summary, the calculation of the environment matrix consists of a forward iteration followed by a backward iteration. The forward iteration is a coarse graining of the environment lattice using TRG, while keeping in memory the S tensors of all iteration steps. In the backward iteration, one starts with the renormalized environment matrix $M^{(N^{\text{env}})}$ and iteratively multiplies to it the missing S tensors until one obtains $M^{(0)}$. We demonstrated the calculation of the environment matrix for pairs of sites that are connected by a horizontal bond. In practice, one has to calculate a different environment matrix for each of the three bond directions of the honeycomb lattice, unless the tensors obey the rotational symmetry of the lattice. The forward iteration is the same for all bond directions, so it has to be calculated only once. However, the backward iteration has to be calculated for each bond direction separately since the periodic boundary conditions (2.42) and the recursion formula (2.41) are different in each case.

2.2.3. Summary of the parameters

The accuracy and the costs of the SRG algorithm depend on the following parameters.

- The cut-off bond dimension D_{cut} . Usually, $D_{\text{cut}} \gtrsim D^2$ (with D the bond dimension of the wave function).
- The number N of renormalization steps (leading to a system size of $2 \cdot 3^N$ sites).
- The number N^{env} of renormalization steps used to calculate the environment lattice (leading to an environment that consists of $6 \cdot 3^{N^{\text{env}}} - 2$ sites).

We studied the influence of the above parameters on the accuracy of the algorithm. See section 4.2.1 for results.

The computationally most expensive steps of the SRG algorithm are the singular value decompositions (2.30), (2.36), and (2.38), as well as the implementations of equations (2.32c), (2.37), and (2.41), which all scale proportionally to D_{cut}^6 . The total CPU time is therefore of order

$$T_{\text{SRG}} \in O(N N^{\text{env}} D_{\text{cut}}^6). \quad (2.43)$$

The memory requirements scale as

$$M_{\text{SRG}} \in O(D_{\text{cut}}^4), \quad (2.44)$$

due to the order-four tensors M , M^{env} , and C . As for the imaginary time evolution algorithm, we can significantly reduce the costs of the SRG algorithm by exploiting symmetries of the wave function (see section 3.1).

2.3. Current state of the TRG method

Improvements and similar methods. Since its introduction by Levin and Nave [1], various authors have proposed improvements or extensions to the TRG method. We already mentioned SRG [2, 39] in section 2.2.2. This thesis is devoted to the study of systems that are defined on the honeycomb lattice. For a summary of similar renormalization procedures for the square, triangular, and kagome lattice, see [37]. The triangular and kagome lattices are of particular interest since one can study frustrated spin systems requiring only nearest-neighbor interactions [50]. A renormalization procedure for three dimensional tensor networks is presented in [51]. It was pointed out by Li et al. [52] that one can use the TRG algorithm in two dimensions to evaluate the Trotter-Suzuki decomposition of a one-dimensional quantum mechanical system.

In the form presented in section 2.2.1, the TRG method calculates local expectation values in spin systems. A generalization to fermionic systems that builds on Grassmann number valued tensors is presented in [3, 53]. We describe an equivalent algorithm in the language of symmetric tensors in section 3.2.

Chen et al. pointed out that one can use a modified variant of TRG to calculate Rényi entropies for small subsystems [38]. This method allows to test a wave function for the presence of topological order. We summarize the proposed algorithm in section 3.3.2 and apply it to ground states of the Hubbard model in section 5.2.2.

Finally, the TRG method was combined with ideas from quantum Monte Carlo simulations in [54]. To evaluate the expectation value of an operator \hat{O} in a PEPS $|\Psi\rangle$, one averages over the expectation values in a large number of randomly chosen product configurations $|\sigma_1, \sigma_2, \dots\rangle$, whose probability distribution is given by $|\langle\sigma_1, \sigma_2, \dots|\Psi\rangle|^2$. This probability is calculated by evaluating a tensor product similar to (2.25) but with bond dimension D rather than D^2 . The computing time therefore scales as $O(D^3)$ rather than $O(D^6)$.

Applications. The TRG algorithm was originally proposed as a method to evaluate partition functions of classical statistical systems. The partition function of any classical lattice system with local interactions can be expressed as a product of tensors of the form (2.25) where the bond dimension is the number d of local degrees of freedom [39]. Due to the small typical bond dimensions, TRG produces highly accurate results for the partition function (see also section 4.1.1). In addition, thermodynamic phases can be categorized by fixed points of the tensor-renormalization flow [55, 56]. Classical statistical systems are studied by TRG in [50, 57–59]

Concerning quantum mechanical systems, TRG was applied to the hard-core boson Hubbard model in [60, 61] and to spin models in [42, 52, 62–65]. Two key areas of interest for strongly correlated quantum mechanical systems are frustrated systems and systems with fermionic degrees of freedom, since those systems are in general problematic for quantum Monte Carlo approaches. Frustrated systems are studied with TRG in [42, 62]. For fermionic systems, we are only aware of benchmarking calculations with exactly solvable test systems [3, 53] and of one recent study of the $t - J$ model [66].

Open fields. Concerning progress in the methods, Zhao et al. [39] proposed to exploit local symmetries of the Hamiltonian in order to reduce computational costs. This has

very recently been announced as an outlook to future work by Gu et al. [66] but we are not aware of an existing implementation that exploits symmetries, apart from \mathbb{Z}_2 symmetry which is implicit in the fermionic formulation of TRG [3]. We discuss local symmetries in section 3.1 and present results from simulations with one and two conserved quantum numbers in sections 5.1 and 5.2, respectively.

Concerning open physical applications, we believe that SRG can be a useful tool in the study of fermionic systems. We present results for the (fermion) Hubbard model in section 5.2.

3. Extending the algorithm

3.1. Exploiting local symmetries

In sections 2.1.3 and 2.2.3, we saw that the CPU times for the imaginary time evolution algorithm and for the SRG scale as $O(D^6)$ and $O(D_{\text{cut}}^6)$, respectively, with D the bond dimension of the PEPS and D_{cut} the cut-off bond dimension used in the renormalization. This high sensitivity to the relevant bond dimension makes calculations with $D_{\text{cut}} \gtrsim 150$ very hard in practice and therefore limits the precision of the results.

In this section, we show that one can significantly reduce the computational costs if the Hamiltonian is invariant under the action of a local symmetry group. For simplicity, we restrict the discussion to abelian symmetries. As an example system, consider the antiferromagnetic spin- $\frac{1}{2}$ Heisenberg model on the honeycomb lattice defined in (2.1). The Hamiltonian conserves the z component of the total spin, $[H, S_{\text{tot}}^z] = 0$, with $S_{\text{tot}}^z = \sum_i S_i^z$ and one can show that the ground state lies in the subspace $S_{\text{tot}}^z = 0$ [67, 68]. The basic idea of the formalism described below is to limit the search for the ground state to this subspace. To do so, we develop a method to impose symmetry constraints (like $S_{\text{tot}}^z = 0$) onto a PEPS. We call the resulting structure a PEPS composed of *symmetric tensors*.

Using symmetric tensors, one can implement the algorithms described in chapter 2 as follows: One starts the imaginary time evolution with an ansatz wave function $|\Psi(\tau = 0)\rangle$ that is a PEPS composed of symmetric tensors and therefore fulfills $S_{\text{tot}}^z = 0$ exactly. This property carries over to the imaginary time evolved wave function $|\Psi(\tau)\rangle = e^{-\tau H}|\Psi(\tau = 0)\rangle$ since the Hamiltonian commutes with S_{tot}^z . As usual, one stops the imaginary time evolution at some large τ and expresses the expectation value of some local operator in the form of a product of tensors T^a and T^b as in (2.27). The symmetry constraints of $|\Psi(\tau)\rangle$ carry over to T^a and T^b , which will also be symmetric tensors. Finally, one uses SRG to evaluate the product of symmetric tensors.

Symmetric tensors give us the usual benefits of a partition of the Hilbert space into symmetry sectors. These are: Firstly, symmetric tensors have a block diagonal structure if expressed in an appropriate basis. An implementation can make use of this structure to reduce both memory requirements and computation time. Secondly, a PEPS composed of symmetric tensors fulfills the imposed symmetry constraint *exactly*. Therefore, we exclude contributions to the ground state from unwanted symmetry sectors right from the beginning, which is expected to increase the accuracy of the results. Finally, the symmetry constraints have a physical meaning and using symmetric tensors allows us to investigate different symmetry sectors of the Hilbert space individually. E.g., we can set the filling factor in Hubbard-like models to a fixed value.

3. Extending the algorithm

3.1.1. From symmetric states to symmetric tensors

We introduce a formalism of *symmetric tensors* that transform in a well-defined way under the action of a local symmetry group G . Any PEPS composed of symmetric tensors is exactly symmetric under the action of G . Conversely, we state that a translationally invariant PEPS that is symmetric under G can be rewritten in terms of symmetric tensors if the PEPS is *injective*. We discuss the condition of injectivity at the end of this section. The purpose of this section is to explain the assumptions that are implicitly imposed by algorithms that operate on symmetric PEPS. For the proofs of the theorems in this section, we refer to [69, 70].

Consider a system defined on the honeycomb lattice $\mathcal{L} = \mathcal{A} \cup \mathcal{B}$ with sublattices \mathcal{A} and \mathcal{B} and local basis $\{|\sigma_i\rangle\}_{\sigma=1,\dots,d}$ on site i . A translationally invariant PEPS with bond dimension D on this system is defined by tensors $A, B \in \mathbb{C}^{d \times D \times D \times D}$ and reads

$$|\Psi\rangle = \text{Tr} \left\{ \prod_{i \in \mathcal{A}, j \in \mathcal{B}} A_{x_i y_i z_i}^{\sigma_i} B_{x_j y_j z_j}^{\sigma_j} |\sigma_i \sigma_j\rangle \right\} \quad (3.1)$$

where the trace is to contract the tensors according to the lattice topology and also to sum over all σ_i and σ_j .

We consider the case in which $|\Psi\rangle$ is symmetric under the action of a local symmetry transformation. In this thesis, we refer to a *local* symmetry as a symmetry that acts on each site of the lattice independently, as opposed to, e.g., translational or rotational symmetry.

Definition 3.1 (Local symmetry). Let $|\Psi\rangle$ be a state defined on a lattice of N sites with local Hilbert space \mathcal{H}_{loc} on each site and let u be a unitary representation of a group G in \mathcal{H}_{loc} with the notation that $u_g \in L(\mathcal{H}_{\text{loc}})$ for all $g \in G$. Then, $|\Psi\rangle$ is said to be *symmetric under the local symmetry G w.r.t. the representation u* if it is symmetric under the simultaneous action of u_g on all sites, i.e.

$$u_g^{\otimes N} |\Psi\rangle = e^{i\theta_g} |\Psi\rangle \quad \forall g \in G \quad \text{with } \theta_g \in \mathbb{R}. \quad (3.2)$$

The state $|\Psi\rangle$ is called *invariant* under G if $e^{i\theta_g} = 1$ for all g and *covariant* otherwise.

Trivially, the PEPS (3.1) is invariant under G if the tensors A and B are invariant under u_g acting on the physical bond, i.e., if $\sum_{\sigma'} \langle \sigma' | u_g | \sigma \rangle A_{xyz}^{\sigma} = A_{xyz}^{\sigma'}$ and $\sum_{\sigma'} \langle \sigma' | u_g | \sigma \rangle B_{xyz}^{\sigma} = B_{xyz}^{\sigma'}$. We can further loosen the restrictions on A and B by allowing invertible matrices X , Y , and Z to be inserted on the virtual bonds, as in figure 3.1a. Generalizing this idea leads to the definition of symmetric tensors.

Definition 3.2 (Symmetric tensor). Let $T : V^1 \times \dots \times V^n \rightarrow \mathbb{C}$ be a tensor of order n with finite dimensional complex bond spaces V^1, \dots, V^n . For all $i = 1, \dots, n$, let $U^i : G \rightarrow L(V^i)$ be a representation of a group G . Then, T is said to be *symmetric under G w.r.t. the representations U^i* if it is invariant under the simultaneous action of the group on all bond spaces, i.e.

$$T(U_g^1, \dots, U_g^n) = T \quad \forall g \in G. \quad (3.3)$$

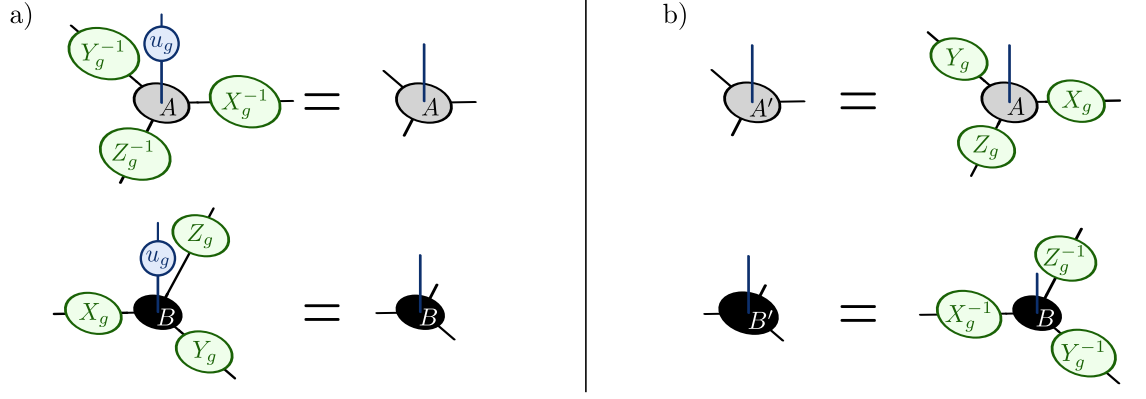


Figure 3.1.: a) A PEPS is invariant under a local symmetry G w.r.t. the representation u if its tensors satisfy the illustrated relation. We call A and B *symmetric tensors* (definition 3.2). b) If two injective PEPS describe the same state then their tensors satisfy the illustrated relation (theorem 3.4).

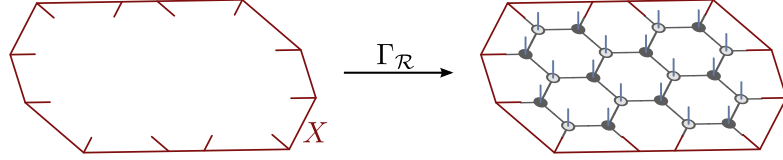


Figure 3.2.: Illustration of the map $\Gamma_{\mathcal{R}}$ that associates boundary conditions X with states defined on a region \mathcal{R} . A PEPS is called *injective in \mathcal{R}* if $\Gamma_{\mathcal{R}}$ is injective (definition 3.3).

Clearly, $|\Psi\rangle$ in (3.1) is invariant under G w.r.t. u if the order four tensors A and B are symmetric w.r.t. u acting on the physical bond of both A and B and some representations X, Y, Z acting on the virtual bonds of B while the inverse representations act on the virtual bonds of A (figure 3.1a).

Therefore, a PEPS composed of symmetric tensors encodes a state with a local symmetry. We will show in the next section that algorithms can reduce computational costs if they operate on symmetric tensors. Therefore, the reverse question is more important: Can a translationally invariant PEPS with a local symmetry be expressed in terms of symmetric tensors? In general, this is not the case. However, the decomposition into symmetric tensors is guaranteed if the state satisfies the condition of *injectivity*. A PEPS is called injective if different boundary conditions define different states. We follow [69] for the definition.

Definition 3.3 (Injective PEPS). For a translationally invariant PEPS defined on a lattice \mathcal{L} as in (3.1) and a simply connected region $\mathcal{R} \in \mathcal{L}$, let $\Gamma_{\mathcal{R}}$ be the map that associates boundary conditions with states defined on \mathcal{R} as in figure 3.2. Explicitly,

$$\Gamma_{\mathcal{R}}(X) := \text{Tr} \left\{ X \prod_{i \in \mathcal{A} \cap \mathcal{R}} A^{\sigma_i} |\sigma_i\rangle \prod_{j \in \mathcal{B} \cap \mathcal{R}} B^{\sigma_j} |\sigma_j\rangle \right\}, \quad (3.4)$$

3. Extending the algorithm

where $X \in \mathbb{C}^{D \times \dots \times D}$ is a combined bond vector of all bonds reaching out of \mathcal{R} and we have dropped the lower indices from A and B to simplify the notation. The PEPS is said to be *injective in \mathcal{R}* if $\Gamma_{\mathcal{R}}$ is injective. A PEPS is called *injective* if it is injective in some region \mathcal{R} .

It is easy to see that if a PEPS is injective in some region \mathcal{R} , then it is also injective in any larger region $\mathcal{R}' \supset \mathcal{R}$. Injectivity is an important property of PEPS as it can be shown that, for an injective PEPS $|\Psi\rangle$, a Hamiltonian H with only short-range interactions exists such that $|\Psi\rangle$ is the unique ground state of H [69].¹ The importance of injectivity in the context of local symmetries is due to a theorem by García et al. [72, theorem 12].

Theorem 3.4 (Uniqueness of tensors for injective PEPS). *Let $|\Psi\rangle$ and $|\Psi'\rangle$ be translationally invariant PEPS defined on the infinitely extended² honeycomb lattice. $|\Psi\rangle$ and $|\Psi'\rangle$ are defined as in (3.1) with tensors A, B , and A', B' , respectively. If both $|\Psi\rangle$ and $|\Psi'\rangle$ are injective, then the following are equivalent.*

(i) $|\Psi\rangle = |\Psi'\rangle$

(ii) *There exist invertible matrices $X, Y, Z \in \mathbb{C}^{D \times D}$ such that $A^\sigma(X, Y, Z) = A'^\sigma$ and $B^\sigma(X^{-1}, Y^{-1}, Z^{-1}) = B'^\sigma$ for all σ (see figure 3.1b). Each of the matrices X, Y , and Z is uniquely defined up to a respective overall factor.*

Theorem 3.4 links a global property (i) of the state to a local property (ii) of two different PEPS representations. The direction “(ii) \Rightarrow (i)” is trivial because it merely states that the state described by a PEPS is invariant under a change of basis in the bond spaces. This can be seen as a gauge degree of freedom of the PEPS representation of a state. The reverse “(i) \Rightarrow (ii)” is a non-trivial statement. It shows that the tensors that constitute an injective PEPS representation of a state are determined by the state up to a trivial *local* gauge degree of freedom.

Let us now consider a PEPS that is invariant under a local symmetry. Since the operator $u_g^{\otimes N}$ of a local symmetry transformation (3.2) acts on each site of the lattice individually, it maps PEPS to PEPS. For an (arbitrary) translationally invariant PEPS $|\Psi\rangle$ defined by tensors A^σ and B^σ , the state $u_g^{\otimes N}|\Psi\rangle$ can be written as a PEPS defined by the tensors $A'^{\sigma'} := \sum_\sigma \langle \sigma' | u_g | \sigma \rangle A^\sigma$ and $B'^{\sigma'} := \sum_\sigma \langle \sigma' | u_g | \sigma \rangle B^\sigma$. If $|\Psi\rangle$ is invariant under G then both PEPS describe the same state. However, due to the gauge degree of freedom, the tensors A' and B' will in general not be equal to A and B , respectively. In the case of injective PEPS, the relationship between A (B) and A' (B') is given by theorem 3.4. Additionally, the following theorem has been proven in [72, theorem 13] for the square lattice. We present its formulation for the honeycomb lattice, which can be proved by the same arguments.

¹Injectivity is equivalent to the uniqueness of the ground state of the parent Hamiltonian in the 1D case [46, 71]. In higher dimensions, injectivity is only a sufficient condition and uniqueness is possible in the absence of injectivity [69].

²More precisely, it suffices if $|\Psi\rangle$ and $|\Psi'\rangle$ are defined on a sufficiently large region in the honeycomb lattice such that both PEPS are injective in a region that is five times smaller than the domain of the PEPS.

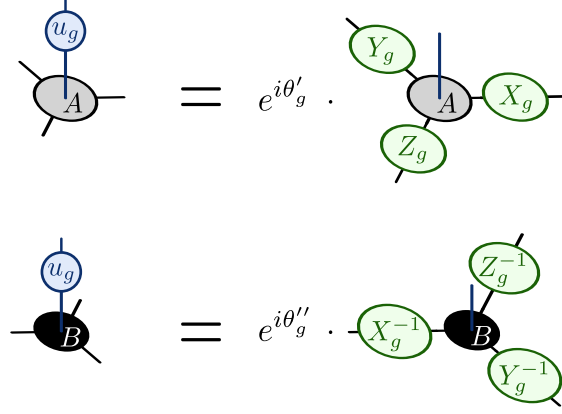


Figure 3.3.: If an injective PEPS is symmetric under a local symmetry G with unitary representation u then its tensors satisfy the illustrated relation (theorem 3.5).

Theorem 3.5 (Symmetric injective PEPS). *Consider a translationally invariant injective PEPS $|\Psi\rangle$ defined on the infinitely extended honeycomb lattice that is symmetric under a local symmetry G w.r.t. a unitary representation u . Then the following hold.*

(i) *The tensors defining the PEPS satisfy the relation in figure 3.3, explicitly*

$$\begin{aligned} \sum_{\sigma} \langle \sigma' | u_g | \sigma \rangle A^{\sigma} &= e^{i\theta'_g} A^{\sigma'}(X_g, Y_g, Z_g) \quad \text{and} \\ \sum_{\sigma} \langle \sigma' | u_g | \sigma \rangle B^{\sigma} &= e^{i\theta''_g} B^{\sigma'}(X_g^{-1}, Y_g^{-1}, Z_g^{-1}) \quad \forall g \in G \end{aligned} \quad (3.5)$$

with invertible matrices $X_g, Y_g, Z_g \in \mathbb{C}^{D \times D}$ and $e^{iN(\theta'_g + \theta''_g)} = e^{i\theta_g}$ where N is the number of unit cells and θ_g is from (3.2). Each of the matrices X , Y , and Z is uniquely defined up to a respective overall factor.

(ii) *If G is a compact connected Lie group, then $X_g, Y_g, Z_g, e^{i\theta'_g}$, and $e^{i\theta''_g}$ in (3.5) are representations of G , for a suitable choice of the overall factor of each matrix.*

Note that (i) is a direct consequence of theorem 3.4. It states that the symmetry transformation in the physical Hilbert space can be replaced by a local transformation of the virtual bond spaces (figure 3.3). The more subtle part (ii) holds only for continuous groups, which include the important examples of the groups $U(n)$ and $SU(n)$. Since $e^{i\theta'_g}$ and $e^{i\theta''_g}$ are representations of G , we can incorporate the phases into the physical representation u by defining representations $\tilde{u}_g := e^{-i\theta'_g} u_g$ and $\tilde{\tilde{u}}_g := e^{-i\theta''_g} u_g$. Equation (3.5) then states that A and B are symmetric tensors.

Discussion of the assumption of injectivity. Theorem 3.5 states that a translationally invariant PEPS with a local symmetry can be expressed in terms of symmetric tensors if the PEPS is injective. To discuss the restriction of injectivity, we first point out that the reverse is not true: One can easily construct a PEPS composed of symmetric tensors that is not injective (e.g. by extending the bond space of an injective PEPS

3. Extending the algorithm

by a bond configuration that has vanishing contribution to all physical configurations). Therefore, an algorithm that uses PEPS composed of symmetric tensors is not limited to injective PEPS. Still, using symmetric tensors may restrict the accessible Hilbert space. For example, it has been proved in [72, theorem 18] that for systems with semi-integer spin per unit cell, a state of zero total magnetization cannot be represented by a translationally invariant PEPS composed of $U(1)$ -symmetric tensors (fortunately, this does not invalidate the use of $U(1)$ -symmetric tensors for the spin- $\frac{1}{2}$ Heisenberg model on the honeycomb lattice, since the lattice has a two-atomic basis). We understand PEPS composed of symmetric tensors as a restricted set of ansatz states for the imaginary time evolution algorithm and will compare results obtained by calculations with symmetric and with normal tensors for the same Hamiltonian in section 4.2.2.

3.1.2. Symmetric tensors for abelian symmetries

The restriction (3.3) reduces the amount of information one can encode in a symmetric tensor. Algorithms that operate on symmetric PEPS can exploit this reduction of information to reduce memory requirements. One can factorize a symmetric tensor into a structure tensor that is completely determined by the symmetry group and a set of degeneracy tensors that contain the remaining degrees of freedom [73]. The factorization takes the simplest form for tensors of order up to three, which is of particular interest for the DMRG algorithm (which operates on MPS). Starting with tensors of order four (as, e.g., the tensors that define a PEPS on the honeycomb lattice), the factorization becomes more involved if the symmetry group is non-abelian. We therefore restrict the following discussion to abelian symmetries.

We first recall some fundamental results from the representation theory of abelian groups, so that readers without any previous knowledge of representation theory should be able to understand this section. We refer to standard textbooks on group theory (e.g. [74]) for the proofs of the fundamental theorems.

We consider a compact connected abelian Lie group G and a representation $U : G \rightarrow L(V)$ on some finite-dimensional complex vector space V . This means that for each $g \in G$, $U_g : V \rightarrow V$ is a linear map from V into itself such that $U_{g_1 g_2} = U_{g_1} U_{g_2}$ for all $g_1, g_2 \in G$ and U_e is the identity in V for e the neutral element in G . Since G is compact, we can define a scalar product in V such that U_g is unitary for all g . Recall that a finite-dimensional unitary representation of a group is completely reducible. Further, the irreducible representations of abelian groups are all one-dimensional. This means that we can choose a basis $\{|e_i\rangle\}$ in V such that the matrices representing U_g are diagonal for all $g \in G$,

$$\langle e_i | U_g | e_j \rangle = \delta_{i,j} D^i(g) \quad \text{with } D^i(g) \in \mathbb{C}, |D^i(g)| = 1 \quad \forall g \in G \quad (3.6)$$

The maps $D^i : G \rightarrow \mathbb{C}$ are called irreducible matrix representation of G . The representation U may contain several copies of the same irreducible matrix representation. We label different irreducible matrix representations by D^ν and relabel the basis states as $|e_i\rangle \equiv |\nu, \alpha_\nu\rangle \equiv |\nu\rangle \otimes |\alpha_\nu\rangle$, where, for each ν , the degeneracy label α_ν runs from one to the number of copies of D^ν that appear in U . We say that the states $\{|\alpha_\nu\rangle\}$ span the *degeneracy space* for states in V that transform w.r.t. D^ν . The matrix elements of U_g

then read

$$\langle \nu, \alpha_\nu | U_g | \mu, \alpha_\mu \rangle = \delta_{\nu, \mu} \delta_{\alpha_\nu, \alpha_\mu} D^\nu(g) \quad \forall g \in G \quad (3.7)$$

It is easy to see that the tensor product of two irreducible matrix representations D^ν and D^μ yields another matrix representation. For abelian groups, the product representation is also one-dimensional and therefore irreducible. We denote the product representation by $D^{\nu \oplus \mu}$.

