

# Photoexcitations of Model Manganite Systems using Matrix-Product States

## Dissertation

zur Erlangung des mathematisch-naturwissenschaftlichen Doktorgrades  
„Doctor rerum naturalium“  
der Georg-August-Universität Göttingen

–

im Promotionsprogramm ProPhys  
der Georg-August University School of Science (GAUSS)

vorgelegt von

**Thomas Köhler**  
aus Bremen

Göttingen, 2018

### **Betreuungsausschuss:**

Priv.-Doz. Dr. Salvatore R. Manmana  
Institut für Theoretische Physik, Georg-August-Universität Göttingen  
Prof. Dr. Stefan Kehrein  
Institut für Theoretische Physik, Georg-August-Universität Göttingen  
Prof. Dr. Stefan Mathias  
I. Physikalisches Institut, Georg-August-Universität Göttingen

### **Mitglieder der Prüfungskommission:**

Referent: Priv.-Doz. Dr. Salvatore R. Manmana  
Institut für Theoretische Physik, Georg-August-Universität Göttingen  
Korreferent: Prof. Dr. Stefan Kehrein  
Institut für Theoretische Physik, Georg-August-Universität Göttingen  
2. Korreferent: Prof. Dr. Andreas Honecker  
Laboratoire de Physique Théorique et Modélisation, Université de Cergy-Pontoise, France

### **Weitere Mitglieder der Prüfungskommission:**

Prof. Dr. Stefan Mathias  
I. Physikalisches Institut, Georg-August-Universität Göttingen  
apl. Prof. Dr. Michael Seibt  
IV. Physikalisches Institut, Georg-August-Universität Göttingen  
Prof. Dr. Matthias Krüger  
Institut für Theoretische Physik, Georg-August-Universität Göttingen

**Tag der mündlichen Prüfung:** 18.01.2019

# Abstract

The emergence of order in materials with strongly-correlated electrons in out-of-equilibrium situations inspires a lot of new research, both experimental and theoretical. The main goal of this theoretical research project is to better understand related questions in the dynamics in a strongly correlated many-body state after a photoexcitation has occurred.

Such situations are usually not fully explainable in a mean-field picture nor analytically solvable. Hence, advanced numerical techniques are necessary. In this work, we investigate non-equilibrium situations after photoexcitations. In order to model a hypothetical one-dimensional (1D) manganite, we chose the 1D Hubbard model with nearest-neighbor interaction and a staggered magnetic field. The photoexcitation is modeled in two different ways: First we investigate sudden, local excitations, and afterwards we study a semi-classical approach by applying the Peierls substitution, which leads to a time-dependent Hamiltonian.

All simulations are performed with an implementation of the time-dependent density-matrix renormalization group (DMRG), which is formulated by tensor-network states (TNSs), namely matrix-product states (MPSs) and matrix-product operators (MPOs). The framework used offers the possibility that every MPO can be externally described by finite-state machines (FSMs), hence it is extremely flexible. In this thesis, we explain how to perform exact FSM arithmetics, and how to compress the resulting FSMs. Based on MPSs and FSMs, a quantum-computer simulator (QCS) is introduced, which is mainly used as a universal tool for (MPS)-quantum-state manipulations.

From the investigations with the sudden, local excitations, we learned that the electron-electron interaction is responsible for a rapid relaxation of the magnetic moment of the individual bands. Nevertheless, this relaxation can be stalled via a stronger magnetic microstructure.

By applying a *spin-selective* photoexcitation via the Peierls substitution, we are able to induce a meta-stable charge-density wave (CDW) pattern if a magnetic microstructure is present. For a small, but finite interaction, we find a decay channel for the doublon-based part of the CDW, which still leaves a finite pattern. For large interaction, nearly no doublons are created by the photoexcitation. In the opposite limit, i.e., the non-interacting case, the two spin species are decoupled. Hence, in both limits the decay channel does not weaken the CDW and we find the pattern to be stable up to the times we can treat.





## Author's Note

Parts of this thesis have been published in peer-reviewed journals. The corresponding references are listed below. While the first two manuscripts have already been published in a peer-reviewed journal, the last two manuscripts are available as preprints.

- [KRS<sup>+</sup>18] T. Köhler, S. Rajpurohit, O. Schumann, S. Paeckel, F. R. A. Biebl, M. Sotoudeh, S. C. Kramer, P. E. Blöchl, S. Kehrein, and S. R. Manmana. Relaxation of photoexcitations in polaron-induced magnetic microstructures. *Phys. Rev. B*, 97:235120, Jun 2018
- [PKM17] S. Paeckel, T. Köhler, and S. R. Manmana. Automated construction of  $U(1)$ -invariant matrix-product operators from graph representations. *SciPost Phys.*, 3:035, 2017
- [KPM18] T. Köhler, S. Paeckel, and S. R. Manmana. Charge density patterns in spin-selectively photoexcited interacting fermions. *ArXiv e-prints*, 1808.02138, Aug 2018
- [PKS<sup>+</sup>19] S. Paeckel, T. Köhler, A. Swoboda, S. R. Manmana, U. Schollwöck, and C. Hubig. Time-evolution methods for matrix-product states. *ArXiv e-prints*, 1901.05824, Jan 2019

Beyond this thesis, there also exists a publication that is based on the author's master's thesis. The publication has also been prepared during the period of PhD work and is listed below:

- [BKT<sup>+</sup>17] J. Becker, T. Köhler, A. C. Tiegel, S. R. Manmana, S. Wessel, and A. Honecker. Finite-temperature dynamics and thermal intraband magnon scattering in Haldane spin-one chains. *Phys. Rev. B*, 96:060403, Aug 2017



# Table of contents

## I Introduction

---

<b>1 Motivation</b>	<b>3</b>
<hr/>	
<b>2 Theoretical Background</b>	<b>5</b>
2.1. Quantum Systems . . . . .	5
2.2. Quantum Computation . . . . .	6
2.3. Phases of Matter . . . . .	7
<hr/>	
<b>3 Modeling</b>	<b>11</b>
3.1. Hubbard Model . . . . .	11
3.2. Heisenberg Model – Half-Filled Strong-Coupling Limit of the Hubbard Model . .	12
3.3. One-Dimensional Praseodymium-Calcium-Manganite . . . . .	15
3.4. Excitations . . . . .	18
3.4.1. Photoexcitation . . . . .	18

## II Numerical Methods

---

<b>4 Motivation, Notation, and Frequently Used Routines</b>	<b>25</b>
4.1. Matrix Decomposition . . . . .	26
4.2. Lanczos Algorithm . . . . .	28
<hr/>	
<b>5 Matrix-Product States</b>	<b>31</b>
5.1. Canonical Form . . . . .	32
5.2. Construction of an Initial MPS . . . . .	33
5.2.1. Construction of an MPS from an Arbitrary State . . . . .	33
5.2.2. Directly Defined MPS . . . . .	33
5.3. Operations on MPSs . . . . .	35
5.3.1. Normalization of MPSs . . . . .	35
5.3.2. Truncation . . . . .	36
5.3.3. Scalar Product . . . . .	40
5.3.4. Addition and Scaling of MPSs . . . . .	40
5.3.5. Reduced Density Operator and Entanglement Entropy . . . . .	41
5.4. Finite-Temperature Representation . . . . .	42
5.4.1. Purification . . . . .	42
5.4.2. Infinite-Temperature States . . . . .	43
<hr/>	
<b>6 Matrix-Product Operators</b>	<b>45</b>
6.1. Application of an MPO to an MPS . . . . .	45

6.1.1. Direct Application . . . . .	46
6.1.2. Variational Application . . . . .	46
6.1.3. Zip-Up . . . . .	47
6.2. Matrix Elements and Expectation Values . . . . .	48
<hr/>	
<b>7 Finite-State Machines</b>	<b>51</b>
7.1. A Language that Describes All Operators . . . . .	52
7.1.1. How to get an MPO from an FSM . . . . .	53
7.1.2. Example: FSM for the Heisenberg Hamiltonian . . . . .	54
7.2. Arithmetic Operations on FSMs . . . . .	55
7.2.1. Addition . . . . .	56
7.2.2. Multiplication . . . . .	56
7.3. Compressing MPOs in Terms of FSMs . . . . .	61
7.4. Precision of the FSM Operations . . . . .	64
7.4.1. Example: Variance for S=1 Heisenberg Chains . . . . .	67
<hr/>	
<b>8 Ground-State Search</b>	<b>71</b>
<hr/>	
<b>9 Time-Evolution Schemata</b>	<b>73</b>
9.1. Trotter Time Evolution . . . . .	73
9.2. MPO $\hat{W}^{I/\Pi}$ . . . . .	74
<hr/>	
<b>10 Universal Quantum-Computer Simulator</b>	<b>77</b>
10.1. Universal Quantum Gates . . . . .	78
10.1.1. Non-Unitary Helper Operators . . . . .	78
10.1.2. Single-Site Gates . . . . .	78
10.1.3. Parametrized Single-Site Gates . . . . .	80
10.1.4. Multi-Site Gates . . . . .	82
10.2. Measurements . . . . .	84
10.3. Implementation . . . . .	85
10.4. Illustrative Applications . . . . .	88
10.4.1. Modular Multiplication Needs No Entanglement? . . . . .	88
10.4.2. (Flipped) CNOT Can Change the Control Qubit . . . . .	89
10.4.3. Benchmark . . . . .	89
10.4.4. Create Canonical Infinite-Temperature States . . . . .	90
10.4.5. Implementing Trotter Time Evolution with Two Site Gates . . . . .	91
<hr/>	
<b>III Photoexcitation Dynamics in 1D PCMO-like Model Systems</b>	
<hr/>	
<b>11 Sudden Local Photoexcitation</b>	<b>95</b>
11.1. Local Density Dynamics After a Photoexcitation . . . . .	95
11.2. Electronic Momentum-Distribution Function . . . . .	100
11.3. Final Temperature of the Excited State . . . . .	103
11.4. Estimation of the Energy Density of the Excitation . . . . .	105
<hr/>	
<b>12 Broad, Continuous, and Intensive Excitation</b>	<b>107</b>
12.1. Periodic Charge-Density Patterns Through Spin-Selective Photoexcitation . . . . .	107
12.1.1. CDW-Pattern for Various Unit Cells and Fillings . . . . .	108

12.2. Mechanism for the Formation of the CDW . . . . .	109
12.3. Effect of Interactions . . . . .	111
12.3.1. CDW Amplitude versus Interaction Strength . . . . .	112
12.3.2. Doublyon-Purified Particle Densities . . . . .	114
12.4. Possible Experimental Realizations . . . . .	115

---

<b>13 Conclusion</b>	<b>117</b>
13.1. Retrospective . . . . .	117
13.2. Perspective . . . . .	118

## Appendices

---

<b>A Appendix FSMs</b>	<b>123</b>
A.1. Entangler FSM . . . . .	123
A.2. Peierls FSM . . . . .	124
A.3. Heisenberg Trotter Gate FSM . . . . .	124
A.4. Full Two Site Gates FSM . . . . .	125
A.4.1. Spin-1/2 . . . . .	125
A.4.2. Fermionic . . . . .	125

---

<b>B Heisenberg Trotter Gate</b>	<b>127</b>
B.1. Exact Heisenberg Trotter Gate . . . . .	127
B.2. Numeric Heisenberg Trotter Gate . . . . .	127

---

<b>References in Alphabetical Order of Labels</b>	<b>131</b>
---	------------



# Introduction

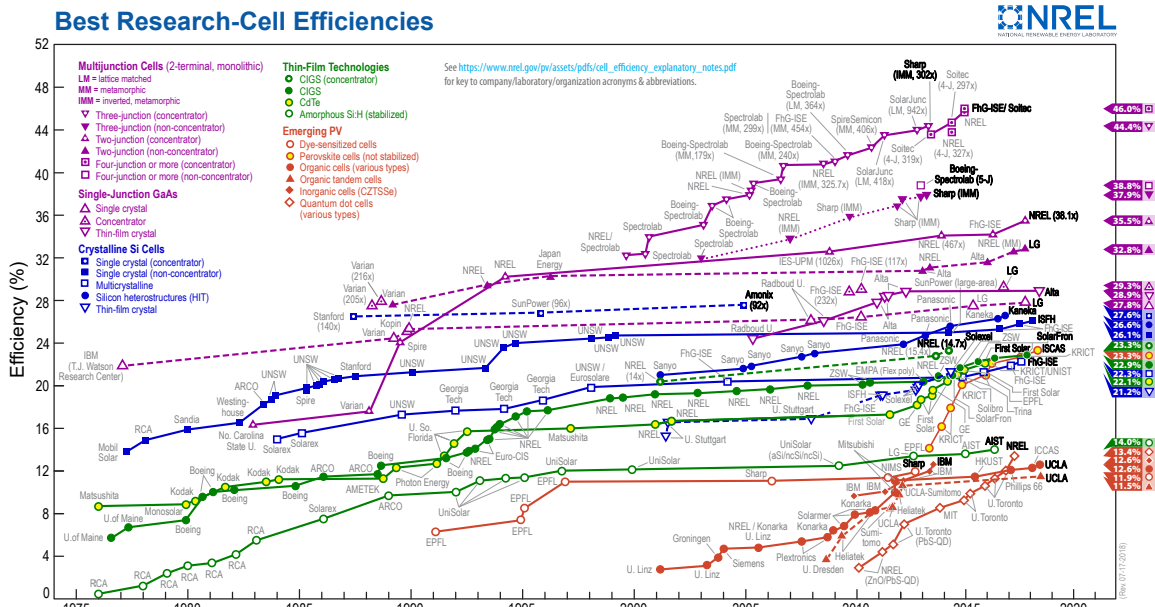






# Motivation | 1

It is the prevalent opinion in the scientific community that the increasing energy demand of humanity can in the long run only be met by renewable energies or fusion reactors. The latter ones are still in development and first results about their economic application are not expected before 2035 [Ite01]. Therefore, optimizing already accessible renewable energy sources is mandatory. One very promising technology is photovoltaics, which continues to increase in efficiency (see Fig. 1.1) and to decrease in price per kWh. In order for those values to continue their current trend, further research on all levels, i.e., from manufacturing details up to fundamental research, is necessary.



**Figure 1.1:** Current best research-cell efficiencies. This plot is courtesy of the National Renewable Energy Laboratory, Golden, CO. [PVE03]

This thesis aims to contribute in the realm of fundamental research. Understanding the exact process of photoexcitations and the subsequent interaction within the photo-excited environment, even in model systems, is of utmost importance. The motivation for this work is to better understand the effect of photoexcitations in interacting, strongly correlated many-body quantum systems. The main focus concerns the question whether or not there are differences due to interaction effects.

Different experiments, e.g., with ultracold gases in optical lattices [Blo05, BDZ08], show that model systems can be realized and studied with increasing accuracy. Hence, the urge for better theoretical methods, with predictive power, is increasing. Fortunately, computational resources still nearly follow Moore’s law [Moo65, Moo17] and therefore numerical tools have become a valid option for problems that resist analytic solvability, e.g., by means of the Bethe ansatz [Bet31].

In fact, a multitude of numerical methods exist that aim at approximate solutions for different classes of problems.

The remainder of this introduction starts with the basic quantum-mechanical notion of strongly correlated systems. Afterwards, quantum computing is briefly introduced and a short summary of phases of matter is presented. Next, all models and excitations that are used in this thesis are introduced.

The second part of this thesis is dedicated to the numerical methods that were used, implemented, and, in some parts, developed for this thesis. It begins with a short introduction of the notation and some algorithms that will be used throughout the following, more detailed, description of matrix-product states (MPSs) and matrix-product operators (MPOs). Afterwards, we introduce finite-state machines (FSMs) as a method to create MPOs and compare operations on FSMs with their numerical counterparts on MPOs. The following three sections make heavy use of the introduced objects and describe the ground-state search, time-evolution schemata, and a universal quantum-computer simulator that is based on MPSs.

The last part combines all of the afore described theory and methods in order to investigate dynamics after sudden, local photoexcitations and spatially and temporally extended photoexcitations. We present the effect of finite interactions and different magnetic microstructures on the propagation of both types of photoexcitations.

In the end we conclude with a retrospective summary and discuss some prospectives for future developments.

# Theoretical Background

2

## Quantum Systems

2.1

Generally, the state of a quantum system on a lattice at temperature  $T = 0$  can be described by a wave function  $|\psi\rangle \in \mathcal{H}^L$  with  $\mathcal{H}$  denoting a local Hilbert space on  $L$  distinct positions. Note that such a state can represent a single particle,  $N = 1$ , but also many particles,  $N > 1$ . Many-particle states give birth to effects like quantum entanglement [Sch35], the EPR paradox [EPR35], and the (not yet proven) quantum supremacy [Pre12].

In order to describe the time evolution of such a state in a system, which is given in terms of a Hermitian Hamiltonian  $\hat{H} : \mathcal{H}^L \rightarrow \mathcal{H}^L$ , *all we need to do* is solve the time-dependent Schrödinger equation [Sch26]

$$i\hbar \frac{\partial}{\partial \tau} |\psi\rangle = \hat{H} |\psi\rangle , \quad (2.1)$$

with  $\hbar$  denoting the reduced Planck constant and  $\tau$  denoting the time. In the case of a time-independent Hamiltonian, we obtain the formal solution

$$|\psi(\tau)\rangle = \underbrace{e^{-\frac{i}{\hbar} \hat{H} \tau}}_{\hat{U}(\tau)} |\psi(0)\rangle , \quad (2.2)$$

which defines the time-evolution operator  $\hat{U}(\tau)$ . For systems for which we can diagonalize the Hamiltonian exactly, we can describe the time evolution for every point in time. Those systems distinguish themselves by being small (diagonalization possible numerically, e.g., [San10]) or by being integrable (diagonalization possible analytically, e.g., [Bet31]). In order to investigate larger, non-integrable systems, further approximations and numerical approaches are necessary. The properties of a quantum state at very low temperatures  $T \approx 0$  can be investigated in the ground state  $|\psi_0\rangle$ . It is defined as the (possibly degenerated) state with the lowest energy  $E_0$ . The energy  $E$ , however, is given by the time-independent Schrödinger equation,

$$E |\psi\rangle = \hat{H} |\psi\rangle . \quad (2.3)$$

In this thesis, we usually start our investigations by exciting a ground state via a local or global excitation.

In principle, there are two operations that can be performed with quantum mechanical states: Applying an arbitrary operator

$$|\tilde{\psi}\rangle = \hat{O} |\psi\rangle \quad (2.4)$$

and measuring overlaps between two states  $|\psi\rangle$  and  $|\psi'\rangle$ ,

$$O = \langle \psi | \psi' \rangle \in \mathbb{C} . \quad (2.5)$$

From these two operations we also derive the measurement of an observable with a Hermitian operator  $\hat{k}$

$$k_{|\psi\rangle} = \langle \psi | \hat{k} | \psi \rangle = \langle \hat{k} \rangle \in \mathbb{R} . \quad (2.6)$$

Note that the limitation to Hermitian operators arises due to the fact that only real numbers can be measured in experiments.

In this thesis, we are particularly interested in the difference between different classes of quantum states, namely weakly correlated and strongly correlated states. Whereas weakly correlated quantum states can be well described by mean-field theory, the description might fail for strongly correlated systems. A general (two-point) correlation function of a state  $\psi$  is defined by

$$c_{|\psi\rangle}^{\hat{O}_j(\tau), \hat{O}'_{j'}(\tau')} = \langle \hat{O}_j(\tau) \hat{O}'_{j'}(\tau') \rangle , \quad (2.7)$$

with the positions  $j$  and  $j'$ , the times  $\tau$  and  $\tau'$ , and the operators  $\hat{O}$  and  $\hat{O}'$ . In order to obtain the strength of the correlations, the bare correlation function is not sufficient. Instead we need the connected correlation function,

$$\tilde{c}_{|\psi\rangle}^{\hat{O}_j(\tau), \hat{O}'_{j'}(\tau')} = \langle \hat{O}_j(\tau) \hat{O}'_{j'}(\tau') \rangle - \langle \hat{O}_j(\tau) \rangle \langle \hat{O}'_{j'}(\tau') \rangle \quad (2.8)$$

$$= \langle (\hat{O}_j(\tau) - \langle \hat{O}_j(\tau) \rangle) (\hat{O}'_{j'}(\tau') - \langle \hat{O}'_{j'}(\tau') \rangle) \rangle , \quad (2.9)$$

which measures the fluctuations around the mean value of the observables.

It is challenging to describe strongly correlated states with methods like density-functional theory (DFT) [HK64] with approximated local (local-density approximation (LDA) [VWN80]) or semi-local functionals. Hence, algorithms were developed that capture the main features of such states, e.g., dynamical mean-field theory (DMFT) [GK92] and extensions of LDA [Bie14]. Furthermore, methods like exact diagonalization (ED) (e.g., [San10]), quantum Monte Carlo (QMC) (e.g., [FMNR01, TW05]), and density-matrix renormalization group (DMRG) (e.g., [Whi92, Whi93, Sch11, PWKH99]) were invented, which all capture the correlations very well but have different, partly severe, restrictions. If one, nevertheless, wants to simulate those systems in an exact way, the only alternative is a quantum simulator [Fey82].

## Quantum Computation | 2.2

The theory of quantum computation has first reached broad publicity in the 1980s after Richard Feynman proposed quantum simulators [Fey82].<sup>1</sup> In the following decades, some theoretical work was done (see [NC04] and references within). The topic got another big boost in attention after Peter Shor proposed his quantum algorithm to effectively<sup>2</sup> factorize numbers in 1994 [Sho94, Sho97]. Since then, several additional quantum algorithms have been developed for theoretical, faultless quantum computers (QCs). On the experimental side, there were two big occurrences, firstly the quantum annealing processors demonstrated by D-Wave in 2007 ([DWa, JAG<sup>+</sup>11]) and secondly the IBM Q Experience [IBM] that was launched in 2016 and made a 5 qubit quantum computer available to the general public. Since the launch, and often in combination with

<sup>1</sup>Actually, the Russian-born mathematician Yuri Manin had proposed a similar idea, but less clearly, already 2 years earlier [Man80]. Because the scientific community was, due to political reasons, not yet globally connected, Feynman introduced the idea independently.

<sup>2</sup>An algorithm is effective if and only if (iff) it does scale polynomially.

that particular QC, an ever growing number of publications have been released [Lis]. Lately, a perceptron [Ros57], which is part of a very simple neural network, was simulated on the 5 qubit system [TMGB18] that got coverage in general media [hei]. In order to further push the boundaries, the Lawrence Berkeley National Laboratory (LBNL) received a \$30 million grant “to build and operate an Advanced Quantum Testbed” [lbn]. The main goal of that project will be to provide different quantum-computing resources to other researchers and to further develop algorithms like variational-quantum eigensolver (VQE) [PMS<sup>+</sup>14]. An overview of different possible experimental realizations of QCs is given in Ref. [NC04]. It includes quantum simulators like ultracold gases in optical lattices, and superconducting quantum computing in which the qubits are realized as Josephson junctions [Jos62, Jos74]. The latter one is used by the IBM Q Experience [IBM].

In chapter 10, we introduce a quantum-computer simulator (QCS) based on matrix-product states (MPSs) and finite-state machines (FSMs). Being aware of the work of Vidal in the early 2000s [Vid03], we do not claim to be able to run every quantum algorithm nor do we expect the speed-up of a QC. Instead, we emphasize the usage of the QCS as a way to check for the necessity of a QC for a given quantum algorithm and as a quantum-algorithm debugger. Furthermore, the question arises how quantum algorithms perform if the qubits are only stored in an approximated manner. At the time of writing, we do not have an answer to that question, but developed the machinery to perform further investigations.

As one main focus of this thesis is the question of the creation of a metastable phase due to a photoexcitation, we next introduce phases of matter; we start with the classical description and afterwards extend it to the quantum world.