We have chosen the basis $\{|\nu, \alpha_\nu\rangle\}$ such that the matrices corresponding to the representation U take the simple form of (3.7). We now show that a symmetric tensor takes a block-diagonal structure in the same basis.

Proposition 3.6 (Block-diagonal structure of abelian symmetric tensors). *Let $T : V^1 \times \dots \times V^n \rightarrow \mathbb{C}$ be a tensor of order n that is symmetric under an abelian group G w.r.t. the representations U^i , $i = 1, \dots, n$. In each bond space V^i , we define a basis $\{|\nu^i, \alpha_\nu^i\rangle\}$ as described above, where ν^i labels the irreducible representation and α_ν^i labels the degeneracy state. Then, we have for the components of T ,*

$$T_{\nu^1 \alpha_\nu^1, \nu^2 \alpha_\nu^2, \dots, \nu^n \alpha_\nu^n} := T(|\nu^1, \alpha_\nu^1\rangle, |\nu^2, \alpha_\nu^2\rangle, \dots, |\nu^n, \alpha_\nu^n\rangle) = 0 \quad (3.8)$$

if $D^{\nu^1 \oplus \nu^2 \oplus \dots \oplus \nu^n}$ is not the trivial matrix representation that maps every group element to $1 \in \mathbb{C}$.

Proof. T is symmetric. Therefore, for all $g \in G$,

$$\begin{aligned} T_{\nu^1 \alpha_\nu^1, \dots, \nu^n \alpha_\nu^n} &= T(|\nu^1, \alpha_\nu^1\rangle, \dots, |\nu^n, \alpha_\nu^n\rangle) = T(U_g^1 |\nu^1, \alpha_\nu^1\rangle, \dots, U_g^n |\nu^n, \alpha_\nu^n\rangle) = \\ &= T(D^{\nu^1}(g) |\nu^1, \alpha_\nu^1\rangle, \dots, D^{\nu^n}(g) |\nu^n, \alpha_\nu^n\rangle) = \\ &= (D^{\nu^1}(g) \cdots D^{\nu^n}(g)) \cdot T(|\nu^1, \alpha_\nu^1\rangle, \dots, |\nu^n, \alpha_\nu^n\rangle) = \\ &= D^{\nu^1 \oplus \dots \oplus \nu^n}(g) \cdot T_{\nu^1 \alpha_\nu^1, \dots, \nu^n \alpha_\nu^n} \end{aligned} \quad (3.9)$$

where we have used multilinearity of T in the fourth equation. For (3.9) to hold for all $g \in G$, the tensor component either has to vanish or $D^{\nu^1 \oplus \dots \oplus \nu^n}$ has to be the trivial representation. \square

Proposition 3.6 can be seen as an analog to quantum mechanical selection rules. An algorithm that operates on symmetric tensors can reduce memory requirements by storing only the tensor components that do not vanish by symmetry.

Physical interpretation. As an example, recall the antiferromagnetic spin- $\frac{1}{2}$ Heisenberg model defined in (2.1). The ground state of this model lies in the subspace $S_{\text{tot}}^z = 0$ and is therefore invariant under the local symmetry operation defined by $u(\phi) := e^{2i\phi S_j^z}$ acting simultaneously on each site j , where S_j^z is the z -component of the spin operator on site j so that $S_{\text{tot}}^z = \sum_j S_j^z$. This is a $U(1)$ symmetry since $u(\phi)u(\phi') = u(\phi + \phi')$ and $u(\phi + 2\pi) = u(\phi)$. The same symmetry group appears in systems with a conserved total particle number, where we have to replace $2S_j^z$ by $n_j - \bar{n}$ where n_j is the local particle number operator and \bar{n} is the (conserved) mean particle number per site. The group $U(1)$ is abelian and its irreducible matrix representations are given by

$$D^\nu : U(1) \ni e^{i\phi} \mapsto e^{i\nu\phi} \in \mathbb{C}; \quad \nu \in \mathbb{Z}. \quad (3.10)$$

3. Extending the algorithm

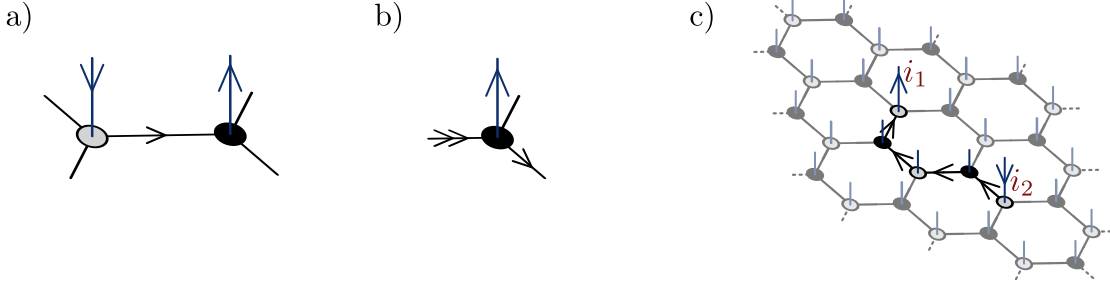


Figure 3.4.: Interpretation of $U(1)$ symmetric tensors. a) Illustration of particle number conservation on bonds. The horizontal arrow denotes a virtual particle flowing from the left to the right tensor. b) The net amount of incoming particles on a tensor must vanish (on the physical bond, we count the excess particle number w.r.t. a bias \bar{n}). c) A PEPS composed of $U(1)$ symmetric tensors still allows fluctuations of the local particle numbers but fluctuations must be balanced globally, i.e. each excess particle on some site i_1 , must be matched by a hole on some other site i_2 .

The trivial representation is D^0 , the product of two irreducible representations D^ν and D^μ is given by $D^{\nu \oplus \mu} = D^{\nu + \mu}$, and the inverse representation of D^ν is $D^{-\nu}$.

We interpret the label ν as an incoming virtual particle number (or incoming amount of magnetization) on the corresponding bond of a tensor. For a given bond configuration $|\nu, \alpha_\nu\rangle$, we decorate the bond with incoming arrows if $\nu > 0$ and outgoing arrows if $\nu < 0$ (figure 3.4). In theorem 3.5, corresponding bonds of neighboring tensors transform w.r.t. mutually inverse representations. This is particle number conservation on bonds, i.e. all particles that flow *out* of one tensor must flow *in* at the neighboring tensor (figure 3.4a). Proposition 3.6 states particle number conservation on vertices and takes the form of Kirchhoff's circuit law of currents, i.e. the total number of incoming particles on a vertex must vanish (figure 3.4b). Particle number conservation on bonds and the Kirchhoff law on vertices enforce that quantum mechanical fluctuations of the local particle numbers are balanced over the entire system so that the total particle number is fixed (figure 3.4c).

3.1.3. Calculations with symmetric tensors & measured performance gain

Proposition 3.6 states that tensors that are symmetric under an abelian symmetry have a block-diagonal structure. We explain how we exploit this structure in our implementation to reduce memory requirements and computation time. We conclude this section by presenting a comparison of measured computation times for a rather complex task with tensors of different symmetries.

One can break down the algorithms described in chapter 2 into a series of basic tensor manipulations, namely storing tensors in memory, permuting and reshaping the bond spaces, multiplying two tensors and factorizing tensors. We explain how to efficiently implement these operations for symmetric tensors.

Storing symmetric tensors. We apply proposition 3.6 to factorize a symmetric tensor T into a structure tensor S and a set of degeneracy tensors $P^{\nu^1, \nu^2, \dots, \nu^n}$.

$$T_{\nu^1 \alpha_\nu^1, \nu^2 \alpha_\nu^2, \dots, \nu^n \alpha_\nu^n} = S_{\nu^1, \nu^2, \dots, \nu^n} \cdot (P^{\nu^1, \nu^2, \dots, \nu^n})_{\alpha_\nu^1, \alpha_\nu^2, \dots, \alpha_\nu^n} \quad (\text{no sum}) \quad (3.11)$$

with $S_{\nu^1, \nu^2, \dots, \nu^n} = \delta_{\nu^1 \oplus \nu^2 \oplus \dots \oplus \nu^n, 0}$

Only those degeneracy tensors for which the corresponding element of the structure tensor does not vanish need to be kept in memory. We point out that the range of α_ν^i in (3.11) depends on ν and therefore the degeneracy tensors all have different dimensions. Although the structure tensor S is completely determined by the group, it is most convenient to calculate S when allocating memory for a symmetric tensor and to store it alongside the degeneracy tensors, as it will frequently be needed for tensor operations. S can be implemented as a table that maps a tuple of quantum numbers (ν^i, \dots, ν^n) to a reference to the corresponding degeneracy tensor if $S = 1$ and to a reserved null value if $S = 0$.

Permutation of tensor bonds. The operation $\nu^i \oplus \nu^j$ is commutative. Therefore, to permute the order of bond spaces in a symmetric tensor, one only needs to simultaneously permute the indices of S and P in (3.11).

Fusing and unfusing bond spaces. These operations frequently appear as substeps of more complex operations. For example, to perform a singular value decomposition of an order four tensor, one would first reshape the tensor to a matrix A by fusing the first two and the last two bonds, respectively, then decompose the matrix in a singular value decomposition $A = U \Lambda V^\dagger$ and finally unfuse the row index of U and the column index of V^\dagger .

A tensor $T : V^1 \times \dots \times V^n \rightarrow \mathbb{C}$ of order n is isomorphic to a tensor $\tilde{T} : (V^1 \otimes V^2) \times V^3 \times \dots \times V^n \rightarrow \mathbb{C}$ of order $n - 1$ defined by the linear extension of

$$\tilde{T}(|v^1\rangle \otimes |v^2\rangle, |v^3\rangle, \dots, |v^n\rangle) := T(|v^1\rangle, |v^2\rangle, |v^3\rangle, \dots, |v^n\rangle) \quad \forall |v^i\rangle \in V^i \quad (3.12)$$

If T is symmetric w.r.t. representations U^i , then direct calculation shows that \tilde{T} is symmetric w.r.t. the product representation $U^1 \otimes U^2$ acting on the first bond and U^i ($i = 3, \dots, n$) acting on the remaining bonds. A basis in the new first bond space $V' := V^1 \otimes V^2$ is given by the product basis and a basis state $|\nu^1, \alpha_\nu^1\rangle \otimes |\nu^2, \alpha_\nu^2\rangle$ transforms under $U^1 \otimes U^2$ with the representation $D^{\nu^1 \oplus \nu^2}$. In general, several different combinations of quantum numbers ν^1 and ν^2 add up to the same combined quantum number $\nu' := \nu^1 \oplus \nu^2$. Therefore, we have to label degeneracy states in V' by a tuple $\alpha'_\nu = (\nu^1, \nu^2, \alpha_\nu^1, \alpha_\nu^2)$ with the restriction $\nu^1 \oplus \nu^2 = \nu'$. The degeneracy tensors of \tilde{T} are then given by

$$(\tilde{P}^{\nu', \nu^3, \dots, \nu^n})_{\alpha'_\nu, \alpha_\nu^3, \dots, \alpha_\nu^n} \equiv (\tilde{P}^{\nu', \nu^3, \dots, \nu^n})_{(\nu^1, \nu^2, \alpha_\nu^1, \alpha_\nu^2), \alpha_\nu^3, \dots, \alpha_\nu^n} = (P^{\nu^1, \nu^2, \nu^3, \dots, \nu^n})_{\alpha_\nu^1, \alpha_\nu^2, \alpha_\nu^3, \dots, \alpha_\nu^n}. \quad (3.13)$$

The mapping from the tuple $(\nu^1, \nu^2, \alpha_\nu^1, \alpha_\nu^2)$ to the (integer) degeneracy index α'_ν is nontrivial since the range of possible values for α_ν^1 and α_ν^2 is different for each combination of ν^1 and ν^2 . It is useful to create a table for this mapping before copying values to the new degeneracy tensors \tilde{P} since it will be needed frequently when applying (3.13) for all values of ν^3, \dots, ν^n . If the bond space is to be split up again later (e.g. after a singular

3. Extending the algorithm

value decomposition), one has to remember the mapping table and apply (3.13) from right to left.

Multiplication of two tensors. One can reduce the contraction of two tensors over any pair of bonds to the contraction over the respective first bonds by appropriate preceding permutations of bond spaces. The costs of this preparative step are of subleading order to the multiplication of two tensors. Preceding permutations also allow one to order the bond spaces such that paging is reduced during the multiplication.

In a network of symmetric tensors, neighboring tensors transform w.r.t. mutually inverse representations on the connecting bond (figure 3.4a). Therefore, a state $|\nu, \alpha_\nu\rangle$ in the bond space of one tensor corresponds to a state $|\nu, \alpha_\nu\rangle$ in the corresponding bond space of the neighboring tensor, where $D^{-\nu}$ labels the inverse of the irreducible representation D^ν , such that $D^{\nu \oplus (-\nu)}$ is the trivial representation. We consider two symmetric tensors T and \tilde{T} of order n and m with degeneracy tensors P and \tilde{P} , respectively. The degeneracy tensors of the result of the contraction of T and \tilde{T} over the first bond are given by

$$\begin{aligned} & (\tilde{P}^{\nu^2, \dots, \nu^n, \mu^2, \dots, \mu^m})_{\alpha_\nu^2, \dots, \alpha_\nu^n, \alpha_\mu^2, \dots, \alpha_\mu^m} = \\ &= \sum_{\nu', \mu', \alpha'} \delta_{\nu' \oplus \mu', 0} \cdot (P^{\nu', \nu^2, \dots, \nu^n})_{\alpha', \alpha_\nu^2, \dots, \alpha_\nu^n} (\tilde{P}^{\mu', \mu^2, \dots, \mu^m})_{\alpha', \alpha_\mu^2, \dots, \alpha_\mu^m} \quad . \end{aligned} \quad (3.14)$$

One can contract two tensors over more than one bond by first fusing the bond spaces together as described in the preceding paragraph. We point out the following subtlety: Due to the Kronecker delta in (3.14), the quantum number μ' effectively runs reverse to ν' in the sum. Similarly, in a preceding fusion of bond spaces, one has to enumerate states $|\nu^1, \nu^2, \alpha_\nu^1, \alpha_\nu^2\rangle$ in the fused degeneracy space of T in such a way that the enumeration matches that of states $|\mu^1, \mu^2, \alpha_\mu^1, \alpha_\mu^2\rangle$ in the fused degeneracy space of \tilde{T} , e.g. by iterating over the pair (μ^1, μ^2) in reverse order.

Factorization of a tensor. To factorize a tensor, e.g. by a singular value decomposition, we first fuse bond spaces such that we have a tensor of order two, which we write as

$$T_{\nu^1 \alpha_\nu^1, \nu^2 \alpha_\nu^2} = \delta_{\nu^1 \oplus \nu^2, 0} \cdot (P^{\nu^1, -\nu^1})_{\alpha_\nu^1, \alpha_\nu^2} \quad (3.15)$$

We perform a singular value decomposition of each degeneracy tensor,

$$(P^{\nu^1, -\nu^1})_{\alpha_\nu^1, \alpha_\nu^2} = \sum_{\alpha'} (\tilde{P}^{\nu^1, -\nu^1})_{\alpha_\nu^1, \alpha'} \Lambda_{\alpha'}^{\nu^1} (\tilde{P}^{\nu^1, -\nu^1})_{\alpha', \alpha_\nu^2} \quad (3.16)$$

where \tilde{P} and $\tilde{\tilde{P}}$ define the degeneracy tensors of two new symmetric tensors \tilde{T} and $\tilde{\tilde{T}}$ of order two. We unfuse the first bond space of \tilde{T} and the last bond space of $\tilde{\tilde{T}}$ to restore the original dimensions. When multiplying the vector $\Lambda_{\alpha'}^{\nu^1}$ of singular values to a tensor, one has to distinguish between multiplication from the left and multiplication from the right. To multiply the singular values to a tensor from the left, $\Lambda_{\alpha'}^{\nu^1}$ has to be multiplied to tensor components with bond configuration $|\nu, \alpha'\rangle$, while, for a multiplication from the right, the corresponding bond configuration of the tensor is $|\nu, \alpha'\rangle$.

Truncation of a bond space. Some singular value decompositions in the algorithms from chapter 2 are followed by a truncation of the new bond space. We truncate the diagonal matrix $\Lambda_{\alpha'}^{\nu^1}$ in (3.16) to a new *total* bond dimension χ by keeping only the χ largest values. In this procedure, some degeneracy spaces (labeled by ν^1 in (3.16)) shrink more than others. We then shrink the degeneracy tensors \tilde{P} and $\tilde{\tilde{P}}$ accordingly. In case the dimension of a degeneracy space shrinks to zero we no longer need to keep the corresponding degeneracy tensors. The idea behind truncating symmetric tensors in this way is to distribute the available total bond dimension among the degeneracy spaces in an optimal way.

Measured performance gain. Exploiting the block-diagonal structure of symmetric tensors reduces the computation time required for tensor multiplications and singular value decompositions since both operations have to operate only on smaller subblocks of the full tensors. On the other hand, a computational overhead is to be expected due to the additional bookkeeping required for fusing and unfusing of bond spaces, which is a trivial operation for plain (non-symmetric) tensors.

We implemented the above operations for tensors that are symmetric under any of the groups \mathbb{Z}_2 , $U(1)$, and $U(1) \times U(1)$, as well as for plain (non-symmetric) tensors. The computationally most expensive step in both the imaginary time evolution algorithm and the SRG (chapter 2) is the singular value decomposition of an order four tensor. We measured the computation times required for an SVD of order four tensors for different symmetries (figure 3.5). The dimension of the bond space is the same on all four bonds and denoted by χ . The measured task includes fusing and unfusing of the first and the last pair of bond spaces. For very low bond dimensions, symmetric tensors perform worse than plain tensors due to the bookkeeping overhead for the reshaping to a matrix. For high bond dimensions, the CPU time has the expected $O(\chi^6)$ scaling for all symmetries and exploiting $U(1) \times U(1)$ symmetry reduces the computation time by more than two orders of magnitude compared to plain tensors. $U(1) \times U(1)$ symmetry arises in systems with two independently conserved particle numbers, e.g. the Hubbard model (see chapter 5). Investigations for the Hubbard model would be out of reach for the SRG algorithm without the use of symmetric tensors.

3.2. Fermionic systems

In chapter 2, we introduced a model composed of spin degrees of freedom to exemplify the imaginary time evolution and the SRG algorithm. We now explain how we can apply the algorithms to systems with fermionic degrees of freedom. The properties of fermionic systems are of particular interest in two spatial dimensions. For example, the state of the fractional quantum Hall effect was the first experimentally observed state that has been identified to develop topological order [6]. Also, high temperature superconductors are composed in a layered structure where the movement of electrons is essentially restricted to parallel planes [4, 5]. Fermionic systems are hard to study with quantum Monte Carlo simulations due to the sign problem [16], except in some special cases (e.g. the Hubbard model at half filling [75]).

In this section, we show why a naive implementation of the SRG algorithm would fail

3. Extending the algorithm

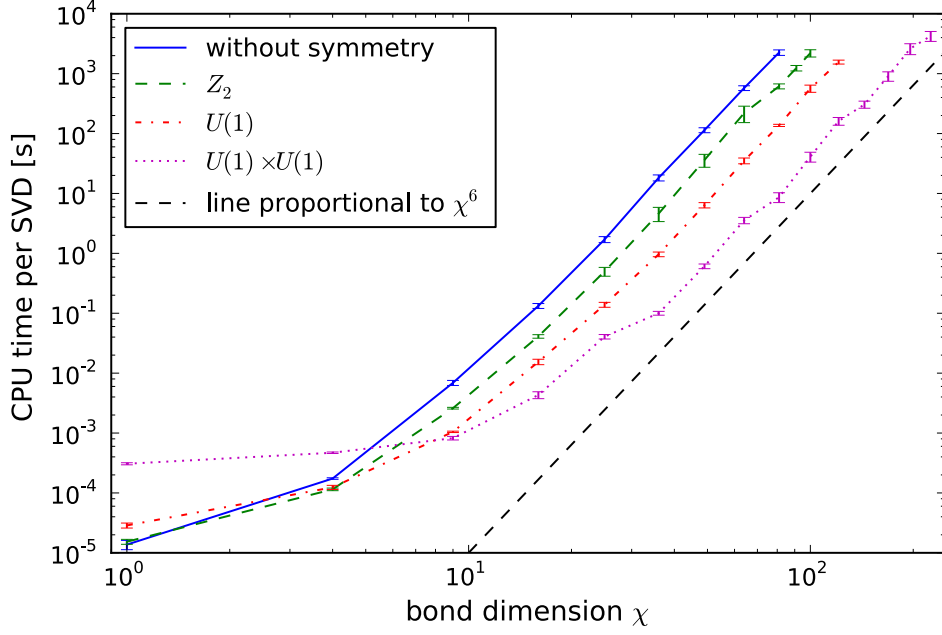


Figure 3.5.: CPU times for the singular value decomposition of an order four tensor with bond dimension χ on all four bonds under the use of different symmetries. To perform the SVD, we first reshape the tensor to a matrix with dimensions $\chi^2 \times \chi^2$. Exploiting the block diagonal structure of symmetric tensors significantly reduces the calculation time for all symmetries as long as $\chi \gtrsim 10$. For $U(1)$ and $U(1) \times U(1)$ symmetry, we assumed that each virtual particle number on the bond spaces can take three different values $-1, 0$, and 1 and distributed the available bond space evenly among all resulting symmetry sectors. This is a conservative assumption. In test runs of the SRG algorithm, we observed even higher fragmented bond spaces.

to handle fermionic systems. We then describe a variant of the algorithm that can be applied to fermionic systems.

3.2.1. Failure of naive PEPS for fermionic systems

We consider a system defined on an (arbitrary) lattice of N sites where a fermionic orbital is located on each site i . We show that it is possible to describe a state in this system by a network of tensors but that the naive way to do so does not allow to apply the algorithms introduced in chapter 2.

Let us label the sites by $i = 1, \dots, N$ and let c_i^\dagger be the creator of a fermion on site i . We introduce the occupation numbers $\sigma_i \in \{0, 1\}$ and define the basis states of the Hilbert space by

$$|\sigma_1, \sigma_2, \dots, \sigma_N\rangle := \left(\prod_{i=1}^N (c_i^\dagger)^{\sigma_i} \right) |0\rangle \quad (3.17)$$

where $|0\rangle$ is the vacuum and we emphasize our convention that the product runs from $i = 1$ to N in ascending order. An arbitrary state $|\Psi\rangle$ expanded in this basis reads

$$|\Psi\rangle = \sum_{\{\sigma_i\}} \psi_{\{\sigma_i\}} |\sigma_1, \sigma_2, \dots, \sigma_N\rangle \quad (3.18)$$

where the sum runs over all configurations of $\sigma_1, \dots, \sigma_N$. The naive way to implement PEPS for fermionic systems is to express the coefficients $\psi_{\{\sigma_i\}}$ as a product of tensors. We define a tensor $A^{[i], \sigma_i}$ where $[i]$ enumerates the lattice sites if the PEPS is not translationally invariant, σ_i is the index for the physical bond and we dropped the lower indices of the tensors for notational simplicity. We write

$$\psi_{\{\sigma_i\}} = \text{Tr} \left[\prod_i A^{[i], \sigma_i} \right] \quad (3.19)$$

where the trace is to contract the product of tensors according to the lattice topology.

An essential operation in both the imaginary time evolution algorithm and the SRG is to apply a bond operator \hat{O}_{rs} to a PEPS $|\Psi\rangle$. In the imaginary time evolution algorithm, we apply the bond evolution operators $e^{-\Delta\tau H_{ij}}$ in (2.14) and in the SRG, we apply the operator whose expectation value we want to calculate in (2.28). Both algorithms rely on the assumption that the resulting state $\hat{O}_{rs}|\Psi\rangle$ has the form of a PEPS except on the two neighboring sites r and s and that the application of \hat{O}_{rs} is a local operation that leaves the tensors on sites $i \neq r, s$ unchanged.

Let r and s label two neighboring sites with $r < s$ for notational simplicity and let us apply the bond operator $\hat{O}_{rs} := c_r^\dagger c_s$ to the state in (3.18)-(3.19).

$$c_r^\dagger c_s |\Psi\rangle = \sum_{\{\sigma_i\}} \text{Tr} \left[\prod_i A^{[i], \sigma_i} \right] c_r^\dagger c_s \left(\prod_{i=1}^N (c_i^\dagger)^{\sigma_i} \right) |0\rangle \quad (3.20)$$

$$= \sum_{\{\sigma_i\}} \eta_{\{\sigma_i\}} \text{Tr} \left[\left(\prod_{i \neq r, s} A^{[i], \sigma_i} \right) \left(\sum_{\sigma'_r, \sigma'_s} O_{\sigma'_r, \sigma'_s}^{\sigma_r, \sigma_s} A^{[r], \sigma'_r} A^{[s], \sigma'_s} \right) \right] |\sigma_1, \dots, \sigma_N\rangle \quad (3.21)$$

3. Extending the algorithm

with

$$O_{\sigma'_r, \sigma'_s}^{\sigma_r, \sigma_s} = \delta_{\sigma_r, 1} \delta_{\sigma_s, 0} \delta_{\sigma'_r, 0} \delta_{\sigma'_s, 1} \quad \text{and} \quad \eta_{\{\sigma_i\}} = \prod_{i=r+1}^{s-1} (-1)^{\sigma_i} \quad (3.22)$$

In the step from (3.20) to (3.21), we moved $c_r^\dagger c_s$ to the right just before the term $(c_r^\dagger)^{\sigma_r}$ in the product and then moved c_s further to the right just before the term $(c_s^\dagger)^{\sigma_s}$, picking up a minus sign for each creation operator in between. This is the origin of the factor $\eta_{\{\sigma_i\}}$ in (3.21).

If the system is one-dimensional then we can number the sites in the natural way so that nearest neighbors are labeled by r and $s = r + 1$. In this case, $\eta_{\{\sigma_i\}} = 1$ and applying a bond operator to a PEPS (an MPS) is a local operation on the tensors.