## Phases of Matter | 2.3

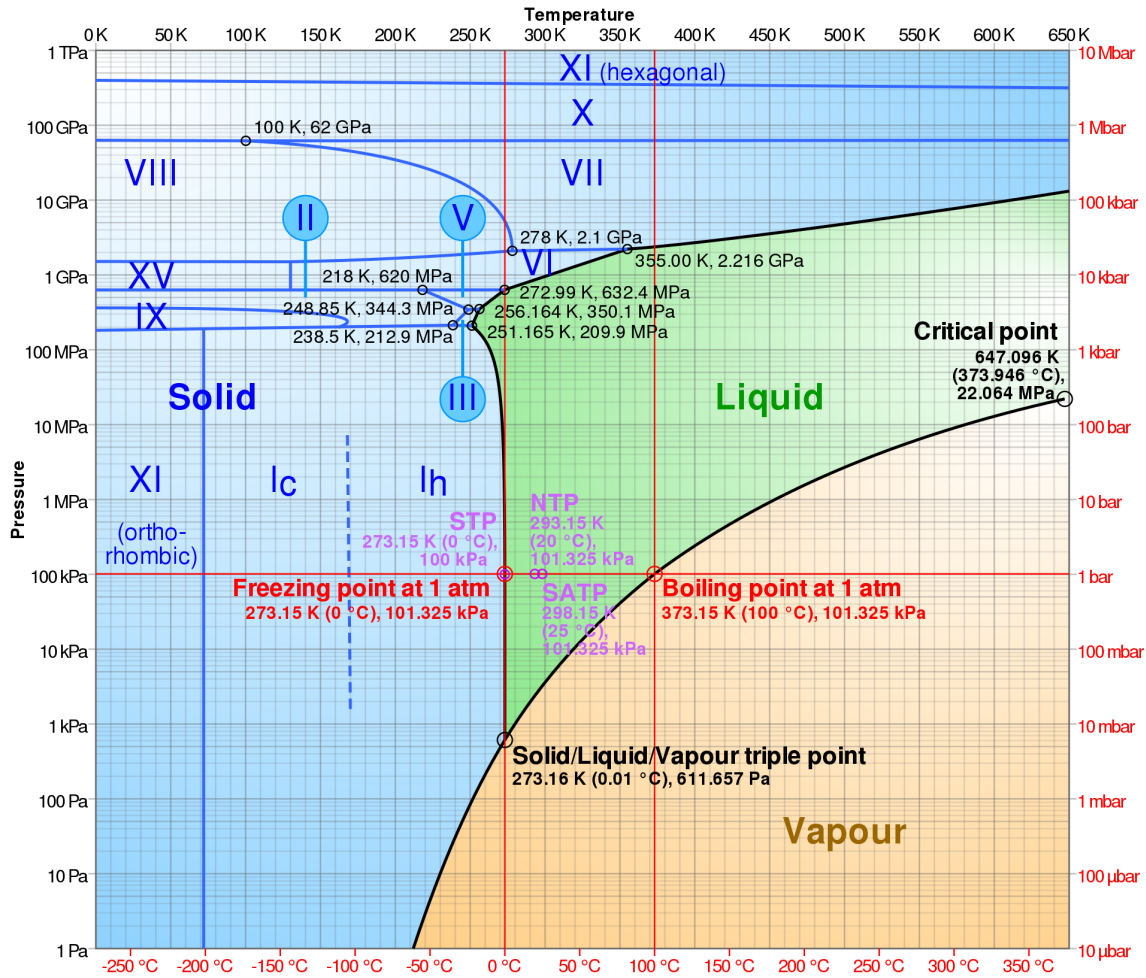
In the classical world, we are able to summarize states of matter into phases, e.g., water can be in different states, which can be described by its thermodynamical properties (temperature  $T$ , pressure  $p$ , volume  $V$ , number of particles  $N$ , entropy  $S$ , and chemical potential  $\mu$ ). These states can be summarized into phases; in case of water, these would be solid, liquid, and vapor.<sup>3</sup> These summaries can be depicted in phase diagrams like Fig. 2.1, in which also the critical point, at which the distinction between liquid and vapor is not valid anymore and the triple point at which all three phases intersect are shown.

Besides these commonly known kinds of phases, the same concept can be applied also to other properties, e.g., magnetic phases of iron or electronic phases of matter. Phase transitions are points in parameter space at which a sudden change of physical properties occurs [Ess]. This statement can be formulated more precisely with the help of the free energy,

$$F(N, V, T) = -k_B T \ln(Z(N, V, T)) , \quad (2.10)$$

where  $k_B$  is the Boltzmann constant and  $Z(N, V, T)$  is the partition function. Consider the thermodynamic limit, then a phase transition is a point in parameter space at which  $F$  becomes a non-analytic function of one of its parameters. Via the Ehrenfest classification it is possible to classify phase transitions of different orders. This order is determined by the lowest derivative of the free energy with respect to the parameter from which the non-analyticity arises [Jae98]. An important quantity for the investigation of phase transitions is the order parameter, which

<sup>3</sup>Plasma, which is also a phase of water, is omitted here, because it is not observed in everyday life and thus does not help the illustration.



**Figure 2.1:** Phase diagram of water with the three phases solid, liquid, and vapor. By Cmglee - Own work, CC BY-SA 3.0 [Wat].

is usually finite in the ordered phase and zero otherwise. The order parameter needs to be chosen appropriately to the phase transition, e.g., the difference in density for liquid-to-vapor transitions or the magnetization for paramagnet-to-ferromagnet transition. States with finite order parameter often originate from broken symmetries. If the Hamiltonian is symmetric and it exists a ground state that is not symmetric, we can encounter a symmetry breaking due to fluctuations. Such a symmetry breaking is called spontaneous.

The Mermin-Wagner theorem, named after David Mermin and Herbert Wagner, states that in low dimensions, i.e., in less than or equal to 2 dimensions, continuous symmetries cannot be spontaneously broken if the classical system is at finite temperature and the interactions are short-ranged. [MW66]

Two phases, in particular, are important for this thesis: The charge-density wave (CDW) and the spin-density wave (SDW). The former can be depicted as a charge density that is periodically modulated with a wavelength  $\lambda$ . The latter can be understood as a combination of two CDWs that are 180° out-of-phase.

Up to this point, we have considered phase transitions that are driven by thermal fluctuations. In the quantum world, we would like to know whether there are still phase transitions in the ground state, i.e., at  $T = 0$ . In fact, due to quantum fluctuations, which arise from the Heisenberg

uncertainty principle, we obtain a quantum phase transition (QPT) at a point in parameter space at  $T = 0$ , which is called quantum critical point (QCP) [Sac99]. It is important to note that in experiments  $T = 0$  is not reachable, but the existence of a QCP can already be deduced in its vicinity, which is also called the quantum critical regime.

In the previous considerations, we presumed that all states were in thermal equilibrium. That means macroscopic, thermodynamical properties are constant in time and there are no macroscopic currents, whereas microscopic observables can vary. An illustrative example for an equilibrium state is an ideal gas in a microcanonical ensemble, in which the pressure is constant but the position of the particles is not.

In contrast, if an equilibrium state is excited, local observables like the local density can show time-dependent behavior, hence it is an out-of-equilibrium state. After an excitation, the state will eventually transition into an equilibrated state. Note that this state does not necessarily need to be in thermal equilibrium, but can also be described by a steady-state. One way to check the kind of equilibrated state is to compare its observables with those of a thermal state that has the same energy. In fact, the question whether or not systems transition into their thermal equilibrium is part of current scientific investigations. One main hypothesis is the eigenstate thermalization hypothesis (ETH) [PGD13, RDO08], which states that for an arbitrary initial state an observable  $\langle \hat{A} \rangle$  will, in time, evolve into its corresponding thermal value if two conditions are met. Those two conditions are: First, the diagonal, i.e.,  $\alpha = \alpha'$ , matrix elements

$$A_{\alpha, \alpha'} = \langle E_{\alpha} | \hat{A} | E_{\alpha'} \rangle , \quad (2.11)$$

with  $|E_{\alpha}\rangle$  being energy eigenstates, only vary smoothly for adjacent  $\alpha$ . And second, the off-diagonal elements  $A_{\alpha, \alpha'}$  with  $\alpha \neq \alpha'$  are much smaller than the diagonal elements. Nevertheless, it is already known that this hypothesis does not always hold, e.g., in the case of an extensive number of conserved quantities [RS12].

In order to be able to investigate properties of explicit systems, we introduce their Hamiltonians in the next chapter.





Due to the exponentially growing Hilbert space, it is in general not possible to describe macroscopic matter by its full wave function. Hence, we always need to apply approximations in order to be able to predict certain observables. In this thesis, we make extensive use of the density-matrix renormalization group (DMRG) [Whi92, Whi93, Sch11, PWKH99] and hence we restrain ourself to the one-dimensional (1D) version of each model, as DMRG works well in 1D. Furthermore, we assume open-boundary conditions (OBC) if not stated otherwise.

We start with the general Hubbard model, which describes interacting electrons on a lattice. Afterwards, we deduce the Heisenberg model, which is used to describe ferro- and antiferromagnetism, as the half-filled strong-coupling limit of the Hubbard model. Note that we use the Heisenberg model in this thesis mainly for illustration. The last model we introduce is the Hubbard-like model for a hypothetical 1D praseodymium-calcium-manganite ( $\text{Pr}_{1-x}\text{Ca}_x\text{MnO}_3$ , if  $x = 0.5$  also PCMO). There are exact solutions for the first two models in some limits, whereas the latter one has, to our knowledge, none<sup>1</sup>.

Furthermore, we present different modeling approaches for excitations. There, we start with a discussion of general excitations and afterwards concentrate on different photoexcitations. Two types of photoexcitations were used to produce the results in part III.

## Hubbard Model

### 3.1

The Hubbard model, named after John Hubbard [Gut63, Kan63, Hub63, Hub64a, Hub64b, Hub65], is a simple model to describe strongly correlated electron systems. Despite its apparent simplicity, it is used widely, e.g., to attempt to describe the electronic properties of high-temperature superconduction [EFG<sup>+</sup>05]. Furthermore, it captures the Mott metal-insulator transition [Hub64b, Geb97]. The 1D Hamiltonian in second quantization is given by

$$\hat{H}_{\text{Hubbard}} = - \sum_{j,\sigma} \left( t_j \hat{c}_{\sigma,j+1}^\dagger \hat{c}_{\sigma,j} + t_j^\dagger \hat{c}_{\sigma,j}^\dagger \hat{c}_{\sigma,j+1} \right) + \sum_j U_j \hat{n}_{\uparrow,j} \hat{n}_{\downarrow,j}, \quad (3.1)$$

in which  $\hat{c}_{\sigma,j}^\dagger$  is the creation operator of an electron with spin  $\sigma \in \{\uparrow, \downarrow\}$  on site  $j$ ,  $\hat{c}_{\sigma,j}$  is the annihilation operator of an electron with spin  $\sigma$  on site  $j$ , and  $\hat{n}_{\sigma,j} = \hat{c}_{\sigma,j}^\dagger \hat{c}_{\sigma,j}$  is the particle-number operator for electrons with spin  $\sigma$  on site  $j$ . The parameter  $t_j$  describes the electron transfer matrix between neighboring sites, and the local Coulomb interaction is parametrized by  $U_j$ . The creation and annihilation operators obey the anticommutator relations,

$$\left\{ \hat{c}_{\sigma,j}, \hat{c}_{\sigma',j}^\dagger \right\} = \hat{c}_{\sigma,i} \hat{c}_{\sigma',j}^\dagger + \hat{c}_{\sigma',j}^\dagger \hat{c}_{\sigma,i} = \delta_{ij} \delta_{\sigma'\sigma} \quad \left\{ \hat{c}_{\sigma,i}^\dagger, \hat{c}_{\sigma',j}^\dagger \right\} = \left\{ \hat{c}_{\sigma,i}, \hat{c}_{\sigma',j} \right\} = 0. \quad (3.2)$$

<sup>1</sup>Of course, except in the limit, in which it is equal to the Hubbard model.

We furthermore define the particle density  $\hat{n}_j = \hat{n}_{\uparrow,j} + \hat{n}_{\downarrow,j}$  and the spin  $\hat{S}_j^z = \frac{1}{2}(\hat{n}_{\uparrow,j} - \hat{n}_{\downarrow,j})$  on site  $j$ . With this, we can define two conserved quantities of Eq. (3.1),

$$\left[ \hat{H}_{\text{Hubbard}}, \hat{N} \right] = 0 \quad \left[ \hat{H}_{\text{Hubbard}}, \hat{S}^z \right] = 0, \quad (3.3)$$

namely the total particle number and the total magnetization,

$$\hat{N} = \sum_j^L \hat{n}_j \quad \hat{S}^z = \sum_j^L \hat{S}_j^z. \quad (3.4)$$

In the non-interacting case ( $U = 0$ ), also the particle number for every momentum is conserved<sup>2</sup>,

$$\hat{n}_{\sigma,k} = \hat{c}_{\sigma,k}^\dagger \hat{c}_{\sigma,k} = \frac{1}{L} \left( \sum_j e^{ijk} \hat{c}_{\sigma,j}^\dagger \right) \left( \sum_j e^{ijk} \hat{c}_{\sigma,j} \right). \quad (3.5)$$

Thus, the Hamiltonian can be diagonalized,

$$\hat{H}_{\text{Hubbard}}^{U=0} = \sum_{\sigma,k} \epsilon(k) \hat{c}_{\sigma,k}^\dagger \hat{c}_{\sigma,k}, \quad (3.6)$$

which yields the dispersion relation

$$\epsilon(k) = -2t \cos(k). \quad (3.7)$$

The Hubbard model in 1D is integrable and (hence) exactly solvable for all fillings and all interaction strengths. Lieb and Wu found that a variant of the Bethe ansatz [Bet31] can be used to find an exact solution for the ground state [LW68]. Later, Takahashi extended this solution in order to describe also thermodynamic properties [Tak72].

Next, we introduce the Heisenberg model as the half-filled strong-coupling limit of the Hubbard model. Afterwards, we present a variation of the Hubbard model, which possesses an additional term that represents an external magnetic field.

## Heisenberg Model – Half-Filled Strong-Coupling Limit of the Hubbard Model

3.2

In this section, we introduce the Heisenberg model and show how to obtain it as the half-filled strong-coupling limit of the Hubbard model. We follow the thorough derivation of Ref. [NR09], but adopt it to the already introduced nearest-neighbor model. The Heisenberg model itself is widely used to describe magnetic insulators and “can be considered to be, till today, the most intensely worked and also the best understood model of magnetism” [NR09]. This statement was underpinned also by the Nobel prize for Haldane in 2016 [Hal17], who worked on topological phases of matter, e.g., on the Affleck-Lieb-Kennedy-Tasaki (AKLT) model [AKLT87, Hal83], which is related to the spin-1 Heisenberg chain. Although we have also investigated the spin-1 Heisenberg model in Ref. [BKT<sup>+</sup>17], in this thesis it serves mainly illustrative purposes.

A flexible formulation of the Hamiltonian with nearest-neighbor interaction is

$$\hat{H} = \sum_{j=1}^{L-1} \left[ \frac{J_j}{2} \left( \hat{S}_j^+ \hat{S}_{j+1}^- + \hat{S}_j^- \hat{S}_{j+1}^+ \right) + J_j^z \hat{S}_j^z \hat{S}_{j+1}^z \right], \quad (3.8)$$

<sup>2</sup>The Fourier transformation is, of course, only exact if periodic-boundary conditions (PBCs) are assumed, which we therefore do here.

in which  $\hat{S}_j^\pm = \hat{S}_j^x \pm i\hat{S}_j^y$  and  $\hat{S}_j^z$  are the common spin operators on site  $j$ . In the case  $J_j = 0$  and  $J_j^z \neq 0 \forall j$ , the Ising model is obtained. In the opposite case,  $J_j \neq 0$  and  $J_j^z = 0 \forall j$ , the model is called the XX-model. If  $J_j = J_j^z \forall j$ , the original Heisenberg model

$$\hat{H}_{\text{Heisenberg}} = \sum_j J_j \hat{S}_j \hat{S}_{j+1}, \quad (3.9)$$

with the spin operator  $\hat{S}_j$  and limited to nearest-neighbor interaction, is obtained [Hei28]. Next, we derive the Heisenberg model as the half-filled strong-coupling limit of the Hubbard model, that means  $\langle \hat{N} \rangle = L$  and  $U \gg t$ . We do this by treating the hopping term as perturbation  $\hat{H}_1$ ,

$$\hat{H}_{\text{Hubbard}} = \underbrace{\sum_j U_j \hat{n}_{\uparrow,j} \hat{n}_{\downarrow,j}}_{\hat{H}_0} + \underbrace{\sum_{j,\sigma} t_j \left( \hat{c}_{\sigma,j+1}^\dagger \hat{c}_{\sigma,j} + \hat{c}_{\sigma,j}^\dagger \hat{c}_{\sigma,j+1} \right)}_{\hat{H}_1} \quad (3.10)$$

of the unperturbed system  $\hat{H}_0$ . For simplicity, we consider the interaction strength  $U_j$  in the following to be site independent, i.e.,  $U_j = U \forall j$ . The eigenstates of  $\hat{H}_0$  can be characterized by the number of doubly occupied sites  $D$ , which leads to highly degenerate eigenstates  $|D, \alpha\rangle$  with energy  $E_D = UD$ , where  $\alpha$  lifts the degeneracy. From first-order perturbation theory<sup>3</sup> we identify  $E_0^{(1)}$  via

$$\det \left( {}^{(0)}\langle 0, \alpha' | \hat{H}_1 | 0, \alpha \rangle^{(0)} - E_0^{(1)} \delta_{\alpha, \alpha'} \right) \stackrel{!}{=} 0. \quad (3.11)$$

Because  $\hat{H}_1$  always creates an empty and a doubly occupied site and

$${}^{(0)}\langle D, \alpha' | \hat{H}_1 | 0, \alpha \rangle^{(0)} \neq 0 \quad \text{if and only if (iff) } D = 1, \quad (3.12)$$

we obtain the still degenerate energy  $E_0^{(1)} = 0$ . Hence, we need to apply second-order perturbation theory, i.e., we need to solve

$$\sum_{\alpha} \left( \sum_{D, \beta}^{D \neq 0} \frac{{}^{(0)}\langle 0, \alpha' | \hat{H}_1 | D, \beta \rangle^{(0)} {}^{(0)}\langle D, \beta | \hat{H}_1 | 0, \alpha \rangle^{(0)}}{E_0^{(0)} - E_D^{(0)}} - E_0^{(2)} \delta_{\alpha, \alpha'} \right) \stackrel{!}{=} 0, \quad (3.13)$$

which, if interpreted as eigenvalue equation, leads to a new Hamiltonian  $\hat{H}_{\text{new}}$ . Because of Eq. (3.12) only summands with  $D = 1$  contribute and hence  $E_D^{(0)} \rightarrow E_1^{(0)}$ . Consequently, we can omit the limitation  $D \neq 0$  and obtain the matrix elements of the new Hamiltonian,

$$\langle 0, \alpha' | \hat{H}_{\text{new}} | 0, \alpha \rangle = \frac{1}{E_0^{(0)} - E_1^{(0)}} {}^{(0)}\langle 0, \alpha' | \hat{H}_1 \left( \underbrace{\sum_{|D, \beta\rangle} |D, \beta\rangle^{(0)} \langle D, \beta|}_{=\hat{\mathbf{1}}} \right) \hat{H}_1 | 0, \alpha \rangle^{(0)} \quad (3.14)$$

$$= -\frac{1}{U} {}^{(0)}\langle 0, \alpha' | \hat{H}_1^2 | 0, \alpha \rangle^{(0)}. \quad (3.15)$$

<sup>3</sup>We denote the order of the object with an upstairs bracketed index.

In order to read off the new Hamiltonian, we introduce the projector  $\hat{P}_0 = \sum_{\alpha} |0, \alpha\rangle \langle 0, \alpha|$ , which projects into the manifold of the ground states of  $\hat{H}_0$ , i.e., all states with  $\langle \hat{n}_j \rangle = 1 \forall j$ . The new Hamiltonian is then given by

$$\hat{H}_{\text{new}} = -\frac{1}{U} \hat{P}_0 \left( \sum_{j,\sigma} t_j \left( \hat{c}_{\sigma,j+1}^{\dagger} \hat{c}_{\sigma,j} + \hat{c}_{\sigma,j}^{\dagger} \hat{c}_{\sigma,j+1} \right) \right) \left( \sum_{j',\sigma'} t_{j'} \left( \hat{c}_{\sigma',j'+1}^{\dagger} \hat{c}_{\sigma',j'} + \hat{c}_{\sigma',j'}^{\dagger} \hat{c}_{\sigma',j'+1} \right) \right) \hat{P}_0 \quad (3.16)$$

$$= -\frac{1}{U} \hat{P}_0 \sum_{\substack{j,j' \\ \sigma,\sigma'}} t_j t_{j'} \left( \underbrace{\hat{c}_{\sigma,j+1}^{\dagger} \hat{c}_{\sigma,j} \hat{c}_{\sigma',j'+1}^{\dagger} \hat{c}_{\sigma',j'}}_{=0} + \underbrace{\hat{c}_{\sigma,j+1}^{\dagger} \hat{c}_{\sigma,j} \hat{c}_{\sigma',j'}^{\dagger} \hat{c}_{\sigma',j'+1}}_{\neq 0 \text{ iff } j=j'} + \underbrace{\hat{c}_{\sigma,j}^{\dagger} \hat{c}_{\sigma,j+1} \hat{c}_{\sigma',j'+1}^{\dagger} \hat{c}_{\sigma',j'}}_{\neq 0 \text{ iff } j=j'} + \underbrace{\hat{c}_{\sigma,j}^{\dagger} \hat{c}_{\sigma,j+1} \hat{c}_{\sigma',j'}^{\dagger} \hat{c}_{\sigma',j'+1}}_{=0} \right) \hat{P}_0, \quad (3.17)$$

in which the underbraced equalities become only valid because of the subsequent projection. In the following, we insert  $\bar{\sigma}$  as the opposite spin to  $\sigma$  and rearrange the terms of Eq. (3.17) that survive the projection in order to omit  $\sigma'$ ,

$$\hat{H}_{\text{new}} = -\frac{1}{U} \hat{P}_0 \left( \sum_{j,\sigma,\sigma'} t_j^2 \left( \hat{c}_{\sigma,j+1}^{\dagger} \hat{c}_{\sigma',j+1} \underbrace{\hat{c}_{\sigma,j} \hat{c}_{\sigma',j}^{\dagger}}_{=\delta_{\sigma,\sigma'} - \hat{c}_{\sigma',j}^{\dagger} \hat{c}_{\sigma,j}} + \hat{c}_{\sigma,j}^{\dagger} \hat{c}_{\sigma',j} \underbrace{\hat{c}_{\sigma,j+1} \hat{c}_{\sigma',j+1}^{\dagger}}_{=\delta_{\sigma,\sigma'} - \hat{c}_{\sigma',j+1}^{\dagger} \hat{c}_{\sigma,j+1}} \right) \right) \hat{P}_0 \quad (3.18)$$

$$= -\frac{1}{U} \hat{P}_0 \left( \sum_{j,\sigma} t_j^2 \left( \hat{n}_{\sigma,j+1} - \hat{n}_{\sigma,j+1} \hat{n}_{\sigma,j} + \hat{n}_{\sigma,j} - \hat{n}_{\sigma,j} \hat{n}_{\sigma,j+1} - \hat{c}_{\sigma,j+1}^{\dagger} \hat{c}_{\bar{\sigma},j+1} \hat{c}_{\bar{\sigma},j}^{\dagger} \hat{c}_{\sigma,j} - \hat{c}_{\sigma,j}^{\dagger} \hat{c}_{\bar{\sigma},j} \hat{c}_{\bar{\sigma},j+1}^{\dagger} \hat{c}_{\sigma,j+1} \right) \right) \hat{P}_0. \quad (3.19)$$

Next, we reintroduce the spin operators,

$$\hat{S}_j^z = \frac{1}{2} (\hat{n}_{\uparrow,j} - \hat{n}_{\downarrow,j}) \quad (3.20)$$

$$\hat{S}_j^+ = \hat{c}_{\uparrow,j}^{\dagger} \hat{c}_{\downarrow,j} \quad \hat{S}_j^- = \hat{c}_{\downarrow,j}^{\dagger} \hat{c}_{\uparrow,j}, \quad (3.21)$$

in terms of the creation and annihilation operators. This leads, in particular, to

$$\hat{S}_j^z \hat{S}_{j+1}^z = \frac{1}{4} (\hat{n}_{\uparrow,j} \hat{n}_{\uparrow,j+1} - \hat{n}_{\uparrow,j} (1 - \hat{n}_{\uparrow,j+1}) - \hat{n}_{\downarrow,j} (1 - \hat{n}_{\downarrow,j+1}) + \hat{n}_{\downarrow,j} \hat{n}_{\downarrow,j+1}) \quad (3.22)$$

$$= \frac{1}{4} \sum_{\sigma} (2\hat{n}_{\sigma,j} \hat{n}_{\sigma,j+1} - \hat{n}_{\sigma,j}). \quad (3.23)$$

Inserting Eqs. (3.21) and (3.22) into Eq. (3.19) we obtain

$$\hat{H}_{\text{new}} = -\frac{1}{U} \hat{P}_0 \left( \sum_j t_j^2 \left( \underbrace{\hat{n}_{\uparrow,j+1} + \hat{n}_{\downarrow,j+1}}_{\hat{1}} - 4 \left( \hat{S}_j^z \hat{S}_{j+1}^z + \frac{1}{2} \left( \hat{S}_j^+ \hat{S}_{j+1}^- + \hat{S}_j^- \hat{S}_{j+1}^+ \right) \right) \right) \right) \hat{P}_0 \quad (3.24)$$

$$= \hat{P}_0 \left( \underbrace{-\sum_j \frac{t_j^2}{U} \hat{1}}_{\text{const}} - \sum_j \frac{4t_j^2}{U} \hat{S}_j \hat{S}_{j+1} \right) \hat{P}_0, \quad (3.25)$$

which is up to an “(actually unimportant) constant”<sup>4</sup> the spin-1/2 Heisenberg Hamiltonian Eq. (3.8) [NR09] with  $J_j = -\frac{4t_j^2}{U}$ . Note that the projectors are still needed in order to use the Heisenberg Hamiltonian, which usually describes systems with only two physical basis states per site, in the larger basis.

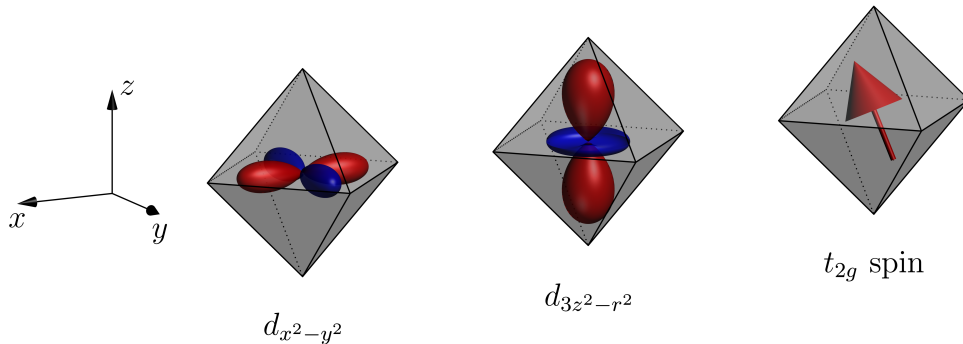
In the next section, we introduce the hypothetical 1D praseodymium-calcium-manganite (PCMO) model, which is used throughout this thesis.

## One-Dimensional Praseodymium-Calcium-Manganite

## 3.3

The main content of this section is published as a regular article in Physical Review B [KRS<sup>+</sup>18]. Whereas the derivations of the model and the band structures were mainly done by my coauthors Sangeeta Rajpurohit and Fabian Biebl, the parts of the final manuscript that are used here were written primarily by me and revised together with the coauthors. It is therefore mainly adopted from the manuscript [KRS<sup>+</sup>18].

In Ref. [KRS<sup>+</sup>18], a hypothetical model for an 1D manganite is introduced. This model is used for all investigations of dynamics after photoexcitations in this thesis. Therefore, we summarize the derivation of the model and the key aspects of this model in this section.



**Figure 3.1:** (Published in [KRS<sup>+</sup>18]) Degrees of freedom of the tight-binding model from which the one-dimensional model is derived. The first two octahedra show the orbital degrees of freedom of the  $e_g$  electrons, which are treated explicitly. The last octahedron shows the degrees of freedom of the  $t_{2g}$  electrons, which are treated classically.

<sup>4</sup>This constant only results in a constant energy shift and is therefore unimportant for almost all uses.

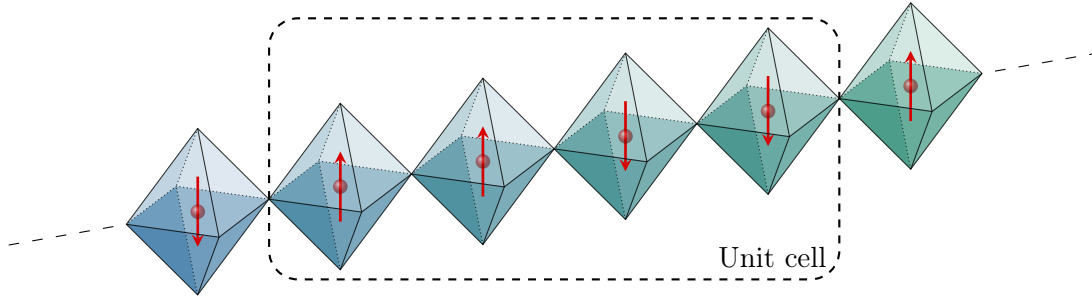
Starting from the model introduced in Ref. [SRB<sup>+</sup>17], we only focus on the electronic degrees of freedom in order to study the electronic relaxation after a photoexcitation. In the following, we therefore freeze the  $t_{2g}$  spin, which is described by a classical spin, and the lattice degrees of freedom in the ground state. Hence, the only remaining dynamical entities in this model are the  $e_g$  electrons. Furthermore, the Hilbert space for the  $e_g$  electrons has been limited to two local  $d_{3z^2-r^2}$  spin orbitals per Mn-site, which makes the model similar to a single-band Hubbard model with spatially varying magnetic fields. For a graphical representation of the mentioned entities, see Fig. 3.1.

We focus on the half-doped system, i.e.,  $x = 0.5$ , because it allows us to study the role of the magnetic microstructures formed by antiferromagnetically coupled Zener polarons on the relaxation dynamics of a photoexcitation. As depicted in Fig. 3.2, such a Zener polaron consists of two neighboring Mn-sites, which share a single  $e_g$  electron that is uniformly delocalized over both sites. The Mn-ions inside a Zener polaron are ferromagnetically aligned, and, without loss of generality, we choose the spins to point along the  $z$ -axis, that is  $S_x = S_y = 0$ .

This leads to the spin configuration on the four Mn-sites of the unit cell, as depicted in Fig. 3.2,

$$(S_{z,1}, S_{z,2}, S_{z,3}, S_{z,4}) = \frac{3\hbar}{2} (-1, -1, +1, +1). \quad (3.26)$$

The spin distribution is periodic, so that  $S_{z,j+4} = S_{z,j}$ . This means that the  $e_g$  electrons experience the  $t_{2g}$  spin and the lattice degrees of freedom as a staggered magnetic field.



**Figure 3.2:** (Combination of Fig. 4.6 in [Bie16] and Fig. 5 in [KRS<sup>+</sup>18]) Representation of the PCMO octahedra with their magnetic microstructure and the delocalized  $t_g$  electrons. Consider the four octahedra in the center as one unit cell.

As a result, the Hamiltonian can be expressed in the form of a one-band Hubbard model with a staggered magnetic field. Formulated in second quantization, we thus obtain the simplified many-electron Hamiltonian for a 1D model, which we connect to a (fictive) half-doped 1D manganite

$$\hat{H} = \sum_j \left\{ -t_{\text{hop}} \sum_{\sigma} \left( \hat{c}_{\sigma,j+1}^{\dagger} \hat{c}_{\sigma,j} + \hat{c}_{\sigma,j}^{\dagger} \hat{c}_{\sigma,j+1} \right) + U \hat{n}_{\uparrow,j} \hat{n}_{\downarrow,j} + \frac{2\Delta}{3\hbar} S_{z,j} \hat{S}_j^z \right\}. \quad (3.27)$$

Using the values of table 3.1, we obtain

$$U \approx 4.3t_{\text{hop}} \quad (3.28)$$

for the Hubbard interaction and

$$\Delta = 2J_H \approx 2.3t_{\text{hop}} \quad (3.29)$$

for the Hund's splitting.

In relation to PCMO, we study the photoexcitation for the set of parameters (3.28) – (3.29). However, the model Eq. (3.27) allows us to go beyond this particular realization and tune the values of  $U/t_{\text{hop}}$  and  $\Delta/t_{\text{hop}}$  independently from each other. In this sense, this model realizes a minimal model for a manganite system to study the effect of the Hund’s coupling in the magnetic microstructure realized by the Zener polarons, and the effect of the electron-electron interaction on the photoexcitation in such systems. In chapters 11 and 12, we hence study the time evolution after a photoexcitation for the parameter values Eqs. (3.28) and (3.29) using matrix-product state (MPS), and also the results when changing the values of  $\Delta/t_{\text{hop}}$  and  $U/t_{\text{hop}}$ .

### Parameters from first-principle calculation

In [SRB<sup>+</sup>17] a first-principle calculation on PCMO is performed and in [KRS<sup>+</sup>18] the 1D model Eq. (3.27) is derived. For completeness, all resulting parameters, as given in [KRS<sup>+</sup>18], are shown in table 3.1.

$J_{AF}$	32.6	meV	$g_{JT}$	2.113	eV/Å
$J_H$	0.653	eV	$k_{JT}$	5.173	eV/Å <sup>2</sup>
$U$	2.514	eV	$t_{\text{hop}}$	0.585	eV
$J_{xc}$	0.692	eV			

**Table 3.1:** (Published in [KRS<sup>+</sup>18]) Model parameters for the 1D model situation, based on the first-principle calculations on PCMO in [SRB<sup>+</sup>17].  $J_{AF}$  describes the antiferromagnetic coupling between the  $t_{2g}$  states of neighboring Mn-sites;  $J_H$  is the Hund’s coupling;  $U$  and  $J_{xc}$  are the Kanamori parameters for electron-electron interaction between  $e_g$  electrons;  $g_{JT}$  and  $k_{JT}$  parametrize the electron-phonon interaction;  $t_{\text{hop}}$  is the hopping amplitude of the  $e_g$  electrons.

### Band structure of non-interacting electrons

In order to understand the results in Sec. 11.1, let us explore the basic features of the model Eq. (3.27) without Coulomb interaction, i.e., the case  $U = 0$ .

The band structure of the non-interacting system elucidates the role of the Hund’s splitting  $\Delta$ , which acts as a staggered magnetic field on the electronic structure. We obtain

$$\epsilon_\nu(k) = \pm t_{\text{hop}} \sqrt{2 + \left(\frac{\Delta}{2t_{\text{hop}}}\right)^2 \pm 2\sqrt{\cos^2(2ka) + \left(\frac{\Delta}{2t_{\text{hop}}}\right)^2}}, \quad (3.30)$$

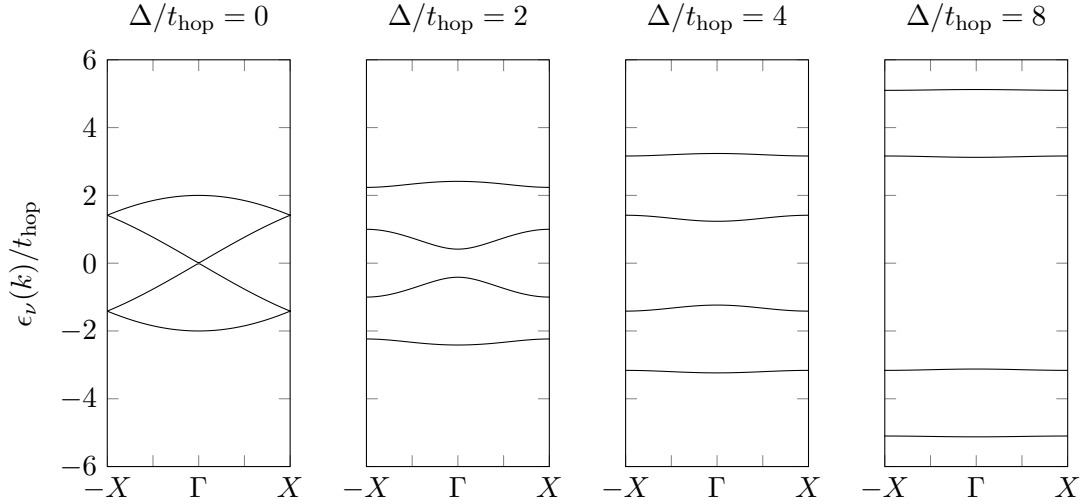
where  $k$  is the momentum in the reduced Brillouin zone,  $\nu$  labels the bands in this reduced Brillouin zone, and we introduced the spacing between the Mn-ions  $a$ . For the details of the derivation, see [KRS<sup>+</sup>18] and [Bie16].

In Fig. 3.3, this band structure is shown for different values of  $\Delta/t_{\text{hop}}$ . Without Hund’s splitting, the system is equivalent to a single-band Hubbard chain (see Sec. 3.1), for which we already obtained the dispersion relation Eq. (3.7).<sup>5</sup> In the setting of the four-site unit cell, this dispersion relation is folded back twice into the smaller reciprocal unit cell as shown in Fig. 3.3. In our case, at quarter filling, the lowest of the four bands is fully occupied in the ground state.

<sup>5</sup>Note that we replaced the electron transfer matrix  $t$  with the  $t_{\text{hop}}$  and included the spacing  $a$ .

In the limit of infinite Hund's splitting  $\Delta$ , the band structure develops into four nearly dispersionless bands.

As seen in Fig. 3.3, intermediate Hund's splitting leads to a coexistence of gaps, flat bands, and bands with large dispersion. Thus, the behavior of the dynamics will be non-trivial for intermediate Hund's splitting. The parameters in table 3.1 show that PCMO lies in this regime.



**Figure 3.3:** (Inspired by Fig. 6 in [KRS<sup>+</sup>18]) One-particle band structure of  $\text{Pr}_{1/2}\text{Ca}_{1/2}\text{MnO}_3$  for different values of the Hund's splitting  $\Delta$ , which is measured in units of  $t_{\text{hop}}$ .  $\Gamma$  denotes the origin of the  $k$ -points, and  $X = \pi/4a$  the zone boundary with the Mn-Mn spacing  $a$ . One can see that the distance between the mean of the upper two bands and the mean of the lower two bands is close to  $\Delta$  for large values of  $\Delta$ . Furthermore, in the same limit, the distance of the upper two bands (as well as the one of the lower two bands) is approximately  $2t_{\text{hop}}$ .

Next, we introduce modeled excitations, in particular various models for photoexcitations.

## Excitations | 3.4

In order to investigate model systems, it is usually of no interest to evolve an eigenstate, e.g., the ground state, of the Hamiltonian in time. Hence, we have, in general, two possible, and more interesting, scenarios: Starting with a somehow previously prepared state that is not an eigenstate of the Hamiltonian or perform the time evolution with a time-dependent Hamiltonian. Examples for both approaches are investigated in the following, focusing on excitations that model photoexcitations. In particular in the case of a sudden local excitation, the propagation of the excitation can be investigated with respect to the Lieb-Robinson bound [LR72, BDS14], which states an upper speed limit for the propagation of information.

### Photoexcitation | 3.4.1

The main idea of a photoexcitation in a strongly correlated system can be expressed by the interaction of photons with electrons. From semi-conductor physics we know that if the energy of a photon  $E_{\text{ph}} = \hbar\omega$  is larger than the gap  $E_g$  between the valence band and the conduction band an electron can be moved from the former into the latter. For photons with much larger



energy  $E_{\text{ph}} \gg E_g$ , hot electrons are created, which lose their energy by emitting phonons, i.e., heating up the system [AHRF<sup>+</sup>08], or via impact ionization [Man10]. Within the description of the Hubbard model, excitations close to the Fermi energy can be described best [AHRF<sup>+</sup>08]. Therefore, we do not describe the creation and the dissipation of the hot electrons but model the state that results afterwards.

In this section, we describe different possibilities, which are frequently considered in the literature, to take into account the light-electron interaction without the need of quantizing the light field. We start by introducing several ad-hoc approaches, which are either local in real or in momentum space. Afterwards, the Peierls substitution, in which the light field is described semi-classically, is presented.

### Operator-based photoexcitations

Conceptually, the simplest approach to modeling a photoexcitation is to apply local operators acting directly on the electron distribution. This has been done in previous works that also extend the Hubbard model, e.g., in [AHRDA13] the one-band Hubbard model at half filling with additional nearest-neighbor interactions is investigated after a whole Gaussian wave packet was applied. Another example is given in [DVB<sup>+</sup>15] in which the Holstein-Hubbard model, which includes phonon degrees of freedom, is treated, starting from an electron at momentum  $k = \pi$  and a phonon vacuum. Here, instead of considering further extensions to the Hubbard model, we quickly summarize how the photoexcitations can be treated “ad hoc” as found in the literature. The first approach we want to consider here is the one by [HP12] that uses a doublon-creation operator

$$\hat{d}_{j,\text{simple}}^\dagger = \hat{c}_{\uparrow,j}^\dagger \hat{c}_{\downarrow,j}^\dagger \quad (3.31)$$

on an empty system at site  $j$ . This allows them to study the decay of a doublon and its spreading with a “light cone effect”.

As we are interested in many-body effects, we can excite a system at its ground state with a particle-hole-like excitation. To achieve this, a more advanced doublon-creation operator

$$\hat{d}_j^\dagger = \frac{1}{\sqrt{2}} \sum_{\sigma} \hat{c}_{\sigma,j}^\dagger \hat{n}_{\bar{\sigma},j} \quad (3.32)$$

as well as a holon-creation operator

$$\hat{h}_j^\dagger = \frac{1}{\sqrt{2}} \sum_{\sigma} \hat{c}_{\sigma,j} (1 - \hat{n}_{\bar{\sigma},j}) \quad (3.33)$$

is defined, e.g., in [DdSAHF<sup>+</sup>10, LP13, LP14, LEP15],  $\bar{\sigma}$  denoting the opposite spin to  $\sigma$ . The photoexcitation is then modeled by applying both operators on the initial state  $|\psi_0\rangle$ , i.e.,  $\hat{d}_j^\dagger \hat{h}_{j'}^\dagger |\psi_0\rangle$  with  $j$  and  $j'$  often being nearest-neighbor or close-by sites. This approach might be a good representation of a photoexcitation at half-filling. However, doublons are very improbable in the quarter-filled case due to the site-dependent magnetic fields and the overall low amount of electrons.

Whereas the previous approaches create a localized excitation, [WBC<sup>+</sup>11] suggest to model the photoexcitation by creating a delocalized holon-doublon pair by applying

$$\hat{X}_L = \sum_{j=1}^{L-1} \sum_{\sigma} \left( \hat{c}_{\sigma,j}^\dagger \hat{c}_{\sigma,j+1} - \hat{c}_{\sigma,j+1}^\dagger \hat{c}_{\sigma,j} \right) \quad (3.34)$$

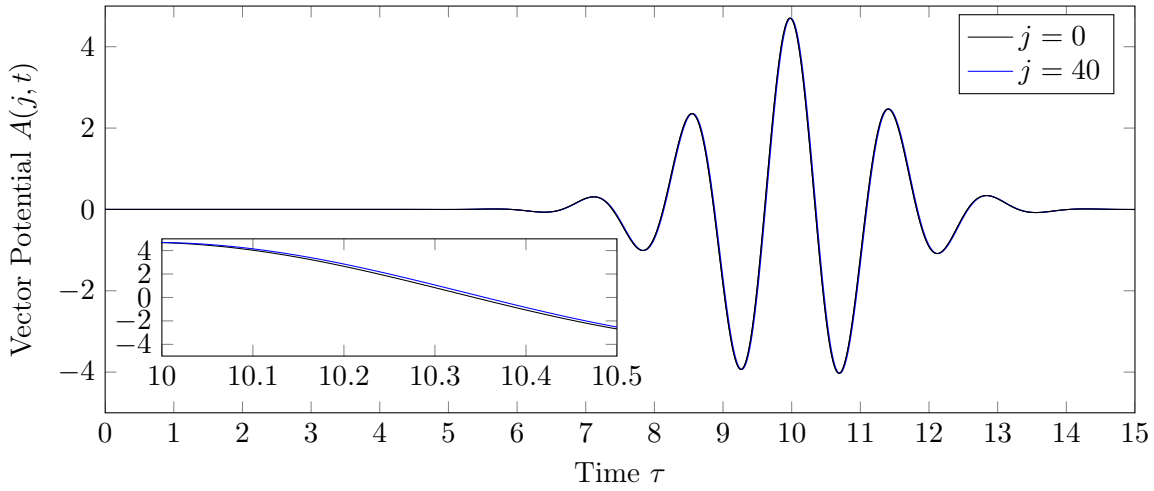
over the complete lattice.

As we want to study lifetimes and the propagation of a photoexcitation, we apply the conceptually simplest excitation and consider the effect of an incoming photon as only moving particles to the neighboring site. The conceptually simplest operator then is

$$\hat{Y}_j = \sum_{\sigma} \hat{c}_{\sigma,j+1}^{\dagger} \hat{c}_{\sigma,j} , \quad (3.35)$$

with  $j$  and  $j + 1$  being lattice sites both located on the same dimer. This operator has two advantages: It can easily be understood as moving particles from site  $j + 1$  to site  $j$ , and it conserves the particle number and the spin.

### Peierls substitution



**Figure 3.4:** (Rescaled version of Fig. 9 in [KPM18]) Vector potential  $A(j, \tau)$  Eq. (3.38) at site  $j = 0$  (black) and  $j = 40$  (blue). The following parameters are used here:  $t_0 = 10$ ,  $E_0 = 20$ ,  $k_0 = 0.00126$ , and  $s = 6000$ . These values correspond to a wavelength  $\lambda \approx 500$  nm.

The vector potential has nearly the same value at all sites  $j$  because the wavelength is much larger than the spatial extent of the chain. In the inset, the small – but finite – difference between the vector potential on the two ends of the system is visible. This small difference is due to fact that the wave packet traverses through the chain only with the speed of light  $c$ .

Another way to model a photoexcitation is via the Peierls-substitution ansatz [Pei33, MBE15, EW13, Sch16]. In this approach, the incident light is considered as a classical field and included in the Hamiltonian via minimal coupling [Gre00]. In the usual modeling, this leads to a position- and time-dependent complex phase in the hopping amplitudes, and to additional local magnetic fields, which are often neglected. In the following, we consider a generalization of the usual ansatz and assume that the effect of the light field can depend on the spin direction of the electrons. This is motivated by the realization of spin-selective photoexcitations, e.g., through the tunability of parameters in experiments on optical lattices [BDZ08].

Our model Hamiltonian Eq. (3.27) is thus modified to

$$\hat{H} = - \sum_{j,\sigma} \left( t_{\sigma,j}^{\text{hop}}(\tau) \hat{c}_{\sigma,j}^{\dagger} \hat{c}_{\sigma,j+1} + \text{h.c.} \right) + \sum_j \left[ U_j \hat{n}_{\uparrow,j} \hat{n}_{\downarrow,j} + \left( \Delta_j + B(j, \tau) \right) \hat{S}_j^z \right] . \quad (3.36)$$

The Peierls substitution ansatz for the photoexcitation leads to a tunneling amplitude

$$t_{\sigma,j}^{\text{hop}}(\tau) = e^{-i\alpha_{\sigma} \frac{e\ell a}{2\hbar} (A(j,\tau) + A(j+1,\tau))} t_{\text{hop}} , \quad (3.37)$$

where  $t_{\text{hop}}$  is the hopping amplitude in equilibrium. The coefficient  $\alpha_\sigma$  takes either the value one, if the light field couples with the electrons of spin direction  $\sigma$ , or zero, if the coupling is suppressed. The functions  $A(j, \tau)$  and  $B(j, \tau)$  describe the time-dependent modulation of the vector potential and the magnetic field by an incident light-pulse in the form of a Gaussian wave packet of width  $s$  and amplitude  $E_0$  at position  $j$ , and are specified by

$$A(j, \tau) = \frac{E_0 \lambda}{2\pi c} e^{-\frac{[aj - c(\tau - \tau_0)]^2}{s^2}} \sin \left[ \frac{2\pi}{\lambda} (aj - c\tau) \right], \quad (3.38)$$

$$B(j, \tau) = \frac{E_0 g_s \mu_B}{c} e^{-\frac{[aj - c(\tau - \tau_0)]^2}{s^2}} \cos \left[ \frac{2\pi}{\lambda} (aj - c\tau) \right]. \quad (3.39)$$

Symbol	Value	Unit	Description
$\hbar$	1.0	$\hbar$	Planck constant
$e_{\text{el}}$	1.0	$e_{\text{el}}$	electron charge
$a$	1.0	$10^{-10}$ m	lattice spacing
$c$	3374.85	$\frac{at_{\text{hop}}}{\hbar}$	speed of light
$g_s \mu_B$	13.04	$\frac{e_{\text{el}} t_{\text{hop}} a^2}{\hbar}$	g-factor times Bohr magneton

**Table 3.2:** (Published in [KPM18]) Constants used in this thesis.

Symbol	Value	Unit	Description
$\lambda$	498.7	nm	wavelength of the light field
$E_0$	20.0	V/m	amplitude
$s$	6000.0	$a$	width
$\tau_0$	10	$\hbar/t_{\text{hop}}$	time of the maximum

**Table 3.3:** (Published in [KPM18]) Parameters of the wave packet used in this thesis.

If not stated otherwise, we use the parameters presented in table 3.3. With these parameters, the magnetic field  $B(j, \tau) \ll \Delta$  so that it will be disregarded in the following. The wavelength  $\lambda$  is chosen to correspond with the wavelength of the highest intensity in the solar spectrum and also lies within the range of lasers that are commonly used in pump-probe experiments [Sch16]. The vector potential  $A(j, \tau)$  with those parameters is shown in Fig. 3.4 for both ends of a system with 40 sites and with the corresponding parameters also within the figures in chapter 12. In Fig. 3.4, the very small difference between the two ends of the system due to the small spatial extent of the chain and the large wavelength can be seen. It is therefore not surprising that similar results can be obtained without the spatial extent.

In the next part, we introduce the numerical methods used to perform calculations on the presented models.



# Numerical Methods





# Motivation, Notation, and Frequently Used Routines

# 4

In chapter 3, we introduced several models and excitations. Their exact description is based on quantum states, which are elements of an exponentially large Hilbert space  $\mathcal{H}$ . Hence, an exact representation is numerically only possible for rather small systems.

To overcome this problem, a truncated quantum state, which is still a good approximation, is desirable.

The density-matrix renormalization group (DMRG) [Whi92, Whi93], which was derived from the numerical renormalization group (NRG)[Wil75], was developed as a ground-state search method. The DMRG quickly became one of the most powerful, and widely used, tools to treat one-dimensional (1D) strongly correlated quantum systems [Sch05]. Around 2005, boosted by quantum-information theory, the matrix-product state (MPS) formalism was explored. The subsequent reformulation of the complete algorithms in terms of tensor-network state (TNS) enhanced the understanding and the development of further methods (see Ref. [Sch11] and references within). It furthermore showed that ground states, which were obtained via DMRG, belong exactly to the class of quantum states for which a good approximation can be found [RO97].

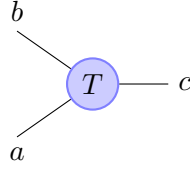
In this chapter, we describe the basic ideas behind MPS and TNS. Furthermore, we introduce all concepts and methods within this framework needed in this thesis. A substantial part of chapters 5 and 6 is used in Ref. [PKS<sup>+</sup>19]. Unless otherwise stated, these chapters are based on the thorough review Ref. [Sch11].

We start this chapter by introducing a compact, but complete, tensor notation. Afterwards, we present some routines, which are extensively used in tensor-network approaches.