In more than one spatial dimensions, the sites have by definition more than two nearest neighbors and it is impossible to find a numbering that assigns successive numbers to all pairs of nearest neighbors. For nearest neighbors r, s with $s - r > 1$, the factor $\eta_{\{\sigma_i\}}$ is a function of the occupation numbers σ_i with $r < i < s$. We could incorporate $\eta_{\{\sigma_i\}}$ into the tensors on the sites i but this would invalidate the assumption of $\hat{O}_{rs}|\Psi\rangle$ being a local operation on the tensors.

3.2.2. Fermionic tensors

The algorithms described in chapter 2 cannot operate on the naive ansatz states defined in (3.18)–(3.19) because applying a bond operator to such a state is not a local operation on the tensors. We now introduce a class of wave functions defined on systems with fermionic degrees of freedom which overcome this problem. We refer to this new set of ansatz states as “fermionic PEPS” and to the local building blocks that constitute a fermionic PEPS as “fermionic tensors”. Applying a bond operator \hat{O}_{kl} to a fermionic PEPS can be written as a local transformation of the fermionic tensors defined on sites k and l only. Our construction of fermionic tensors is similar to the Grassmann number valued tensors introduced in [3] and a fermionic tensor is equivalent to a Grassmann number valued tensor if all bond dimensions are powers of two. However, our construction explicitly allows arbitrary bond dimensions as well as the use of symmetric tensors (definition 3.2) for any abelian symmetry group that has \mathbb{Z}_2 as a subgroup. Alternative approaches to study fermions with tensor network states include the introduction of an extra “sign” bond in a single bond direction with periodic boundaries [76], which works best on the square lattice, and an effective rearranging of the tensors to a one-dimensional chain by the use of so-called swap gates [77–80], which is hard to generalize to infinite systems.

We recall the projective construction of a PEPS with spin degrees of freedom from section 1.2 and generalize this idea to systems with fermionic degrees of freedom. We construct a *fermionic PEPS* by setting an entangled pair of auxiliary bosonic or fermionic degrees of freedom on each link and then projecting onto the physical Hilbert space. On each link l , we define two identical auxiliary Hilbert spaces $\mathcal{H}_{\text{aux}}^{[la]} \cong \mathcal{H}_{\text{aux}}^{[lb]}$ (figure 3.6a). To construct $\mathcal{H}_{\text{aux}}^{[la/lb]}$, we define a vacuum state $|0_{\text{aux}}^{[la/lb]}\rangle \in \mathcal{H}_{\text{aux}}^{[la/lb]}$ and creation operators $d_{\nu, \alpha_\nu}^{[la/lb] \dagger}$ for auxiliary particles. In the simplest case, $\nu \in \{0, 1\}$ and $\alpha_\nu \in \{1, \dots, D_\nu\}$ with some bond dimensions D_0 and D_1 . Then, $d_{\nu, \alpha_\nu}^{[la/lb] \dagger}$ creates an auxiliary bosonic particle if

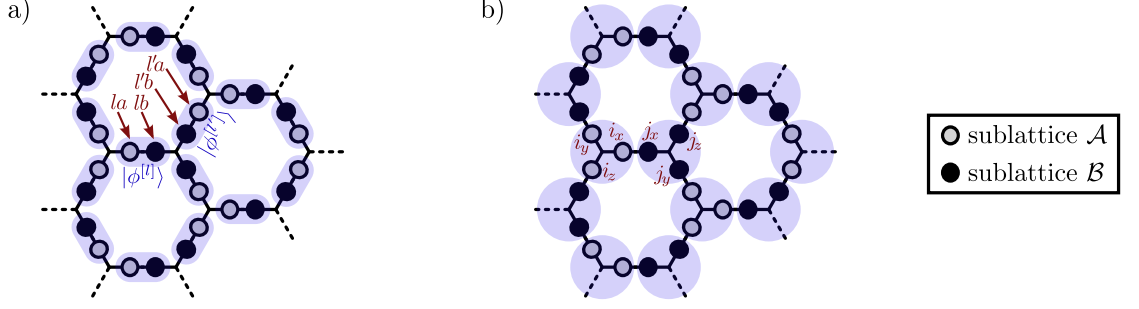


Figure 3.6.: a) We define two identical auxiliary Hilbert spaces $\mathcal{H}_{\text{aux}}^{[la]} \cong \mathcal{H}_{\text{aux}}^{[lb]}$ on each link l of the honeycomb lattice and put an entangled state $|\phi^{[l]}\rangle$ on each link (3.23). Since the lattice is bipartite, we can define an order of fermions in the two auxiliary Hilbert spaces. b) We associate three auxiliary Hilbert spaces to each vertex i and label them by i_x , i_y , and i_z . Then, each link contains one auxiliary Hilbert space associated to a vertex in sublattice \mathcal{A} and one associated to sublattice \mathcal{B} .

$\nu = 0$ and an auxiliary fermionic particle if $\nu = 1$. On each link l , we define an entangled state

$$|\phi^{[l]}\rangle := \sum_{\nu=0,1} \sum_{\alpha_\nu=1}^{D_\nu} d_{\nu,\alpha_\nu}^{[la]\dagger} d_{\nu,\alpha_\nu}^{[lb]\dagger} (|0_{\text{aux}}^{[la]}\rangle \otimes |0_{\text{aux}}^{[lb]}\rangle) \in \mathcal{H}_{\text{aux}}^{[la]} \otimes \mathcal{H}_{\text{aux}}^{[lb]} \quad (3.23)$$

We point out that $|\phi^{[l]}\rangle$ is not a *fully* entangled state between $\mathcal{H}_{\text{aux}}^{[la]}$ and $\mathcal{H}_{\text{aux}}^{[lb]}$ since it contains only configurations with exactly one (bosonic or fermionic) auxiliary particle in each half.

For $\nu = 1$, the order of the two fermion creation operators in (3.23) is important. We restrict our description to bipartite lattices and denote the two sublattices by \mathcal{A} and \mathcal{B} . Then each link l connects a vertex $i \in \mathcal{A}$ with a vertex $j \in \mathcal{B}$ and we attribute $\mathcal{H}_{\text{aux}}^{[la]}$ to vertex i and $\mathcal{H}_{\text{aux}}^{[lb]}$ to vertex j . Consider the example of a honeycomb lattice and let i_x , i_y , and i_z label the three auxiliary Hilbert spaces attributed to site i , where x , y , and z denote the three bond directions (figure 3.6b).

Let $\mathcal{H}_{\text{loc}}^{[i]}$ be the local physical Hilbert space on site i , which contains a vacuum state $|0^{[i]}\rangle$. There is no need to define an extra unphysical vacuum state as for the auxiliary Hilbert spaces. We define operators $b_{\nu,\alpha_\nu}^{[i]\dagger}$ such that the set of states $\{b_{\nu,\alpha_\nu}^{[i]\dagger} |0^{[i]}\rangle\}_{\nu,\alpha_\nu}$ is a basis of $\mathcal{H}_{\text{loc}}^{[i]}$. Again, $b_{\nu,\alpha_\nu}^{[i]\dagger}$ maps the vacuum to a bosonic state if $\nu = 0$ and to a fermionic state if $\nu = 1$. For example, if each site can be occupied by electrons with two spin orientations whose creators are c_{\uparrow}^\dagger and c_{\downarrow}^\dagger , then we have two bosonic operators $b_{0,1}^{[i]\dagger} = \mathbb{1}$ and $b_{0,2}^{[i]\dagger} = c_{\uparrow}^\dagger c_{\downarrow}^\dagger$ and two fermionic operators $b_{1,1}^{[i]\dagger} = c_{\uparrow}^\dagger$ and $b_{1,2}^{[i]\dagger} = c_{\downarrow}^\dagger$. We refer to $b_{0,1}^{[i]\dagger}$ as a bosonic creation operator even though it is hermitian and therefore commutes with the annihilator $b_{0,1}^{[i]}$.

On each site i of the lattice, we now define a projector from the three auxiliary Hilbert

3. Extending the algorithm

spaces to the local Hilbert space,

$$P^{[i]} := \sum_{\nu, \alpha_\nu} A_{\nu^0 \alpha_\nu^0, \nu^x \alpha_\nu^x, \nu^y \alpha_\nu^y, \nu^z \alpha_\nu^z}^{[i]} b_{\nu^0 \alpha_\nu^0}^{[i] \dagger} |0^{[i]}\rangle \langle 0_{\text{aux}}^{[i]}| d_{\nu^x \alpha_\nu^x}^{[i_x]} d_{\nu^y \alpha_\nu^y}^{[i_y]} d_{\nu^z \alpha_\nu^z}^{[i_z]} \quad (3.24)$$

with $A_{\nu^0 \alpha_\nu^0, \nu^x \alpha_\nu^x, \nu^y \alpha_\nu^y, \nu^z \alpha_\nu^z}^{[i]} \in \mathbb{C}$ and $\langle 0_{\text{aux}}^{[i]}| = \langle 0_{\text{aux}}^{[i_x]}| \otimes \langle 0_{\text{aux}}^{[i_y]}| \otimes \langle 0_{\text{aux}}^{[i_z]}|$

where the sum runs over all combinations of indices of the tensor $A^{[i]}$.

The entangled auxiliary states $|\phi^{[l]}\rangle$ and the projectors $P^{[i]}$ form the building blocks of what can be regarded as a generalization of PEPS for fermionic systems. If we define a numbering $i = 1, \dots, N$ of the lattice sites i then we can define a state in the combined physical Hilbert space by

$$|\Psi\rangle := \left(\prod_{i=1}^N P^{[i]} \right) \left(\prod_l |\phi^{[l]}\rangle \right). \quad (3.25)$$

The order of the entangled auxiliary states $|\phi^{[l]}\rangle$ in (3.25) is not important because $|\phi^{[l]}\rangle$ contains only configurations with an even number of fermions. In contrast, $|\Psi\rangle$ is well-defined only if one specifies an order of the projectors $P^{[i]}$ since the sum in (3.24) contains terms with odd numbers of fermionic operators. For the same reason, applying a bond operator to the state $|\Psi\rangle$ defined in (3.25) is not a local operation on the tensors $A^{[i]}$.

We now restrict the description to states $|\Psi\rangle$ with an even total number of (physical) fermions. This implies that the Hamiltonian creates and annihilates fermions in pairs only. If n_i denotes the operator for the number of (physical) fermions on site i , then $|\Psi\rangle$ contains an even total number of fermions if it is invariant under the transformation $|\Psi\rangle \mapsto (\prod_i \exp(i\pi n_i)) |\Psi\rangle$. This is a local \mathbb{Z}_2 symmetry in the sense of definition 3.1 (section 3.1.1). In section 3.1, we showed for regular (non-fermionic) PEPS that we can ensure that a PEPS has a local symmetry by imposing a symmetry constraint onto its tensors (theorem 3.5). In a similar way, we ensure that $|\Psi\rangle$ from (3.25) contains only configurations with an even number of fermions by imposing the following condition onto the tensor $A^{[i]}$ in (3.24)

$$A_{\nu^0 \alpha_\nu^0, \nu^x \alpha_\nu^x, \nu^y \alpha_\nu^y, \nu^z \alpha_\nu^z}^{[i]} = 0 \quad \text{if } \nu^0 \oplus \nu^x \oplus \nu^y \oplus \nu^z \neq 0, \quad (3.26)$$

where \oplus denotes addition modulo two. This is just another way of saying that the projector $P^{[i]}$ contains only terms with an even number of fermionic operators. With this restriction, (3.25) states that the state $|\Psi\rangle$ is constructed by starting from a bosonic state $\prod_l |\phi^{[l]}\rangle$, applying an even number of fermionic operators, and projecting on the subspace of zero auxiliary particles. Therefore, $|\Psi\rangle$ contains only configurations with an even number of fermions, as desired.

With the restriction (3.26), all projectors $P^{[i]}$ commute with each other and the order of the projectors in (3.25) is no longer important. Therefore, applying a bond operator \hat{O}_{rs} to $|\Psi\rangle$ is now a local operation on the tensors $A^{[r]}$ and $A^{[s]}$. We show how to apply a generic bond operator to a fermionic PEPS in the next section.

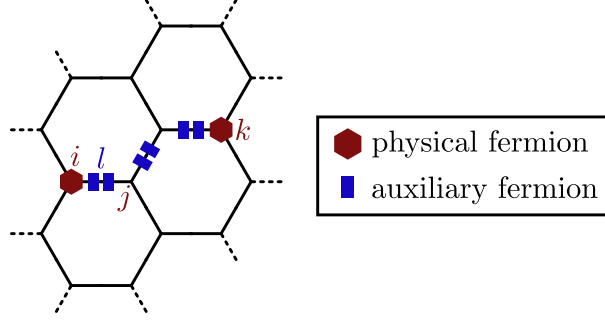


Figure 3.7.: \mathbb{Z}_2 symmetric fermionic tensors on each vertex ensure that all fermions appear in pairs.

We point out that theorem 3.5 holds for regular (non-fermionic) PEPS and not for the fermionic PEPS defined in (3.25).³ However, it has been demonstrated in [3, 81] that one can write a wide variety of physically relevant states, ranging from Fermi-liquid states to strongly correlated states, in a notation that is equivalent to the one introduced here. This motivates the use of fermionic PEPS as ansatz states for the imaginary time evolution algorithm.

Physical interpretation. In section 3.1.2, we interpreted the use of $U(1)$ symmetric tensors with the picture of virtual particles that flow between the tensors, ensuring that for every excess (physical) particle that exists on some site i , a corresponding hole must exist on a different site k . We interpret fermionic PEPS in a similar way. We constructed the projectors $P^{[i]}$ by “complementing the fermionic operators”, i.e. if $P^{[i]}$ creates a physical fermion, then it also has to annihilate an auxiliary fermion on some link l . Since $|\phi^{[l]}\rangle$ is a bosonic state, the neighboring projector $P^{[j]}$ also has to annihilate an auxiliary fermion. By condition (3.26), $P^{[j]}$ then either has to create a physical fermion or it has to annihilate a second auxiliary fermion, thus continuing the chain of auxiliary fermion pairs in figure 3.7. On a finite lattice, the chain has to end at some site k , where a second physical fermion is created. Therefore, fermions can only be created in pairs.

In the naive definition of fermionic PEPS (3.18)–(3.19), one has to define an order of all sites i . By complementing the fermionic operators, we no longer need this kind of *global* order. Instead, we reduced the problem to defining (i) a *local* order of the physical and auxiliary fermions on each site i (i.e. an order of the operators in 3.24) and (ii) an order for all pairs of *neighboring* sites (i.e. an order of the operators in 3.23). The latter is easy to define on a bipartite lattice.

Arbitrary symmetries. Condition (3.26) states that $A^{[i]}$ is a \mathbb{Z}_2 symmetric tensor in the sense of definition 3.2 w.r.t. the representation that maps the bond configuration $|\nu, \alpha_\nu\rangle$ to $(-1)^\nu |\nu, \alpha_\nu\rangle$. We can generalize the construction of fermionic PEPS for any symmetry

³Also, part (ii) of theorem 3.5 does not apply to the discrete symmetry group \mathbb{Z}_2 . This is more of a formal restriction since we use $U(1)$ or $U(1) \times U(1)$ symmetry instead of \mathbb{Z}_2 symmetry in our applications anyway (see chapter 5).

3. Extending the algorithm

group that has \mathbb{Z}_2 as a subgroup. For example, consider a system where the Hamiltonian conserves not only the parity $\prod_i (-1)^{n_i}$ of the state but rather the total number $\sum_i n_i$ of fermions. The ground state $|\Psi\rangle$ then has a well-defined (even) total particle number N_{tot} , i.e. it is invariant under the transformation $|\Psi\rangle \mapsto (\prod_i e^{i\varphi(n_i - \bar{n}_i)}) |\Psi\rangle$ for all $\varphi \in \mathbb{R}$, where $\bar{n}_i \in 2\mathbb{N}_0$ denote constant even biases chosen such that $\sum_i \bar{n}_i = N_{\text{tot}}$. We construct the auxiliary Hilbert spaces $\mathcal{H}_{\text{aux}}^{[la/lb]}$ by introducing creation operators for auxiliary particles $d_{\nu, \alpha_\nu}^{[la/lb]\dagger}$ where now $\nu \in \mathbb{Z}$. The operators $d_{\nu, \alpha_\nu}^{[la/lb]\dagger}$ create an auxiliary bosonic state if ν is even and a fermionic state if ν is odd. The entangled state $|\phi^{[l]}\rangle$ on link l reads (cf. (3.23))

$$|\phi^{[l]}\rangle := \sum_{\nu} \sum_{\alpha_\nu} d_{\nu, \alpha_\nu}^{[la]\dagger} d_{-\nu, \alpha_\nu}^{[lb]\dagger} (|0_{\text{aux}}^{[la]}\rangle \otimes |0_{\text{aux}}^{[lb]}\rangle) \quad (3.27)$$

We interpret ν in the sum in (3.27) as a number of virtual fermions flowing from $\mathcal{H}_{\text{aux}}^{[lb]}$ to $\mathcal{H}_{\text{aux}}^{[la]}$ if $\nu > 0$ and in the reverse direction if $\nu < 0$. Then, the minus sign in $d_{-\nu, \alpha_\nu}^{[lb]\dagger}$ ensures conservation of the virtual particle number on links.

On each vertex i , we label the constructors of physical states by $b_{\nu, \alpha_\nu}^{[i]\dagger}$ with $\nu \in \mathbb{N}_0$ such that $b_{\nu, \alpha_\nu}^{[i]\dagger} |0^{[i]}\rangle$ is a state with ν (physical) fermions. Therefore, $b_{\nu, \alpha_\nu}^{[i]\dagger}$ is a product of ν fermionic creation operators and commutes with all other operators if ν is even but anticommutes with all fermionic operators if ν is odd. The condition for the tensors $A^{[i]}$ becomes (cf. (3.26))

$$A_{\nu^0 \alpha_\nu^0, \nu^x \alpha_\nu^x, \nu^y \alpha_\nu^y, \nu^z \alpha_\nu^z}^{[i]} = 0 \quad \text{if } \bar{n}_i - \nu^0 + \nu^x + \nu^y + \nu^z \neq 0 \quad (3.28)$$

that is, the number ν^0 of physical fermions created by $P^{[i]}$ is equal to the bias \bar{n}_i plus the number $\nu^x + \nu^y + \nu^z$ of virtual fermions $P^{[i]}$ annihilates. Since the entangled states $|\phi^{[l]}\rangle$ contain only configurations with a net number of zero virtual fermions, a fermionic PEPS constructed this way contains $N_{\text{tot}} = \sum_i \bar{n}_i$ fermions, as desired. Note that, since \bar{n}_i is even, restriction (3.28) ensures that $P^{[i]}$ contains only configurations with an even number of fermionic operators so that all projectors commute with each other, as in the \mathbb{Z}_2 symmetric case.

Restriction (3.28) states that $A^{[i]}$ is a $U(1)$ symmetric tensor in the sense of definition 3.2 w.r.t. the usual representations acting on the virtual bonds and the representation $D(e^{i\varphi}) : |\nu^0, \alpha_\nu^0\rangle \mapsto e^{i\varphi(\bar{n}_i - \nu^0)} |\nu^0, \alpha_\nu^0\rangle$ acting on the physical bond. We can exploit this symmetry to reduce memory requirements and calculation times of our implementation, as described in section 3.1.3.

The algorithms described in chapter 2 operate on translationally invariant states. For a translationally invariant fermionic PEPS, the bias \bar{n}_i must not depend on the unit cell. Since \bar{n}_i is even, this means that translationally invariant fermionic PEPS can only describe states with an even average number of fermions per unit cell (note that quantum mechanical fluctuations with an odd number of fermions in a given unit cell are still allowed, though).

3.2.3. Calculations with fermionic tensors

Algorithms that operate on PEPS are typically composed of a set of basic tensor operations. These operations are different for fermionic tensors than they are for normal

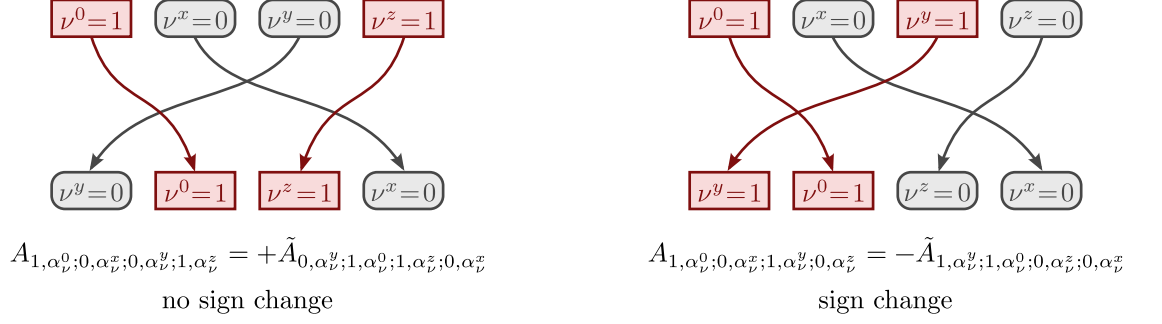


Figure 3.8.: Different components of a fermionic tensor pick up different signs under the same permutation of bonds.

tensors. We describe how to perform calculations with fermionic tensors and focus especially on the appearances of minus signs in each operation. The description has to be very explicit and therefore rather technical in some cases because some minus signs appear in rather subtle steps. To the authors' knowledge, a comprehensive description of operations on fermionic tensors (or, equivalently, Grassmann number valued tensors) of this kind has not been published yet.⁴

Storing fermionic PEPS. A fermionic PEPS is defined by the entangled auxiliary states $|\phi^{[i]}\rangle$ and the projectors $P^{[i]}$. An algorithm that operates on fermionic PEPS does not need to store a representation of $|\phi^{[i]}\rangle$ in memory since it is completely defined by the bond spaces and the order convention for neighboring sites. The projectors $P^{[i]}$ are completely defined by the tensors $A^{[i]}$ (see (3.24)), which are symmetric under a symmetry group that has \mathbb{Z}_2 as a subgroup. An algorithm that operates on fermionic PEPS has to store the tensors $A^{[i]}$ in memory and can use the optimizations for symmetric tensors described in section 3.1.3 to reduce memory requirements and computational costs.

Permutation of tensor bonds. We use the convention that, whenever we rearrange the order of the indices of a fermionic tensor $A_{\nu^0\alpha_\nu^0,\nu^x\alpha_\nu^x,\nu^y\alpha_\nu^y,\nu^z\alpha_\nu^z}$, then we reorder the operators $b_{\nu,\alpha_\nu}^\dagger$ and d_{ν,α_ν} in (3.24) in the same way and absorb any minus signs from this permutations in the tensor elements. This kind of permutation of operators is necessary, e.g. as a preparation for the multiplication of two tensors (see below). The creators $b_{\nu,\alpha_\nu}^\dagger$ and the annihilators d_{ν,α_ν} behave in the same way under permutations. They anticommute if ν labels a fermionic state (i.e. if $\nu = 1$ in the case of only \mathbb{Z}_2 symmetry). Reordering the operators gives rise to a minus sign if the permutation of the operators, restricted to only those operators with fermionic ν , has negative parity. We point out that different components of $A_{\nu^0\alpha_\nu^0,\nu^x\alpha_\nu^x,\nu^y\alpha_\nu^y,\nu^z\alpha_\nu^z}$ pick up different signs under the same permutation, depending on the values of ν^0 , ν^x , ν^y , and ν^z (see figure 3.8).

⁴See, however, [82] for a discussion of contraction schemes in a more general framework of fermionic operator circuits.

3. Extending the algorithm

Multiplication of two tensors. We consider two fermionic tensors $A^{[1]}$ and $A^{[2]}$ which we want to contract over the z -bond. We write the corresponding projectors as

$$P^{[1]} := \langle 0_{\text{aux}}^{[1]} | \sum_{\nu, \alpha_\nu} A_{\nu^0 \alpha_\nu^0, \nu^x \alpha_\nu^x, \nu^y \alpha_\nu^y, \nu^z \alpha_\nu^z}^{[1]} b_{\nu^0 \alpha_\nu^0}^{[1] \dagger} d_{\nu^x \alpha_\nu^x}^{[1]} d_{\nu^y \alpha_\nu^y}^{[1]} d_{\nu^z \alpha_\nu^z}^{[1]} | 0^{[1]} \rangle \quad \text{and} \quad (3.29)$$

$$P^{[2]} := \langle 0_{\text{aux}}^{[2]} | \sum_{\mu, \alpha_\mu} A_{\mu^0 \alpha_\mu^0, \mu^x \alpha_\mu^x, \mu^y \alpha_\mu^y, \mu^z \alpha_\mu^z}^{[2]} b_{\mu^0 \alpha_\mu^0}^{[2] \dagger} d_{\mu^x \alpha_\mu^x}^{[2]} d_{\mu^y \alpha_\mu^y}^{[2]} d_{\mu^z \alpha_\mu^z}^{[2]} | 0^{[2]} \rangle \quad . \quad (3.30)$$

The multiplication of two fermionic projectors is defined by

$$\begin{aligned} P^{[12]} &= \langle 0_{\text{aux}}^{[12]} | \sum_{\substack{\nu, \alpha_\nu \\ \mu, \alpha_\mu}} A_{\nu^0 \alpha_\nu^0, \nu^x \alpha_\nu^x, \nu^y \alpha_\nu^y, \mu^0 \alpha_\mu^0, \mu^x \alpha_\mu^x, \mu^y \alpha_\mu^y}^{[12]} \\ &\quad b_{\nu^0 \alpha_\nu^0}^{[1] \dagger} d_{\nu^x \alpha_\nu^x}^{[1]} d_{\nu^y \alpha_\nu^y}^{[1]} b_{\mu^0 \alpha_\mu^0}^{[2] \dagger} d_{\mu^x \alpha_\mu^x}^{[2]} d_{\mu^y \alpha_\mu^y}^{[2]} (| 0^{[1]} \rangle \otimes | 0^{[2]} \rangle) \\ &:= P^{[1]} P^{[2]} | \phi^{[12]} \rangle \end{aligned} \quad (3.31)$$

with

$$| \phi^{[12]} \rangle = \sum_{\rho=0,1} \sum_{\alpha_\rho} d_{\rho, \alpha_\rho}^{[1z] \dagger} d_{\rho, \alpha_\rho}^{[2z] \dagger} (| 0_{\text{aux}}^{[1z]} \rangle \otimes | 0_{\text{aux}}^{[2z]} \rangle) \quad (3.32)$$

where $| 0_{\text{aux}}^{[12]} \rangle$ is the vacuum of all auxiliary Hilbert spaces attributed to sites 1 and 2 except $\mathcal{H}_{\text{aux}}^{[1z]}$ and $\mathcal{H}_{\text{aux}}^{[2z]}$ and we assumed that site 1 is in the sublattice \mathcal{A} and site 2 in the sublattice \mathcal{B} . The problem at hand is to find the elements of the tensor $A^{[12]}$. Permuting the operator $d_{\rho, \alpha_\rho}^{[1z] \dagger}$ from the construction of $| \phi^{[12]} \rangle$ to the left past $P^{[2]}$ in (3.31) does not change the sign since $P^{[2]}$ contains only terms with an even number of fermionic operators. We then have, at the right end of $P^{[1]}$, the factor $\langle 0_{\text{aux}}^{[1z]} | d_{\nu^z \alpha_\nu^z}^{[1z]} d_{\rho, \alpha_\rho}^{[1z] \dagger} | 0_{\text{aux}}^{[1z]} \rangle = \delta_{\nu^z, \rho} \delta_{\alpha_\nu^z, \alpha_\rho}$ and, at the right end of $P^{[2]}$ the factor $\langle 0_{\text{aux}}^{[2z]} | d_{\mu^z \alpha_\mu^z}^{[2z]} d_{\rho, \alpha_\rho}^{[2z] \dagger} | 0_{\text{aux}}^{[2z]} \rangle = \delta_{\mu^z, \rho} \delta_{\alpha_\mu^z, \alpha_\rho}$. This means that we get the new tensor $A^{[12]}$ simply by contracting $A^{[1]}$ and $A^{[2]}$ over the last bond, without having to deal with any minus signs. It is easy to see that $P^{[12]}$ defined this way contains only terms with an even number of fermionic operators.