In the next two chapters, we give a detailed introduction to MPSs and matrix-product operators (MPOs). Afterwards, we present a general construction scheme for MPOs based on finite-state machines (FSMs) and describe additions and multiplications directly in the realm of those FSMs. Then, two applications of the introduced methods are presented, namely the ground-state search and two different time-evolution algorithms. In the last chapter of this part, the universal quantum-computer simulator is introduced.

## Tensor Notation

A tensor is a multi-dimensional collection of numbers; intuitively accessible examples are vectors (one dimension, rank 1) and matrices (two dimensions, rank 2). The rank describes the number of dimensions and every dimension is labeled by an index. Graphically, tensors are represented as blobs (circles, triangles, squares) with one leg per tensor index (see Fig. 4.1). Note that the type of the blob and the coloring are chosen to describe certain properties, which are defined when needed. Consider a rank 3 tensor  $T$  with associated vector spaces  $A$ ,  $B$ , and  $C$ . It has three indices  $a = [1, \dots, \dim(A)]$ ,  $b = [1, \dots, \dim(B)]$ , and  $c = [1, \dots, \dim(C)]$ , hence its scalar coefficients are given by  $T_{a,b,c}$ . We explicitly do not consider quantum-number conservation in the method description and hence do not have to distinguish between incoming/outgoing tensor



**Figure 4.1:** The graphical representation of the example tensor  $T_{a,b,c}$  with three indices (legs)  $a$ ,  $b$  and  $c$ .

legs (or, conversely, (dual) vector spaces).<sup>1</sup> Thus, the upstairs/downstairs location of tensor indices is meaningless and  $T_{a,b,c} = T_{b,c}^a = T^{a,b,c}$ . Nevertheless, it is important to note that indices are always separated by a comma and unseparated indices are consequently combined (merged) indices. Through complex conjugation of every element of  $T$ , we obtain a new tensor  $\bar{T}$  with elements  $\bar{T}_{\bar{a},\bar{b},\bar{c}} = T_{a,b,c}^*$ . We use  $\bar{a}$  to denote indices of conjugated tensors. Contracting  $\bar{T}$  and  $T$  over  $a$  and  $b$ , we write:

$$\sum_{a,b} \bar{T}_{a,b,\bar{c}} T_{a,b,c} = X_{\bar{c},c}, \quad (4.1)$$

which is equivalent to

$$\sum_{\bar{a},a,\bar{b},b} \delta_{\bar{a},a} \delta_{\bar{b},b} \bar{T}_{\bar{a},\bar{b},\bar{c}} T_{a,b,c}. \quad (4.2)$$

Given two tensors  $A_{a,b,c}$  and  $B_{b,d,c}$ , the shorthand  $A \cdot B$  denotes the contraction over all shared indices:

$$A \cdot B = \sum_{b,c} A_{a,b,c} B_{b,d,c} = Y_{a,d}. \quad (4.3)$$

Tensor contractions can also be represented graphically by drawing tensors with connected legs.

## Matrix Decomposition | 4.1

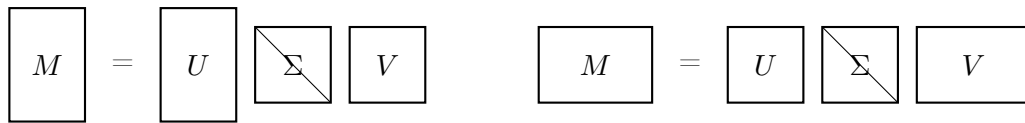
One class of time-consuming operations in the framework of tensor networks are matrix decompositions, in particular, the singular-value decomposition (SVD). The SVD decomposes a matrix  $M \in \mathbb{C}^{m \times n}$  into

$$M_{\alpha,\beta} = \sum_{s,s'} U_{\alpha,s} \Sigma_{s,s'} V_{s',\beta} \quad (4.4)$$

with  $U \in \mathbb{C}^{m \times d}$  consisting of orthonormal columns, i.e.,  $\sum_{\alpha} \bar{U}_{\bar{s},\alpha} U_{\alpha,s} = \mathbb{1}_{\bar{s},s}$ , and  $V \in \mathbb{C}^{d \times n}$  consisting of orthonormal rows, i.e.,  $\sum_{\beta} V_{s',\beta} \bar{V}_{\beta,\bar{s}'} = \mathbb{1}_{s',\bar{s}'}$ . Furthermore,  $\Sigma \in \mathbb{R}^{d \times d}$  is a diagonal matrix that contains the  $r = \min(m, n)$  singular values  $\Sigma_s = \Sigma_{s,s}$ . The dimensions described above are depicted in Fig. 4.2 and the graphical representation is given in Fig. 4.3. Note that the matrix rank  $r$  of the matrix  $M$ , which should not be confused with the rank of a tensor, is given by the number of non-zero singular values  $\Sigma_s$ . In fact, determining the higher dimensional analog of the matrix rank for tensors is a NP-hard problem [Hå90].

<sup>1</sup>For a treatment with symmetries, see, e.g., [PKM17, Hub17].





**Figure 4.2:** Pictorial representation of the singular value decomposition of a matrix  $M$  into  $U \cdot \Sigma \cdot V$ . The case  $m > n$  is depicted left and the case  $m < n$  is depicted right.

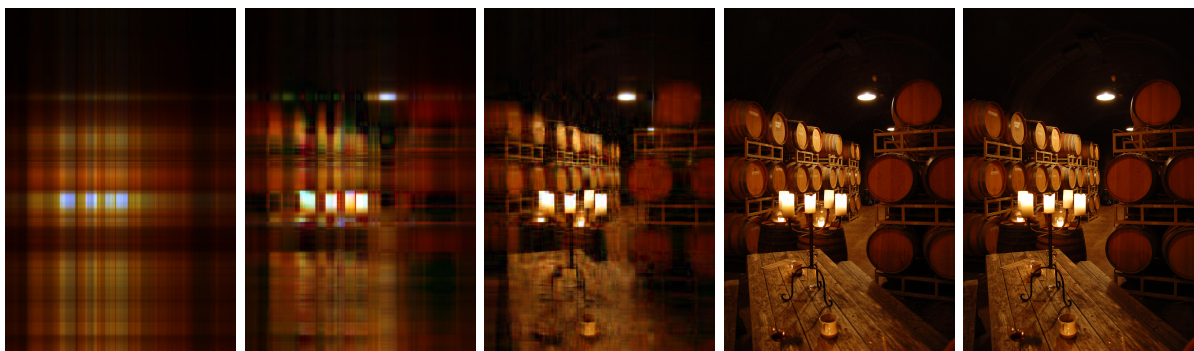


**Figure 4.3:** Pictorial representation of the singular value decomposition of a matrix  $M \in \mathbb{C}^{m \times n}$  into  $U \cdot \Sigma \cdot V$ . The result consists of a right orthonormal tensor (red triangle), a diagonal matrix (grey diamond), and a left orthonormal tensor (green triangle).

Although it is not strictly defined, the singular values are usually sorted to be in descending order. This is important, as the SVD can be used to optimally approximate a matrix  $M$  with rank  $r$  by a matrix  $M'$  with rank  $r'$  by considering the largest singular values. Optimal is here regarding to the Frobenius norm  $\|M\|_F^2 = \sum_{j,j'} |M_{j,j'}|^2$ . The approximation is performed by setting all but the first  $r'$  singular values to zero ( $\Sigma_s = 0 \forall s > r'$ ). This becomes important for the truncation of MPS in Sec. 5.3.2. Note that the complexity of the SVD is  $\mathcal{O}(mnr)$ . Hence, it is one of the operations with leading order for most tensor-network algorithms.

The compression properties of the SVD can also be exemplified by applying it to a photograph. For this, we decompose a photograph into its color channels and interpret those pixel maps as matrices in order to apply the SVD on them. Afterwards, we reduce the number of singular values, perform the resulting matrix product, and recombine the color channels back into a photograph.

In Fig. 4.4, we show the same picture but with a different number of singular values, i.e., different ranks of the matrices that represent the picture. A single singular value is obviously not sufficient to restore the information of the original photo. While four singular values still lead to a rather fuzzy picture, 16 singular values are sufficient in order for a human to understand most of the content of the picture. In order to make the wine glass in the lower left corner and the other details visible, a larger number is necessary. This procedure is arguably not the best compression scheme for photographs as the number of singular values in this case needs to be less than 1201 to have any compression and  $\sim 46$  to achieve a similar file size as the original JPEG.



**Figure 4.4:** Application of an SVD compression to a photograph. From left to right,  $r' = 1, 4, 16, 128$  and finally the original photo with dimension  $2000 \times 3008$  (file size: 685 KiB [IEC08]).

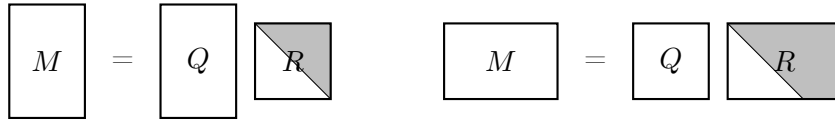
Another widely used decomposition is the QR decomposition. It decomposes a matrix  $M \in \mathbb{C}^{m \times n}$  into

$$M_{\alpha,\beta} = \sum_{\gamma} Q_{\alpha,\gamma} R_{\gamma,\beta}, \quad (4.5)$$

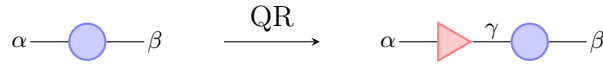
where  $Q \in \mathbb{C}^{m \times d}$  has orthonormal columns, i.e.,  $\sum_{\alpha} \bar{Q}_{\bar{\gamma},\alpha} Q_{\alpha,\gamma} = \mathbb{1}_{\bar{\gamma},\gamma}$ ,  $R \in \mathbb{C}^{d \times n}$  is an upper triangular matrix (see Figs. 4.5 and 4.6).

Note that the QR decomposition can also be used to decompose a matrix into a right part with orthonormal rows by first forming the adjoint of  $M$ , then performing the decomposition, and finally taking the adjoint of the result,

$$M = \overline{(\bar{M})} \stackrel{QR(\bar{M})}{=} \overline{(QR)} = \bar{R}\bar{Q}. \quad (4.6)$$



**Figure 4.5:** Pictorial representation of the QR decomposition of a matrix  $M \in \mathbb{C}^{m \times n}$  into  $QR$ . The case  $m < n$  is depicted left and the case  $m > n$  is depicted right.



**Figure 4.6:** Pictorial representation of the QR decomposition of a matrix  $M \in \mathbb{C}^{m \times n}$  into  $QR$ . The result consists of a right orthonormal tensor (red triangle) and general tensor (blue circle).

The QR decomposition is mainly used whenever a re-orthonormalization is needed but the singular values are not relevant. It is favorable to use the QR decomposition although it also scales with  $\mathcal{O}(mnr)$ . However, it has a smaller prefactor ( $\approx \frac{2}{3}$  instead of  $\approx 2$  for the SVD).

Another alternative for the SVD is the adaptive cross-approximation (ACA) [Beb00, BR03], which iteratively approximates a matrix  $M$  with another matrix  $M'$  with lower rank [Thy17].

## Lanczos Algorithm | 4.2

Because the Lanczos algorithm is used within all ground-state calculations in this thesis, we summarize it in the following.

The Lanczos algorithm [Lan50, BDD<sup>+</sup>00, CW02] is an iterative method to find the smallest eigenvalues and the corresponding eigenvectors of a matrix  $M$ . The main idea is a projection of the matrix into a Krylov subspace that is defined as

$$\mathcal{K} \equiv \text{span} \{ \vec{v}_0, M\vec{v}_0, M^2\vec{v}_0, M^3\vec{v}_0, \dots \} \quad (4.7)$$

with a non-trivial initial vector  $\vec{v}_0$ . In order to ensure the orthogonality of the basis vectors, the following recursion is used to create the so-called Lanczos vectors:

$$\vec{v}_{n+1} = M\vec{v}_n - \alpha_n \vec{v}_n - \beta_n^2 \vec{v}_{n-1} \quad (4.8)$$

with

$$\alpha_n = \frac{\vec{v}_n^t M \vec{v}_n}{\vec{v}_n^t \vec{v}_n} \quad , \quad \beta_n^2 = \frac{\vec{v}_n^t \vec{v}_n}{\vec{v}_{n-1}^t \vec{v}_{n-1}} \quad , \quad \beta_0 = 0 . \quad (4.9)$$

After  $n$  iterations, the matrix  $M$  is represented in the orthogonal Krylov space by the tridiagonal matrix

$$M_{\text{eff}} = \begin{pmatrix} \alpha_0 & \beta_1 & 0 & \cdots & 0 \\ \beta_1 & \alpha_1 & \beta_2 & \ddots & \vdots \\ 0 & \beta_2 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \beta_{n-1} \\ 0 & \cdots & 0 & \beta_{n-1} & \alpha_{n-1} \end{pmatrix} . \quad (4.10)$$

The matrix  $M_{\text{eff}}$  can be easily diagonalized due to its tridiagonal nature, and thus reveals the eigenvalues [PTVF07]. In order to obtain the eigenvectors of  $M$ , all Lanczos vectors need to be stored because the eigenvectors of  $M_{\text{eff}}$  are represented in the Lanczos basis.



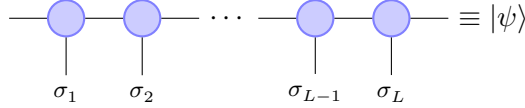
Matrix-product states are efficient representations for one-dimensional weakly entangled [Has07] quantum states  $|\psi\rangle \in \mathcal{H} = \mathcal{H}_d^L$ . The concept is closely related to tensor trains [Ose09] that are used in mathematics and computer science [Gel17]. The main idea is to represent the coefficients  $c_{\sigma_1 \dots \sigma_L}$  of a general quantum state

$$|\psi\rangle = \sum_{\sigma_1, \dots, \sigma_L} c_{\sigma_1 \dots \sigma_L} |\sigma_1 \dots \sigma_L\rangle, \quad (5.1)$$

by a product of matrices  $M_j^{\sigma_j}$ ,

$$|\psi\rangle = \sum_{\substack{\sigma_1, \dots, \sigma_L, \\ m_0, \dots, m_L}} M_{1; m_0, m_1}^{\sigma_1} \cdots M_{L; m_{L-1}, m_L}^{\sigma_L} |\sigma_1 \dots \sigma_L\rangle, \quad (5.2)$$

which can also be interpreted as a product of rank-3 tensors, as depicted in Fig. 5.1.



**Figure 5.1:** Schematic of the tensor network of a matrix-product state. Horizontal lines denote the internal indices with bond dimension  $m$ , whereas the vertical lines denote physical indices with dimension  $d$ .

The matrix product Eq. (5.2) consists of  $d \cdot L$  matrices, where  $d$  denotes the local (physical) dimension and  $L$  the number of physical sites, i.e., the size of the system. The matrix dimensions at site  $j$ , which are – at the very most – necessary to represent every state exactly, are given by

$$M_j \in \mathbb{C}^{d \times d^l \times d^{l'}} \quad \text{with} \quad \begin{cases} l = j - 1 & , \text{ if } j \leq \lceil L/2 \rceil \\ l = L - (j - 1) & , \text{ if } j > \lceil L/2 \rceil \end{cases} \quad \text{and} \quad \begin{cases} l' = j & , \text{ if } j < \lceil L/2 \rceil \\ l' = L - j & , \text{ if } j \geq \lceil L/2 \rceil \end{cases}. \quad (5.3)$$

At first glance, the amount of values in this representation Eq. (5.2) is larger than the amount of values in Eq. (5.1) by a factor of  $d$ . But one of the most important benefits of matrix-product states (MPSs) is that they can be truncated, i.e., that the matrix dimension, also called bond dimension,  $m$  of the matrices  $M_j^{\sigma_j}$  is reduced dramatically while still representing the coefficients very accurately. Therefore, the maximal bond dimension  $m_{\max}$ , which is an upper bound for the bond dimension  $m$ , is an important parameter of the MPS. Note that we add another position index (mostly  $j$ ) to every tensor separated by a semicolon in order to denote its corresponding site. This additional information allows us to use the abbreviation  $M_j \equiv M_{j; m_{j-1}, m_j}^{\sigma_j}$  whenever the indices and the contractions are given by the context or are not relevant. If a tensor is located between two sites, e.g., site  $j$  and  $j + 1$ , the position index is underlined, i.e.,  $\underline{j}$ .

This section is structured as follows. First, we introduce left-/right-canonical MPSs, which is a very convenient representation as we see further below. Afterward, we give a short overview of ways to create an MPS either by constructing the MPS from an arbitrary state or by writing down the corresponding matrices. Next, we explain how to perform operations on MPSs, including different truncation schemata and the optimal evaluation of scalar products. This chapter is closed with the introduction of the representation of finite-temperature states within the MPS framework.

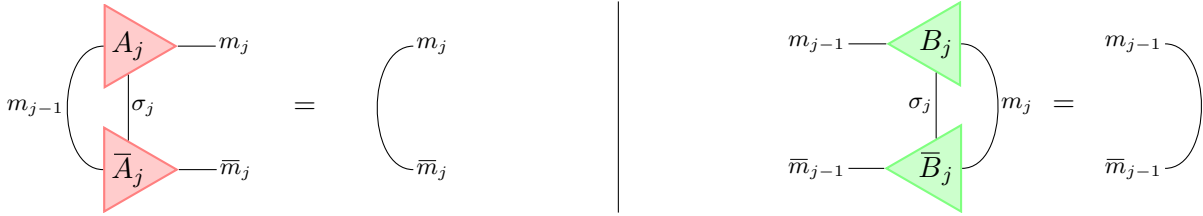
## Canonical Form | 5.1

Now we explicitly employ a gauge invariance to introduce one of the most important representations of MPSs: the left-/right canonical form. Afterwards, we explain in detail how to obtain such MPS representations.

At first, we define left-normalized MPS site tensors  $A_j$  and right-normalized MPS site tensors  $B_j$  via

$$\sum_{\sigma_j, m_{j-1}} \bar{A}_{j; \bar{m}_j, m_{j-1}}^{\sigma_j} A_{j; m_{j-1}, m_j}^{\sigma_j} = \mathbb{1}_{\bar{m}_j, m_j} \quad , \quad \sum_{\sigma_j, m_j} B_{j; m_{j-1}, m_j}^{\sigma_j} \bar{B}_{j; m_j, \bar{m}_{j-1}}^{\sigma_j} = \mathbb{1}_{m_{j-1}, \bar{m}_{j-1}} \quad . \quad (5.4)$$

Graphically, they are represented by red (left-normalized) or green (right-normalized) triangles (see Fig. 5.2), where the orientation of the triangles also indicates the representation.<sup>1</sup>



**Figure 5.2:** (Left) left-normalized tensor  $A_j$  (red) contracted with its adjoint resulting in an identity. (Right) analogous, right-normalized tensor  $B_j$  (green) contracted with its adjoint resulting in an identity. See Eq. (5.4) and note that the order of the tensors only defines the non-physical index  $m_{j-1}$  ( $m_j$ ) over which the contraction takes place.

Left-/right-canonical MPSs are now specified to consist of left-/right-normalized tensors only. Furthermore, a mixed canonical MPS is defined by fixing a non-normalized site tensor at site  $j$  and demanding all site tensors to the left/right to be left-/right-normalized tensors only (see Fig. 5.5). The non-normalized site is often called active site or orthogonality center. This definition can be extended to  $k$ -sites mixed canonical MPSs in the sense that a connected subset  $\{j_1 \cdots j_k\}$  of sites is permitted to be non-normalized, whereas the remaining sites to the left/right are left-/right-normalized. In Sec. 5.3.1, the procedure for obtaining a canonical MPS from an arbitrary MPS is explained.

<sup>1</sup>The color choice is motivated by the coloring of the navigation lights of maritime vessels. Port is left-hand side and red; starboard is right-hand side and green.

## Construction of an Initial MPS | 5.2

In order to work with MPSs, one always needs an initial state. It is possible to convert an arbitrary quantum state into an MPS as long as the source state can be stored exactly and a singular-value decomposition (SVD) can be performed on the data representing the source state. The size of systems in which this procedure can be performed is comparably small. Furthermore, those systems can also be treated with exact diagonalization techniques.

To overcome this size constraint, it is possible to write down specific states directly in terms of MPSs. In the following, we first present the conversion of an arbitrary state and then we write down the specific states that are used in this thesis. Furthermore, more complex states can be constructed from those initial states, see Sec. 5.4 for an infinite temperature state and chapter 8 for the ground state.

### Construction of an MPS from an Arbitrary State | 5.2.1

The main idea behind constructing an MPS from an arbitrary state Eq. (5.1) is to factorize the global coefficients into local tensors. To perform this factorization, the first step is to store the coefficients in a matrix  $C \in \mathbb{C}^{d \times d^{L-1}}$ ,

$$C_{1;m_0\sigma_1,\sigma_2\dots\sigma_L m_L} = c_{\sigma_1\dots\sigma_L}, \quad (5.5)$$

in which  $m_0 = 1$  and  $m_L = 1$  are dummy indices added for convenience. Next, a QR decomposition (see Sec. 4.1) is performed on  $C_1$ ,

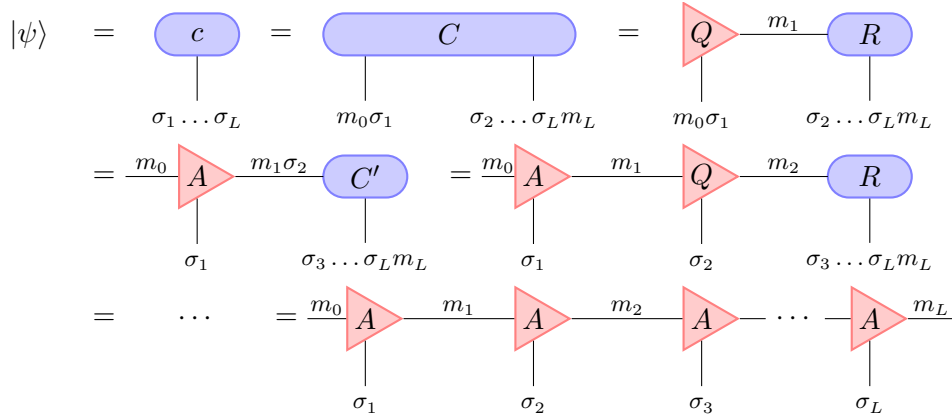
$$C_{1;m_0\sigma_1,\sigma_2\dots\sigma_L m_L} = \sum_{m_0,m_1} Q_{1;m_0\sigma_1,m_1} R_{1;m_1,\sigma_2\dots\sigma_L m_L}, \quad (5.6)$$

resulting in the first left-normalized MPS tensor  $A_{1;m_0,d_1}^{\sigma_1} = Q_{1;\sigma_1 m_0,d_1}$  of the MPS and a matrix  $C'_{2;\sigma_2 m_1,\sigma_3\dots\sigma_L} = R_{1;m_1,\sigma_2\dots\sigma_L}$ . With the new matrix  $C'_2$ , which contains the remaining degrees of freedom, the same procedure needs to be applied until the full left-canonical MPS is obtained. This procedure is also depicted in Fig. 5.3.

Note that within this process a truncation is not possible because the right side is not right-normalized, hence arbitrary parts of the states would be truncated. Therefore, it is not possible to convert a state with an arbitrary number of sites into an MPS in this way. An alternative to the method above is given in Ref. [FW15] in which an algorithm is presented that efficiently produces MPS representations from correlation matrices.

### Directly Defined MPS | 5.2.2

Another way to obtain an initial state is to write down the corresponding tensors directly. Here we introduce all initial states that are used in this thesis. Two different product states are used as initial states for the quantum-computer simulator (QCS) (chapter 10). The random state is necessary as initial state for the ground-state search (chapter 8), whereas the canonical infinite-temperature state is used for finite-temperature calculations.



**Figure 5.3:** Schematic of the construction of an MPS from an arbitrary quantum state. Starting from the coefficients  $c_{\sigma_1 \dots \sigma_L}$ , a matrix  $C$  is created. Then, a QR decomposition is performed to gain the tensor  $A_1$  at site 1 and the new matrix  $C'_2$ .

### Product States

Product states are particularly useful as initial states, because they have zero entanglement and can therefore be represented by  $1 \times 1$  matrices. On the other hand, starting from a product state necessitates the use of algorithms that are capable of increasing the bond dimension adaptively. Otherwise, the eventually increasing entanglement cannot be represented correctly.

**Néel State** The Néel state in one dimension is defined by

$$|\psi\rangle_{\text{Néel}} = |\uparrow\downarrow \cdots \uparrow\downarrow\rangle \quad (5.7)$$

and is used to represent the initial spin-1/2 state  $|0_p 1_q \cdots 0_p 1_q\rangle$  for universal quantum-gate simulations. Here,  $p$  denotes physical sites that are used for the actual computation and  $q$  denotes auxiliary sites, which act as a bath to fulfill quantum-number conservation, see chapter 10. This state is constructed by the following  $1 \times 1$  matrices:

$$A_{p(j)}^{\uparrow} = \begin{pmatrix} 1 \end{pmatrix}, \quad A_{p(j)}^{\downarrow} = \begin{pmatrix} 0 \end{pmatrix}, \quad A_{q(j)}^{\uparrow} = \begin{pmatrix} 0 \end{pmatrix}, \quad A_{q(j)}^{\downarrow} = \begin{pmatrix} 1 \end{pmatrix}, \quad (5.8)$$

where  $p(j)$  again denotes physical sites and  $q(j)$  auxiliary sites.

**Vacuum State** The vacuum state, i.e., a state in which the particle density is zero, is defined by

$$|\psi\rangle_{\text{vacuum}} = |0 \cdots 0\rangle. \quad (5.9)$$

This state is mainly used in order to create initial states for the application of creation operators in a controlled way (see Sec. 5.4.2). The state is given by the following  $1 \times 1$  matrices on all sites,

$$A_j^0 = \begin{pmatrix} 1 \end{pmatrix}, \quad A_j^{\downarrow} = \begin{pmatrix} 0 \end{pmatrix}, \quad A_j^{\uparrow} = \begin{pmatrix} 0 \end{pmatrix}, \quad A_j^{\uparrow\downarrow} = \begin{pmatrix} 0 \end{pmatrix}. \quad (5.10)$$

### Random State

In order to obtain a ground state, an initial state is variationally optimized (see chapter 8). The simplest initial state is a random state created by filling the matrices of the MPS with random numbers. Note that it is beneficial to scale the random numbers appropriately to the system size and the bond dimension in order to avoid numerical problems in the subsequent normalization.

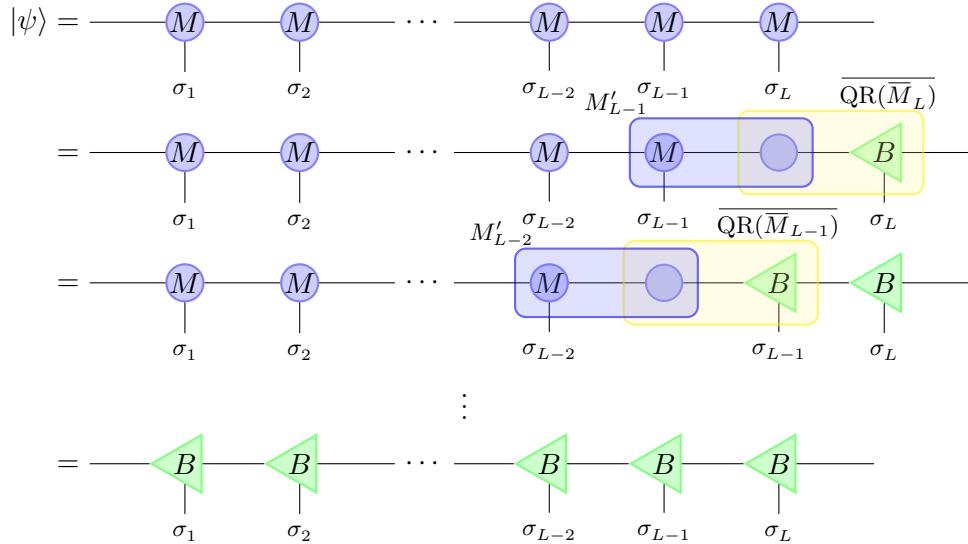


## Operations on MPSs | 5.3

In this section, we present the most commonly used operations performed on MPSs. This includes the more technical truncation of an MPS, which is nevertheless essential to the use of MPSs, and also arithmetic operations such as the scalar product.

### Normalization of MPSs | 5.3.1

Here we explain how to obtain a right-canonical MPS. There are infinite ways to transform MPSs by inserting arbitrary unitary transformations  $\sum_{m'_j} U_{j;m_j,m'_j} \bar{U}_{j;m'_j,\bar{m}_j} = \mathbb{1}_{j;m_j,\bar{m}_j}$  between the site tensors. In particular, we can choose them in a way that the new site tensors fulfill normalization constraints.



**Figure 5.4:** Schematic of the transformation of a non-canonical MPS into a right-canonical MPS.

Starting from an MPS with non-normalized tensors Eq. (5.2), we reshape the rightmost tensor,

$$M'_{L;m_{L-1},(\sigma_L \bar{m}_L)} = M_{L;m_{L-1},\bar{m}_L}^{\sigma_L}. \quad (5.11)$$

As before,  $m_0 = \bar{m}_0 = m_L = \bar{m}_L = 1$  are dummy indices. Then,  $\bar{M}'_L$  is decomposed via the QR decomposition,

$$M'_{L;m_{L-1},(\sigma_L \bar{m}_L)} = \sum_{\bar{m}'_{L-1}} \bar{R}_{L-1;m_{L-1},\bar{m}'_{L-1}} \bar{Q}_{L;\bar{m}'_{L-1},(\sigma_L \bar{m}_L)}, \quad (5.12)$$

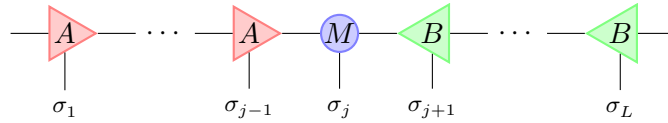
in which  $B_{L;\bar{m}_{L-1},\bar{m}_L}^{\sigma_L} = \bar{Q}_{L;\bar{m}_{L-1},(\sigma_L \bar{m}_L)}$  fulfills the condition Eq. (5.4) for right-normalized tensors due to the orthonormal rows of  $\bar{Q}_L$ . In the next step, the residual part of the decomposition  $\bar{R}_L$  is incorporated into the site tensor  $M'_{L-1}$  on the left,

$$M'_{L-1;m_{L-2},\sigma_{L-1} \bar{m}_{L-1}} = \sum_{m_{L-1}} M_{L-1;m_{L-2},m_{L-1}}^{\sigma_{L-1}} \bar{R}_{m_{L-1},\bar{m}_{L-1}}. \quad (5.13)$$

This is repeated with the new site tensor  $M'_{L-1}$  until the first site is reached. The complete procedure is illustrated in Fig. 5.4. Once the system is normalized, i.e., the first site is reached, the residual of the decomposition of the first site will be a scalar number. The square of the absolute value of this number is the norm of the state and can be discarded if a state with norm  $\langle\psi|\psi\rangle = 1$  is desired.

The procedure to obtain a left-canonical MPS is analogous and also similar to the construction of an MPS from an arbitrary state (see Sec. 5.2.1).

In order to obtain a mixed-canonical state (see Sec. 5.1 and Fig. 5.5) with active site  $j$  it is necessary to normalize the state from the left and from the right up to site  $j$ . The residuals  $R_{j-1}$  and  $\bar{R}_j$  are both incorporated into the site tensor  $M_j$ .



**Figure 5.5:** MPS with active site  $j$  and consequently left (right) normalized site tensor left (right) of site  $j$  as defined in Eq. (5.4).

## Truncation | 5.3.2

One of the main benefits of MPSs is the possibility to restrict the bond dimension  $m$ , but to still obtain a very good approximation for certain classes of many-body quantum states. This is particularly important as  $m$  grows with, e.g., the application of a matrix-product operator (MPO) (see Sec. 6.1). The problem we want to tackle is therefore to approximate  $|\psi\rangle$  with bond dimension  $m$  by  $|\tilde{\psi}\rangle$  with bond dimension  $\tilde{m} < m$  so that

$$\| |\psi\rangle - |\tilde{\psi}\rangle \|_2^2 \quad (5.14)$$

is minimized. Note that if Eq. (5.14) was zero, it would be equivalent to  $\langle\psi|\tilde{\psi}\rangle = 1$  (assuming both states are normalized), which would also be equivalent to  $|\psi\rangle = |\tilde{\psi}\rangle$ . In this section, we present two procedures to obtain such a truncated MPS.

The first procedure is based on the SVD and gives direct access to the error induced by the truncation; the second is a variational approach that results in an optimally truncated representation of the input state.

### Direct Truncation via SVD

The truncation via the SVD is a well established method and described in the literature several times [Sch13, Man17, Hub17]. Nevertheless, we explain it here in full detail in order to deliver a clean and complete description.

Consider a general decomposition of a pure state  $|\psi\rangle \in \mathcal{H} = \mathcal{H}^A \otimes \mathcal{H}^B$ ,

$$|\psi\rangle = \sum_{a,b} \Psi_{a,b} |a\rangle_A \otimes |b\rangle_B, \quad (5.15)$$

with orthonormal bases  $\{|a\rangle_A\} \in \mathcal{H}^A$ ,  $\{|b\rangle_B\} \in \mathcal{H}^B$ , and the coefficient matrix  $\Psi_{a,b}$  with rank  $r = \min(\dim(\mathcal{H}^A), \dim(\mathcal{H}^B))$ . The goal is to reduce the number of used basis states of the

Hilbert spaces  $\mathcal{H}^A$  and  $\mathcal{H}^B$  while still representing the initial state as accurately as possible. In the following, we assume, without loss of generality, that the parts denoted with  $A$  and  $B$  represent the left and the right side of an one-dimensional (1D) system, which is cut at bond  $j$ . Because the bases are orthonormal, the 2-norm of the state  $\|\psi\rangle\|_2^2$  and the Frobenius norm of the coefficient matrix  $\|\Psi_{a,b}\|_F^2$  are equal:

$$\|\psi\rangle\|_2^2 = \langle\psi|\psi\rangle = \sum_{\substack{a,b, \\ a',b'}} \Psi_{a,b}^\dagger \Psi_{a',b'} \underbrace{\langle a|a'\rangle}_{\delta_{a,a'}} \underbrace{\langle b|b'\rangle}_{\delta_{b,b'}} = \sum_{a,b} |\Psi_{a,b}|^2 = \|\Psi_{a,b}\|_F^2. \quad (5.16)$$

Hence, it is sufficient to find an approximation  $\tilde{\Psi}_{a,b}$  for  $\Psi_{a,b}$  with a lower rank in order to obtain a representation of  $|\psi\rangle$  with a reduced number of used basis states. In Sec. 4.1, we have shown that the SVD provides exactly this approximation. Applied on Eq. (5.15), we obtain

$$|\psi\rangle = \sum_{a,b,s} U_{a,s} \Sigma_{s,s} V_{s,b} |a\rangle_A \otimes |b\rangle_B = \sum_s \Sigma_s |s\rangle_A \otimes |s\rangle_B, \quad (5.17)$$

with  $|s\rangle_A = \sum_a U_{a,s} |a\rangle_A$  and  $|s\rangle_B = \sum_b V_{s,b} |b\rangle_B$  again being orthonormal basis elements. Note that the right side of Eq. (5.17) is the Schmidt decomposition, which states that every pure state can be decomposed into two sets of orthonormal basis elements, one for each side, while sharing the same eigenvalues, which are called Schmidt values.

At this point, we can limit  $s$  and therefore restrict the bond dimension between the subsystem  $A$  and the subsystem  $B$ . The error  $\epsilon$  introduced by this truncation is approximately given by the square root of the discarded weight

$$\epsilon = \sum_{s=\tilde{m}+1}^m \Sigma_s^2. \quad (5.18)$$

If the Schmidt values decay fast enough, the discarded weight is small and we obtained a good approximation in terms of Eq. (5.14).

It is surprisingly simple to incorporate these considerations into an algorithm that sequentially truncates every bond of an MPS. As a pre-requirement, we have to ensure that the state is in the mixed-canonical form (see Fig. 5.5). The truncation sweeps in practice usually start from one of the edges, but in the following we nevertheless keep the assumption that  $j$  is the active site.

Depending on the direction in which we want to sweep through the system, we need to adjust the operations of the procedure. In the following, we concentrate on the case of a right sweep, whereas Figs. 5.6 and 5.7 cover both cases. The procedure starts by reshaping the site tensor,

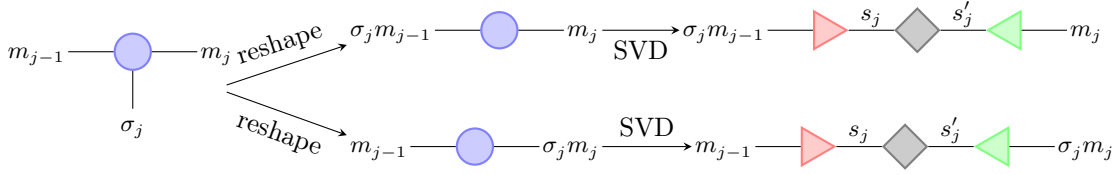
$$M_{j;(\sigma_j m_{j-1}), m_j} = M_{j; m_{j-1}, m_j}^{\sigma_j}. \quad (5.19)$$

Next, an SVD is performed and only the largest  $m_{\max}$  singular values are kept,<sup>2</sup>

$$M_{j;(\sigma_j m_{j-1}), m_j} = \sum_{s_j, s'_j}^{m=\text{rk}(M)} U_{j;(\sigma_j m_{j-1}), s_j} S_{\underline{j}; s_j, s'_j} V_{\underline{j}; s'_j, m_j} \approx \sum_{s_j, s'_j}^{m_{\max} \ll m} U_{j;(\sigma_j m_{j-1}), s_j} S_{\underline{j}; s_j, s'_j} V_{\underline{j}; s'_j, m_j}. \quad (5.20)$$

Up to this point, we have only obtained a representation of  $M$  with a reduced rank.

<sup>2</sup>Note that here we use  $m_{\max}$  instead of  $\tilde{m}$  in order to indicate that this parameter is not specific to the matrix but a parameter of the MPS.

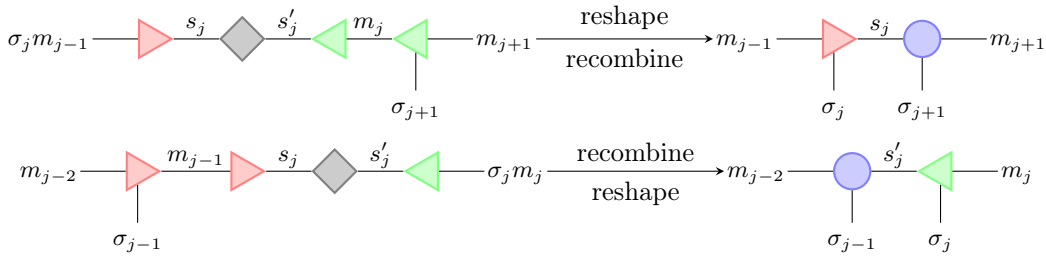


**Figure 5.6:** Pictorial representation of the singular-value decomposition of a rank-three tensor  $M$  into  $U \cdot S \cdot V^\dagger$  within a truncation sweep to the right (top) or to the left (bottom).

In order to reduce the bond dimension  $m$  between site  $j$  and  $j + 1$ , the parts of the matrix decomposition are assigned to the different site tensors (see Fig. 5.7),

$$A_{j;m_{j-1},s_j}^{\sigma_j} = U_{j;(\sigma_j m_{j-1}),s_j} \quad (5.21)$$

$$M_{j+1;s_j,m_{j+1}}^{\sigma_{j+1}} = \sum_{s'_j, m_j}^{m_{\max} \ll m} S_{j;s_j,s'_j} V_{j;s'_j,m_j} B_{j+1;m_j,m_{j+1}}^{\sigma_{j+1}}. \quad (5.22)$$



**Figure 5.7:** Pictorial representation of the assignment of the result of the singular-value decomposition from Fig. 5.6 into the new left (right) normalized rank-three tensor  $A^j$  ( $B^j$ ) and the new active site  $M^{j+1}$  ( $M^{j-1}$ ) at the top (bottom). Note that the index  $s_j$  becomes the new  $m_j$  and the index  $s'_j$  becomes the new  $m_{j-1}$ .

Now, the active site has been moved one site to the right and the procedure can restart with  $j = j + 1$  until the right edge is reached.

Due to this sweeping through the system, the truncation of site  $L - 1$  becomes dependent on the truncation of site 1 but not vice versa. In the case of small truncations, the error resulting from this asymmetry is small and can be ignored, but for a better approximation further steps, e.g., a variational truncation, are necessary.

## Variational Truncation

The variational truncation is an iterative method that starts from an initial guess state  $|\tilde{\psi}\rangle$  with a chosen bond dimension  $m_{\max}$  and variationally minimizes the distance to the untruncated state  $|\psi\rangle$ . A good choice for the initial state is the result of the previous paragraph.

The distance is given by

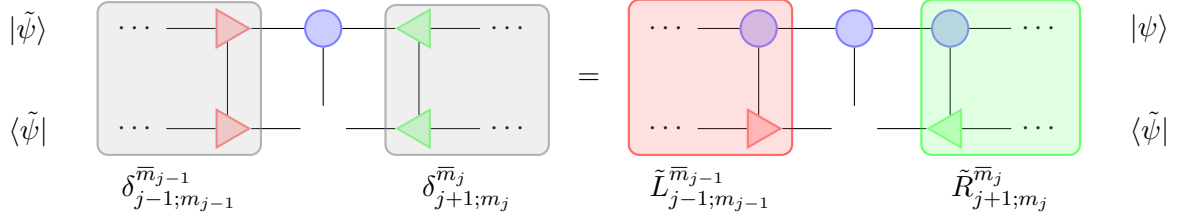
$$\| |\psi\rangle - |\tilde{\psi}\rangle \|_2^2 = \langle \psi | \psi \rangle - \langle \psi | \tilde{\psi} \rangle - \langle \tilde{\psi} | \psi \rangle + \langle \tilde{\psi} | \tilde{\psi} \rangle. \quad (5.23)$$

In order to minimize this global distance, we sweep through the system and minimize only locally with respect to a single site tensor  $\tilde{M}_j$ . Because this tensor only occurs in the second half of

Eq. (5.23), the new (optimized)  $\tilde{M}_j$  can be obtained via

$$\frac{\partial}{\partial \tilde{M}_{j;\bar{m}_{j-1},\bar{m}_j}^{\sigma_j}} \left( \langle \tilde{\psi} | \tilde{\psi} \rangle - \langle \tilde{\psi} | \psi \rangle \right) = 0. \quad (5.24)$$

Let us now consider that the truncated state is in a mixed canonical form with the active site



**Figure 5.8:** Summary of the iterative truncation considering the truncated state to be in a mixed canonical form. The left-hand side can then be reduced to the active site  $j$  that we want to obtain. The right-hand side, which needs to be considered completely, can nevertheless be computed iteratively via the bond tensors  $L$  and  $R$ .

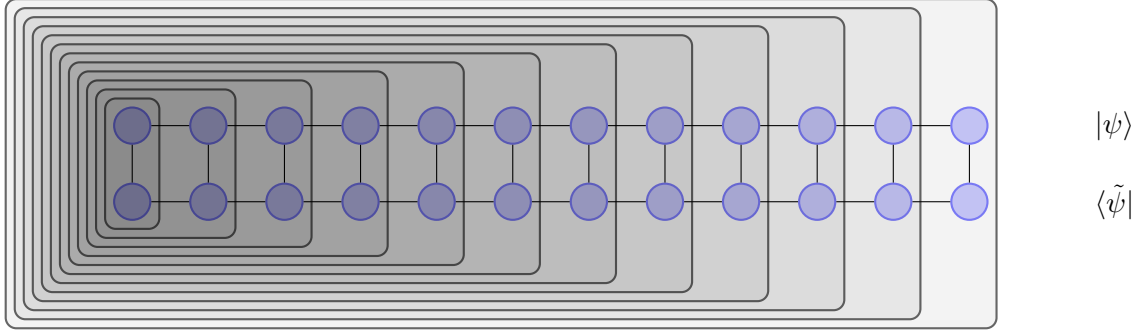
at position  $j$ . Then, the new tensor is given by

$$\tilde{M}_{j;m_{j-1},m_j}^{\sigma_j} = \sum_{\bar{m}_{j-1}} L_{j-1;m_{j-1}}^{\bar{m}_{j-1}} \left( \sum_{\bar{m}_j} R_{j+1;m_j}^{\bar{m}_j} M_{j;m_{j-1},m_j}^{\sigma_j} \right), \quad (5.25)$$

in which the left (right) part of the tensor network (Fig. 5.8, right hand side) are contracted into  $\tilde{L}_{j-1}$  ( $\tilde{R}_{j+1}$ ),

$$\begin{aligned} \tilde{L}_{j-1;m_{j-1}}^{\bar{m}_{j-1}} &= \sum_{\substack{\sigma_{j-1}, \\ \bar{m}_{j-2}, \\ m_{j-2}}} \tilde{A}_{j-1;\bar{m}_{j-2},\bar{m}_{j-1}}^{\sigma_{j-1}} \left( \cdots \left( \sum_{\substack{\sigma_1, \\ \bar{m}_0, \\ m_0}} \tilde{A}_{1;\bar{m}_0,\bar{m}_1}^{\sigma_1} M_{1;m_0,m_1}^{\sigma_1} \right) \cdots \right) M_{j-1;m_{j-2},m_{j-1}}^{\sigma_{j-1}} \quad (5.26) \\ \tilde{R}_{j+1;m_j}^{\bar{m}_j} &= \sum_{\substack{\sigma_{j+1}, \\ \bar{m}_{j+1}, \\ m_{j+1}}} \tilde{B}_{j+1;\bar{m}_j,\bar{m}_{j+1}}^{\sigma_{j+1}} \left( \cdots \left( \sum_{\substack{\sigma_L, \\ \bar{m}_L, \\ m_L}} \tilde{B}_{L;\bar{m}_{L-1},\bar{m}_L}^{\sigma_L} M_{L;m_{L-1},m_L}^{\sigma_L} \right) \cdots \right) M_{j+1;m_j,m_{j+1}}^{\sigma_{j+1}}. \end{aligned} \quad (5.27)$$

Based on this optimization, we start with a canonical state and sweep back and forth through the system until the distance is smaller than a threshold. Note that except at the edges, it is never necessary to calculate the complete contraction of the boundary tensors Eqs. (5.26) and (5.27), because the next tensor in sweep direction has already been calculated in the sweep before, and the other tensor is obtained by enlarging the one from the previous sweep step. Additionally to this single-site variational truncation, there is also a two-site variant, which is more stable against local minima but less performant. Nevertheless, the two-site variant is often used, because it offers the possibility to dynamically change the bond dimension.



**Figure 5.9:** The scalar product  $\langle \tilde{\psi} | \psi \rangle$  of two (possibly different) states represented as MPS. The optimal contraction order is sideways, e.g., from left to right, as indicated by the shading and Eq. (5.30).

### Scalar Product | 5.3.3

In order to calculate the overlap of two states, which is needed, e.g., to calculate the norm of a state, we need to evaluate the scalar product between two MPSs that is defined as

$$\langle \tilde{\psi} | \psi \rangle = \sum_{\substack{\sigma_1 \cdots \sigma_L, \\ \sigma'_1 \cdots \sigma'_L}} \langle \sigma'_1 \cdots \sigma'_L | \sum_{m_0, \dots, m_L} \bar{M}_{L; m_{L-1}, m_L}^{\sigma'_L} \cdots \bar{M}_{1; m_0, m_1}^{\sigma'_1} M_{1; m_0, m_1}^{\sigma_1} \cdots M_{L; m_{L-1}, m_L}^{\sigma_L} | \sigma_1 \cdots \sigma_L \rangle . \quad (5.28)$$

Because the result of every tensor contraction in Eq. (5.28) is a number and  $\langle \sigma'_1 \cdots \sigma'_L | \sigma_1 \cdots \sigma_L \rangle = \delta_{\sigma'_1 \cdots \sigma'_L, \sigma_1 \cdots \sigma_L}$ , the sum shrinks to

$$\langle \tilde{\psi} | \psi \rangle = \sum_{\substack{\sigma_1 \cdots \sigma_L, \\ m_0, \dots, m_L}} \bar{M}_{L; m_{L-1}, m_L}^{\sigma_L} \cdots \bar{M}_{1; m_0, m_1}^{\sigma_1} M_{1; m_0, m_1}^{\sigma_1} \cdots M_{L; m_{L-1}, m_L}^{\sigma_L} . \quad (5.29)$$

The computation of this expression is costly, because it consists of  $d^L \cdot (2L - 1)$  tensor contractions. Hence, the complexity increases exponentially with the system size. It is therefore beneficial to reformulate the expression into

$$\langle \tilde{\psi} | \psi \rangle = \sum_{\substack{\sigma_L, m_{L-1}, \\ m_L}} \bar{M}_{1; m_{L-1}, m_L}^{\sigma_L} \left( \cdots \left( \sum_{\sigma_1, m_0} \bar{M}_{1; m_0, m_1}^{\sigma_1} M_{1; m_0, m_1}^{\sigma_1} \right) \cdots \right) M_{L; m_{L-1}, m_L}^{\sigma_L} , \quad (5.30)$$

as depicted in Fig. 5.9. For the evaluation of Eq. (5.30), only  $dL$  additions and  $2dL - d$  tensor contractions are necessary, i.e., the evaluation scales polynomially with the system size. Note that the sum over the dummy index  $m_L$  is distinct, because it does not follow the rule for the intermediate sums.