We point out that the reason why no minus signs appear in the multiplication of $A^{[1]}$ and $A^{[2]}$ is because we have contracted over the respective last bonds. One can show that contracting instead over the first bonds does not give rise to any minus signs either but contracting over a bond that is neither the first nor the last results in different signs depending on the values of ν and μ on the other bonds. It is advisable to always rearrange the order of the bonds before multiplying two fermionic tensors such that one does not have to deal with minus signs in the tensor multiplication. We also point out that in the multiplication $A^{[1]} \cdot A^{[2]}$, site 1 has to be in the sublattice \mathcal{A} and site 2 in the sublattice \mathcal{B} . Otherwise, the order of the creation operators in (3.32) would be reversed, which would give rise to a minus sign if ρ labels a fermionic state.

Fusing and unfusing bond spaces. We consider the case that we have to contract two fermionic tensors $A^{[1]}$ and $A^{[2]}$ over two bonds simultaneously (figure 3.9). Assume that we have already rearranged the order of the bonds of both tensors in such a way that the tensor contraction is over the first two bonds. We proceed as follows: First, we consider each tensor individually and fuse the first two bond spaces to one as described in section

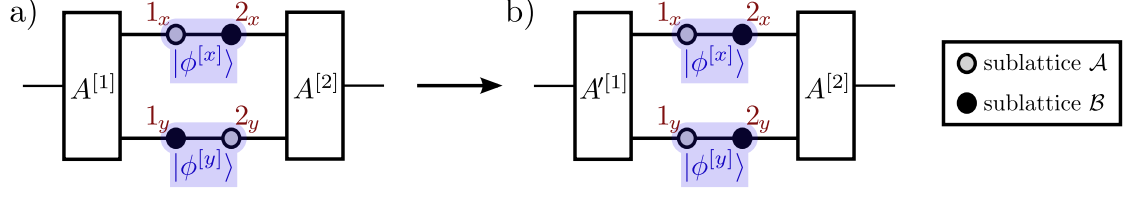


Figure 3.9.: When fusing two tensor bonds x and y , we also have to fuse the entangled states $|\phi^{[x]}\rangle$ and $|\phi^{[y]}\rangle$. This leads to minus signs if the auxiliary Hilbert spaces on the two bonds are associated to different sublattices as in (a). In the illustration, we swap the order of the auxiliary Hilbert spaces on the y -bond and incorporate the minus signs into the tensor $A^{[1]}$ (b).

3.1.3. This procedure creates tensors $\tilde{A}^{[1]}$ and $\tilde{A}^{[2]}$. Then, we multiply $\tilde{A}^{[1]}$ and $\tilde{A}^{[2]}$ over the first bond.

We have to multiply $\tilde{A}^{[1]}$ and $\tilde{A}^{[2]}$ in the correct order. If the auxiliary Hilbert spaces on the first two bonds of $A^{[1]}$ are both attributed to sites from the same sublattice (say, \mathcal{A}), then the correct order is $\tilde{A}^{[1]} \cdot \tilde{A}^{[2]}$. However, special care needs to be taken if $A^{[1]}$ and $A^{[2]}$ are already results of tensor multiplications. In this case, the auxiliary Hilbert spaces attributed to the bonds of $A^{[i]}$ do not all come from the same original lattice site. We label the auxiliary Hilbert spaces attributed to the first two bonds of $A^{[1]}$ by 1_x and 1_y and the corresponding auxiliary Hilbert spaces attributed to $A^{[2]}$ by 2_x and 2_y . Let us assume that $\mathcal{H}_{\text{aux}}^{[1_x]}$ was originally attributed to sublattice \mathcal{A} and $\mathcal{H}_{\text{aux}}^{[1_y]}$ to sublattice \mathcal{B} (figure 3.9a). Then, for the tensor multiplication to be well-defined, $\mathcal{H}_{\text{aux}}^{[2_x]}$ and $\mathcal{H}_{\text{aux}}^{[2_y]}$ have to originate from sublattices \mathcal{B} and \mathcal{A} , respectively. This situation appears e.g. in equations (2.32a) and (2.32b). In this case, the multiplication of $\tilde{A}^{[1]}$ and $\tilde{A}^{[2]}$ is not well defined. To overcome this problem, we reattribute $\mathcal{H}_{\text{aux}}^{[1_y]}$ to sublattice \mathcal{A} and $\mathcal{H}_{\text{aux}}^{[2_y]}$ to sublattice \mathcal{B} (figure 3.9b) by flipping the order of the creation operators in the corresponding entangled auxiliary state $|\phi^{[y]}\rangle$. Specifically, we write

$$\begin{aligned}
 |\phi^{[y]}\rangle &= \sum_{\nu^y=0,1} \sum_{\alpha_\nu^y} d_{\nu^y \alpha_\nu^y}^{[2_y] \dagger} d_{\nu^y \alpha_\nu^y}^{[1_y] \dagger} (|0_{\text{aux}}^{[2_y]}\rangle \otimes |0_{\text{aux}}^{[1_y]}\rangle) = \\
 &= \sum_{\nu^y=0,1} (-1)^{\nu^y} \sum_{\alpha_\nu^y} d_{\nu^y \alpha_\nu^y}^{[1_y] \dagger} d_{\nu^y \alpha_\nu^y}^{[2_y] \dagger} (|0_{\text{aux}}^{[1_y]}\rangle \otimes |0_{\text{aux}}^{[2_y]}\rangle) \quad . \quad (3.33)
 \end{aligned}$$

We incorporate the factor $(-1)^{\nu^y}$ into *one* of the tensors $A^{[1]}$ or $A^{[2]}$ before fusing the first two bonds. Then, the correct order for the multiplication of $\tilde{A}^{[1]}$ and $\tilde{A}^{[2]}$ is $\tilde{A}^{[1]} \cdot \tilde{A}^{[2]}$.

Factorization of a tensor. Consider the result $P^{[12]}$ of (3.31). To invert the multiplication of $P^{[1]}$ and $P^{[2]}$, we factorize the order six tensor $A^{[12]}$ into two tensors $\tilde{A}^{[1]}$ and $\tilde{A}^{[2]}$ of order four and a diagonal matrix Λ by a singular value decomposition as described in section 3.1.3. We multiply $\sqrt{\Lambda}$ to each resulting tensor. In (3.16) we defined the singular value decomposition of a symmetric tensor in such a way that the two resulting tensors are also symmetric. Since $A^{[12]}$ is symmetric under \mathbb{Z}_2 , so are $\tilde{A}^{[1]}$ and $\tilde{A}^{[2]}$. Therefore, $\tilde{A}^{[1]}$ and $\tilde{A}^{[2]}$ define two projectors $\tilde{P}^{[1]}$ and $\tilde{P}^{[2]}$ that contain only terms with an even

3. Extending the algorithm

number of fermionic operators and are defined such that $\tilde{P}^{[1]}\tilde{P}^{[2]}|\phi^{[12]}\rangle = P^{[12]}$. If the bond that connects $\tilde{A}^{[1]}$ and $\tilde{A}^{[2]}$ is not the first or the last bond of both tensors then the same signs as in the multiplication of two fermionic tensors have to be taken into account.

Applying a bond operator to a fermionic PEPS. We consider a fermionic PEPS $|\Psi\rangle$ as in (3.25) where the projectors satisfy condition (3.26) and therefore all commute with each other. For two neighboring sites r and s connected by a link l_{rs} , applying a generic bond operator \hat{O}_{rs} to $|\Psi\rangle$ reads

$$|\Psi'\rangle := \hat{O}_{rs}|\Psi\rangle = \underbrace{\left(\hat{O}_{rs}P^{[r]}P^{[s]}|\phi^{[l_{rs}]}\rangle\right)}_{=: \tilde{P}^{[rs]}} \left(\prod_{i \neq r,s} P^{[i]}\right) \left(\prod_{l \neq l_{rs}} |\phi^{[l]}\rangle\right) \quad (3.34)$$

where $\tilde{P}^{[rs]}$ is a projector from the product of the auxiliary Hilbert spaces attributed to sites r and s (except those on the link l_{rs}) to the product of the local Hilbert spaces on sites r and s . We require that \hat{O}_{rs} may change the total number of fermions by an even number only. Then, $\tilde{P}^{[rs]}$ contains only terms with an even number of fermionic operators. We can bring $|\Psi'\rangle$ back to the form of a fermionic PEPS by splitting $\tilde{P}^{[rs]}$ into two projectors $\tilde{P}^{[r]}$ and $\tilde{P}^{[s]}$ defined on the sites r and s as described in the preceding paragraph.

3.3. Rényi entropy and topological order

A central objective of condensed matter physics is the classification of different phases and phase transitions. In the Landau theory of spontaneous symmetry breaking [83], ordered phases are characterized by the emergence of a *local* order parameter. We can use SRG to check for the existence of a local order parameter, e.g. the magnetization in a ferromagnet, by applying a small external field h that breaks the spin-rotational symmetry of the Hamiltonian. The system is in an ordered phase if the susceptibility at $h = 0$ diverges (see section 5.1). However, Landau theory does not apply to all phases and phase transitions. In particular, topologically ordered phases [7], which appear for example in the fractional quantum Hall effect [6], cannot be characterized by a local parameter. The existence of topological order manifests itself in a number of non-local properties, such as degeneracy of the ground state, fractional statistics of excitations, the existence of edge states, and a correction to the area law of entanglement entropy introduced in section 1.1.

In a recent work by Meng et al. [75], quantum Monte Carlo calculations showed indications of a topologically ordered phase in the ground state of the Hubbard model on the honeycomb lattice. It is therefore intriguing to extend the TRG algorithm beyond the calculations of local observables in order to test for the existence of topological order.

We give an intuitive picture of topological order and the correction to the area law by means of an example system. Then, we define the topological entanglement Rényi entropy S_{tp} as a measure of topological order and describe a method to calculate S_{tp} for PEPS using a modified version of the TRG algorithm.

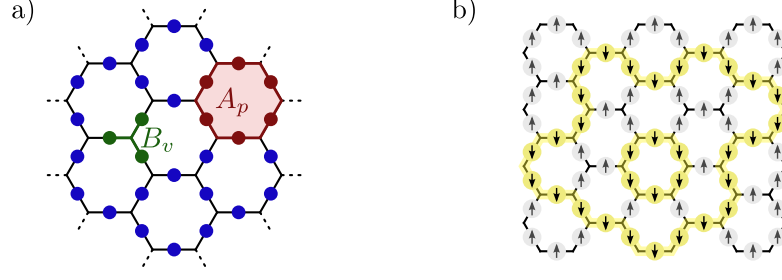


Figure 3.10.: a) Illustration of the plaquette terms A_p and the vertex terms B_v of the toric code (3.36). b) The ground state of the toric code contains only configurations where down spins lie along closed strings.

3.3.1. An example system with topological order

We consider the ground state of the toric code [84] on the honeycomb lattice as an example state with topological order. The system is defined on a honeycomb lattice where a spin- $\frac{1}{2}$ degree of freedom occupies each link. The Hamiltonian reads

$$H = - \sum_p A_p - \sum_v B_v \quad (3.35)$$

with

$$A_p = \prod_{i \in p} \sigma_i^x; \quad B_v = \prod_{j \in v} \sigma_j^z; \quad \sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}; \quad \sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad (3.36)$$

where p labels plaquettes (hexagons) and $\prod_{i \in p}$ runs over all links around a plaquette. Further, v labels vertices and $\prod_{j \in v}$ runs over all links connected to the vertex (figure 3.10a).

It is easy to see that $[A_p, A_{p'}] = [B_v, B_{v'}] = [A_p, B_v] = 0$ for all p, p', v, v' . The ground state is therefore an eigenstate of all A_p and B_v . Since $A_p^2 = B_v^2 = 1$, the highest eigenvalue of both operators is one. We can indeed construct a state $|\Psi_{\text{sn}}\rangle$ that is an eigenstate of all A_p and B_v with eigenvalue one and therefore the ground state of H . The vertex terms B_v ensure that $|\Psi_{\text{sn}}\rangle$ contains only configurations with either zero or two down spins on the links connected to each vertex. In a finite system \mathcal{S} , this means that down spins can only exist along closed strings around some region $\mathcal{R} \subset \mathcal{S}$ (figure 3.10b). Acting with A_p on all plaquettes inside \mathcal{R} annihilates (or creates) the string around \mathcal{R} but leaves the eigenstate $|\Psi_{\text{sn}}\rangle$ unchanged. Therefore, $|\Psi_{\text{sn}}\rangle$ is the equal weight superposition of all configurations of an arbitrary number of closed strings. We call this state a string-net condensate.⁵

The string-net condensate has an interesting non-local property. If we cut out any subsystem $\mathcal{S}' \subset \mathcal{S}$ as in figure 3.11a then all strings that enter on the boundary of \mathcal{S}' also have to leave the subsystem, since string ends or crossings are prohibited by the vertex terms B_v . Therefore, the number of down spins on the boundary of any subsystem is even. This is a non-local property that gives an intuitive picture of the topological order in the string-net condensate state and has a direct consequence to the scaling of

⁵See [85] for a classification of different string-net condensate states.

3. Extending the algorithm

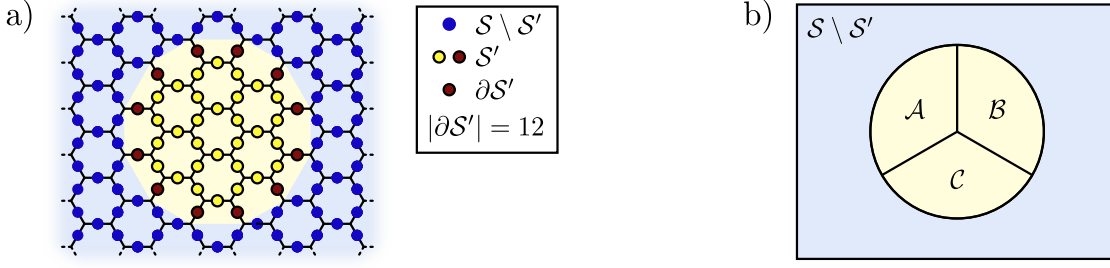


Figure 3.11.: a) Illustration of a subsystem $S' \subset S$ with boundary $\partial S'$. The string net condensate state contains only configurations with an even number of down spins on the boundary. b) Partitioning of the subsystem S' into three regions \mathcal{A} , \mathcal{B} , and \mathcal{C} for the definition of the topological entanglement Rényi entropy (3.40).

the entanglement entropy. We define the Rényi entanglement entropy of order n of a subsystem $S' \subset S$,

$$S_{S'}^{(n)} := \frac{1}{1-n} \log_2[\text{Tr}(\rho_{S'}^n)] \quad (3.37)$$

where

$$\rho_{S'} := \text{Tr}_{S \setminus S'} |\Psi_{\text{sn}}\rangle \langle \Psi_{\text{sn}}| = \sum_{\varphi_{S \setminus S'}} \langle \varphi_{S \setminus S'} | \Psi_{\text{sn}} \rangle \langle \Psi_{\text{sn}} | \varphi_{S \setminus S'} \rangle \quad (3.38)$$

is the reduced density matrix of the state $|\Psi_{\text{sn}}\rangle$ for the subsystem S' . The trace in (3.38) needs to be taken over an orthogonal basis $|\varphi_{S \setminus S'}\rangle$ of the Hilbert space of the complementary subsystem $S \setminus S'$. The Rényi entropy is a generalization of the von Neumann entropy and measures the entanglement between two subsystems S' and $S \setminus S'$. It converges to the von Neumann entropy in the limit $n \rightarrow 1$. In appendix A, we show that the Rényi entropy of $|\Psi_{\text{sn}}\rangle$ is

$$S_{S'}^{(n)} = |\partial S'| - 1 \quad \forall n \quad (3.39)$$

where $|\partial S'|$ is the number of spins on the boundary of S' (figure 3.11a). From the area law, one would expect the entanglement to scale proportionally to $|\partial S'|$. However, the effective size of the boundary is reduced by one spin due to the condition of an even number of down spins on the boundary. This gives rise to a constant correction to the area law.

Let us now partition the subsystem S' into three disjunct regions \mathcal{A} , \mathcal{B} , and \mathcal{C} . Consider the geometry illustrated in figure 3.11b. Analogous to the topological entanglement Entropy defined in [86] in terms of von Neumann entropies,⁶ we define the topological entanglement Rényi entropy of order n ,

$$S_{\text{tp}}^{(n)} := -S_{\mathcal{A}}^{(n)} - S_{\mathcal{B}}^{(n)} - S_{\mathcal{C}}^{(n)} + S_{\mathcal{A} \cup \mathcal{B}}^{(n)} + S_{\mathcal{B} \cup \mathcal{C}}^{(n)} + S_{\mathcal{A} \cup \mathcal{C}}^{(n)} - S_{\mathcal{A} \cup \mathcal{B} \cup \mathcal{C}}^{(n)} \quad (3.40)$$

⁶In [86], S_{tp} has been defined as the negative of our definition. We flip all signs to get positive topological entropies. This is in conformance with [38].

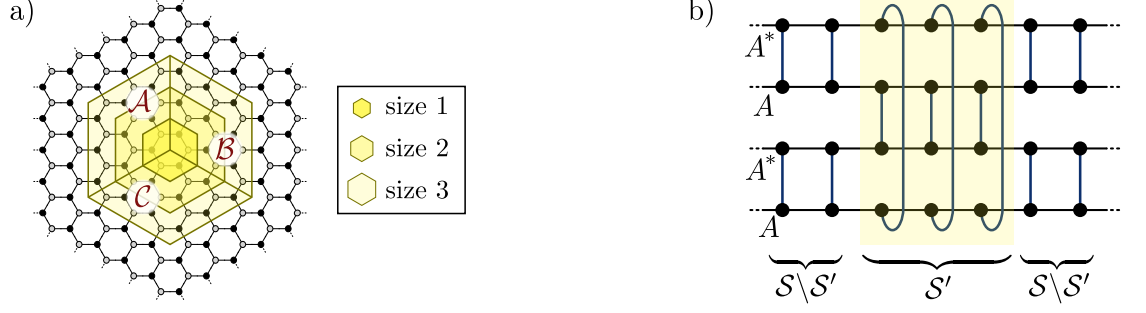


Figure 3.12.: a) Three subsystems of growing size, each partitioned into three disjunct sets \mathcal{A} , \mathcal{B} , and \mathcal{C} . b) Calculation of the Rényi entropy of order two for an MPS. The generalization to PEPS is straight forward.

It has been shown in [87] that $S_{tp}^{(n)}$ is a measurement of topological order. Any contribution to the Rényi entropy that is proportional to the size of the boundary cancels in (3.40). In case of the string net condensate state, only the constant correction survives and we have $S_{tp}^{(n)} = 1$, independent of the size of the subsystem \mathcal{S}' and the exact geometry of the partitioning into \mathcal{A} , \mathcal{B} , and \mathcal{C} . In other systems with topological order, the entanglement entropy can have more structure at short ranges and one has to evaluate $S_{tp}^{(n)}$ in the limit of large \mathcal{S}' .

3.3.2. Calculating Rényi entropies

We will use the string-net condensate state again in section 4.1.2. For now, let us return to states whose degrees of freedoms are defined on the vertices rather than the links of the honeycomb lattice and consider a translationally invariant PEPS defined by tensors $A_{x_i y_i z_i}^{\sigma_i}$ and $B_{x_j y_j z_j}^{\sigma_j}$. A modified version of the TRG algorithm that calculates the Rényi entropy of order two defined in (3.37) has been proposed in [38].

We choose one of the three subsystem sizes in figure 3.12a and calculate each of the seven Rényi entropies in (3.40) separately. We first express the Rényi entropy $S_{\mathcal{S}'}^{(2)}$ of a subsystem $\mathcal{S}' \subset \mathcal{S}$ as a tensor network. This procedure is illustrated in figure 3.12b for the one-dimensional case with MPS. The generalization to PEPS in two dimensions is straight-forward and defines a tensor network with tensors $\tilde{T}^a = T^a \otimes T^a$ and $\tilde{T}^b = T^b \otimes T^b$ on the sites in $\mathcal{S} \setminus \mathcal{S}'$ and different impurity tensors on the sites in \mathcal{S}' (figure 3.13a), i.e. the tensor network is not translationally invariant. To evaluate the product of tensors, one therefore has to modify the TRG algorithm. Figure 3.13 illustrates the first four renormalization steps for the calculation of $S_{AUC}^{(2)}$ with subsystem size three. All subsequent renormalization steps are similar to the fourth step and transform the tensor network to a configuration equivalent to that in figure 3.13d.

For a PEPS with bond dimension D , the tensors in figure 3.12b have bond dimension D^4 . Calculating accurate Rényi entropies is therefore much harder than calculating expectation values of local observables, where the bond dimension is only D^2 . Also, due to the complex geometry of the impurity tensors in figure 3.13, the environment of a tensor depends strongly on the position in the lattice, which makes it difficult to implement the improved renormalization step of the SRG algorithm.

3. Extending the algorithm

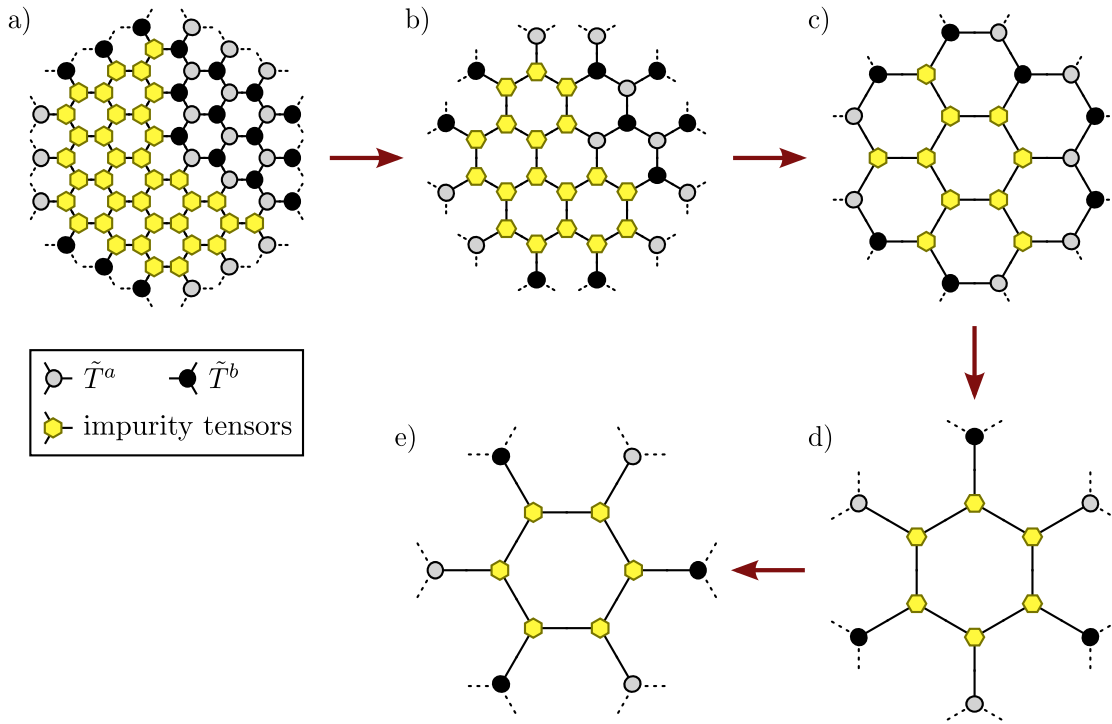


Figure 3.13.: First four TRG renormalization steps to calculate the Rényi entropy of an example subsystem geometry. All following steps are equivalent to the fourth step.

4. Results I: characterization of the algorithms

The algorithms described in chapter 2 depend on a number of non-physical parameters (see sections 2.1.3 and 2.2.3). We now analyze how the accuracy of numerical results depends on the choice of these parameters. As described in chapter 2, the algorithm for ground state properties of lattice Hamiltonians consists of two steps: the search of the ground state using imaginary time evolution and the calculation of observables using SRG. We first verify qualitatively that the SRG algorithm and its extensions described in chapter 3 produce correct results, independently from the imaginary time evolution. To do this, we introduce a number of analytically solvable test systems (section 4.1). We then turn to a quantitative characterization of both the imaginary time evolution and the SRG algorithm in section 4.2.

4.1. Qualitative verification of TRG & SRG

4.1.1. Classical statistical system

In section 2.2, we described the TRG method (and its improvement SRG) as an algorithm to approximately calculate local expectation values of a PEPS. At its heart, the algorithm is an efficient method to approximately calculate products of tensors of the form

$$\text{Tr} \prod_{\substack{i \in \mathcal{A} \\ j \in \mathcal{B}}} T_{x_i y_i z_i}^a T_{x_j y_j z_j}^b \quad (4.1)$$

where the tensors T^a and T^b are defined on the sites of the sublattices \mathcal{A} and \mathcal{B} of the honeycomb lattice, respectively, and the trace is to sum over connecting bonds (cf. (2.25)). In the original proposal of the TRG algorithm [1], it has been introduced to calculate partition functions of classical statistical lattice systems, which can also be expressed in the form of (4.1).