### Addition and Scaling of MPSs | 5.3.4

In order to add two MPSs  $|\psi\rangle$  with site tensors  $M_j$  and  $|\tilde{\psi}\rangle$  with site tensors  $\tilde{M}_j$ , as it is necessary in the Lanczos algorithm (Sec. 4.2), the direct sum of all site tensors needs to be performed. That

means

$$|\psi\rangle + |\tilde{\psi}\rangle = \sum_{\substack{\sigma_1, \dots, \sigma_L, \\ m'_0, \dots, m'_L}} M'_{1; m'_0, m'_1}{}^{\sigma_1} \cdots M'_{L; m'_{L-1}, m'_L}{}^{\sigma_L} |\sigma_1 \dots \sigma_L\rangle \quad (5.31)$$

with

$$M'_{j; m'_{j-1}, m'_j}{}^{\sigma_j} = \begin{pmatrix} M_{j; m_{j-1}, m_j}{}^{\sigma_j} & 0 \\ 0 & \tilde{M}_{j; \tilde{m}_{j-1}, \tilde{m}_j}{}^{\sigma_j} \end{pmatrix} = M_{j; m_{j-1}, m_j}{}^{\sigma_j} \oplus \tilde{M}_{j; \tilde{m}_{j-1}, \tilde{m}_j}{}^{\sigma_j} \quad (5.32)$$

and the two exceptions

$$M'_{1; m'_0, m'_1}{}^{\sigma_1} = \begin{pmatrix} M_{1; m_0, m_1}{}^{\sigma_1} & \tilde{M}_{1; \tilde{m}_0, \tilde{m}_1}{}^{\sigma_1} \end{pmatrix} \quad M'_{L; m'_{L-1}, m'_L}{}^{\sigma_L} = \begin{pmatrix} M_{L; m_{L-1}, m_L}{}^{\sigma_L} \\ \tilde{M}_{L; \tilde{m}_{L-1}, \tilde{m}_L}{}^{\sigma_L} \end{pmatrix}. \quad (5.33)$$

Note that the dimension of the new state  $|\psi'\rangle$  is given by the sum of the dimensions of the initial states. The indices of the new state are therefore given by  $m'_j = m_j + \tilde{m}_j$ .

If the goal is only to scale the MPS, this approach is far from optimal, because the dimension is doubled while it could stay constant by rescaling the norm of the MPS. Such a rescaling can for example be performed at the end of a truncation sweep.

### Reduced Density Operator and Entanglement Entropy | 5.3.5

A last quantity that can be accessed directly from an MPS is the reduced density operator for a specific partition and the closely related von Neumann entanglement entropy or short entanglement entropy. The density operator of a pure state  $|\psi\rangle$  is given by

$$\hat{\rho} = |\psi\rangle \langle \psi|. \quad (5.34)$$

In order to obtain the reduced density operator of a specific part, e.g., part  $A$  that includes all sites left of site  $j$ , the other part ( $B$  with all sites right of site  $j - 1$ ) need to be traced out

$$\hat{\rho}_A = \text{Tr}_B |\psi\rangle \langle \psi|. \quad (5.35)$$

This quantity becomes particularly accessible if we express the state in its Schmidt decomposition (Eq. (5.17)),

$$\hat{\rho}_A = \text{Tr}_B \left( \sum_s \Sigma_s |s\rangle_A |s\rangle_B \sum_{s'} \Sigma_{s'} \langle s'|_B \langle s'|_A \right) \quad (5.36)$$

$$= \sum_{s, s'} \Sigma_s |s\rangle_A \langle s'|_A \underbrace{\text{Tr}(\Sigma_{s'} \langle s'|_B |s\rangle_B)}_{\delta_{s, s'}} \quad (5.37)$$

$$= \sum_s \Sigma_s^2 |s\rangle_A \langle s|. \quad (5.38)$$

And analogously, we obtain the reduced density operator for the part  $B$ ,

$$\hat{\rho}_B = \text{Tr}_A |\psi\rangle \langle \psi| = \sum_s \Sigma_s^2 |s\rangle_B \langle s|. \quad (5.39)$$

Note that the reduced density operators share the same spectrum, but of course, act on different parts of the system. Furthermore, this spectrum is given by the square of the singular values of the SVD of an active site  $j$ , if a truncation sweep to the left is performed.<sup>3</sup>

At this point, we can directly read off the von-Neumann entanglement entropy

$$S_{A|B}(|\psi\rangle) = -\text{Tr}\rho_A \log_2 \rho_A = -\sum_s \Sigma_s^2 \log_2 \Sigma_s^2, \quad (5.40)$$

which is a quantity that describes how much  $|\psi\rangle$  differs from a product state, i.e., how entangled the parts  $A$  and  $B$  of the system are with each other.

## Finite-Temperature Representation | 5.4

For calculations at a finite temperature  $T = \frac{1}{\beta k_B}$ , which are presented in Sec. 11.3, two ingredients are necessary. First, a state for which we can define a temperature, with respect to a given Hamiltonian, and second, a way to change the corresponding temperature. In this section, we summarize a way to represent a mixed state with an MPS that is called purification and present two different ways to obtain a state at infinite temperature, i.e.,  $\beta = 0$ . Afterwards, the imaginary-time evolution, which is analog to the real-time evolution (see chapter 9), can be used as a tool to reduce the temperature of such a state.

### Purification | 5.4.1

A quantum-mechanical state at a finite temperature is represented by an ensemble, which can be represented by a mixed state. MPSs are wave functions and hence can only represent pure states. It is therefore necessary to use a subsystem of an MPS, because if two subsystems are entangled, their reduced density operators represent mixed states. To be more precise, in the purification approach it is necessary to double the system and define one half of the sites as the physical subsystem  $P$  and the other half as the auxiliary subsystem  $Q$ . The desired properties then follow directly from the Schmidt decomposition,

$$|\psi_{\beta=0}\rangle = \sum_s \Sigma_s |s\rangle_P |s\rangle_Q \Rightarrow \hat{\rho}_P = \sum_s \Sigma_s^2 |s\rangle_P \langle s|_P. \quad (5.41)$$

For the following consideration, we need to assume that we can obtain a state at infinite temperature  $|\psi_{\beta=0}\rangle$  with the corresponding density operator  $\hat{\rho}_0$ , which we show below. The thermal density operator  $\hat{\rho}_\beta$  can then be derived via

$$\hat{\rho}_\beta = \frac{e^{-\beta\hat{H}}}{Z(\beta)} \stackrel{Z(0)\hat{\rho}_0 = \hat{\text{Id}}}{=} \frac{Z(0)}{Z(\beta)} \text{Tr}_Q \left( e^{-\beta/2\hat{H}} |\psi_{\beta=0}\rangle \langle\psi_{\beta=0}| e^{-\beta/2\hat{H}} \right), \quad (5.42)$$

with the partition function  $Z(\beta) = \text{Tr}_P e^{-\beta\hat{H}}$ . The partial trace can be expanded around the exponentials, because the Hamiltonian does not act on the auxiliary subsystem  $Q$ . Because  $\beta \in \mathbb{R}$ , we can define  $|\psi_\beta\rangle = e^{-\frac{\beta}{2}\hat{H}} |\psi_{\beta=0}\rangle$ , which shows that by using an imaginary-time evolution up to time  $\tau = -i\frac{\beta}{2}$ , we obtain a state at temperature  $T = \frac{1}{\beta k_B}$ .

<sup>3</sup>If a truncation sweep to the right is performed and the spectrum at bond  $\underline{j}$  shall be obtained, the singular values of the SVD of the active site  $j+1$  need to be considered.



In order to evaluate expectation values, we need to employ the fact that the density operator contains all physically measurable information. Hence, we can obtain thermally averaged expectation values via

$$\langle \hat{O} \rangle_\beta = \frac{Z(0)}{Z(\beta)} \text{Tr}_P \left( \hat{O} \hat{\rho}_\beta \right) = \frac{Z(0)}{Z(\beta)} \langle \psi_\beta | \hat{O} | \psi_\beta \rangle . \quad (5.43)$$

Note that  $Z(0) = d^L$ , and  $Z(\beta)$  is given by the normalization factors of the purified state in the progress of the imaginary-time evolution. Hence, we can use the method introduced in Sec. 6.2 in order to calculate expectation values of the thermal ensemble.

## Infinite-Temperature States | 5.4.2

In order to fulfill the assumption that an infinite-temperature state can be obtained, here we present three different ways to accomplish this task. Note that we identify an infinite-temperature state with a maximally entangled mixed state.

Consider the factorization

$$\hat{\rho}_0 = \frac{\hat{\text{Id}}}{d^L} = \left( \frac{\hat{\text{Id}}}{d} \right)^{\otimes L} , \quad (5.44)$$

which yields an enlarged local site  $j$  that consists of one physical site  $p(j)$  and one auxiliary site  $q(j)$ ,

$$\frac{\hat{\text{Id}}_j}{d} = \text{Tr}_Q |\psi\rangle_j \langle \psi|_j \quad |\psi\rangle_j = \frac{1}{\sqrt{d}} \sum_\sigma |\sigma\rangle_{p(j)} \otimes |\sigma\rangle_{q(j)} . \quad (5.45)$$

At this point, the methodically simplest approach for the infinite-temperature state is to directly apply Eq. (5.45). This means, for a spin-1/2 system

$$|\psi\rangle_j = \frac{1}{\sqrt{2}} \left( |\uparrow\rangle_{p(j)} \otimes |\uparrow\rangle_{q(j)} + |\downarrow\rangle_{p(j)} \otimes |\downarrow\rangle_{q(j)} \right) . \quad (5.46)$$

This strategy is surely expandable to arbitrary systems, but the resulting states also lack properties necessary to employ multiple conserved quantum numbers. Hence, for systems with larger local basis or multiple conserved quantum numbers, more sophisticated methods are needed to obtain a canonical infinite-temperature state. Two of those methods are presented in the following.

### Entangler Hamiltonian

In Ref. [NA16], a way to obtain a state that conserves multiple  $U(1)$  symmetries of a model, e.g., total spin and particle-number conservation, independently within the physical and the auxiliary system is presented. The idea is to formulate a so-called entangler-Hamiltonian or, short, entangler, whose ground state is the desired state at  $\beta = 0$ , and to perform a ground-state search (see chapter 8) with that operator. Note, that the entangler is constructed only by fixing the particle statistics, e.g., spin-1/2 fermions. Therefore, the same entangler can be used for the construction of infinite-temperature states as long as the local basis matches.

Here we follow [NA16] and construct the entangler-Hamiltonian for a system consisting of spin-1/2 fermions by choosing

$$\hat{H}_{\text{C2}}^{\text{Spin-}\frac{1}{2}\text{-fermions}} = - \sum_{\substack{j \neq j', \\ \sigma = \uparrow, \downarrow}} \left( \hat{\Lambda}_{\sigma, j}^\dagger \hat{\Lambda}_{\sigma, j'} + \text{h.c.} \right), \quad (5.47)$$

with

$$\hat{\Lambda}_{\sigma, j} = \hat{c}_{\sigma, p(j)} \hat{c}_{\bar{\sigma}, q(j)} \hat{P}_j^\sigma \quad (5.48)$$

and

$$\hat{P}_j^\sigma = |1 - \hat{n}_{\bar{\sigma}, p(j)} - \hat{n}_{\sigma, q(j)}|. \quad (5.49)$$

Still,  $p(j)$  labels physical sites and  $q(j)$  the corresponding auxiliary sites.  $\bar{\sigma}$  denotes the opposite spin direction of  $\sigma$ .

In Ref. [NA16], further entangler-Hamiltonians are presented, but we refrain from reviewing them here, because they are not relevant for this thesis. Moreover, the following approach turns out to be conceptually easier and computationally faster, because no ground-state search is necessary.

### Iterative Filling

Another method to create a canonical infinite-temperature state is introduced in Ref. [Bar16]. The main idea is to start from a vacuum state and to homogenously fill the system iteratively with maximally entangled states on the corresponding rungs until the desired quantum numbers are realized. This filling is accomplished by applying a global operator, which, in the case of a spin-1/2-fermion system, is given by

$$\hat{C}_{\text{tot}}^\dagger = \sum_{j=1}^L \left[ \hat{c}_{\uparrow, p(j)}^\dagger \otimes \hat{c}_{\downarrow, q(j)}^\dagger + \hat{c}_{\downarrow, p(j)}^\dagger \otimes \hat{c}_{\uparrow, q(j)}^\dagger \right]. \quad (5.50)$$

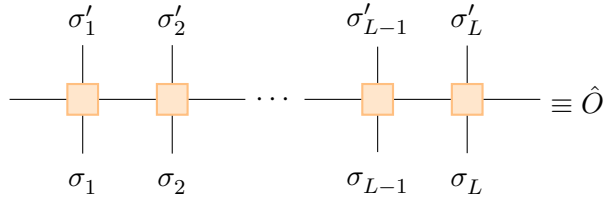
This operator already ensures that the total spin is zero in the full system as well as in the individual subsystems. In order to assure the constant spin, it is necessary to add two particles in either subsystem with every application of the operator Eq. (5.50) distributed over all sites. If a different total spin is desired, another operator is needed. In Sec. 10.4.4, we review this approach in the context of our QCS and present an intuitive way how to perform the necessary operations.

Analogous to quantum states, we can express every operator  $\hat{O} : \mathcal{H} \rightarrow \mathcal{H}$  as a matrix-product operator (MPO). We define:

$$\hat{O} = \sum_{\substack{\sigma_1, \dots, \sigma_L, \\ \sigma'_1, \dots, \sigma'_L}} c_{\sigma_1 \dots \sigma_L, \sigma'_1 \dots \sigma'_L} |\sigma_1 \dots \sigma_L\rangle \langle \sigma'_1 \dots \sigma'_L| \quad (6.1)$$

$$= \sum_{\substack{\sigma_1, \dots, \sigma_L, \\ \sigma'_1, \dots, \sigma'_L, \\ w_0, \dots, w_L}} W_{1;w_0,w_1}^{\sigma_1,\sigma'_1} \dots W_{L;w_{L-1},w_L}^{\sigma_L,\sigma'_L} |\sigma_1 \dots \sigma_L\rangle \langle \sigma'_1 \dots \sigma'_L| . \quad (6.2)$$

The  $W$ -tensors will do the same job as the  $M$ -tensors of matrix-product states (MPSs), the only difference being the number of physical indices, which is two ( $\sigma_j, \sigma'_j$ ) instead of one ( $\sigma_j$ ) per site (see Fig. 6.1). We denote the bond dimension of the matrix-product operator (MPO) with  $w$ .



**Figure 6.1:** Schematic of the tensor network of a matrix-product operator (MPO). Horizontal lines denote the internal indices with bond dimension  $w$ , whereas the vertical lines denote physical indices with dimension  $d$ .

In the following, we present different ways to apply an MPO to an MPS. Afterwards, we discuss in detail how to create MPOs from finite-state machines (FSMs) in chapter 7, before we finally show the two main uses of the MPS/MPO framework, the ground-state search, and two different time-evolution schemata.

## Application of an MPO to an MPS

## 6.1

One of the most important operations within the framework of MPSs/MPOs is the application of an MPO to an MPS. In general, this can be done by a straightforward tensor product of the corresponding site tensors of the MPS and MPO. In practice, this is not the method of choice as in most applications the resulting target state  $\hat{O}|\psi\rangle$  has a much higher bond dimension, which however, is not required to represent the target state efficiently. This becomes apparent when looking at time-evolution operators represented by the MPOs time step  $\hat{U}(\mathcal{T})$  with  $\mathcal{T} \ll 1$ , because only a moderate growth of entanglement entropy per time step is expected (see

chapter 9). It is therefore helpful to look at different approaches. Nevertheless, we begin with the direct application before we turn to more elaborate application schemes.

## Direct Application | 6.1.1

The direct application of an MPO to an MPS is obtained by regrouping the contractions so that the target MPS tensor can be obtained by performing the tensor product of the individual site tensors

$$\hat{O}|\psi\rangle = \sum_{\substack{\sigma_1, \dots, \sigma_L, m_0, \dots, m_L, \\ \sigma'_1, \dots, \sigma'_L, w_0, \dots, w_L \\ \sigma''_1, \dots, \sigma''_L}} \left( W_{1;w_0,w_1}^{\sigma_1 \sigma'_1} \cdots W_{L;w_{L-1},w_L}^{\sigma_L \sigma'_L} \right) |\sigma_1 \cdots \sigma_L\rangle \langle \sigma'_1 \cdots \sigma'_L| \\ \times \left( M_{1;m_0,m_1}^{\sigma''_1} \cdots M_{L;m_{L-1},m_L}^{\sigma''_L} \right) |\sigma''_1 \cdots \sigma''_L\rangle \quad (6.3)$$

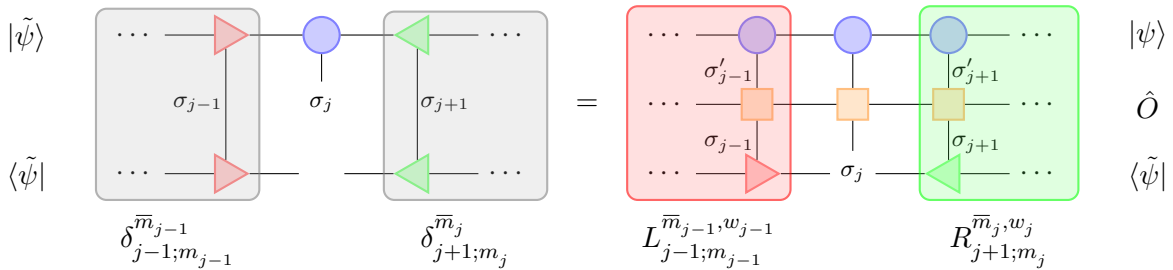
$$= \sum_{\substack{\sigma_1, \dots, \sigma_L, \\ m_0, \dots, m_L, \\ w_0, \dots, w_L}} \tilde{M}_{1;(w_0 m_0), (w_1 m_1)}^{\sigma_1} \cdots \tilde{M}_{L;(w_{L-1} m_{L-1}), (w_L m_L)}^{\sigma_L} |\sigma_1 \cdots \sigma_L\rangle = |\tilde{\psi}\rangle, \quad (6.4)$$

with the tensors  $\tilde{M}_j$  given by

$$\tilde{M}_{j;(w_{j-1} m_{j-1}), (w_j m_j)}^{\sigma_j} = \sum_{\sigma'_j} W_{j;w_{j-1}, w_j}^{\sigma_j \sigma'_j} M_{j;m_{j-1}, m_j}^{\sigma'_j}. \quad (6.5)$$

The resulting state  $|\tilde{\psi}\rangle$  is therefore again an MPS, but with a larger dimension  $\tilde{m} = m \cdot w$ . Repeated application of an operator onto a state strongly increases the dimension of the state, and truncation becomes necessary.

## Variational Application | 6.1.2



**Figure 6.2:** Summary of the variational application of an MPO to an MPS considering the truncated target state  $|\tilde{\psi}\rangle$  to be in a mixed canonical form. The left-hand side can then be reduced to the active site  $j$  that we want to obtain. The right-hand side, which needs to be considered completely, can nevertheless be calculated iteratively via the bond tensors  $L_{j-1}$  and  $R_{j+1}$ .

In the spirit of variationally compressing a state towards a target state, we can apply the same considerations to the application of an MPO to a source state  $|\psi\rangle$  with the subsequent compression of the state in one operation. Therefore, we seek to minimize the distance between a target state  $|\tilde{\psi}\rangle$  with bond dimension  $\tilde{m}$  and the source state with the MPO applied to it, which we denote as  $\hat{O}|\psi\rangle$ . The distance is in this case given by

$$\| |\tilde{\psi}\rangle - \hat{O}|\psi\rangle \|_2^2 = \langle \psi | \hat{O}^\dagger \hat{O} | \psi \rangle - \langle \psi | \hat{O}^\dagger | \tilde{\psi} \rangle - \langle \tilde{\psi} | \hat{O} | \psi \rangle + \langle \tilde{\psi} | \tilde{\psi} \rangle . \quad (6.6)$$

Hence, we can obtain the new tensor  $M_j$ , on which the operator is already applied and which is compressed, via

$$\frac{\partial}{\partial \overline{M}_{j;\overline{m}_{j-1},\overline{m}_j}^{\sigma_j}} \left( \langle \tilde{\psi} | \tilde{\psi} \rangle - \langle \tilde{\psi} | \hat{O} | \psi \rangle \right) = 0 . \quad (6.7)$$

If we keep the target state in a mixed canonical form, then the above set of coupled equations again reduces to a local update scheme for the target tensors

$$\overline{M}_{j;\overline{m}_{j-1},\overline{m}_j}^{\sigma_j} = \sum_{\overline{m}_{j-1},w_{j-1}} L_{j-1;\overline{m}_{j-1},w_{j-1}}^{\overline{m}_{j-1},w_{j-1}} \left( \sum_{\sigma'_j,\overline{m}_j,w_j} W_{j;\overline{m}_{j-1},w_j}^{\sigma_j,\sigma'_j} M_{j;\overline{m}_{j-1},\overline{m}_j}^{\sigma'_j} R_{j+1;\overline{m}_j,w_j}^{\overline{m}_j,w_j} \right) , \quad (6.8)$$

where the boundary tensors  $L_{j-1}, R_{j+1}$  can be built recursively by sweeping through the system and evaluating the contractions

$$L_{j-1;\overline{m}_{j-1},w_{j-1}}^{\overline{m}_{j-1},w_{j-1}} = \sum_{\substack{\sigma_{j-1},\sigma'_{j-1}, \\ m_{j-2},w_{j-2}}} \overline{A}_{j-1;\overline{m}_{j-2},\overline{m}_{j-1}}^{\sigma'_{j-1}} L_{j-2;\overline{m}_{j-2},w_{j-2}}^{\overline{m}_{j-2},w_{j-2}} W_{j-1;\overline{m}_{j-2},w_{j-1}}^{\sigma_{j-1},\sigma'_{j-1}} M_{j-1;\overline{m}_{j-2},\overline{m}_{j-1}}^{\sigma_{j-1}} \quad (6.9)$$

$$R_{j+1;\overline{m}_j,w_j}^{\overline{m}_j,w_j} = \sum_{\substack{\sigma_{j+1},\sigma'_{j+1}, \\ w_{j+1},m_{j+1}}} W_{j+1;\overline{m}_j,w_{j+1}}^{\sigma_{j+1},\sigma'_{j+1}} M_{j+1;\overline{m}_j,m_{j+1}}^{\sigma'_{j+1}} R_{j+2;\overline{m}_j,w_{j+1}}^{\overline{m}_j,w_{j+1}} \overline{B}_{j+1;\overline{m}_j,\overline{m}_{j+1}}^{\sigma'_{j+1}} . \quad (6.10)$$

The corresponding onsite contractions are depicted in Fig. 6.2. Analogous to the variational truncation, it is helpful to start with a good guess for the target state and optimize it via several sweeps until the distance is small enough. Next, a method to obtain a good initial state is provided.

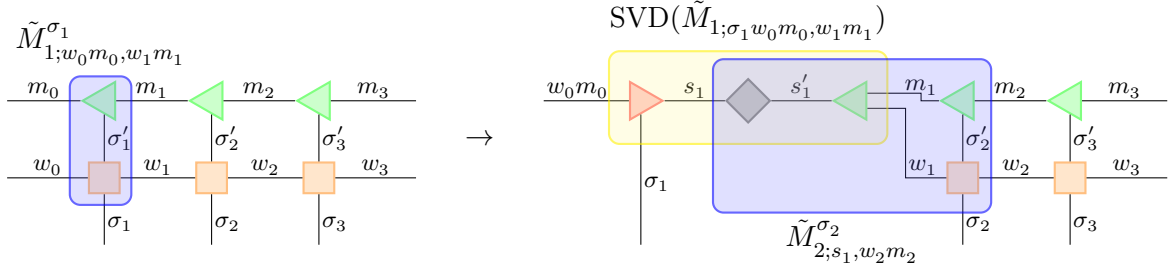
### Zip-Up | 6.1.3

A powerful alternative to the direct application of an MPO to an MPS is the zip-up method proposed in [SW10]. The central assumption is that the operator, e.g., a time-evolution operator (see chapter 9), only gently destroys the canonical form of the MPS. Hence, a modest truncation is already possible within the contraction process.

The first step is similar to Eq. (6.5), but includes the right-normalized tensor  $B$ ,

$$M_{1;w_0,m_0,w_1,m_1}^{\sigma_1} = \sum_{\sigma'_1} W_{1;w_0,w_1}^{\sigma_1,\sigma'_1} B_{1;m_0,m_1}^{\sigma'_1} . \quad (6.11)$$

Note that  $m_0$  and  $w_0$  are dummy indices and can therefore easily be fused into a new dummy index. Applying the singular-value decomposition (SVD) with a relaxed truncation criterion as



**Figure 6.3:** Graphical summary of the essential steps of the zip-up method proposed in Ref. [SW10]. (Left) initial tensor network that shall be contracted, consisting of a right-canonical MPS and a MPO that only gently destroys the canonical form. The first step is to interpret the combination of the first MPS tensor and the first MPO tensor as a new tensor  $M'_{1;w_0m_0,w_1m_1}$ , which is slightly non-normalized and therefore framed in blue. (Right) the next step is to apply a SVD (including a gentle truncation) on the tensor  $M'_1$  and finally build the next tensor  $M'_2$  leaving the contracted, left-normalized MPS tensor  $A_{1;m_0w_0,s_1}$  on the left side.

described in Eqs. (5.19) to (5.21), we obtain the left-normalized tensor  $A_1$ , now with a single right index  $s_1$ ,

$$M'_{1;w_0m_0,w_1m_1} = \sum_{s_1,s'_1} A_{1;m_0w_0,s_1}^{\sigma_1} S_{\perp;s_1,s'_1} V_{\perp;s'_1,w_1m_1}. \quad (6.12)$$

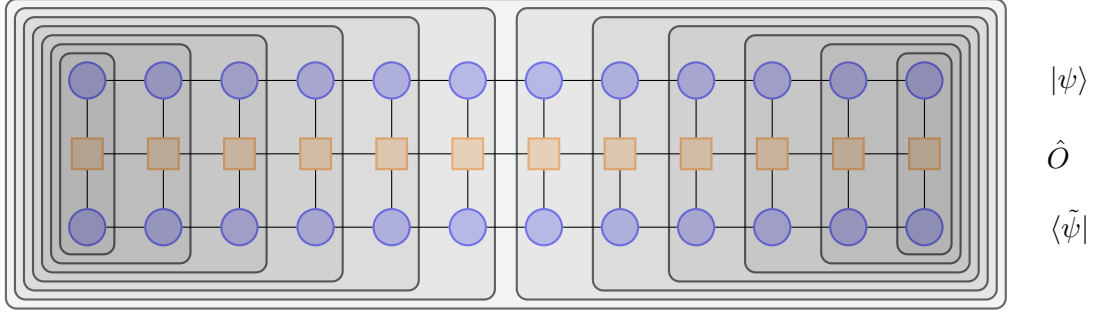
In the next step, the remaining parts of the result of the SVD are incorporated as in Eq. (5.22), but also including the MPO tensor, to obtain the next slightly non-normalized tensor

$$M'_{2;s_1,w_2m_2} = \sum_{\sigma'_2,s'_1,m_1,w_1} S_{\perp;s_1,s'_1} V_{\perp;s'_1,w_1m_1} W_{2;w_1,w_2}^{\sigma_2\sigma'_2} B_{2;m_1,m_2}^{\sigma'_2}. \quad (6.13)$$

This procedure, which is depicted in Fig. 6.3, is repeated until the right end of the system is reached and hence the complete operator is applied. For the relaxed truncation scheme, a maximal growth factor of the MPS bond dimension  $m' = 2m$  and a truncated weight of  $1/10$  of the target weight turns out to be a suitable choice. Afterwards, a sweep back applying a tougher truncation is possible as the new MPS is fully left-canonical.

## Matrix Elements and Expectation Values | 6.2

In order to compute matrix elements, we only need to include an MPO into the tensor network of the scalar product. Analogous to the evaluation of the scalar product (Sec. 5.3.3), the complexity



**Figure 6.4:** The matrix element  $\langle \tilde{\psi} | \hat{O} | \psi \rangle$  of an operator between two states represented as MPO and MPS respectively. If  $|\tilde{\psi}\rangle = |\psi\rangle$ , we obtain an expectation value. The optimal contraction order is sideways, e.g., from left to right, as indicated by the shading. An easy option for two-fold parallelization is the concurrent evaluation of  $L_{j-1}$  and  $R_j$  from left and right, respectively.

of this contraction can be drastically reduced by changing the contraction order,

$$\langle \tilde{\psi} | \hat{O} | \psi \rangle = \sum_{\substack{\sigma_1 \dots \sigma_L, \\ \sigma'_1 \dots \sigma'_L, \\ m_0, \dots, m_L, \\ w_0, \dots, w_L}} \bar{M}_{L; m_{L-1}, m_L}^{\sigma_L} \dots \bar{M}_{1; m_0, m_1}^{\sigma_1} W_{1; w_0, w_1}^{\sigma_1, \sigma'_1} \dots W_{L; w_{L-1}, w_L}^{\sigma_L, \sigma'_L} M_{1; m_0, m_1}^{\sigma'_1} \dots M_{L; m_{L-1}, m_L}^{\sigma'_L} \quad (6.14)$$

$$= \sum_{\substack{\sigma'_L, \\ \sigma_L, \\ m_{L-1}, \\ m_L}} W_{L; w_{L-1}, w_L}^{\sigma_L, \sigma'_L} \bar{M}_{1; m_{L-1}, m_L}^{\sigma_L} \left( \dots \left( \sum_{\substack{\sigma_1, \\ m_0}} W_{1; w_0, w_1}^{\sigma_1, \sigma'_1} \bar{M}_{1; m_0, m_1}^{\sigma_1} M_{1; m_0, m_1}^{\sigma'_1} \right) \dots \right) M_{L; m_{L-1}, m_L}^{\sigma'_L} \cdot \quad (6.15)$$

Furthermore, an easy two-fold parallelization strategy is to concurrently evaluate this contraction from both sides, as shown in Fig. 6.4. If the goal is to calculate an expectation value of a local operator, i.e., only a single site tensor of the corresponding MPO is unequal to the identity, and the MPS is in a mixed canonical form with its active sites being the site of the local operator, then the evaluation boils down to

$$\langle \psi | \hat{O} | \psi \rangle = \sum_{\sigma_j, \sigma'_j} W_{j; w_{j-1}, w_j}^{\sigma_{j-1}, \sigma'_j} \bar{M}_{j; m_{j-1}, m_j}^{\sigma_j} M_{j; m_{j-1}, m_j}^{\sigma'_j} \cdot \quad (6.16)$$

At this point we have learnt how to use MPOs, but the question arises how to actually build them. One answer to this question is given in the next chapter, which shows how to represent arbitrary states as FSMs and demonstrates the relationship between FSMs and MPOs.





The main content of this chapter is published as a regular article in SciPost Physics [PKM17] and therefore some parts of the following chapter are directly adopted from the manuscript of Ref. [PKM17]. The parts of the manuscript used here were written cooperatively by Sebastian Paeckel and me, and revised together with Salvatore R. Manmana. The usage of finite-state machines (FSMs) to represent matrix-product operators (MPOs) is discussed in the original works Ref. [CB08] and Ref. [CDV08]. Here, those parts are reviewed for completeness, since they are necessary to explain the graph-manipulation-based arithmetics and the compression schemes. How to include  $U(1)$  symmetries into the FSM formalism is a contribution of Sebastian Paeckel and is mentioned here to show the benefits of this representation.

Finite-state machines are a class of abstract machines from theoretical computer science. They are less powerful than Turing machines, which can in principle simulate every (classical) algorithm. The limitation of an FSM is due to the limited amount of memory, namely only the current state can be interpreted as memory. On the other hand, the fact that FSMs have memory makes them more powerful than plain combinational logic. Although there are several classifications of FSMs in computer science, we restrict ourself in the following to acceptors and their capability to define a regular language.<sup>1</sup> In order to explain FSMs, it is important to note that states in this context are states of the machine and not quantum-mechanical wave functions.

Throughout this section, we consider the Heisenberg model (see Sec. 3.2) as example. Furthermore, we use three different ways to describe FSMs within this thesis. All of them contain the full amount of information, hence they can be converted to each other.<sup>2</sup> Nevertheless, for different considerations different representations are more convenient. Those representations will be presented next, followed by the description of how to get from an FSM to an MPO (including a detailed example). Afterwards, the arithmetic operations and graph-based compressions are introduced. This chapter closes with an example that shows the precision of the introduced techniques.

### Mathematical Definition

A finite-state machine  $M$ , and therefore a regular language  $\Sigma$ , is fully specified by a 5-tuple  $M(\mathcal{K}, \mathcal{L}, \delta, I, \mathcal{F})$ .  $\mathcal{K}$  is the alphabet of the FSM and contains all symbols the described language consists of. The set  $\mathcal{L}$  contains all states the FSM can be in, including the initial state  $I$  and all final states  $\mathcal{F}$ .  $\delta$  is a set of transition functions  $\delta : \mathcal{L} \times \mathcal{K} \rightarrow \mathcal{L}$  that describes all permitted transitions between states. Note that we usually only allow for a single final state, hence we reduce the set  $\mathcal{F}$  to its only element, the final state  $F$ .

### Graph Representation

In order to represent  $M$  graphically, it is common to draw a graph in which every node represents a state  $q \in \mathcal{L}$  and every edge represents a permitted transition  $\delta$ . In this thesis, a graph

<sup>1</sup>A language is regular if there is a FSM that accepts exactly all words of the language.

<sup>2</sup>Of course, there is an exception to this rule, which is curable and is discussed when needed.

representation is denoted with  $\Lambda(M)$  and the initial/final state will be colored in green/red. Examples for this representation are given throughout the thesis, e.g., Figs. 7.3 and 10.15. This representation will be used to a great extent in Secs. 7.2 and 7.3 in order to illustrate the algorithmic ideas.

### State-Transition-Table Representation

This representation nearly encapsulates the complete information into a single table with  $n$  rows and  $n$  columns, where  $n$  is the amount of states. Every element in this table is a permitted transition from the state denoted by the row to the state denoted by the column. There is no possibility to specify an initial or final state in this representation, but we usually use the first/last row and column to denote the initial/final state.

In Sec. 7.1.1, we show that this representation is very closely related to the site tensors of MPOs. Furthermore, an example for this representation is presented in Fig. 7.2.

## A Language that Describes All Operators | 7.1

The goal of this section is to provide a recipe how to transform any operator to an FSM and from there to an MPO. We therefore need to switch between the mathematical definition and the graph representation several times, hence we use the words graph and FSM interchangeably from this point on.

Let  $\mathcal{K} = \{\hat{k}_1^{(j)}, \dots, \hat{k}_m^{(j)}\}$  be a set of  $m \in \mathbb{N}$  local operators  $\hat{k}^{(j)} : \mathcal{H}_d \rightarrow \mathcal{H}_d$ , i.e., operators acting only on the  $d$ -dimensional local Hilbert space on site  $j$ . This set of operators will become the alphabet of the FSM. Any global operator can be decomposed into the general form

$$\hat{O} = \sum_{\nu,r} \hat{O}_{\nu,r} = \sum_{\nu,r} \sum_j \hat{\delta}_{\nu,r}^{(j)}, \quad (7.1)$$

with

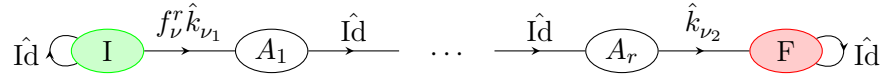
$$\hat{\delta}_{\nu,r}^{(j)} = f_{\nu_1 \dots \nu_n}^r \cdot \hat{k}_{\nu_1}^{(j)} \otimes \dots \otimes \hat{k}_{\nu_n}^{(j+r)} \quad (7.2)$$

being a string of  $n$  local operators  $\hat{k}_{\nu_l}^{(j')} \in \mathcal{K}$  with  $l \in [1, n]$  and  $j' \in [j, j+r]$ . It couples lattice site  $j$  with range  $r+1 \geq n$  and amplitude  $f_{\nu_1 \dots \nu_n}^r \in \mathbb{C}$ . Note that we abbreviate the index set  $\nu = (\nu_1 \dots \nu_n)$ . Furthermore, we call  $\hat{\delta}_{\nu,r}^{(j)}$  *lattice-ordered  $n$ -point  $r+1$ -ranged operator strings*.<sup>3</sup> The set of all lattice-ordered  $n$ -point  $r+1$ -ranged operator strings  $\Sigma = \{\hat{\delta}_{\nu,r}^{(j)}\}_{\nu,r}$  defines a regular language. A single local operator can be represented in the same way, resulting in a single lattice-ordered 1-point 1-ranged operator string.

An important insight is that the sum over every starting point  $j$  in Eq. (7.1) can be described by an FSM that can be represented by a *single-branched graph* with identity loops on the initial and final nodes, see Fig. 7.1. These identity loops<sup>4</sup> represent an undefined amount of identities (0 to  $L-r-1$ ). Because we demand that only operator strings with the correct number of operators are accepted, such an FSM accepts all operator strings that include the structure of  $\hat{\delta}_{\nu,r}^{(j)}$  on every site, but only a single occurrence per string.

<sup>3</sup>Lattice ordered: The operators are ordered by their physical location;  $n$ -point: Only on  $n$  sites operators different from the identity are applied. Note that they are not necessarily different operators;  $r+1$ -ranged: Maximal distance between two operators is  $r+1$ .

<sup>4</sup>In our daily discussions, we started to call them ‘Öhrchen’, which is the German word for ‘little ear’.



**Figure 7.1:** (Adopted from Fig. 3 in [PKM17]) General 2-point  $r + 1$ -ranged single-branched graph that represents the sum over every starting point  $j$   $\hat{O}_{\nu,r} = \sum_j \hat{o}_{\nu,r}^{(j)}$ .

In Sec. 7.2.1, we show how to add FSMs and in particular single-branched graphs. Applying this technique, we can build the outer sum of Eq. (7.1), hence every global operator can be represented by an FSM. Because FSMs built up this way become large very quickly, we introduce compression techniques in Sec. 7.3.

## How to get an MPO from an FSM | 7.1.1

To get from an FSM to an MPO, we need to change the representation again, i.e., we need to transfer a graph to its state-transition-table representation. In Fig. 7.2 this transfer is depicted for an exemplary FSM. As already mentioned, this transfer is possible but lossy, whereas the opposite direction is not possible due to the missing information. To fix this problem, we refine the definition of the initial state by demanding that all the previous transitions have to be identities from itself. Analogously, we refine the definition of the final state by demanding that all the subsequent transitions have to be identities to itself.

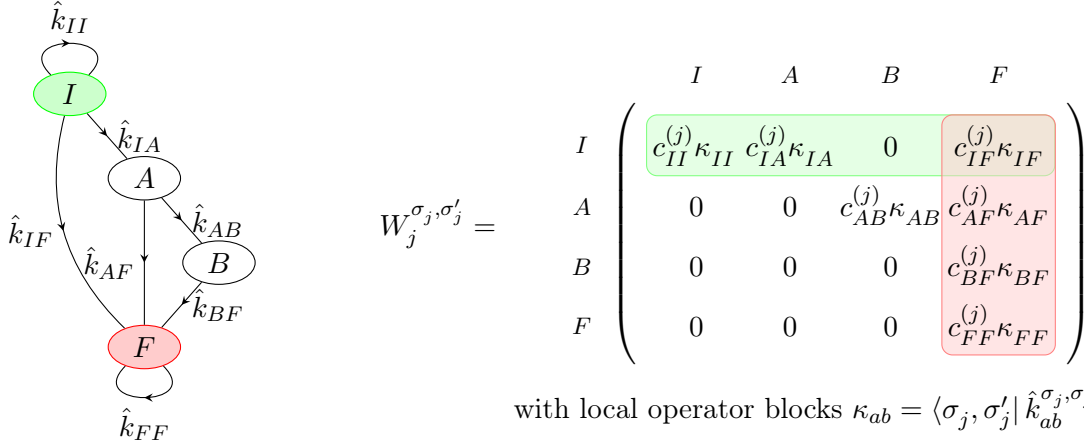
In order to transfer the FSM, we start with an empty table with  $n$  rows and  $n$  columns, where  $n$  is given by the number of nodes in the graph. Every row and every column is denoted by one node, see the right side of Fig. 7.2. Next, we transfer the edges by identifying the starting node, the transition, and the ending node and by including the transition into the table in the row of the starting node and the column of the ending node.

From this state-transition table, we can finally construct the MPO representation, i.e., the MPO site tensors  $W_j$  with dimension  $|\mathcal{L}| \times |\mathcal{L}|$  as depicted in Fig. 7.2. The corresponding boundary tensors are obtained by projecting out (a) the transition from the initial state into the bulk for  $j = 1$  and (b) the transitions from the bulk into the final state for  $j = L$ . We emphasize that the site-dependent coefficients  $c_{ab}^{(j)} \in \mathbb{C}$  are free parameters and therefore can be chosen independently for every site. Furthermore, we implemented these coefficients using a Turing-complete expression parser that is evaluated individually for every entry of every  $W_j$  tensor and if necessary before every time step<sup>5</sup>. This way, the coefficients  $c_{ab}^{(j)}$  can depend on several simulation parameters, e.g., current site  $j$ , current simulation time  $\tau$ , system size  $L$ , and the expectation value  $\langle \hat{H} \rangle$ , leading to  $c_{ab}^{(j)}(j, \tau, L, \langle \hat{H} \rangle, \dots)$ . This way, e.g., the Peierls substitution (see Eqs. (3.37) and (3.38)) is implemented completely within the FSM framework, see App. A.2 for the complete FSM.

Note that the whole complexity of index shifting, as required by quantum-number conservation or in an implementation of fermionic anticommutation rules, is moved from the tensor-network level to the FSM evaluation level. This fact is important to keep in mind, because FSMs resulting from arithmetic operations and subsequent compression conserve those properties from their input FSMs.

As MPOs are not unique representations for operators, the graph representation via FSMs is also not unique; for every operator  $\hat{H}$ , there is a set of corresponding FSMs  $\left\{ \Lambda(\hat{H}) \right\}_{\Lambda}$ . Therefore, we

<sup>5</sup>Note that in this context time step is an abbreviation for a ground-state-search sweep, an imaginary-time-evolution time step, or a real-time-evolution time step.



**Figure 7.2:** (Adopted from Fig. 4 in [PKM17]) Left: An FSM defined on states  $\mathcal{L} = \{I, A, B, F\}$  with transitions  $\hat{k}_{ab} \in \mathcal{K}$ . Note that we omit the upstairs indices within the graph representation for brevity. The initial and the final state is highlighted in green or red, respectively. Transitions between states are denoted by arrows between nodes with the corresponding symbol  $\hat{k}_{ab}$ . Right: Bulk MPO site-tensor block  $W_j^{\sigma_j, \sigma'_j}$ , obtained from the FSM on the left. The initial and the final site tensors are marked by a green or red background, respectively. The coefficients  $c_{ab}^{(j)}$  are site-dependent weight functions.

are free to choose one representation  $\Lambda(\hat{H})$ , which makes it easier to perform operator arithmetics and then switch to another representation  $\tilde{\Lambda}(\hat{H})$  to find the most compact MPO.

### Example: FSM for the Heisenberg Hamiltonian

### 7.1.2

Starting from the Heisenberg Hamiltonian (see Sec. 3.2), we describe here the steps to obtain a graph representation and finally an MPO.

The first important step is to write down the Hamiltonian in its full length,

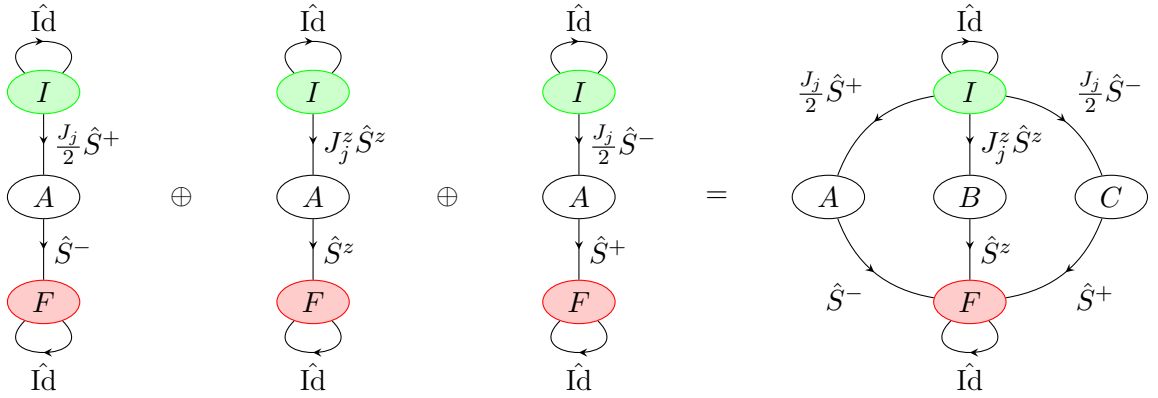
$$\begin{aligned}
 \hat{H} &= \sum_{j=1}^{L-1} \frac{J_j}{2} \hat{S}_j^+ \hat{S}_{j+1}^- + \frac{J_j}{2} \hat{S}_j^- \hat{S}_{j+1}^+ + J_j^z \hat{S}_j^z \hat{S}_{j+1}^z \\
 &= \sum_j \hat{\text{Id}}_1 \otimes \dots \otimes \hat{\text{Id}}_{j-1} \otimes \frac{J_j}{2} \hat{S}_j^+ \otimes \hat{S}_{j+1}^- \otimes \hat{\text{Id}}_{j+2} \otimes \dots \otimes \hat{\text{Id}}_L \\
 &\quad + \sum_j \hat{\text{Id}}_1 \otimes \dots \otimes \hat{\text{Id}}_{j-1} \otimes \frac{J_j}{2} \hat{S}_j^- \otimes \hat{S}_{j+1}^+ \otimes \hat{\text{Id}}_{j+2} \otimes \dots \otimes \hat{\text{Id}}_L \\
 &\quad + \sum_j \hat{\text{Id}}_1 \otimes \dots \otimes \hat{\text{Id}}_{j-1} \otimes J_j^z \hat{S}_j^z \otimes \hat{S}_{j+1}^z \otimes \hat{\text{Id}}_{j+2} \otimes \dots \otimes \hat{\text{Id}}_L.
 \end{aligned} \tag{7.3}$$

$$\tag{7.4}$$

Note that the amount of identities in front and after the operators can also be smaller than two. In Sec. 7.2.1, we show that it is possible to add FSMs; it is therefore sufficient to restrict this example to only one of the sums in Eq. (7.4), which will be represented by a single-branched graph. Without loss of generality, we consider the last term,

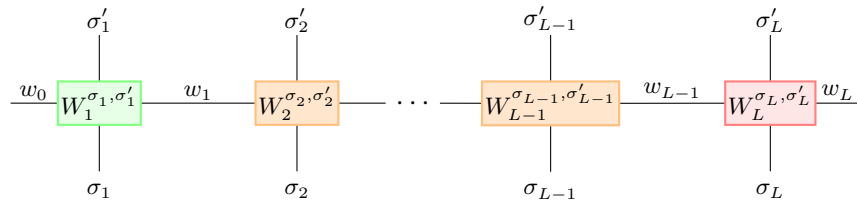
$$\sum_j \hat{\text{Id}}_1 \otimes \dots \otimes \hat{\text{Id}}_{j-1} \otimes J_j^z \hat{S}_j^z \otimes \hat{S}_{j+1}^z \otimes \hat{\text{Id}}_{j+2} \otimes \dots \otimes \hat{\text{Id}}_L, \tag{7.5}$$

for which we need to ask the question how an FSM would look like that accepts exactly all terms in this sum. The uninterrupted and undefined number of identities already suggests a graph with an initial node that has an identity loop onto itself. The only other option for the first term is the  $\hat{S}^z$  operator, which must be followed by another  $\hat{S}^z$  operator. Hence, an edge labeled with  $J_i^z \hat{S}^z$  to a node  $A$  and a sequential edge labeled with  $\hat{S}^z$  from node  $A$  to another node is necessary. From this node the only edge is again an uninterrupted and undefined number of identities that leads to another identity loop to itself. Hence no further edges are attached to this node; it is the final node  $F$ . In Fig. 7.3, all three sums of Eq. (7.4) are shown on the left side, whereas the sum of them is shown on the right side.



**Figure 7.3:** Graph representation  $\Lambda(\hat{H}_{\text{Heisenberg}})$  of the Heisenberg Hamiltonian Eq. (7.3). On the left-hand side, the three sums of Eq. (7.4) are represented individually. On the right-hand side, they are combined into a single graph.

Analogous to Fig. 7.2, we can now build the MPO site-tensor blocks from the state-transition table of the summed graph. Afterwards, we can directly read off the complete MPO representation as depicted in Fig. 7.4.



**Figure 7.4:** MPO tensor network with colored first and last site tensor, which can be directly read off from the state transition table of the corresponding FSM.

## Arithmetic Operations on FSMs | 7.2

In this section, we introduce arithmetic operations executed directly on graph representations of FSMs. We start with the conceptually easier and more often used addition and afterwards present the multiplication.

## Addition | 7.2.1

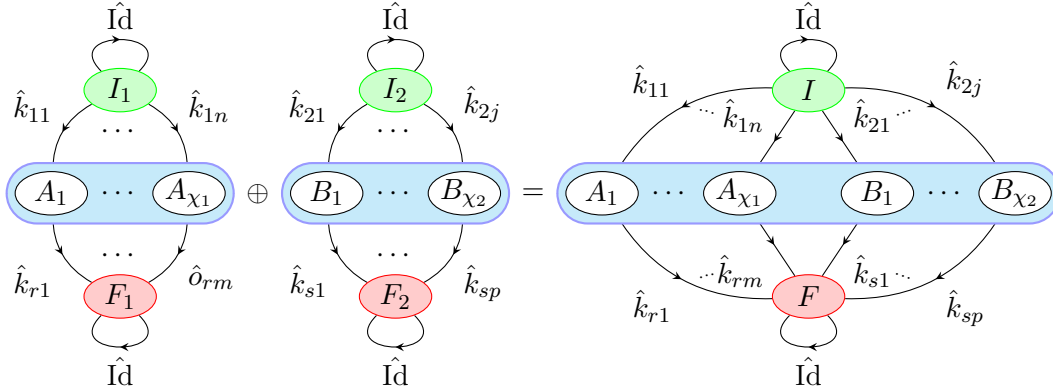
As described before, every global operator can be formulated as a sum over lattice-ordered  $n$ -point  $r + 1$ -ranged operator strings

$$\hat{O} = \sum_{\nu,r} \hat{O}_{\nu,r}. \quad (7.6)$$

A natural translation to the graph representation of  $\hat{O}$  can be obtained by introducing a commutative map  $\oplus$  between graph representations of sums of operators  $\hat{O}_1, \hat{O}_2$  via

$$\oplus : \quad \Lambda(\hat{O}_1 + \hat{O}_2) = \Lambda(\hat{O}_1) \oplus \Lambda(\hat{O}_2). \quad (7.7)$$

This is possible because of the intrinsic summation, which is due to the MPO construction over every operator string accepted by the FSM. The realization of  $\oplus$  in terms of graphs is obtained by taking the graph representations of the operators  $\hat{O}_1, \hat{O}_2$  and by then merging the initial and final states as depicted in Fig. 7.5. This means for example for the initial nodes that a new node  $I$  is created and all edges from  $I_1$  and  $I_2$  are moved to start from the new node  $I$ . Note that the identity loops are an exception, i.e., they are not moved but a single identity loop is added to the new node.