We test our implementations of TRG and SRG by calculating the partition function of the ferromagnetic Ising model on the triangular lattice. This model can be solved analytically [88]. We briefly explain how we express the partition function in the form of (4.1). For a more extensive explanation, see [39].

Expressing the partition function as a tensor product. We consider a classical spin $S_i \in \{-1, 1\}$ defined on each vertex i of the triangular lattice (figure 4.1a). The partition function of the Ising model is given by

$$Z = \text{Tr}_{\{S_i\}} e^{-\beta E} = \text{Tr}_{\{S_i\}} \prod_{\langle i, j \rangle} e^{\beta S_i S_j} = \text{Tr}_{\{S_i\}} \prod_{\Delta_{ijk}} e^{\beta(S_i S_j + S_j S_k + S_k S_i)/2} \quad (4.2)$$

4. Results I: characterization of the algorithms

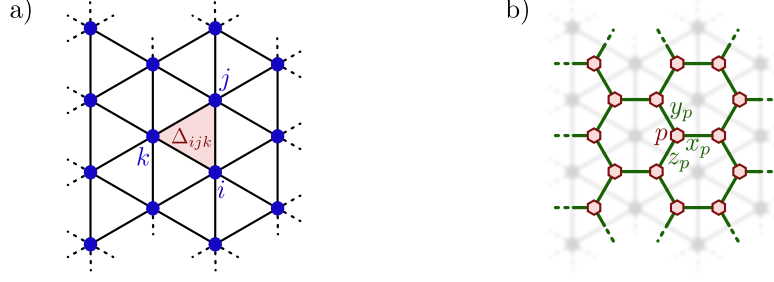


Figure 4.1.: a) The triangular lattice with vertices i, j, k and plaquette Δ_{ijk} . b) The dual of the triangular lattice is the honeycomb lattice. Each plaquette of the triangular lattice corresponds to a vertex of the honeycomb lattice.

where β is the inverse temperature, the trace is to sum over all spin configurations and the last product runs over all plaquettes (triangles). We label the plaquettes by p and define the domain wall configurations $x_p := S_i S_j$, $y_p := S_j S_k$ and $z_p := S_k S_i$, which take values -1 and 1 . We rewrite Z as

$$Z = \text{Tr}_{\{x_p, y_p, z_p\}} \prod_p e^{\beta(x_p + y_p + z_p)/2} \quad (4.3)$$

where the trace is to sum over all domain wall configurations x_p, y_p, z_p with the constraint that neighboring plaquettes have the same domain wall configuration on the shared domain wall and

$$x_p y_p z_p = S_i^2 S_j^2 S_k^2 = 1 \quad \forall p. \quad (4.4)$$

The dual lattice is defined by setting a vertex on each plaquette of the original lattice and connecting vertices corresponding to neighboring original plaquettes. This procedure transforms the triangular lattice to the honeycomb lattice (figure 4.1b). We define a tensor on each site p of the dual lattice,

$$T_{x_p y_p z_p} := \begin{cases} e^{\beta(x_p + y_p + z_p)/2} & \text{if } x_p y_p z_p = 1 \\ 0 & \text{else.} \end{cases} \quad (4.5)$$

The partition function can then be expressed as a product of tensors,

$$Z = \text{Tr}_{\{x_p, y_p, z_p\}} \prod_p T_{x_p y_p z_p} \quad (4.6)$$

which is of the same form as (4.1) if we define $T^a := T^b := T$.

Results. We calculate the free energy $F = \log(Z)$ for $3^{30} \approx 10^{14}$ sites using both TRG and SRG, and present the relative errors compared to the analytical solution in figure 4.2. The cut-off bond dimension is $D_{\text{cut}} = 30$ in both cases and the environment matrices for the SRG were calculated using $N^{\text{env}} = 3$ renormalization steps. Note that the elements of the tensor defined in (4.5) vanish if an odd number of x_p, y_p, z_p are negative. This is a \mathbb{Z}_2 symmetry of the tensor, which we have exploited to speed up the calculations.

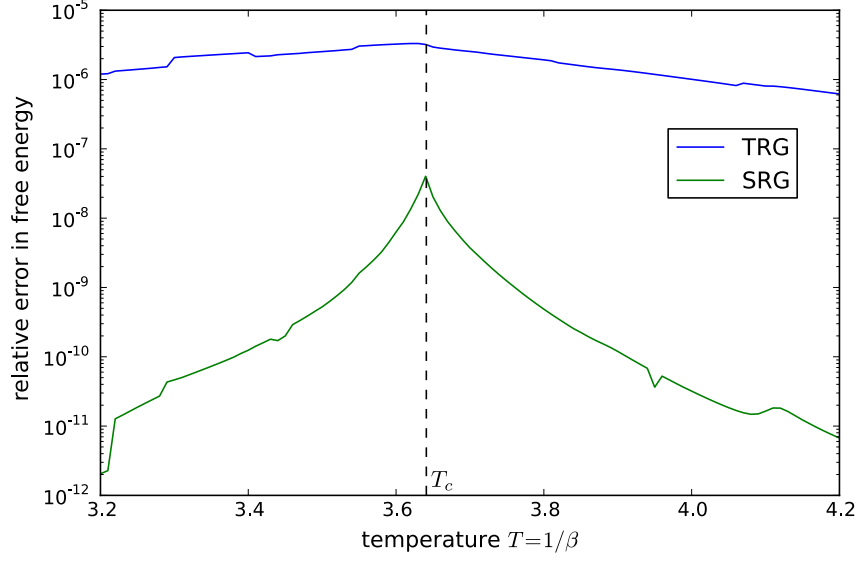


Figure 4.2.: Error of the free energy for the Ising model on the triangular lattice for calculations using TRG and SRG with $D_{\text{cut}} = 30$. The dashed line indicates the critical temperature $T_c = 4/\log(3) \approx 3.64$. The SRG error is the largest close to the transition, as the correlation length exceeds the size of the considered environment. Far from T_c , the consideration of the environment in the SRG algorithm significantly reduces the error in comparison to the simpler TRG algorithm.

The consideration of the environment in the SRG algorithm increases the accuracy by about two orders of magnitude in the vicinity of the critical temperature T_c , and even more further from the transition. We observe that the error of the SRG has a peak at T_c , while the error of the TRG is approximately independent of the temperature. Our interpretation is that near the critical temperature, the correlation length becomes larger than the size of the environment considered in the truncation step of the SRG. Therefore, the SRG uses a poor estimate of the environment matrix if $T \approx T_c$.

4.1.2. String-net condensate and topological entropy

Test wave functions. We use the string-net condensate state $|\Psi_{\text{sn}}\rangle$ described in section 3.3.1 to test our implementation of the calculation of Rényi entropies (see section 3.3.2). The state is defined on a system where a spin- $\frac{1}{2}$ degree of freedom occupies each link of the honeycomb lattice (figure 4.3a). We transform $|\Psi_{\text{sn}}\rangle$ to a new state defined on a system where two spins occupy each link. This is done by acting on each original spin with the transformation

$$U = |\uparrow\uparrow\rangle\langle\uparrow| + |\downarrow\downarrow\rangle\langle\downarrow|. \quad (4.7)$$

U is a local isometry from the original local Hilbert space of one spin to the subspace spanned by $\{|\uparrow\uparrow\rangle, |\downarrow\downarrow\rangle\}$ within the new local Hilbert space of two spins. It therefore does

4. Results I: characterization of the algorithms

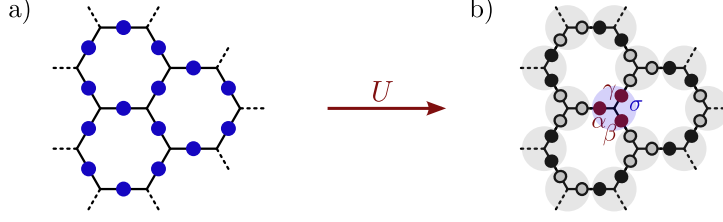


Figure 4.3.: We double the spins of the string net condensate state and then group three spins α, β, γ around a vertex to a single degree of freedom σ .

not change the entanglement entropy between two subsystems. We distribute the two spins on each link among the two connected vertices and group the three spins around a vertex i to a single degree of freedom labeled by a tuple $\sigma_i = (\alpha_i \beta_i \gamma_i)$ (figure 4.3b). It is easy to see that the resulting state $|\Psi'_{\text{sn}}\rangle$ can be expressed as a PEPS (cf. [38]),

$$|\Psi'_{\text{sn}}\rangle = \text{Tr} \left\{ \prod_{i \in \mathcal{A}, j \in \mathcal{B}} A_{x_i y_i z_i}^{\sigma_i} B_{x_j y_j z_j}^{\sigma_j} |\sigma_i \sigma_j\rangle \right\} \quad (4.8)$$

where $x, y, z \in \{\uparrow, \downarrow\}$ and the tensors are defined by

$$A_{xyz}^{(\alpha\beta\gamma)} = B_{xyz}^{(\alpha\beta\gamma)} = \begin{cases} \delta_{\alpha,x} \delta_{\beta,y} \delta_{\gamma,z} & \text{if an even number of } x, y, z \text{ are down spins} \\ 0 & \text{else.} \end{cases} \quad (4.9)$$

We refer to the condition of an even number of x, y, z pointing down as the condition of string conservation on vertices. It ensures that $|\Psi'_{\text{sn}}\rangle$ contains only configurations where the down spins lie on closed strings that do not cross or end anywhere on the lattice.

In [38], different kinds of perturbations of the tensors A and B in (4.9) were studied. It was shown that the state remains topologically ordered for small perturbations of the tensors as long as the condition of string conservation on vertices is maintained. On the other hand, topological order breaks down immediately for any perturbations of the tensors that violates this condition.

Results. We use TRG to calculate the topological Rényi entanglement entropy $S_{\text{tp}}^{(2)}$ defined in (3.40) for the string-net condensate state $|\Psi'_{\text{sn}}\rangle$ defined in (4.8)-(4.9). Even for a very low cut-off bond dimension $D_{\text{cut}} = 10$, the numerical results agree up to nine digits with the analytical value of $S_{\text{tp}}^{(2)} = 1$ for all three subsystem sizes illustrated in figure 3.12a.

We also calculate S_{tp} for 100 translationally invariant PEPS whose tensors A and B are chosen randomly in the neighborhood of the one defined in (4.9). For one half of these test states, we allow only perturbations that do not violate the condition of string conservation on vertices. For the other half, we allow arbitrary (but small) perturbations. In both cases, we restrict the perturbations to the region $\pm 0.1 \pm 0.1i$ around the unperturbed values. We use TRG with $D_{\text{cut}} = 60$ to calculate S_{tp} .

We present results in figure 4.4. Both kinds of perturbations create small variations in the short-range entanglement spectrum. With growing size of the subsystem, the

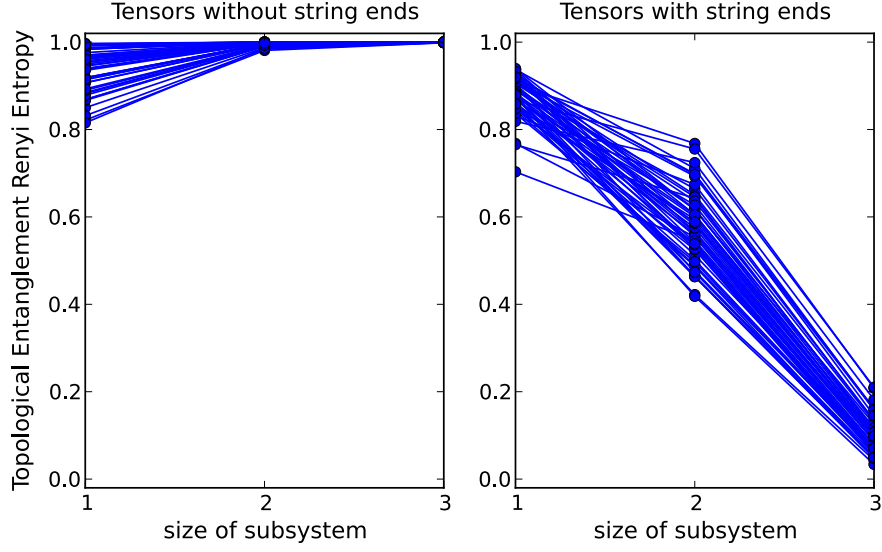


Figure 4.4.: Topological entanglement Rényi entropy calculated using TRG for PEPS whose tensors were chosen randomly in the neighborhood of the string-net condensate state (4.9). Left: perturbations that satisfy string conservation on vertices. Right: arbitrary perturbations. The subsystem sizes correspond to the three geometries shown in figure 3.12a.

topological entropy approaches one in the case of string conservation on vertices and zero in the case of arbitrary perturbations. This agrees with the theoretical predictions.

4.1.3. Fermion pairing states

Test wave functions. To test our implementation of fermionic PEPS (see section 3.2), we calculate local expectation values in the two-dimensional fermion pairing state introduced in [3]. The system consists of fermions that are located on the vertices i of a honeycomb lattice and are created by c_i^\dagger . For $u \geq 0$, the state is defined by

$$|\Psi_u^{2D}\rangle := \exp\left(u \sum_{\langle i,j \rangle} c_i^\dagger c_j^\dagger\right) |0\rangle = \prod_{\langle i,j \rangle} \left(1 + u c_i^\dagger c_j^\dagger\right) |0\rangle, \quad (4.10)$$

where $|0\rangle$ is the vacuum and $\langle i,j \rangle$ denotes pairs of nearest neighboring lattice sites with $i \in \mathcal{A}$ and $j \in \mathcal{B}$, where \mathcal{A} and \mathcal{B} denote the two sublattices. It is easy to see that $|\Psi_u^{2D}\rangle$ can be written as a fermionic PEPS in the form of (3.25) with

$$P^{[i]} = \langle 0_{\text{aux}}^{[i]} | \left(d_{0,1}^{[i_x]} d_{0,1}^{[i_y]} d_{0,1}^{[i_z]} + \sqrt{u} c_i^\dagger \left(d_{1,1}^{[i_x]} d_{0,1}^{[i_y]} d_{0,1}^{[i_z]} + d_{0,1}^{[i_x]} d_{1,1}^{[i_y]} d_{0,1}^{[i_z]} + d_{0,1}^{[i_x]} d_{0,1}^{[i_y]} d_{1,1}^{[i_z]} \right) \right) |0^{[i]}\rangle, \quad (4.11)$$

where $d_{0,1}$ annihilate bosonic and $d_{1,1}$ annihilate fermionic virtual particles.

An interpretation of the state $|\Psi_u^{2D}\rangle$ has been given in [3], where it has been shown that, up to a particle-hole conjugation on the \mathcal{B} sublattice, $|\Psi_u^{2D}\rangle$ is the ground state of

4. Results I: characterization of the algorithms

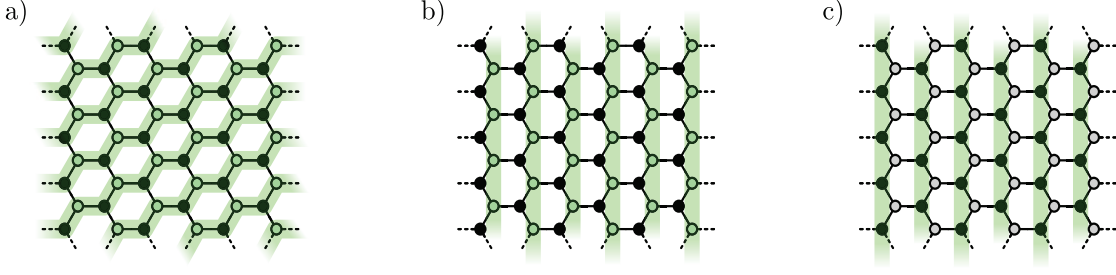


Figure 4.5.: Three ways to partition the honeycomb lattice into parallel chains. In the cases b and c, only one sublattice contains particles.

the Hamiltonian

$$H = - \sum_{\langle i,j \rangle} (c_i^\dagger c_j + h.c.) - \sum_{\langle\langle i,i' \rangle\rangle} (-)^i \frac{u}{2} c_i^\dagger c_{i'} + \sum_i (-)^i \frac{1-3u^2}{2u} c_i^\dagger c_i \quad (4.12)$$

where, in the first sum, again $i \in \mathcal{A}$ and $j \in \mathcal{B}$. The second sum runs over all ordered pairs of next-to-nearest neighbors and $(-)^i$ is 1 if $i \in \mathcal{A}$ and -1 if $i \in \mathcal{B}$. The Hamiltonian describes the dynamics of non-interacting fermions in a staggered external potential of magnitude $(1-3u^2)/(2u)$ that can tunnel to neighboring sites with tunneling amplitude one and to next-to-nearest neighboring sites with an amplitude of $(-)^i u/2$.

The state $|\Psi_u^{2D}\rangle$ is invariant under a rotation of the lattice by 120° around a vertex. In the implementation of the SRG algorithm for fermionic tensors, many conceivable sign errors would cancel out for rotationally invariant wave functions. We therefore introduce an additional set of test wave functions that are not rotationally invariant. We partition the honeycomb lattice into parallel chains in either of the ways illustrated in figure 4.5. On each chain, we relabel the sites by $l \in \mathbb{Z}$ and define the one-dimensional pairing state

$$|\Psi_u^{1D}\rangle = \exp \left(u \sum_{l \in \mathbb{Z}} \left(c_{2l}^\dagger c_{2l+1}^\dagger + c_{2l}^\dagger c_{2l-1}^\dagger \right) \right) |0\rangle. \quad (4.13)$$

A translationally invariant but non rotationally invariant test wave function on the honeycomb lattice is then given by the product of $|\Psi_u^{1D}\rangle$ defined on each chain.

We derive analytic expressions for local expectation values and for the scalar product of two different pairing states for $|\Psi_u^{1D}\rangle$ and $|\Psi_u^{2D}\rangle$ in Appendix B.

Results. We calculate the expectation value of the particle number operator $n_i = c_i^\dagger c_i$ and the nearest neighbor pairing amplitude $\langle c_i^\dagger c_j^\dagger \rangle$ with $i \in \mathcal{A}$ and $j \in \mathcal{B}$ (the pairing amplitude is real for real u and equal to the expectation value of the nearest-neighbor hopping term in the particle-hole conjugated system (4.12)). We use the SRG algorithm with $D_{\text{cut}} = 40$. The results agree well with the analytical solution (figure 4.6).

We also use SRG to calculate the overlap of two different pairing states. For a system of N sites, we define the infidelity per site as:

$$\bar{f}_{\Delta u}(u) := 1 - \left(\frac{|\langle \Psi_{u-\Delta u/2} | \Psi_{u+\Delta u/2} \rangle|}{\sqrt{\langle \Psi_{u-\Delta u/2} | \Psi_{u-\Delta u/2} \rangle \cdot \langle \Psi_{u+\Delta u/2} | \Psi_{u+\Delta u/2} \rangle}} \right)^{1/N}. \quad (4.14)$$

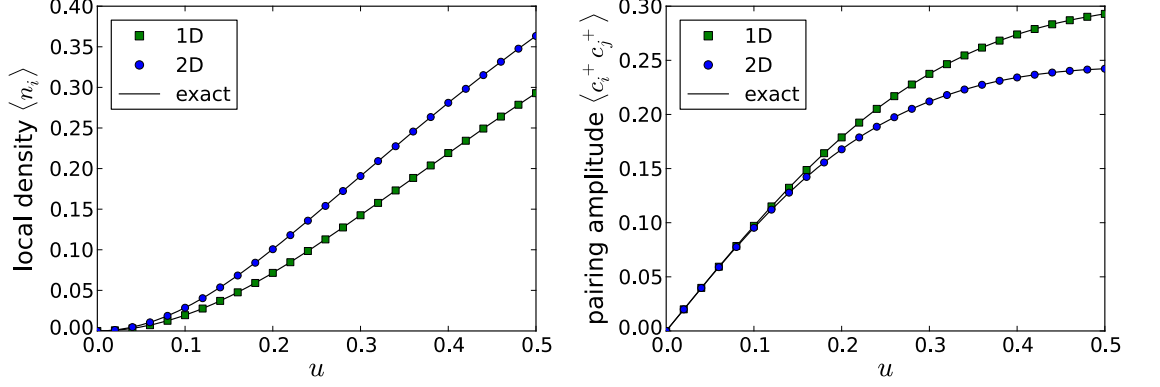


Figure 4.6.: Expectation value of the particle number (left) and the nearest neighbor pairing amplitude ($\langle c_i^\dagger c_j^\dagger \rangle$, right) for the pairing state in one and two dimensions. We compare results obtained by SRG to the exact values and find good agreement. In the one-dimensional case, numerical results for the different geometries illustrated in figure 4.5 would be indistinguishable at this resolution.

In section 5.2.2, we analyze \bar{f} for the Hubbard model to look for indications of a phase transition. Here, we only verify the correctness of our implementation. We use SRG to calculate the scalar product in the enumerator of (4.14) in a similar way as we usually calculate the norm of a wave function in order to normalize expectation values. The results for $\Delta u = 0.02$ agree well with the analytical values (figure 4.7) .

4.2. Quantitative analysis of the parameter space

4.2.1. SRG: N , N^{env} , and D_{cut}

We analyze the parameter space of the SRG before we turn to the imaginary time evolution in the next section because we can only assess the quality of wave functions produced by the imaginary time evolution by calculating expectation values in these wave functions. Therefore, we first need to know how reliable the calculation of expectation values is. We use the 2D fermion pairing state defined in (4.10) as a test wave function. The fermion pairing state is a PEPS with a bond dimension of only $D = 2$ (4.11). It can therefore be only weakly entangled. To obtain a second, more meaningful test case, we also generate approximate ground state wave functions of the spin- $\frac{1}{2}$ Heisenberg model (2.1) using imaginary time evolution with both plain (non-symmetric) tensors and with $U(1)$ -symmetric tensors that conserve magnetization in one spatial direction.

System size (N). In each renormalization step, the SRG algorithm transforms the honeycomb lattice to a similar lattice of three times the original size. After N renormalization steps, the algorithm has calculated the expectation value in a system with $2 \cdot 3^N$ sites. We calculate the pairing amplitude $\langle c_i^\dagger c_j^\dagger \rangle$ of the 2D fermion pairing state (4.10) with $u = 0.5$ and compare the result after each renormalization step to the exact value in the thermodynamic limit (figure 4.8a). The expectation value converges after four

4. Results I: characterization of the algorithms

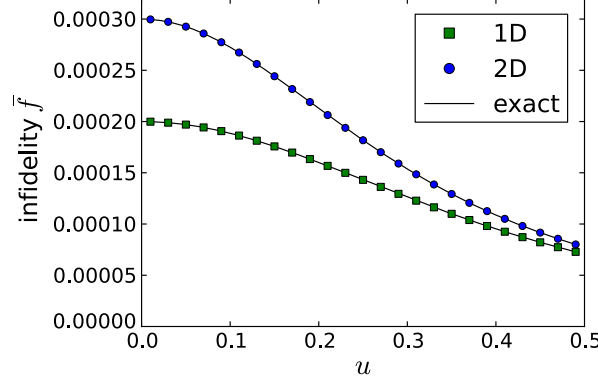


Figure 4.7.: Infidelity $\bar{f}_{\Delta u}(u)$ as defined in (4.14) with $\Delta u = 0.02$ for the fermion pairing states. The results obtained by SRG agree well with the exact values. In the one-dimensional case, numerical results for the different geometries illustrated in figure 4.5 would be indistinguishable at this resolution.

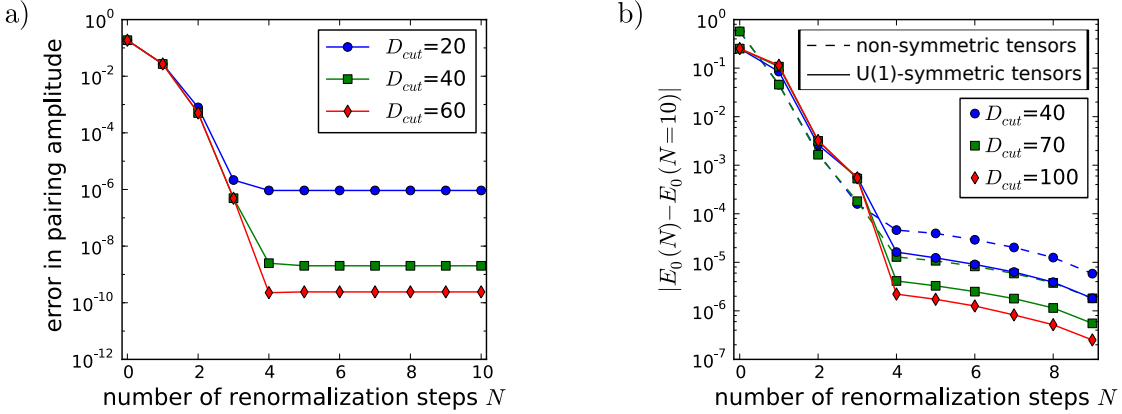


Figure 4.8.: Dependency on the system size. a) Error (compared to the exact value in the thermodynamic limit) of the pairing amplitude in the 2D fermion pairing state (4.10) after N renormalization steps. b) Convergence of the renormalization for the calculation of the ground state energy of the Heisenberg model. The bond dimension of the ground state wave functions is $D = 9$.

renormalization steps. Figure 4.8b shows the corresponding data for the ground state Energy of the Heisenberg model. Since this model cannot be exactly solved, we compare the energy after each renormalization step to the after for $N = 10$ renormalization steps. The renormalization approaches its end-value quickly in the first four steps and then changes only slowly. It is unclear whether the drift for large N is a physical effect due to growing system size or whether this is owing to the accumulation of truncation errors.

If not explicitly stated otherwise, we obtained all results that we present in this thesis with $N = 10$.

Size of environment (N^{env}). We demonstrated in section 4.1.1 (figure 4.2) that the consideration of the environment matrix in the renormalization step of the SRG algorithm improves the accuracy compared to the simpler renormalization step use in the TRG algorithm. We calculate the environment matrix by performing N^{env} TRG steps, which leads to an environment size of $6 \cdot 3^{N^{\text{env}}} - 2$ sites. We generally expect the SRG algorithm to produce more accurate results for growing environment size. However, the renormalization step of the SRG algorithm contains an inversion of the environment matrix (2.39). For very large N^{env} , one might expect the environment matrix to contain very large and very small singular values at the same time, which would make its inversion numerically unstable.

We calculate the local density $\langle n_i \rangle$ and the pairing amplitude $\langle c_i^\dagger c_j^\dagger \rangle$ of the 2D fermion pairing state defined in (4.10) with $u = 0.5$ for different environment sizes (figure 4.9). The SRG algorithm is unreliable for $N^{\text{env}} = 1$ but produces more accurate results than TRG as soon as $N^{\text{env}} \geq 2$, independent of the cutoff bond dimension D_{cut} . A larger value of N^{env} does not improve the accuracy but we also do not run into numerical instabilities for large N^{env} .

If not explicitly stated otherwise, we obtained all results that we present in this thesis with $N^{\text{env}} = 3$.

Cut-off bond dimension (D_{cut}). We present the D_{cut} -dependency of the result for the pairing amplitude $\langle c_i^\dagger c_j^\dagger \rangle$ of the 2D fermion pairing state (4.10) in figure 4.9. The stepwise convergence points to a degenerate singular value spectrum of the product of two tensors in the renormalization. Figure 4.10 shows the D_{cut} -dependency of the ground state energy of the Heisenberg model. With our available resources, we can reach convergence for $D = 9$ only if we exploit the $U(1)$ -symmetry.

As demonstrated, the D_{cut} -dependency varies strongly from system to system and one has to verify that one uses high enough D_{cut} for each system to which one applies the SRG algorithm individually. We investigated how the accuracy of calculated expectation values relates to the magnitude of the singular values that are neglected in each renormalization step but did not yet find a coherent picture. We regard the lack of an error estimate for results obtained by SRG as one of the major disadvantages of the algorithm.

4. Results I: characterization of the algorithms

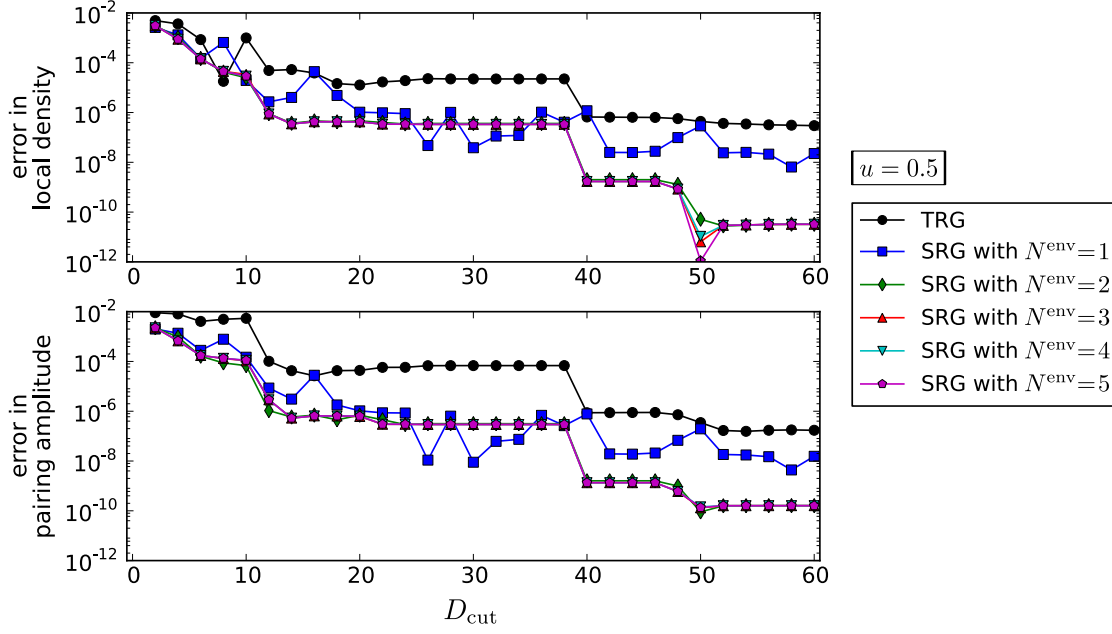


Figure 4.9.: Accuracy of local observables calculated for the 2D fermion pairing state (4.10) at $u = 0.5$ using TRG and SRG with different environment sizes. Independently of the cutoff bond dimension D_{cut} , the SRG algorithm converges w.r.t. N^{env} as soon as $N^{\text{env}} \geq 2$ and shows no numerical instability for large N^{env} . The stepwise convergence w.r.t. D_{cut} points to a degenerate singular value spectrum of the product of two tensors.

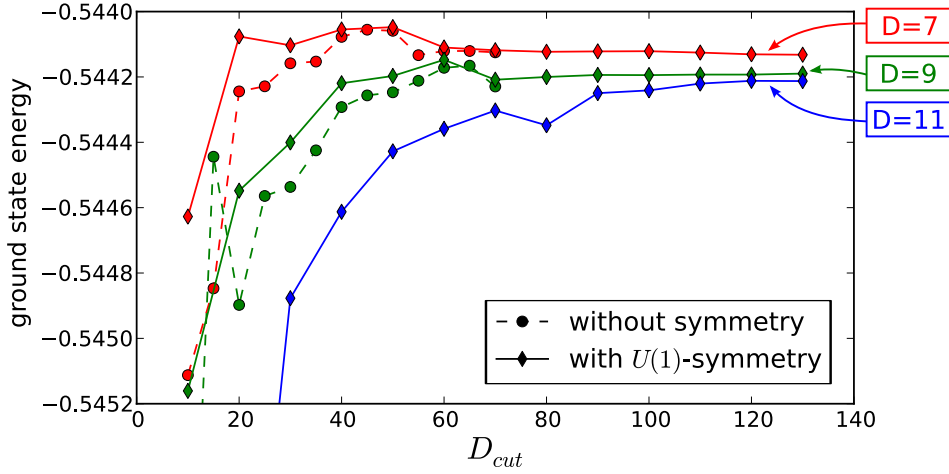


Figure 4.10.: Calculated ground state energy for the spin- $\frac{1}{2}$ Heisenberg model for wave functions with bond dimensions $D = 7, 9, 11$. Exploiting symmetries allows us to reach higher D , as well as higher D_{cut} . For growing bond dimension D of the wave function, one needs higher D_{cut} to reach convergence of the calculated expectation value.

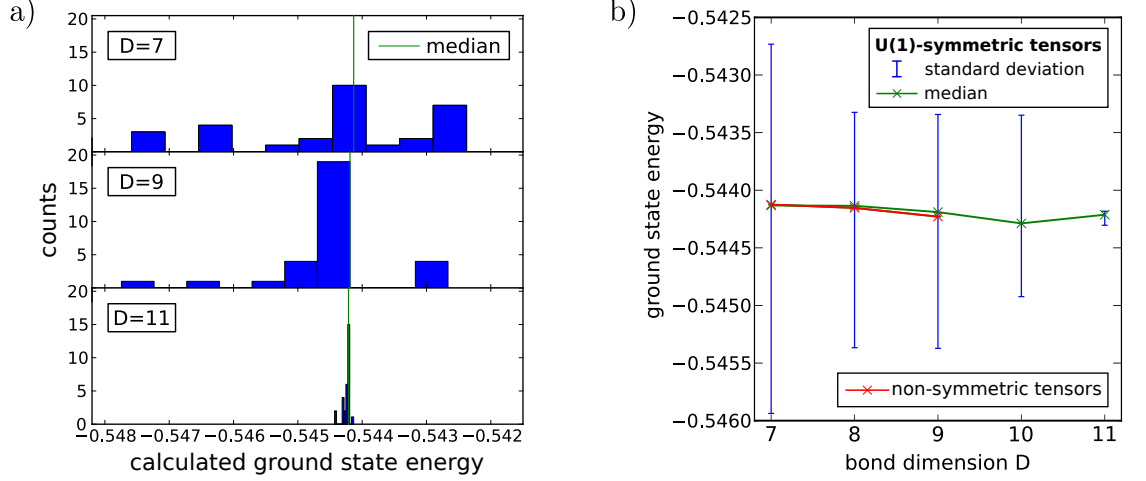


Figure 4.11.: Sensitivity of the imaginary time evolution to the start wave function. For each bond dimension D , we apply the imaginary time evolution algorithm to 30 random start wave functions and then calculate the energy expectation value using SRG. a) Histograms of the energy when using $U(1)$ -symmetric tensors. For small D , the imaginary time evolution algorithm tends to get stuck in local energy minima. This problem is less pronounced for higher bond dimensions. b) Comparison of results with $U(1)$ -symmetric tensors to results with plain (non-symmetric) tensors. The error bars for the calculations with non-symmetric tensors are too small to be seen at this resolution. Parameters for the SRG are $D_{\text{cut}} = 70$ for non-symmetric tensors and $D_{\text{cut}} = 130$ for $U(1)$ -symmetric tensors.

4.2.2. Imaginary time evolution: $|\Psi(\tau = 0)\rangle$, D , $\Delta\tau$, and τ

Start wave function ($|\Psi(\tau = 0)\rangle$) and bond dimension (D). In the imaginary time evolution algorithm, we project a random start wave function $|\Psi(\tau = 0)\rangle$ onto the ground state. We do this by following a trajectory $|\Psi(\tau)\rangle$ through the Hilbert space. The trajectory converges to the ground state unless $|\Psi(\tau = 0)\rangle$ is orthogonal to it, which happens with vanishing probability for a randomly chosen state. However, we go through the trajectory in small steps and perform an approximation after each step: We first translate $|\Psi(\tau)\rangle$ in imaginary time to $|\Psi(\tau + \Delta\tau)\rangle$. Then, we truncate the bond dimension of $|\Psi(\tau + \Delta\tau)\rangle$. Due to this approximation after each step, the imaginary time evolution algorithm is not an exact projective method and therefore can get stuck in a local energy minimum.

To check whether the imaginary time evolution is vulnerable to getting stuck in a local minimum, we test the dependency on the start wave function for the spin- $\frac{1}{2}$ Heisenberg model. We first test the algorithm without exploiting any symmetries. For each of the bond dimensions $D = 7, 8, 9$, we generate 30 random start wave functions. We perform an imaginary time evolution until convergence for each start wave function and then use SRG to calculate the energy expectation value. We present results in figure 4.11b (red line). The values for the 30 different start wave functions lie within an interval of width

4. Results I: characterization of the algorithms

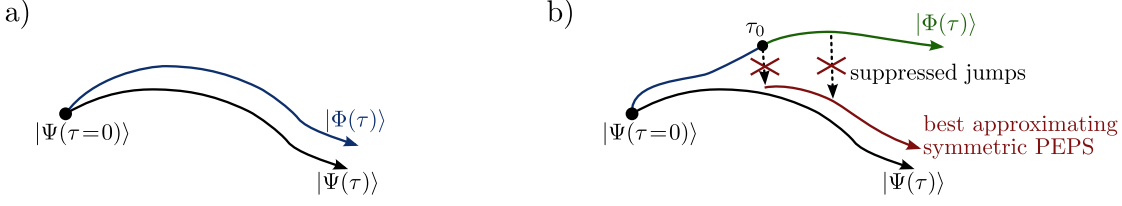


Figure 4.12.: Imaginary time evolution. a) Due to the truncation after each imaginary time step, the wave function $|\Phi(\tau)\rangle$ follows the exact trajectory $|\Psi(\tau)\rangle$ of the imaginary time evolution only approximately. b) Imaginary time evolution with PEPS composed of symmetric tensors. At the imaginary time τ_0 , the exact trajectory $|\Psi(\tau)\rangle$ would be better approximated if the partitioning of the bond spaces into symmetry sectors changes for the tensors that define $|\Phi(\tau)\rangle$. This is however suppressed for small imaginary time steps $\Delta\tau$.

2×10^{-7} or less for all D and would be indistinguishable in the graph. As the calculated energies agree well with results obtained by other methods (see section 5.1), there is no indication that the imaginary time evolution with non-symmetric tensors is vulnerable to getting stuck in a local energy minimum.

A different picture emerges for calculations with $U(1)$ -symmetric tensors. For each $D = 7, \dots, 11$, we perform imaginary time evolution for 30 random start wave functions and ensure convergence of the imaginary time evolution by the method described in the subsequent paragraph. We then use SRG to calculate the energy expectation value of each produced wave function. For small bond dimensions D , the resulting energy spectra (figure 4.11a) show a number of results that lie far off from the main peak. This is a clear sign that, for small D , the imaginary time evolution with symmetric tensors sometimes converges to states that are not a good approximation of the ground state. We present the standard deviation of the 30 calculated ground state energies for each bond dimension in figure 4.11b. The spreading narrows down for growing D .

A possible explanation why the imaginary time evolution might be more vulnerable to getting stuck in a local minimum for symmetric PEPS than for non-symmetric PEPS goes as follows: Due to the truncation after each imaginary time step, the algorithm does not exactly follow the trajectory $|\Psi(\tau)\rangle$ but rather an approximate trajectory $|\Phi(\tau)\rangle$ of states that can be represented by PEPS with fixed bond dimension (figure 4.12). In each truncation, $|\Phi(\tau)\rangle$ goes a small step away from $|\Psi(\tau)\rangle$. However, since the imaginary time evolution operator $e^{-\Delta\tau H}$ always produces a flow in the direction of the ground state, $|\Phi(\tau)\rangle$ is pushed back towards $|\Psi(\tau)\rangle$ in the next imaginary time step, so that the trajectories stay close to each other (figure 4.12a). Now, if $|\Phi(\tau)\rangle$ is a *symmetric* PEPS, then the bond spaces of its tensors are partitioned into symmetry sectors. In some imaginary time step from $\tau = \tau_0$ to $\tau = \tau_0 + \Delta\tau$, the symmetric PEPS that approximate $|\Psi(\tau_0)\rangle$ and $|\Psi(\tau_0 + \Delta\tau)\rangle$ the best might have differently partitioned bond spaces. This means that $|\Phi(\tau)\rangle$ must change the partitioning of its bond spaces if it is to stay close to $|\Psi(\tau)\rangle$ (figure 4.12b, red trajectory). Even if this corresponds to only a small change in the Hilbert space of states, it is a non-continuous jump in the tensor space and therefore exponentially suppressed for small $\Delta\tau$. Thus, it can happen that $|\Phi(\tau)\rangle$ gets stuck in

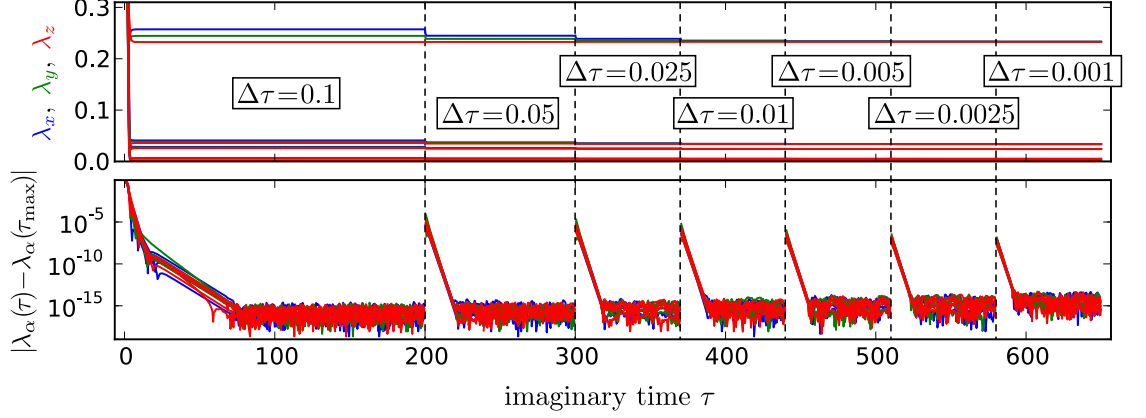


Figure 4.13.: Convergence of the imaginary time evolution with $U(1)$ -symmetric tensors of bond-dimension $D = 11$ for the Heisenberg model. We start imaginary time evolution with step size $\Delta\tau = 0.1$ and decrease the step size until $\Delta\tau = 0.001$. For each $\Delta\tau$, we perform imaginary time evolution until the wave function converges. Top: Values of the bond vectors of the wave function (2.12). Bottom: Distance of the values of the bond vectors to the corresponding values at the largest imaginary time (dashed lines), for each $\Delta\tau$. The bond vectors are clearly converged for each $\Delta\tau$.

wrongly partitioned symmetry sectors (figure 4.12b, green trajectory). For growing D , the distribution of bond dimensions to the symmetry sectors can be more generous and the partitioning into symmetry sectors is less stiff. Therefore, symmetric PEPS with higher bond dimensions are less likely to get stuck in a local energy minimum.

We experimented with differently partitioned bond spaces for the start wave function but did not find a reliable best choice. For further calculations with the Heisenberg model, we have to pick one of the 30 wave functions for each $D = 7, \dots, 11$. It would be tempting to use the wave function with the lowest energy for each D . However, we observe that for wave functions that lie far off from the main peak in figure 4.11a, the calculation of the energy expectation value (SRG) tends to converge worse and the truncation error of the SRG tends to be higher than for the other wave functions. It is therefore likely that the wave functions that correspond to far-off lying values in figure 4.11a are in reality poor approximations of the ground state and that the calculated energy expectation values for these wave functions are simply wrong. Rather than picking the wave function with the lowest energy for each D , we exclude outliers in the energy spectrum by taking the median. For $D \leq 9$, these energies also agree well with the results from the unbiased calculations with non-symmetric tensors (figure 4.11b).

We regard the sensitivity to the start wave function for small bond dimensions as a major problem of the imaginary time evolution algorithm with symmetric tensors. As long as the cause of this effect is not better understood, the only solution is to use high enough bond dimensions.

4. Results I: characterization of the algorithms

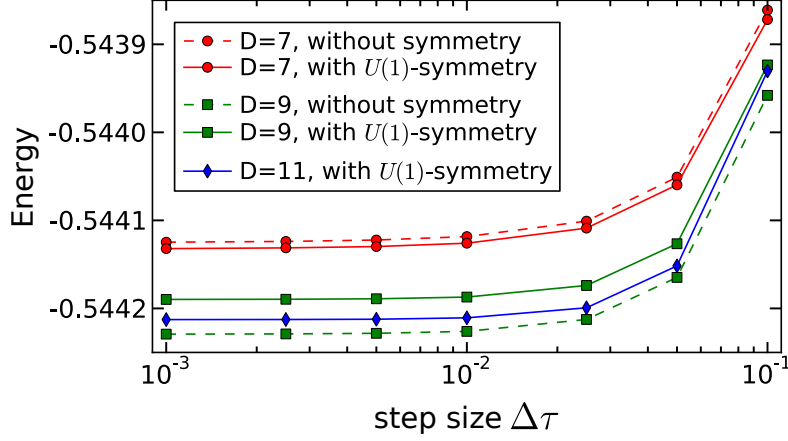


Figure 4.14.: Convergence of the imaginary time evolution algorithm w.r.t. the step size $\Delta\tau$. We perform imaginary time evolution for the Heisenberg model with step size $\Delta\tau$ until the wave function converges and then calculate the energy expectation value using SRG. Decreasing $\Delta\tau$ leads to a better approximation of the ground state. Parameters for the SRG are $D_{\text{cut}} = 70$ for non-symmetric tensors and $D_{\text{cut}} = 130$ for $U(1)$ -symmetric tensors.

Step size ($\Delta\tau$) and imaginary time (τ). We use second order Trotter-Suzuki decomposition (2.11) of the imaginary time evolution operator $e^{-\tau H}$ so that the Trotter error is proportional to $O(\Delta\tau^3)$. For best accuracy, a small step size $\Delta\tau$ is preferable. However, the computing time is proportional to $1/\Delta\tau$. Additionally, very small step sizes increase the problem of suppressed jumps for simulations with symmetric tensors (see preceding paragraph). We therefore begin the imaginary time evolution with a rather large step size of $\Delta\tau = 0.1$ and decrease $\Delta\tau$ in steps until we reach $\Delta\tau = 0.001$. We always perform imaginary time evolution until the wave function converges before we decrease $\Delta\tau$. We ensure this by checking that the values of the bond vectors λ_x , λ_y , and λ_z of the wave function (2.12) converge (see figure 4.13 for an example).

For each $\Delta\tau$ in figure 4.13, we take the converged wave function and calculate the energy expectation value (figure 4.14). As expected, decreasing $\Delta\tau$ leads to a better approximation of the ground state and therefore to a lower energy. We do not expect any benefit from decreasing $\Delta\tau$ even further since the expectation values are already converged.

5. Results II: the Heisenberg model and the Hubbard model

5.1. The Heisenberg model

Definition of the model and ground state energy. The Heisenberg model [8] describes antiferromagnetism in a lattice system. We consider a system where a spin- $\frac{1}{2}$ degree of freedom occupies each vertex of the honeycomb lattice. The Hamiltonian reads

$$H_0 = J \sum_{\langle i,j \rangle} \mathbf{S}_i \cdot \mathbf{S}_j \quad (5.1)$$

where $J > 0$ defines the energy scale, \mathbf{S}_i is the spin operator on site i , and the sum runs over all pairs of neighboring sites. We find approximate ground states of H_0 with different bond dimensions by imaginary time evolution and calculate the energy expectation value per lattice site using SRG. For a detailed description of our proceeding, see section 4.2.1. We compare results for the ground state energy to results obtained by other methods in table 5.1.

Staggered magnetization. The ground state of the Hamiltonian (5.1) breaks the $SU(2)$ symmetry of spin rotations and shows long-range antiferromagnetic order. We define the operator for the staggered magnetization in z -direction,

$$M^+ := \sum_i (-)^i S_i^z, \quad (5.2)$$

where $(-)^i$ is 1 if the site i is in the sublattice \mathcal{A} and -1 if $i \in \mathcal{B}$. Since M^+ is not invariant under $SU(2)$, its expectation value is different for different ground states. To find the ground state where the magnetic order points in the z direction, we apply a small external staggered magnetic field of magnitude h ,

$$H(h) := H_0 - hM^+. \quad (5.3)$$

We find an approximate ground state of $H(h)$ for small values of h by imaginary time evolution and then calculate the expectation value of M^+ per site (figure 5.1). The extrapolation of the expectation value to $h \rightarrow 0$ gives an estimate of the staggered magnetization of the ground state of H_0 . We compare our results to results obtained by other methods in table 5.1. In contrast to the ground state energy, the staggered magnetization is not yet converged w.r.t. the bond dimension D of the wave function. This is due to the existence of gapless spin excitations, leading to long-ranged spin-spin correlations in the ground state, which cannot be captured by a PEPS.

5. Results II: the Heisenberg model and the Hubbard model

Method	E_0/J	$\langle M^+ \rangle$
Exact diagonalization (extrapolated, maximally 38 sites) [89]	−0.5448	
Quantum Monte Carlo (extrapolated, maximally 128 sites) [90]	−0.5445	0.33
Spin wave [91]	−0.5489	0.363
Series expansion [92]	−0.5443	0.399
TRG ($D = 8$) [36]	−0.5506	0.318
SRG ($D = 8$) [2]	−0.544	0.321
Ours (without symmetry, $D = 7$)	−0.5441	0.322
Ours (without symmetry, $D = 8$)	−0.5442	0.322
Ours (without symmetry, $D = 9$)	−0.5442	0.320
Ours (exploiting $U(1)$ -symmetry, $D = 7$)	−0.5441	0.322
Ours (exploiting $U(1)$ -symmetry, $D = 8$)	−0.5441	0.322
Ours (exploiting $U(1)$ -symmetry, $D = 9$)	−0.5442	0.320
Ours (exploiting $U(1)$ -symmetry, $D = 10$)	−0.5443	0.314
Ours (exploiting $U(1)$ -symmetry, $D = 11$)	−0.5442	0.313

Table 5.1.: Comparison of results for the ground state energy E_0 and the staggered magnetization $\langle M^+ \rangle$ (5.2) per lattice site for the spin- $\frac{1}{2}$ Heisenberg model (5.1).

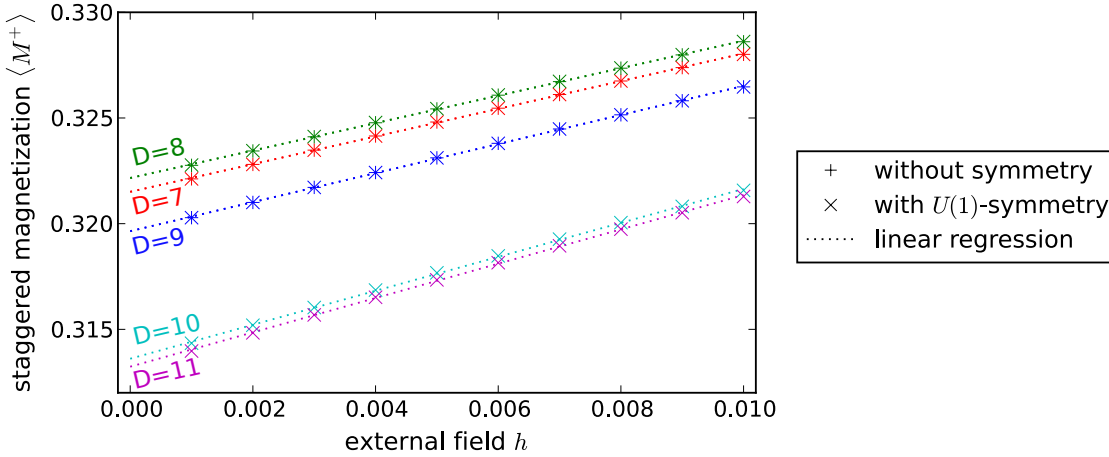


Figure 5.1.: Staggered magnetization (5.2) per lattice site for the Heisenberg model in a small external field. We calculate expectation values for non-symmetric PEPS with $D_{\text{cut}} = 70$ and for $U(1)$ -symmetric PEPS with $D_{\text{cut}} = 130$. For $D = 7, 8, 9$, the values for calculations with non-symmetric tensors and with $U(1)$ -symmetric tensors lie on top of each other.

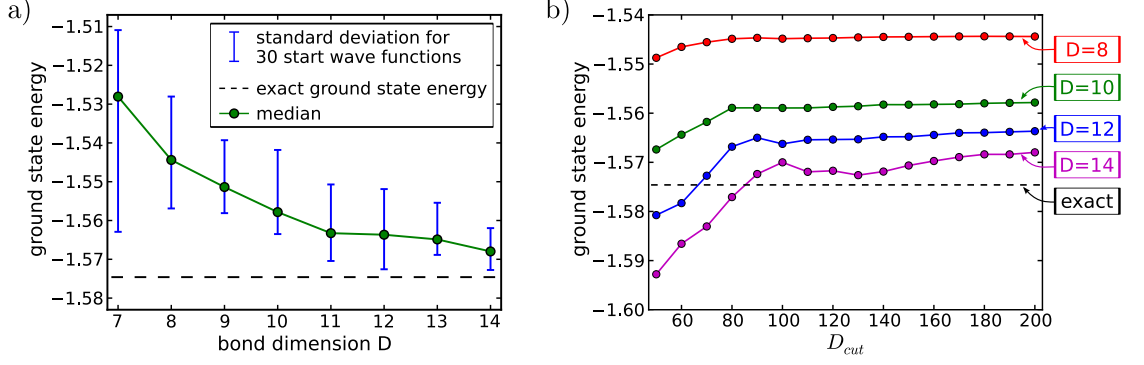


Figure 5.2.: Results for free fermions. a) Convergence w.r.t. the bond dimension D of the wave function and sensitivity of the imaginary time evolution to the start wave function. The energy expectation value is calculated using SRG with $D_{\text{cut}} = 200$. b) Convergence w.r.t. the cutoff bond dimension of the renormalization for wave functions with different bond dimensions.

5.2. The Hubbard model

We consider the Hubbard model of interacting fermions on the honeycomb lattice. The Hamiltonian reads

$$H(U) = \hat{T} + U\hat{D} = - \sum_{\langle i,j \rangle} \left(c_{i\uparrow}^\dagger c_{j\uparrow} + c_{j\uparrow}^\dagger c_{i\uparrow} + c_{i\downarrow}^\dagger c_{j\downarrow} + c_{j\downarrow}^\dagger c_{i\downarrow} \right) + U \sum_i n_{i\uparrow} n_{i\downarrow} \quad (5.4)$$

where $U \geq 0$ is the interaction strength, $c_{i,\sigma}^\dagger$ ($c_{i,\sigma}$) is the creation (annihilation) operator of a fermion with spin $\sigma \in \{\uparrow, \downarrow\}$ on site i of the lattice, $n_{i,\sigma} = c_{i,\sigma}^\dagger c_{i,\sigma}$ is the particle number operator, and the sum over $\langle i, j \rangle$ sums over all pairs of nearest neighbors.

The Hubbard model has gained interest in condensed matter physics since it is believed to give insight into mechanism behind high-temperature superconductivity. It can be shown analytically [93] that the system is a semi-metal for small interaction strengths U and an antiferromagnet for large U . Recent quantum Monte Carlo studies show indications of a topologically ordered spin-liquid phase ranging from $U \approx 3.5$ to $U \approx 4.3$ [75]. These results are still under discussion. Vaezi and Wen [94,95] find a topologically ordered phase from $U \approx 2.2$ to $U \approx 3.0$ using a slave-particle theory.

5.2.1. Quality of the simulations

The Hamiltonian (5.4) conserves the total particle number $N = \sum_i (n_{i\uparrow} + n_{i\downarrow})$ and the magnetization $M = \frac{1}{2} \sum_i (n_{i\uparrow} - n_{i\downarrow})$. We use imaginary time evolution and SRG with $U(1) \times U(1)$ -symmetric tensors to study the ground state properties at half filling without magnetization. We first consider the non-interacting case ($U = 0$) in order to test convergence w.r.t. the bond dimensions D of the wave function and D_{cut} of the renormalization. $H(U = 0)$ is diagonal in Fourier space and the ground state energy per site is $E_0/N \approx -1.5746$. For each bond dimension $D = 7, \dots, 14$, we generate 30 random start wave functions and project them onto the ground state by imaginary time evolution.

5. Results II: the Heisenberg model and the Hubbard model

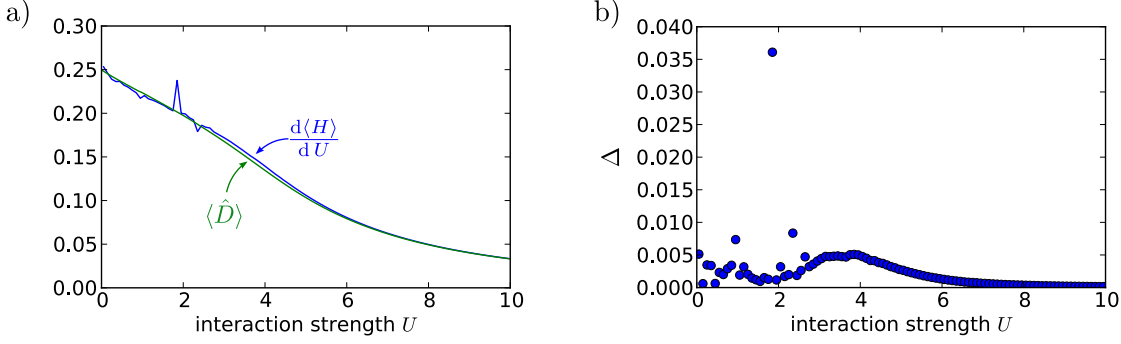


Figure 5.3.: Quality of the approximate ground state wave functions for the Hubbard model. We use imaginary time evolution with bond dimension $D = 14$ to generate ground state wave functions of the Hubbard model (5.4) for U in steps of 0.1. a) Expectation value of the double occupancy \hat{D} (5.4) and derivative of the calculated ground state energy. b) Measure Δ of the wave function quality, as defined in (5.7).

We then calculate the energy expectation value for each wave function using SRG with $D_{\text{cut}} = 200$. Figure 5.2a shows the standard deviation and the median of the calculated ground state energy for each D . As discussed for the Heisenberg model in section 4.2.2, the imaginary time evolution with small bond dimensions is highly sensitive to the start wave function. For growing bond dimension, the standard deviation decreases and the medians approach the exact value. At $D = 14$, the calculated ground state energy (median) lies 0.4% above the exact value. Figure 5.2b demonstrates that the calculation of the energy is well converged at $D_{\text{cut}} = 200$. Thus, the error of our calculations is controlled primarily by the quality of the wave functions rather than by the accuracy of the calculation of expectation values.

We now turn to the interacting regime ($U > 0$) where the Hubbard model cannot be solved analytically. We derive a condition to assess the quality of approximate ground state wave functions produced by the imaginary time evolution, since this seems to be the primary source of errors. For each value of U in some range, let $|\Psi_0(U)\rangle$ be a candidate to approximate the ground state of $H(U)$. The ground state energy is then approximately $E_0(U) \approx \langle \Psi_0(U) | H(U) | \Psi_0(U) \rangle$ and its derivative reads

$$\frac{dE_0(U)}{dU} \approx \frac{d(\langle \Psi_0(U) | H(U) | \Psi_0(U) \rangle)}{dU} = \quad (5.5)$$

$$= \langle \Psi_0(U) | \hat{D} | \Psi_0(U) \rangle + \frac{\partial |\Psi_0(U)\rangle}{\partial U} \underbrace{\frac{d(\langle \psi | H(U) | \psi \rangle)}{d\psi}}_{\approx 0} \bigg|_{|\psi\rangle=|\Psi_0(U)\rangle}, \quad (5.6)$$

where the last term would vanish exactly if $|\Psi_0(U)\rangle$ was an *exact* ground state. We move all terms that can be calculated by SRG to the left hand side and define

$$\Delta(U) := \left| \frac{d(\langle \Psi_0(U) | H(U) | \Psi_0(U) \rangle)}{dU} - \langle \Psi_0(U) | \hat{D} | \Psi_0(U) \rangle \right| \approx 0. \quad (5.7)$$

We generate approximate ground state wave functions $|\Psi_0(U)\rangle$ with bond dimension $D =$

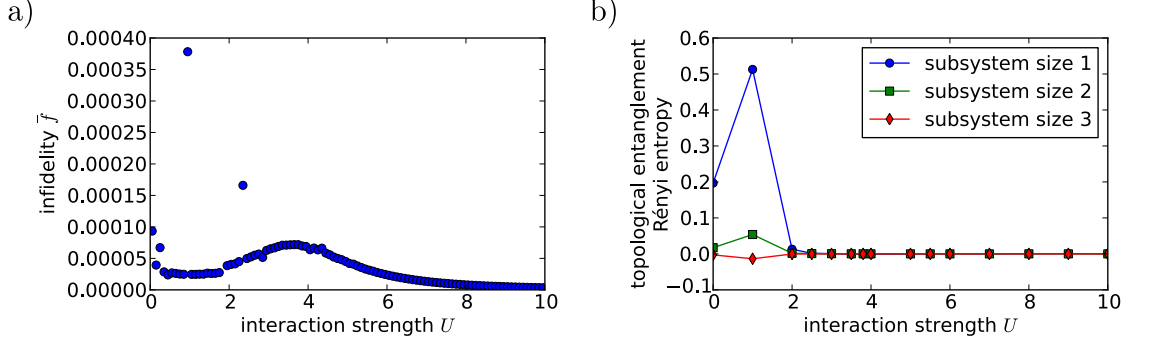


Figure 5.4.: Results for the Hubbard model on the honeycomb lattice. a) The infiltrity (5.8) shows indications to only one phase transition. b) The topological entanglement Rényi entropy (3.40) points to no topological order in the range $U \in [0, 10]$. The three subsystem sizes correspond to the geometries in illustrated in figure 3.12a.

14 for $U \in [0, 10]$ in steps of $\Delta U = 0.1$, calculate the expectation values of $H(U)$ and \hat{D} in these states using SRG with $D_{\text{cut}} = 200$ and evaluate the derivative in (5.7) numerically (figure 5.3). The wave functions meet the criterion $\Delta = 0$ to good accuracy for high interaction strengths but we observe high numerical errors around $U = 3 \dots 5$, which is roughly where the transition from the semi-metal to the antiferromagnet is expected. For $U \lesssim 3$, the numerical derivation of the energy expectation value is problematic.

5.2.2. Results

Due to the still considerable dependency of the imaginary time evolution on the start wave function even at $D = 14$ and rather poor agreement of $\Delta \approx 0$ around the expected transition point (see preceding section), the following results for the Hubbard model should be taken with care.

Phase transitions. Similar to (4.14), we define the infiltrity for a system of N sites as

$$\bar{f}_{\Delta U}(U) := 1 - \left(\frac{|\langle \Psi_0(U - \frac{\Delta U}{2}) | \Psi_0(U + \frac{\Delta U}{2}) \rangle|}{\sqrt{\langle \Psi_0(U - \frac{\Delta U}{2}) | \Psi_0(U - \frac{\Delta U}{2}) \rangle \cdot \langle \Psi_0(U + \frac{\Delta U}{2}) | \Psi_0(U + \frac{\Delta U}{2}) \rangle}} \right)^{1/N} \quad (5.8)$$

where $|\Psi_0(U)\rangle$ is the ground state of $H(U)$. We use SRG with $D_{\text{cut}} = 150$ to calculate $\bar{f}_{\Delta U}(U)$ for $\Delta U = 0.1$ in the range $U \in [0, 10]$ (figure 5.4a). The bond dimension of the wave function is $D = 14$. The results show indications for only one phase transition in this range, pointing to no existence of an extended intermediary spin-liquid phase.

Rényi entropy. We calculate the topological entanglement Rényi entropy (3.40) of the ground state of $H(U)$ using the method described in section (3.3.2). The calculation requires renormalization of a tensor network with bond dimension D^4 , where D is the bond dimension of the wave function (cf. figure 3.12b). We use wave functions with

5. Results II: the Heisenberg model and the Hubbard model

$D = 4$ and calculate Rényi entropies using $D_{\text{cut}} = 200$. Our results (figure 5.4b) show no topological entanglement Rényi entropy in the limit of large subsystem sizes (the subsystem sizes in figure 5.4b correspond to the geometries illustrated in figure 3.12a).

Conclusions and outlook

Summary of the presented work

In this thesis, we discussed the tensor entanglement renormalization group (TRG) and its improvement, the second renormalization group of tensors (SRG). Our main use of the method was to efficiently calculate approximate local expectation values in the ground states of quantum mechanical systems on the honeycomb lattice. We used imaginary time evolution to find approximate ground states in the form of PEPS for given Hamiltonians.

The work on this thesis included the implementation of the algorithms in a way that uses local symmetries of the Hamiltonian to reduce computational costs. This allows to perform more accurate calculations with considerably higher bond dimensions. For two simultaneously conserved quantum numbers, calculations with $D_{\text{cut}} = 200$ can be carried out at moderate costs. As a comparison, the highest bond dimension used so far in SRG calculations is, to our knowledge, $D_{\text{cut}} = 152$ [66]. The performance gain due to the use of symmetries comes from a partitioning of the bond spaces of the tensors into symmetry sectors, leading to a block diagonal structure of the tensors. We reviewed assumptions that are implicitly imposed onto the system by this method and explained in detail the formalism of symmetric tensors.

A recently proposed variation of the TRG algorithm uses Grassmann number valued tensors to simulate systems with fermionic degrees of freedom. We reviewed this approach from the perspective of symmetric tensors.

Tests of the implemented algorithm demonstrated that the SRG produces accurate results for a variety of analytically solvable example systems. This includes the calculation of the partition function of a classical statistical system, and of local expectation values and Rényi entropies in a given quantum mechanical state. Investigations of the convergence of calculated results w.r.t. the parameters of the algorithm showed that the use of symmetries comes at the price that the imaginary time evolution algorithm is vulnerable to getting stuck in a local energy minimum, especially at low bond dimensions D of the wave function. This effect is not yet fully understood.

The developed algorithm estimates the ground state energy of the spin- $\frac{1}{2}$ Heisenberg model on the honeycomb lattice to be $E_0 \approx -0.5442$, which is in good agreement with results from quantum Monte Carlo simulations. Calculations of the staggered magnetization in the ground state provide inconclusive results as the method cannot capture long-ranged spin-spin correlations. Results for the Hubbard model on the honeycomb lattice give no indication of the existence of a topologically ordered phase but suffer from high numerical errors.

Outlook

The sensitivity of the imaginary time evolution with symmetric tensors to the start wave function greatly derogates the reliability of the method. More insight into the cause of this effect is needed in order to gain full advantage of conserved quantum numbers. Other possible improvements to the method include (i) a better understanding of the relationship between the truncation error of the renormalization and the accuracy of calculated expectation values, and (ii) a generalization to the calculation of long-range observables. The latter may be achieved by the use of tensor product operators [96], a generalization of PEPS that encodes operators rather than states in the form of a product of tensors.

Concerning physical applications, fermionic systems have not yet been extensively studied by the SRG method. Since our simulations of the Hubbard model suffer from large numerical errors, the starting point for such studies should be the $t - J$ model. Gu et al. [66] studied this model on the Honeycomb lattice. Future work should go into the study of the $t - J$ model on other lattices.

A. Rényi Entropy of the string-net condensate state

The string-net condensate state is defined on a system with spin- $\frac{1}{2}$ degrees of freedom on the links of the honeycomb lattice and given by

$$|\Psi_{\text{sn}}\rangle = \sum_{\text{cl}} |\psi_{\text{cl}}\rangle \quad (\text{A.1})$$

where the sum runs over all configurations where down spins lie along closed loops only (figure 3.10b). We calculate the Rényi entanglement entropy of order n of a subsystem $\mathcal{S}' \subset \mathcal{S}$,

$$S_{\mathcal{S}'}^{(n)} := \frac{1}{1-n} \log_2 [\text{Tr}(\rho^n)] \quad (\text{A.2})$$

with

$$\rho := \text{Tr}_{\mathcal{S} \setminus \mathcal{S}'} |\Psi_{\text{sn}}\rangle \langle \Psi_{\text{sn}}| = \sum_{\varphi_{\mathcal{S} \setminus \mathcal{S}'}} \langle \varphi_{\mathcal{S} \setminus \mathcal{S}'} | \Psi_{\text{sn}} \rangle \langle \Psi_{\text{sn}} | \varphi_{\mathcal{S} \setminus \mathcal{S}'} \rangle \quad (\text{A.3})$$

where the states $|\varphi_{\mathcal{S} \setminus \mathcal{S}'}\rangle$ form an orthogonal basis of the Hilbert space of the complementary subsystem $\mathcal{S} \setminus \mathcal{S}'$. We choose the product wave functions on $\mathcal{S} \setminus \mathcal{S}'$ for $|\varphi_{\mathcal{S} \setminus \mathcal{S}'}\rangle$. Since strings cannot end at the boundary, the spins on the boundary of \mathcal{S}' in the states $|\Psi_{\text{sn}}\rangle$ and $\langle \Psi_{\text{sn}}|$ in (A.3) are completely determined by $\varphi_{\mathcal{S} \setminus \mathcal{S}'}$. This means that matrix elements of ρ between two states with different boundary configurations vanish. The reduced density matrix splits into a direct sum $\rho = \bigoplus_{\alpha} \rho_{(\alpha)}$ where α labels the boundary configuration and we have

$$S_{\mathcal{S}'}^{(n)} = \frac{1}{1-n} \log_2 \left[\sum_{\alpha} \text{Tr}(\rho_{(\alpha)}^n) \right] \quad (\text{A.4})$$

with

$$\rho_{(\alpha)} = \sum_{\varphi_{\mathcal{S} \setminus \mathcal{S}'}} \langle \varphi_{\mathcal{S} \setminus \mathcal{S}'}^{\alpha} | \Psi_{\text{sn}} \rangle \langle \Psi_{\text{sn}} | \varphi_{\mathcal{S} \setminus \mathcal{S}'}^{\alpha} \rangle \quad (\text{A.5})$$

where $\{|\varphi_{\mathcal{S} \setminus \mathcal{S}'}^{\alpha}\rangle\}$ is an orthogonal basis of the subspace of states with boundary configuration α within the Hilbert space of the system $\mathcal{S} \setminus \mathcal{S}'$.

Only boundary configurations α with an even number of down spins on the boundary of \mathcal{S}' contribute to the sum in (A.4). All of these allowed configurations contribute with equal amount since for any two allowed configurations α and α' , $\rho_{(\alpha)}$ and $\rho_{(\alpha')}$ are isomorphic. This can be seen by the following argument: We can create or annihilate a pair of down spins on any two sites of the boundary of \mathcal{S}' by acting with plaquette operators A_p from (3.36) as illustrated in figure A.1. Using only operations of this kind, we can build a transformation $\pi_{\alpha',\alpha}$ that maps states in the Hilbert space of \mathcal{S} with

A. Rényi Entropy of the string-net condensate state

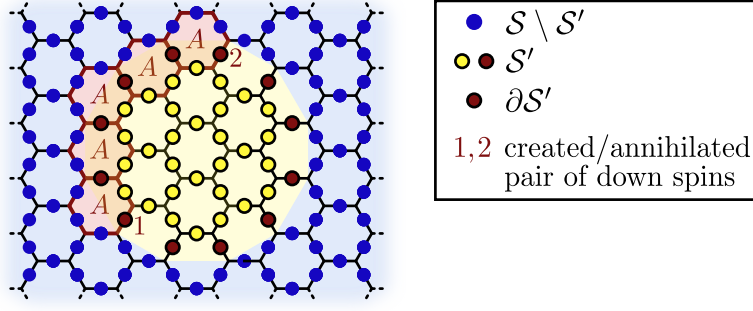


Figure A.1.: We can create or annihilate a pair of down spins on any two sites of the boundary without changing the state $|\Psi_{\text{sn}}\rangle$.

configuration α to states with configuration α' . The transformation $\pi_{\alpha',\alpha}$ is unitary since A_p is and it leaves the eigenstate $|\Psi_{\text{sn}}\rangle$ unchanged. Inserting $\pi_{\alpha,\alpha'}$ and $\pi_{\alpha',\alpha}$ in between the scalar products on the right hand side of (A.3) transforms $\rho_{(\alpha)}$ to $\rho_{(\alpha')}$.

Since $\rho_{(\alpha)}$ are isomorphic for all allowed α , $\text{Tr}(\rho_{(\alpha)}^n)$ is independent of α . Consider the boundary configuration $\alpha = 0$ of no strings crossing the boundary of \mathcal{S}' . It is easy to see that $\rho_{(0)}$ is proportional to the projector onto the string-net condensate defined on the subsystem \mathcal{S}' . Thus, $\rho_{(0)}$ has only one non-vanishing eigenvalue and we have $\text{Tr}(\rho_{(0)}^n) = (\text{Tr} \rho_{(0)})^n$. We denote the number of spins on the boundary of \mathcal{S}' by $|\partial\mathcal{S}'|$. There are $2^{|\partial\mathcal{S}'|}$ possible boundary configurations α , half of which fulfill the required condition of string conservation. For a normalized state, we have

$$\begin{aligned} 1 &= \text{Tr} \rho = \sum_{\alpha} \text{Tr} \rho_{(\alpha)} = 2^{|\partial\mathcal{S}'|-1} \text{Tr} \rho_{(0)} \\ \Rightarrow \text{Tr}(\rho_{(\alpha)}^n) &= \text{Tr}(\rho_{(0)}^n) = (\text{Tr} \rho_{(0)})^n = 2^{n-|\partial\mathcal{S}'|} \quad \text{for all allowed } \alpha. \end{aligned} \quad (\text{A.6})$$

We insert (A.6) into (A.4) and conclude for the Rényi entropy of the string-net condensate,

$$S_{\mathcal{S}'}^{(n)} = \frac{1}{1-n} \log_2 \left[2^{|\partial\mathcal{S}'|-1} \cdot 2^{n-|\partial\mathcal{S}'|} \right] = |\partial\mathcal{S}'| - 1 \quad (\text{A.7})$$

B. Analytic solution of the fermion pairing state

We derive exact analytical expressions for the expectation values of local operators in the fermion pairing state defined on a chain (4.13) and on the honeycomb lattice (4.10). In both geometries, the lattice can be divided into two sublattices \mathcal{A} and \mathcal{B} such that all nearest neighbors of any site $i \in \mathcal{A}$ belong to the sublattice \mathcal{B} and vice versa (figure B.1). The general fermion pairing state is then given by

$$|\Psi_u\rangle := \exp \left(u \sum_{\langle i,j \rangle} c_i^\dagger c_j^\dagger \right) |0\rangle, \quad (\text{B.1})$$

where the sum runs over all pairs (i, j) of nearest neighboring lattice sites with $i \in \mathcal{A}$ and $j \in \mathcal{B}$. The parameter u is real-valued, c_i^\dagger creates a fermion on site i and $|0\rangle$ is the vacuum state.

B.1. One-dimensional case

On a chain of length N , the two sublattices are formed by the even and the odd numbered sites, respectively. With periodic boundary conditions ($c_{N+1}^\dagger \equiv c_1^\dagger$), the pairing state (B.1) reads

$$|\Psi_u^{1D}\rangle = \exp \left(u \sum_{i=1}^{N/2} \left(c_{2i}^\dagger c_{2i+1}^\dagger + c_{2i}^\dagger c_{2i-1}^\dagger \right) \right) |0\rangle. \quad (\text{B.2})$$

We define the creation operators in momentum space

$$c_k^{A\dagger} := \sqrt{\frac{2}{N}} \sum_{j=1}^{N/2} e^{ikj} c_{2j}^\dagger; \quad c_k^{B\dagger} := \sqrt{\frac{2}{N}} \sum_{j=1}^{N/2} e^{ikj} c_{2j+1}^\dagger \quad (\text{B.3})$$

where $k = 2\pi n/(N/2)$ with $n = -N/4, \dots, N/4 - 1$. Note that the operator $c_k^{A\dagger}$ ($c_k^{B\dagger}$) creates a fermion that lives on the sublattice \mathcal{A} (\mathcal{B}) only.

In momentum space, (B.2) simplifies to

$$|\Psi_u^{1D}\rangle = \exp \left(\sum_k \underbrace{u(1 + e^{ik})}_{=: \alpha_k} c_{-k}^{A\dagger} c_k^{B\dagger} \right) |0\rangle = \prod_k \left(1 + \alpha_k c_{-k}^{A\dagger} c_k^{B\dagger} \right) |0\rangle \quad (\text{B.4})$$

B. Analytic solution of the fermion pairing state

The right hand side of equation (B.4) still contains a term that mixes fermions living on different sublattices. To decouple the two particle types we perform a Bogoliubov transformation by defining new field operators

$$\begin{pmatrix} \psi_{1,-k} \\ \psi_{2,k}^\dagger \end{pmatrix} := \begin{pmatrix} u_k & -v_k \\ v_k^* & u_k^* \end{pmatrix} \begin{pmatrix} c_{-k}^A \\ c_k^{B\dagger} \end{pmatrix} \quad (\text{B.5})$$

with coefficients u_k, v_k such that

$$|u_k|^2 + |v_k|^2 = 1 \quad \text{and} \quad \left(1 + \alpha_k c_{-k}^{A\dagger} c_k^{B\dagger}\right) |0\rangle \propto \psi_{2,k} \psi_{1,-k} |0\rangle \quad (\text{B.6})$$

A simple calculation shows that the conditions (B.6) are satisfied by

$$u_k = \frac{1}{\sqrt{1 + |\alpha_k|^2}}; \quad v_k = \frac{\alpha_k}{\sqrt{1 + |\alpha_k|^2}}. \quad (\text{B.7})$$

The operators $\psi_{1,k}$ and $\psi_{2,k}$ fulfill fermionic anticommutation relations due to the unitarity of the transformation (B.5). We will refer to the particles annihilated by these operators as “Bogoliubov particles”. Inserting (B.6) into (B.4) we see that the one-dimensional pairing state is the vacuum with respect to the Bogoliubov particles, i.e.

$$|\Psi_u^{1D}\rangle \propto \prod_k \psi_{2,k} \psi_{1,-k} |0\rangle \quad \implies \quad \psi_{1,k} |\Psi_u^{1D}\rangle = \psi_{2,k} |\Psi_u^{1D}\rangle = 0 \quad \forall k \quad (\text{B.8})$$

Local density. We use translational invariance of $|\Psi_u^{1D}\rangle$ to write

$$\begin{aligned} \langle \Psi_u^{1D} | c_i^\dagger c_i | \Psi_u^{1D} \rangle &= \frac{2}{N} \sum_{j=1}^{N/2} \langle \Psi_u^{1D} | c_{2j}^\dagger c_{2j} | \Psi_u^{1D} \rangle = \frac{2}{N} \sum_k \langle \Psi_u^{1D} | c_k^{A\dagger} c_k^A | \Psi_u^{1D} \rangle \\ &= \frac{2}{N} \sum_k \langle \Psi_u^{1D} | (u_{-k} \psi_{1,k}^\dagger + v_{-k}^* \psi_{2,-k}) (u_{-k}^* \psi_{1,k} + v_{-k} \psi_{2,-k}^\dagger) | \Psi_u^{1D} \rangle \\ &\stackrel{(B.8)}{=} \frac{2}{N} \sum_k |v_{-k}|^2 \langle \Psi_u^{1D} | \psi_{2,-k} \psi_{2,-k}^\dagger | \Psi_u^{1D} \rangle \\ &= \frac{2}{N} \sum_k |v_{-k}|^2 \langle \Psi_u^{1D} | \underbrace{(\{\psi_{2,-k}, \psi_{2,-k}^\dagger\} - \psi_{2,-k}^\dagger \psi_{2,-k})}_{=1} | \Psi_u^{1D} \rangle \\ &\stackrel{(B.8)}{=} \frac{2}{N} \sum_k |v_{-k}|^2 \langle \Psi_u^{1D} | \Psi_u^{1D} \rangle \end{aligned} \quad (\text{B.9})$$

We take the thermodynamic limit by replacing $\frac{2}{N} \sum_k$ by $\frac{1}{2\pi} \int_{-\pi}^{\pi} dk$ and conclude for the local particle density

$$\langle n_i \rangle \equiv \frac{\langle \Psi_u^{1D} | c_i^\dagger c_i | \Psi_u^{1D} \rangle}{\langle \Psi_u^{1D} | \Psi_u^{1D} \rangle} \xrightarrow{N \rightarrow \infty} \frac{1}{2\pi} \int_{-\pi}^{\pi} |v_k|^2 dk = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{dk}{1 + \frac{1}{|\alpha_k|^2}} = 1 - \frac{1}{\sqrt{1 + 4u^2}}. \quad (\text{B.10})$$

Pairing amplitude. We define the pairing amplitude as $\langle c_{2j}^\dagger c_{2j+1}^\dagger \rangle$. A calculation analogous to (B.9) leads to

$$\langle c_{2j}^\dagger c_{2j+1}^\dagger \rangle \equiv \frac{\langle \Psi_u^{1D} | c_{2j}^\dagger c_{2j+1}^\dagger | \Psi_u^{1D} \rangle}{\langle \Psi_u^{1D} | \Psi_u^{1D} \rangle} \xrightarrow{N \rightarrow \infty} \frac{1}{2\pi} \int_{-\pi}^{\pi} v_k^* u_k dk = \frac{1 - (1 + 4u^2)^{-1/2}}{2u}. \quad (\text{B.11})$$

Norm and Overlap. We calculate the scalar product of two fermion pairing states with respective pairing parameters u and u' .

$$\langle \Psi_{u'}^{1D} | \Psi_u^{1D} \rangle \stackrel{(B.4)}{=} \prod_k \langle 0 | (1 + \alpha_k'^* c_k^B c_{-k}^A) (1 + \alpha_k c_{-k}^A c_k^{B\dagger}) | 0 \rangle = \prod_k (1 + \alpha_k'^* \alpha_k) \quad (\text{B.12})$$

where $\alpha_k' = u'(1 + e^{ik})$. Since the scalar product scales exponentially with growing system size it is more convenient to consider the logarithm of the modulus of the overlap per site.

$$\frac{\log(|\langle \Psi_{u'}^{1D} | \Psi_u^{1D} \rangle|)}{N} = \frac{1}{N} \sum_k \log |1 + \alpha_k'^* \alpha_k| \xrightarrow{N \rightarrow \infty} \frac{1}{4\pi} \int_{-\pi}^{\pi} \log |1 + \alpha_k'^* \alpha_k| dk \quad (\text{B.13})$$

See figure 4.7 for results from numerical evaluation of the integral in (B.13).

B.2. Two-dimensional case

We consider the fermion pairing state defined in (B.1) on the honeycomb lattice (see [3]). The honeycomb lattice is an equilateral triangular Bravais lattice with a two-atomic basis (figure B.1). We set the distance of two nearest neighbors to one and choose as lattice vectors

$$\mathbf{a}_1 = \begin{pmatrix} 0 \\ \sqrt{3} \end{pmatrix}; \quad \mathbf{a}_2 = \begin{pmatrix} 3/2 \\ \sqrt{3}/2 \end{pmatrix}. \quad (\text{B.14})$$

Therefore, the reciprocal lattice is spanned by

$$\mathbf{b}_1 = 2\pi \begin{pmatrix} -1/3 \\ 1/\sqrt{3} \end{pmatrix}; \quad \mathbf{b}_2 = 2\pi \begin{pmatrix} 2/3 \\ 0 \end{pmatrix}. \quad (\text{B.15})$$

We define the Fourier transformed creation operators

$$c_{\mathbf{k}}^{A\dagger} := \sqrt{\frac{2}{N}} \sum_{i \in \mathcal{A}} e^{i\mathbf{k} \cdot \mathbf{r}_i} c_i^\dagger; \quad c_{\mathbf{k}}^{B\dagger} := \sqrt{\frac{2}{N}} \sum_{j \in \mathcal{B}} e^{i\mathbf{k} \cdot \mathbf{r}_j} c_j^\dagger \quad (\text{B.16})$$

where N is the number of sites in the direct lattice, \mathbf{r}_i is the position vector of site i in the direct lattice and \mathbf{k} can take $N/2$ different values $\mathbf{k} = k_1 \mathbf{b}_1 + k_2 \mathbf{b}_2$ with $k_1, k_2 \in [-\frac{1}{2}, \frac{1}{2}]$.

With these definitions, the two-dimensional pairing state defined in (B.1) can be written as (cf. (B.4), see [3])

$$|\Psi_u^{2D}\rangle = \prod_{\mathbf{k}} \left(1 + \alpha_{\mathbf{k}} c_{-\mathbf{k}}^{A\dagger} c_{\mathbf{k}}^{B\dagger} \right) |0\rangle \quad \text{with } \alpha_{\mathbf{k}} = u(e^{-i\mathbf{k} \cdot \delta \mathbf{x}} + e^{-i\mathbf{k} \cdot \delta \mathbf{y}} + e^{-i\mathbf{k} \cdot \delta \mathbf{z}}). \quad (\text{B.17})$$

B. Analytic solution of the fermion pairing state

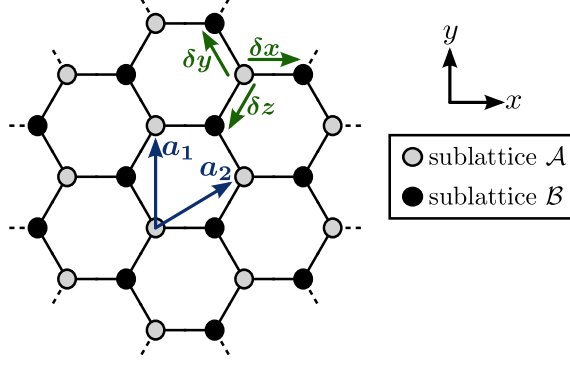


Figure B.1.: The honeycomb lattice can be divided into two sublattices, \mathcal{A} and \mathcal{B} such that sites $i \in \mathcal{A}$ have only nearest neighbors in the sublattice \mathcal{B} and vice versa. Both sublattices form an equilateral triangular Bravais lattice. We illustrate lattice vectors \mathbf{a}_1 and \mathbf{a}_2 as defined in (B.14) and vectors connecting nearest neighbors as defined in (B.18).

where $\delta\mathbf{x}$, $\delta\mathbf{y}$, and $\delta\mathbf{z}$ are illustrated in figure B.1.

$$\delta\mathbf{x} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \quad \delta\mathbf{y} = \begin{pmatrix} -1/2 \\ \sqrt{3}/2 \end{pmatrix}; \quad \delta\mathbf{z} = \begin{pmatrix} -1/2 \\ -\sqrt{3}/2 \end{pmatrix} \quad (\text{B.18})$$

Beginning from equation (B.5), all remaining calculations we outlined for the one-dimensional pairing state can be adopted directly to the two-dimensional case. We only have to use the new definition (B.17) for $\alpha_{\mathbf{k}}$ and, in the thermodynamic limit, we now substitute $\int_{-1/2}^{1/2} dk_1 \int_{-1/2}^{1/2} dk_2$ for $\frac{2}{N} \sum_{\mathbf{k}}$. We evaluated the resulting two-dimensional integrals for the particle density, the pairing amplitude, and the overlap of two states numerically (figures 4.6 and 4.7).

C. Short manual to the developed programs

C.1. Usage of the programs

Overview of the executables. Table C.1 gives an overview of the applications we wrote to implement the algorithms described in this thesis. The foundation of all applications is a library of symmetric tensors. Table C.2 lists the symmetry groups for which specialized tensors exist. For each application, only one type of symmetric tensors is compiled in, depending on the filename. This allows us to do some important optimization at compile time. All applications print out usage instructions when invoked with the “-h” option. We explain the basic usage of the “itgs-” and the “qmsrg-” applications by means of an (unrealistic) example session. Usage of the “renyisrg-” applications is similar to “qmsrg-” and the remaining applications should be self-explanatory.

Imaginary time evolution. To find, e.g. an approximate ground state of the Ising model in a transverse field of magnitude 0.3, we invoke

```
itgs-nonsym --param transversefield=0.3 -D7 -t0.01 -m200 -o itime \
--model tim
```

This uses PEPS with bond dimension $D = 7$ and imaginary time steps $\Delta\tau = 0.01$ to calculate $|\Psi(\tau = 200)\rangle$. It will save the resulting wave function, as well as some intermediary wave functions and the convergence of the bond vectors, to files whose names begin with “itime”. We selected the transverse ising model (TIM) by specifying “--model tim”. Alternatively, we could have specified the Hamiltonian explicitly using the “-H” option, i.e.

```
itgs-nonsym --param transversefield=0.3 -D7 -t0.01 -m200 -o itime \
-H '- SUM_{ij}>(Sz[i]*Sz[j]) - transversefield * SUM_i(Sx[i])'
```

For more examples for the syntax of Hamiltonians, invoke the program with the “-h” option and look at the description of the “--model” option. The token “transversefield” is a parameter of the Hamiltonian, which we set to the value 0.3 with the “--param” option. Using parameters in Hamiltonians simplifies post-processing of the output data because the values of the parameters will be stored in all output files. All words that start with a lower case letter in the formula for the Hamiltonian will be interpreted as parameters and parameters default to zero if no value is specified.

SRG. To calculate some expectation values of the wave function produced by the imaginary time evolution from the preceding paragraph, we invoke

```
qmsrg-nonsym -i itime_wf200 --ops 'H,SUM_i(Sx[i])' -D70 -o srg.py
```

C. Short manual to the developed programs

Filename	Application
<code>itgs-system</code>	Imaginary time evolution
<code>qmsrg-system</code>	Calculation of local observables with SRG or TRG
<code>renyisrg-symmetry</code>	Calculation of Rényi entropies (currently uses TRG)
<code>wfviewer-symmetry</code>	Inspect binary wave function files generated by <code>itgs-*</code>
<code>test-identifier</code>	Unit tests for the underlying library of symmetric tensors

Table C.1.: Naming pattern for the applications. “Symmetry” is a placeholder for an identifier in table C.2. “System” is a placeholder that describes a physical system, such as spin systems or systems with one or two particle species that propagate through the lattice. Most physical systems are named after the used symmetry group, except in some cases where two different kinds of systems use the same symmetry group.

Identifier	Symmetry group
<code>nonsym</code>	Trivial symmetry group (“non-symmetric tensors”)
<code>z2b</code> & <code>z2f</code>	\mathbb{Z}_2 for bosonic and fermionic systems
<code>u1b</code> & <code>u1f</code>	$U(1)$ for bosonic and fermionic systems
<code>u1u1b</code> & <code>u1u1f</code>	$U(1) \times U(1)$ for bosonic and fermionic systems

Table C.2.: Symmetry groups for which specialized tensors exist.

This loads a wave function from the file “`itime_wf200`”, calculates the energy and the magnetization in the x direction using SRG with $D_{\text{cut}} = 70$ and writes the output to the file “`srg.py`”. Expectation values are calculated *per bond*. Multiply by $\frac{3}{2}$ to get expectation values per site. The operators are specified as a comma separated list using the “`--ops`” option. The same notation as for the Hamiltonian in the imaginary time evolution is used. `H` is a predefined operator that stands for the Hamiltonian used in the imaginary time evolution.

Output format. The applications produce output files in two different formats. Wave functions are stored in a binary format to save disk space. The file names follow the pattern “`basefilename_wfimaginarytime`” where `basefilename` is specified with the “`-o`” option and `imaginarytime` is τ rounded to integer. The “`wfviewer-`” applications may be used to inspect those files. Wave function files are currently not portable between different computer architectures.

All other output data is written to text files in the syntax of the Python programming language. The files should be human-readable and self-explanatory due to automatically generated comments within the files. No specialized parser for the output files is required if post-processing is done in Python scripts. Simply load files into a Python program either with an `import` statement or programmatically with `imp.load_source()`.

C.2. Organization of the source code

Overview. We wrote the programs in the C++ programming language. The implemented classes may be grouped into four categories: a library for symmetric (and optionally fermionic) tensors; implementations of the algorithms, independent of the symmetry group (the symmetry group is a compile-time parameter to these implementations); definitions of physical systems; And finally a set of small drivers that specialize the algorithms to each physical system. All individual methods of the classes are documented at the point of declaration. The primary documentation for the interplay between the methods are the unit tests, implemented in files whose names begin with “test-”.

Tensor library. The tensor library builds the foundation of all applications. Itself is, however, on purpose independent of the remaining parts of the application suite and we believe that it can be reused easily in other applications.

The template classes `SymDiagMatrix` and `SymTensor` provide the interface to the tensor library and implement the basic tensor operations described in sections 3.1.3 and 3.2.3. All manipulations of large data are delegated to the BLAS and LAPACK libraries. The classes are templated by a typename `Sym` that defines the symmetry group. Each symmetry group is defined in a class in the namespace `symmetries`. The implementations of these classes contain a fair amount of boiler-plate code because the interface has been designed with the primary goal to allow high optimization. To implement symmetric tensors for a new abelian symmetry group, copy the file “z2.h” and adjust it to your needs.

Symmetry-group-agnostic implementations of the algorithms. The algorithms are implemented in the classes `BasicWaveFunction` (for the imaginary time evolution algorithm), `SrgIterator` (TRG and SRG), and `SrgRenyiIterator` (calculation of Rényi entropies). The classes are templated by a typename that defines a symmetry group.

Physical Systems. The class `BasicPhysops` implements an abstract parser for physical operators. It basically defines the syntax for formulas that may, for example, be used to specify the Hamiltonian for the imaginary time evolution applications (see section C.1). `BasicPhysops` does not make any assumptions on the underlying physical system (e.g. the dimension of the local Hilbert space) or the conserved quantum numbers. This is done in a set of classes that derive from `BasicPhysops` and whose names end in `Physops`. For example, the class `TwoFermU1U1Physops` defines a physical system of spin- $\frac{1}{2}$ fermions with conserved particle number and magnetization. It also defines local operators, such as creators and annihilators and maps them to keywords that can be used in formulas.

Drivers for the executables. The abstract classes `Itgs`, `Qmsrg`, and `Renyisrg` implement the interpretation of command-line parameters and the generation of output files for the corresponding type of application. They are specialized to each physical system in a set of very light-weight subclasses.

Bibliography

- [1] M. Levin and C. P. Nave, Physical Review Letters **99**, 120601 (2007).
- [2] Z. Y. Xie, H. C. Jiang, Q. N. Chen, Z. Y. Weng, and T. Xiang, Physical Review Letters **103**, 160601 (2009).
- [3] Z. Gu, F. Verstraete, and X. Wen, 1004.2563 (2010).
- [4] J. G. Bednorz and K. A. Müller, Zeitschrift für Physik B Condensed Matter **64**, 189 (1986).
- [5] E. Dagotto, Reviews of Modern Physics **66**, 763 (1994).
- [6] X. G. Wen and Q. Niu, Physical Review B **41**, 9377 (1990).
- [7] X. G. Wen, International Journal of Modern Physics B **4**, 239 (1990).
- [8] W. Heisenberg, Zeitschrift für Physik **49**, 619 (1928).
- [9] J. Hubbard, Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences **276**, 238 (1963).
- [10] M. Lewenstein et al., Advances in Physics **56**, 243 (2007).
- [11] I. Bloch, J. Dalibard, and W. Zwerger, Reviews of Modern Physics **80**, 885 (2008).
- [12] A. F. Albuquerque et al., 1102.5325 (2011).
- [13] D. M. Ceperley and B. J. Alder, Physical Review Letters **45**, 566 (1980).
- [14] W. M. C. Foulkes, L. Mitas, R. J. Needs, and G. Rajagopal, Reviews of Modern Physics **73**, 33 (2001).
- [15] G. Sugiyama and S. Koonin, Annals of Physics **168**, 1 (1986).
- [16] M. Troyer and U. Wiese, Physical Review Letters **94**, 170201 (2005).
- [17] K. G. Wilson, Reviews of Modern Physics **47**, 773 (1975).
- [18] S. R. White, Physical Review Letters **69**, 2863 (1992).
- [19] U. Schollwöck, Reviews of Modern Physics **77**, 259 (2005).
- [20] S. Liang and H. Pang, Physical Review B **49**, 9214 (1994).
- [21] T. Xiang, J. Lou, and Z. Su, Physical Review B **64**, 104414 (2001).

Bibliography

- [22] E. M. Stoudenmire and S. R. White, arXiv:1105.1374 (2011).
- [23] S. R. White and A. L. Chernyshev, Physical Review Letters **99**, 127004 (2007).
- [24] S. Yan, D. A. Huse, and S. R. White, 1011.6114 (2010).
- [25] A. Weichselbaum and S. R. White, arXiv:1110.1406 (2011).
- [26] M. Srednicki, Physical Review Letters **71**, 666 (1993).
- [27] J. Eisert, M. Cramer, and M. B. Plenio, Reviews of Modern Physics **82**, 277 (2010).
- [28] S. Östlund and S. Rommer, Physical Review Letters **75**, 3537 (1995).
- [29] S. Rommer and S. Östlund, Physical Review B **55**, 2164 (1997).
- [30] U. Schollwöck, Annals of Physics **326**, 96 (2011).
- [31] T. Nishino et al., Progress of Theoretical Physics **105**, 409 (2001).
- [32] F. Verstraete and J. I. Cirac, arXiv:cond-mat/0407066 (2004).
- [33] V. Murg, F. Verstraete, and J. I. Cirac, Physical Review A **75**, 033605 (2007).
- [34] G. Vidal, Physical Review Letters **101**, 110501 (2008).
- [35] G. Vidal, Physical Review Letters **99**, 220405 (2007).
- [36] H. C. Jiang, Z. Y. Weng, and T. Xiang, Physical Review Letters **101**, 090603 (2008).
- [37] Z. Gu, M. Levin, and X. Wen, Physical Review B **78**, 205116 (2008).
- [38] X. Chen, B. Zeng, Z. Gu, I. L. Chuang, and X. Wen, 1003.1774 (2010), Phys. Rev. B **82**, 165119 (2010).
- [39] H. H. Zhao et al., Physical Review B **81**, 174411 (2010).
- [40] A. Shitade et al., Physical Review Letters **102**, 256403 (2009).
- [41] J. Chaloupka, G. Jackeli, and G. Khaliullin, Physical Review Letters **105**, 027204 (2010).
- [42] L. Wang, Z. Gu, X. Wen, and F. Verstraete, arXiv:1112.3331 (2011).
- [43] C. Huang and F. Lin, arXiv:0911.4670 (2009).
- [44] M. Nielsen, *Quantum computation and quantum information*, Cambridge University Press, Cambridge, 2000.
- [45] G. Vidal, Physical Review Letters **91**, 147902 (2003).
- [46] M. Fannes, B. Nachtergaele, and R. F. Werner, Communications in Mathematical Physics **144**, 443 (1992).

- [47] P. Calabrese and J. Cardy, International Journal of Quantum Information **4**, 429 (2006), Int.J.Quant.Inf. 4 (2006) 429.
- [48] K. Okunishi, Y. Hieida, and Y. Akutsu, Physical Review E **59**, R6227 (1999).
- [49] J. Jordan, R. Orús, G. Vidal, F. Verstraete, and J. I. Cirac, Physical Review Letters **101**, 250602 (2008).
- [50] W. Li et al., Physical Review B **82**, 134434 (2010).
- [51] A. Garcia-Saez and J. I. Latorre, arXiv:1112.1412 (2011).
- [52] W. Li et al., Physical Review Letters **106**, 127202 (2011).
- [53] Z. Gu, arXiv:1109.4470 (2011).
- [54] L. Wang, I. Pizorn, and F. Verstraete, Physical Review B **83**, 134421 (2011).
- [55] M. Hinczewski and A. N. Berker, Physical Review E **77**, 011104 (2008).
- [56] Z. Gu and X. Wen, arXiv:0903.1069 (2009), Phys.Rev.B80:155131,2009.
- [57] M. Chang and M. Yang, Physical Review B **79**, 104411 (2009).
- [58] W. Meng-Xiong et al., Chinese Physics Letters **27**, 076402 (2010).
- [59] C. Güven, M. Hinczewski, and A. N. Berker, Physical Review E **82**, 051110 (2010).
- [60] L. Bonnes, H. Büchler, and S. Wessel, New Journal of Physics **12**, 053027 (2010).
- [61] P. Chen and M. Yang, Physical Review B **82**, 180510 (2010).
- [62] P. Chen, C. Lai, and M. Yang, Physical Review B **81**, 020409 (2010).
- [63] W. Li, S. Gong, Y. Zhao, and G. Su, Physical Review B **81**, 184427 (2010).
- [64] H. H. Zhao et al., arXiv:1105.2716 (2011).
- [65] H. Jiang, Z. Gu, X. Qi, and S. Trebst, Physical Review B **83**, 245104 (2011).
- [66] Z. Gu et al., arXiv:1110.1183 (2011).
- [67] W. Marshall, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **232**, 48 (1955).
- [68] A. Auerbach, *Interacting electrons and quantum magnetism*, Springer-Verlag, New York, 1994.
- [69] D. Perez-Garcia, F. Verstraete, J. I. Cirac, and M. M. Wolf, 0707.2260 (2007), Quant. Inf. Comp. 8, 0650-0663 (2008).
- [70] M. Sanz, M. M. Wolf, D. Pérez-García, and J. I. Cirac, Physical Review A **79**, 042308 (2009).

Bibliography

- [71] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac, Quantum Information and Computation **7**, 401 (2007), Quantum Inf. Comput. **7**, 401 (2007).
- [72] D. Perez-Garcia, M. Sanz, C. E. Gonzalez-Guillen, M. M. Wolf, and J. I. Cirac, 0908.1674 (2009), New J. Phys. **12** (2010) 025010.
- [73] S. Singh, R. N. C. Pfeifer, and G. Vidal, Physical Review A **82**, 050301 (2010).
- [74] W. Tung and W. K. Tung, *Group Theory in Physics*, World Scientific Publishing Company, 1985.
- [75] Z. Y. Meng, T. C. Lang, S. Wessel, F. F. Assaad, and A. Muramatsu, Nature **464**, 847 (2010).
- [76] C. V. Kraus, N. Schuch, F. Verstraete, and J. I. Cirac, 0904.4667 (2009), Phys. Rev. A **81**, 052338 (2010).
- [77] P. Corboz, J. Jordan, and G. Vidal, Physical Review B **82**, 245119 (2010).
- [78] P. Corboz, R. Orús, B. Bauer, and G. Vidal, Physical Review B **81**, 165104 (2010).
- [79] P. Corboz and G. Vidal, Physical Review B **80**, 165129 (2009).
- [80] I. Pizorn and F. Verstraete, Physical Review B **81**, 245110 (2010).
- [81] B. Béri and N. R. Cooper, 1101.5610 (2011).
- [82] T. Barthel, C. Pineda, and J. Eisert, Physical Review A **80**, 042333 (2009).
- [83] L. D. Landau, Phys. Z. Sowjetunion **11**, 545 (1937).
- [84] K. Alexei, Annals of Physics **321**, 2 (2006).
- [85] M. A. Levin and X. Wen, Physical Review B **71**, 045110 (2005).
- [86] A. Kitaev and J. Preskill, Physical Review Letters **96**, 110404 (2006).
- [87] S. T. Flammia, A. Hamma, T. L. Hughes, and X. Wen, Physical Review Letters **103**, 261601 (2009).
- [88] G. H. Wannier, Physical Review **79**, 357 (1950).
- [89] J. Richter, J. Schulenburg, and A. Honecker, Quantum magnetism in two dimensions: From semi-classical néel order to magnetic disorder, in *Quantum Magnetism*, volume 645 of *Lecture Notes in Physics*, pages 85–153, Springer Berlin / Heidelberg, 2004.
- [90] J. D. Reger, J. A. Riera, and A. P. Young, Journal of Physics: Condensed Matter **1**, 1855 (1989).
- [91] Z. Weihong, J. Oitmaa, and C. J. Hamer, Physical Review B **44**, 11869 (1991).
- [92] J. Oitmaa, C. J. Hamer, and Z. Weihong, Physical Review B **45**, 9834 (1992).

- [93] A. Giuliani, 1102.3881 (2011).
- [94] A. Vaezi and X. Wen, 1010.5744 (2010).
- [95] A. Vaezi and X. Wen, 1101.1662 (2011).
- [96] B. Pirvu, V. Murg, J. I. Cirac, and F. Verstraete, *New Journal of Physics* **12**, 025012 (2010).

Acknowledgments

I wish to thank Ulrich Schollwöck for the chance to write my diploma thesis in his group, the earlier possibility to work with him, and his direct help with any questions.

Thanks also to Wilhelm Zwerger for supervising from my home university and for the uncomplicated assistance with all technicalities.

I owe special thanks to Tassilo Keilmann for overseeing my work and for fruitful discussions; to Fabian Heidrich-Meisner for proofreading the thesis and for his literature recommendations; to Stefan Depenbrock for sharing with me his wisdom on PEPS algorithms, for proofreading parts of this thesis and for being my wingman at the DPG spring conference; to Anton Wöllert for helpful discussions on numerical methods; and to Cordula Weber for great administrative support.

Furthermore, I wish to thank Frank Pollmann for fruitful discussions about the studied algorithms.

I thank Andreas Läuchli for helpful discussions on the Heisenberg model on the honeycomb lattice.

Thanks also to Roger Melko for his tips for investigating topological order.

I also wish to thank Sebastian Thomas for helping me out with L^AT_EX-related issues.

Finally I owe special thanks to my brother, Richard Heiner Bamler, for proofreading this thesis at record time and to my parents for their unconditional support at any time.