**Figure 7.5:** (Published as Fig. 5 in [PKM17]) Realization of the operator sum  $\hat{O}_1 + \hat{O}_2$  in terms of graph representations  $\Lambda(\hat{O}_1 + \hat{O}_2) = \Lambda(\hat{O}_1) \oplus \Lambda(\hat{O}_2)$ . Graph representations of operators  $\hat{O}_{1,2}$  are illustrated by transitions from the initial state into the graph's bulk ( $\hat{k}_{11} \dots \hat{k}_{1n}$  and  $\hat{k}_{21} \dots \hat{k}_{2j}$ ) and from the graph's bulk to the final state ( $\hat{k}_{r1} \dots \hat{k}_{rm}$  and  $\hat{k}_{s1} \dots \hat{k}_{sp}$ ). Blue boxes denote the bulk of the graph representations  $\Lambda(\hat{O}_{1,2})$  and  $\Lambda(\hat{O}_1 + \hat{O}_2)$ .

## Multiplication | 7.2.2

In order to present the multiplication of graphs, we define the notion of a *maximally branched* graph representation, which is given by the graph  $\Lambda_{\max}(\hat{O})$  satisfying the conditions: a) the initial node  $I$  is the only node with more than one child node, b) the final node  $F$  is the only node with more than one parent node.  $\Lambda_{\max}(\hat{O})$  satisfies the equation

$$\Lambda_{\max}(\hat{O}) = \bigoplus_{\nu,r} \Lambda(\hat{O}_{\nu,r}). \quad (7.8)$$

This representation has several advantages; most importantly for our discussion, we only need to consider two branches that do not branch off again.

Let us consider the product of two global operators  $\hat{O}_1, \hat{O}_2$  in terms of their maximally branched representations

$$\hat{O}_1 \cdot \hat{O}_2 = \left( \sum_{\nu^1, r^1} \hat{O}_{\nu^1, r^1} \right) \cdot \left( \sum_{\nu^2, r^2} \hat{O}_{\nu^2, r^2} \right) \quad (7.9)$$

and in particular a single summand that is the product of two lattice-ordered string operators<sup>6</sup>

$$\hat{O}_{\nu^1, r^1} \cdot \hat{O}_{\nu^2, r^2} = \left( \sum_{j_1} \hat{o}_{\nu^1, r^1}^{(j_1)} \right) \cdot \left( \sum_{j_2} \hat{o}_{\nu^2, r^2}^{(j_2)} \right). \quad (7.10)$$

Note that we have introduced superscripts  $\nu^{1,2}, r^{1,2}$  to distinguish the index sets of the global operator.<sup>7</sup> A representation of this product in terms of a FSM and therefore its graph representation requires a reformulation in terms of lattice-ordered string operators. Although the product of two lattice-ordered operators  $\hat{O}_{r^1, \nu^1} \cdot \hat{O}_{\nu^2, r^2}$  is no longer lattice-ordered, a careful inspection of the terms violating the lattice order reveals how to build a graph representation generating the product  $\hat{O}_{r^1, \nu^1} \cdot \hat{O}_{r^2, \nu^2}$ . It turns out to be useful to define a non-commutative  $\wedge$  product, which maps two single-branched graphs to a single-branched graph via

$$\Lambda \left( \hat{O}_{\nu^1, r^1} \right) \wedge \Lambda \left( \hat{O}_{\nu^2, r^2} \right) = \Lambda \left( \sum_{j_1 + r^1 < j_2} \hat{o}_{\nu^1, r^1}^{(j_1)} \hat{o}_{\nu^2, r^2}^{(j_2)} \right). \quad (7.11)$$

A graph realization of  $\wedge$  is obtained by identifying the final state of  $\Lambda(\hat{O}_{\nu^1, r^1})$  with the initial state of  $\Lambda(\hat{O}_{\nu^2, r^2})$ , see Fig. 7.6 for an exemplary evaluation.

Next, we introduce the algorithm to construct the graph representation of the operator product  $\hat{\hat{O}} = \Lambda(\hat{O}_{\nu^1, r^1}) \cdot \Lambda(\hat{O}_{\nu^2, r^2})$  in terms of generating a new graph. This procedure can then be applied to all branches to construct a new graph for  $\hat{O} = \hat{O}_1 \cdot \hat{O}_2$ .

Let  $\mathcal{I}(\mathcal{K}, b = 1, n)$  be the set of all single-branch graphs representing lattice-ordered  $n$ -point operators. Then, with a proper  $\beta \geq b$ , we look for a realization of the non-commutative map

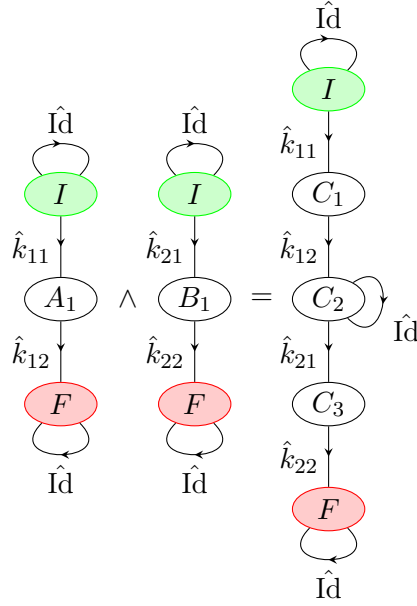
$$\begin{aligned} \otimes : \mathcal{I}(\mathcal{K}, 1, n) \times \mathcal{I}(\mathcal{K}', 1, m) &\longrightarrow \mathcal{I}(\mathcal{K} \times \mathcal{K}', 2 + \beta, n + m) \\ \Lambda(\hat{O}_{\nu^1, r^1}) \otimes \Lambda(\hat{O}_{\nu^2, r^2}) &\longmapsto \Lambda(\hat{O}_{\nu^1, r^1} \cdot \hat{O}_{\nu^2, r^2}), \end{aligned} \quad (7.12)$$

with  $\mathcal{K} \times \mathcal{K}'$  denoting the symmetrized on-site tensor-product set of  $\mathcal{K}$  and  $\mathcal{K}'$ . For this purpose, we apply the definition of  $\oplus$  in Eq. (7.7) and search for a graph representation of  $\Lambda(\hat{O}_{\nu^1, r^1} \cdot \hat{O}_{\nu^2, r^2})$  by ordering the appearing types of terms in the resulting sum of the operator product according to the lattice treated. We construct single-branch graph representations for all different types of generated lattice-ordered operator strings, which we denote by  $\hat{\gamma}$ . Then, a graph representation is obtained by summing up all these strings

$$\Lambda(\hat{O}_{\nu^1, r^1} \cdot \hat{O}_{\nu^2, r^2}) = \bigoplus_{\hat{\gamma}} \Lambda(\hat{\gamma}). \quad (7.13)$$

<sup>6</sup>We use the notation introduced in Eq. (7.1).

<sup>7</sup>Expanding the indices results in  $\hat{O}_j = \sum_{\nu_1^j \dots \nu_n^j, r^j} \hat{O}_{\nu_1^j \dots \nu_n^j, r^j}$ .



**Figure 7.6:** (Adopted from Fig. 7a in [PKM17]) Graph representation of  $\Lambda \left( \hat{O}_{\nu^1, r^1} \right) \wedge \Lambda \left( \hat{O}_{\nu^2, r^2} \right)$  for 2-point 2-ranged interacting operator strings.

From now on, we focus on the special case of 2-point operators, i.e., operators of the form

$$\hat{O}_{\nu^n, r^n} = \sum_{j_n} \hat{o}_{\nu_1^n \nu_2^n, r^n}^{(j_n)} = \sum_{j_n} f_{\nu_1^n \nu_2^n}^{r^n} \hat{k}_{\nu_1^n}^{(j_n)} \hat{k}_{\nu_2^n}^{(j_n+r^n)}. \quad (7.14)$$

Nevertheless, the generalization to arbitrary  $n$ -point string operators is straightforward: Simply replace identities with additional local operators. Decomposing the operator product, we find

$$\begin{aligned} \hat{O}_{r^1, \nu^1} \cdot \hat{O}_{r^2, \nu^2} &= \sum_{j_1, j_2} \hat{o}_{\nu_1^1 \nu_2^1, r^1}^{(j_1)} \hat{o}_{\nu_1^2 \nu_2^2, r^2}^{(j_2)} \quad (7.15) \\ &= \underbrace{\sum_{j_1} \sum_{j_2 > j_1 + r^1} \hat{o}_{\nu_1^1 \nu_2^1, r^1}^{(j_1)} \hat{o}_{\nu_1^2 \nu_2^2, r^2}^{(j_2)}}_{\hat{O}_A = \text{---} \times \text{---} \text{---} \text{---} \times \text{---}} + \underbrace{\sum_{j_1} \sum_{j_2 < j_1 - r^2} \hat{o}_{\nu_1^1 \nu_2^1, r^1}^{(j_1)} \hat{o}_{\nu_1^2 \nu_2^2, r^2}^{(j_2)}}_{\hat{O}_C = \text{---} \times \text{---} \text{---} \text{---} \times \text{---}} \\ &\quad + \sum_{j_1} \sum_{j_2 = i_1 - r^2}^{j_1 + r^1} \underbrace{\hat{o}_{\nu_1^1 \nu_2^1, r^1}^{(j_1)} \hat{o}_{\nu_1^2 \nu_2^2, r^2}^{(j_2)}}_{\hat{o}_B(j_1, j_2) \equiv \text{Overlaps}}. \quad (7.16) \end{aligned}$$

Note that in Eq. (7.16), and in the following, the red chain represents  $\hat{o}_{\nu_1^1 \nu_2^1, r^1}^{(j_1)}$ , whereas the blue chain represents  $\hat{o}_{\nu_1^2 \nu_2^2, r^2}^{(j_2)}$ . In order to take care of different commutation relations of local operators acting on different sites, we include a factor  $\epsilon_{\nu_1 \nu_2}$  with which

$$\left[ \hat{k}_{\nu_1}^{(j_1)}, \hat{k}_{\nu_2}^{(j_2)} \right]_{\epsilon_{\nu_1 \nu_2}} = \hat{k}_{\nu_1}^{(j_1)} \hat{k}_{\nu_2}^{(j_2)} - \epsilon_{\nu_1 \nu_2} \hat{k}_{\nu_2}^{(j_2)} \hat{k}_{\nu_1}^{(j_1)} = 0 \quad (7.17)$$

is fulfilled. Next, we can decompose the product into lattice-ordered sums by commuting local operators acting on strictly unequal sites.



The first lattice-ordered contribution is given via  $\hat{O}_A$ . The corresponding diagram is a single-branch graph obtained by identifying the final state of the graph  $\Lambda(\hat{O}_{\nu^1, r^1})$  with the initial state of the graph  $\Lambda(\hat{O}_{\nu^2, r^2})$  by introducing an intermediate state  $E$  (see Fig. 7.7a). We now make use of the wedge product  $\wedge$  for single-branched graphs as defined above to rewrite  $\hat{O}_A$  in short as

$$\Lambda(\hat{O}_A) = \Lambda(\hat{O}_{r^1, \nu^1}) \wedge \Lambda(\hat{O}_{r^2, \nu^2}). \quad (7.18)$$

Swapping the operators, we obtain another lattice-ordered sum by commuting all local operator contributions, so that the corresponding graph picks up two factors,  $\epsilon_{\nu_1^1 \nu_1^2} \epsilon_{\nu_2^1 \nu_2^2}$  for commuting  $\hat{k}_{\nu_1^1}^{(j_2)}$  and  $\epsilon_{\nu_1^1 \nu_2^2} \epsilon_{\nu_2^1 \nu_2^2}$  for commuting  $\hat{k}_{\nu_2^2}^{(j_2+r^2)}$  with  $\hat{k}_{\nu_1^1}^{(j_1)} \hat{k}_{\nu_2^1}^{(j_1+r^1)}$  (see Fig. 7.7b). Again, the corresponding graph can be expressed via a wedge product,

$$\Lambda(\hat{O}_C) = \Lambda(\text{sgn}(\hat{o}_{\nu_1^1 \nu_2^1, r^1}, \hat{o}_{\nu_1^2 \nu_2^2, r^2}) \hat{O}_{r^2, \nu^2}) \wedge \Lambda(\hat{O}_{r^1, \nu^1}), \quad (7.19)$$

where we have introduced  $\text{sgn}(\hat{o}_{\nu_1^1 \nu_2^1, r^1}, \hat{o}_{\nu_1^2 \nu_2^2, r^2}) \equiv \epsilon_{\nu_1^1 \nu_1^2} \epsilon_{\nu_2^1 \nu_2^2} \epsilon_{\nu_1^1 \nu_2^2} \epsilon_{\nu_2^1 \nu_2^2}$  for brevity.

The remaining sums over  $\hat{o}_B(j_1, j_2)$  correspond to overlapping interaction terms, i.e., all those lattice indices  $j_1, j_2$  that fulfill

$$\{j_1, \dots, j_1 + r^1\} \cap \{j_2, \dots, j_2 + r^2\} \neq \emptyset. \quad (7.20)$$

In order to generate all these terms using one algorithm, we write the 2-point operator expressions graphically by representing a local operator  $\hat{k}_\nu$  with a cross (“×”) and the intermediate vacant operator sites with a circle (“○”), e.g.,

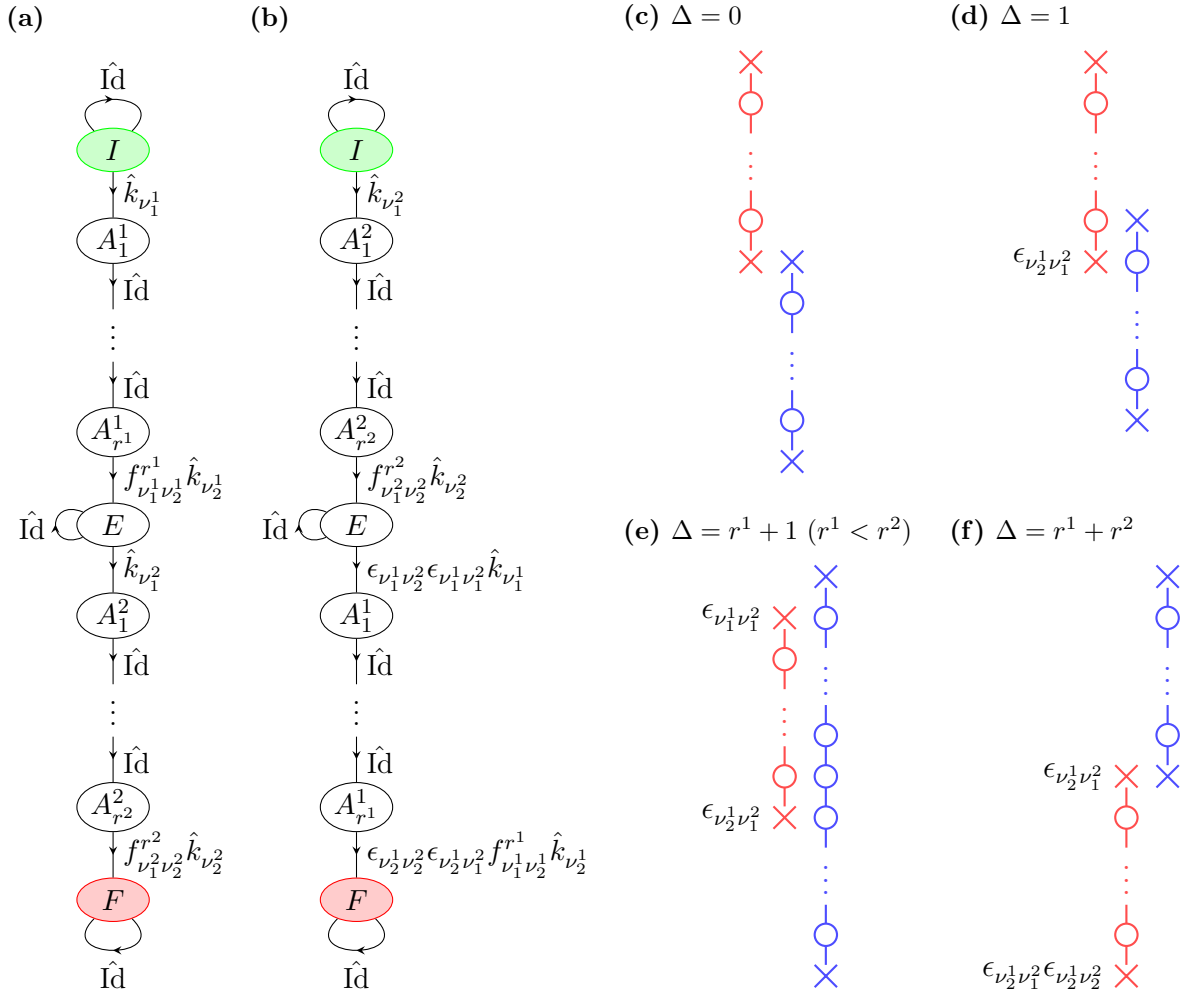
$$\hat{k}_{\nu_1^1}^{(j)} \hat{k}_{\nu_2^2}^{(j+r)} \rightarrow \times \text{---} \underbrace{\text{---} \text{---} \text{---} \text{---} \text{---}}_{r-1 \text{ times}} \text{---} \times. \quad (7.21)$$

Employing this condensed notation, all lattice-ordered combinations of local operator strings can be generated by placing the graphical representations of  $\hat{o}_{\nu_1^1 \nu_2^1, r^1}^{(j_1)}$  and  $\hat{o}_{\nu_1^2 \nu_2^2, r^2}^{(j_2)}$  next to each other by aligning the last operator of  $\hat{o}_{\nu_1^1 \nu_2^1, r^1}^{(j_1)}$  with the first operator of  $\hat{o}_{\nu_1^2 \nu_2^2, r^2}^{(j_2)}$  (see Fig. 7.7c). Subsequently, the right string is shifted upward until the initial operator of  $\hat{o}_{\nu_1^1 \nu_2^1, r^1}^{(j_1)}$  is aligned with the final operator of  $\hat{o}_{\nu_1^2 \nu_2^2, r^2}^{(j_2)}$  (see Fig. 7.7f). While the right string’s position is shifted by one step, the local operators  $\hat{k}_{\nu_{j_1}^1}$  of the left operator string pick up a sign factor whenever they pass a local operator  $\hat{k}_{\nu_{j_2}^2}$  in the right string, which is denoted by adding a factor  $\epsilon_{\nu_{j_1}^1 \nu_{j_2}^2}$  to the left condensed representation (see Figs. 7.7d and 7.7e).

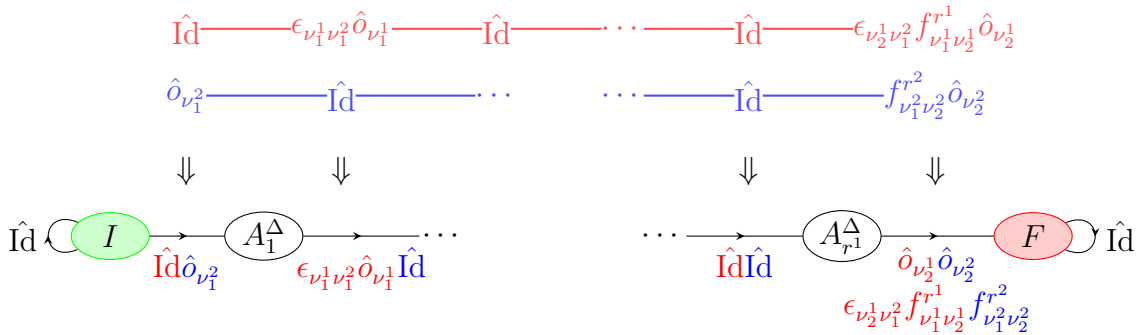
For each such step  $\Delta$ , with  $0 \leq \Delta \leq r^1 + r^2$ , this algorithm generates a string  $\hat{\gamma}_\Delta$  of local operators by merging the shifted right string into the left and resubstituting the original operators:  $\times \rightarrow \hat{k}_\nu$ . Missing sites are replaced with identities, whereas two local operators per site are contracted into a new local site operator  $\hat{u}_{\nu_{12}} = \hat{k}_{\nu_1^1} \hat{k}_{\nu_2^2}$ . Note that, in the latter case, the local operators are ordered in a way that avoids evaluation of on-site commutators  $\left[ \hat{k}_{\nu_{j_1}^1}^{(l)}, \hat{k}_{\nu_{j_2}^2}^{(l)} \right]$ . Finally, each string  $\hat{\gamma}_\Delta$  is converted into a single-branch graph by introducing a set of states  $\left\{ A_j^\Delta \right\}_j$  with the transitions between the  $A_j^\Delta$ ’s properly chosen from the corresponding new local site operators  $\left\{ \hat{u}_{\nu_{12}} \right\}$  (see Fig. 7.8 for an example).

The algorithm above yields the following graph representation

$$\Lambda(\hat{O}_{r^1, \nu^1} \cdot \hat{O}_{r^2, \nu^2}) = \left[ \Lambda(\hat{O}_{r^1, \nu^1}) \wedge_S \Lambda(\hat{O}_{r^2, \nu^2}) \right] \oplus \bigoplus_{\Delta=0}^{r^1+r^2} \hat{\gamma}_\Delta \equiv \Lambda(\hat{O}_{r^1, \nu^1}) \otimes \Lambda(\hat{O}_{r^2, \nu^2}), \quad (7.22)$$



**Figure 7.7:** (Adopted from Fig. 3 in [PKM17]) (a) Tree-diagram representation of  $\hat{O}_A$ . (b) Tree-diagram representation of  $\hat{O}_C$ . (c-f) The figures depict stages of the algorithm for determining all summands  $\hat{o}_B(j_1, j_2)$  by shifting the operator string  $\hat{o}_{\nu_1^2 \nu_2^2, r^2}^{(j_2)}$  by  $\Delta$  steps up until its terminating local operator is aligned with the initial local operator of the other operator string  $\hat{o}_{\nu_1^1 \nu_2^1, r^1}^{(j_1)}$  at step  $\Delta = r^1 + r^2$ .



**Figure 7.8:** (Adopted from Fig. 12 in [PKM17]) Resubstituted local operators at step  $\Delta = r^1 + 1$  with  $r^2 = r^1 + 1$ . The resulting operator string is  $\hat{o}_{\nu_1^2} \otimes \epsilon_{\nu_1^1 \nu_1^2} \hat{o}_{\nu_1^1} \otimes \hat{\text{Id}} \otimes \dots \otimes \hat{\text{Id}} \otimes \epsilon_{\nu_2^1 \nu_1^2} f_{\nu_1^1 \nu_2^1}^{r^1} \hat{o}_{\nu_2^1} f_{\nu_1^2 \nu_2^2}^{r^2} \hat{o}_{\nu_2^2}$ , which is represented by the graph at the bottom of this figure.

with the symmetrized wedge product

$$\begin{aligned} \Lambda(\hat{O}_{r^1, \nu^1}) \wedge_S \Lambda(\hat{O}_{r^2, \nu^2}) &\equiv \left[ \Lambda(\hat{O}_{r^1, \nu^1}) \wedge \Lambda(\hat{O}_{r^2, \nu^2}) \right] \\ &\oplus \left[ \Lambda(\text{sgn}(\hat{o}_{\nu^1, r^1}, \hat{o}_{\nu^2, r^2}) \hat{O}_{r^2, \nu^2}) \wedge \Lambda(\hat{O}_{r^1, \nu^1}) \right]. \end{aligned} \quad (7.23)$$

Employing linearity of the graph representation for addition, the product of two general operators can then be formulated via

$$\begin{aligned} \Lambda(\hat{O}_1 \cdot \hat{O}_2) &= \Lambda\left(\sum_{r^1, \nu^1} \sum_{r^2, \nu^2} \hat{O}_{r^1, \nu^1} \cdot \hat{O}_{r^2, \nu^2}\right) \\ &= \bigoplus_{\nu^1, r^1} \bigoplus_{\nu^2, r^2} \left[ \Lambda(\hat{O}_{r^1, \nu^1}) \otimes \Lambda(\hat{O}_{r^2, \nu^2}) \right]. \end{aligned} \quad (7.24)$$

Despite the compact form, we emphasize that Eq. (7.22) and Eq. (7.24) describe a graph representation that is maximally expanded, so that there is no branching below the initial node, and the bond dimension of the generated MPO is very large. However, the size of the graph can be reduced very efficiently by compressing it into a more compact form.

## Compressing MPOs in Terms of FSMs 7.3

We have already used the fact that FSMs and MPOs are not unique, and for the arithmetics we introduced the maximally branched representation. On the other hand, MPOs with a small bond dimension  $w$  are desirable in order to decrease the necessary matrix-product state (MPS) truncation after the application of the MPO. Note that the bond dimension of an MPO is equal to the number of nodes of an FSM by which it is created. We are therefore interested in a lossless<sup>8</sup> way to obtain a representation with as few nodes as possible, especially for FSMs that result of FSM arithmetics.

The topic of obtaining the optimal MPO representation of a general operator is also discussed in [HMS17] in terms of MPO arithmetics. Here, we present similar compression schemes based on graph manipulations and show the connection to the schemes in [HMS17].

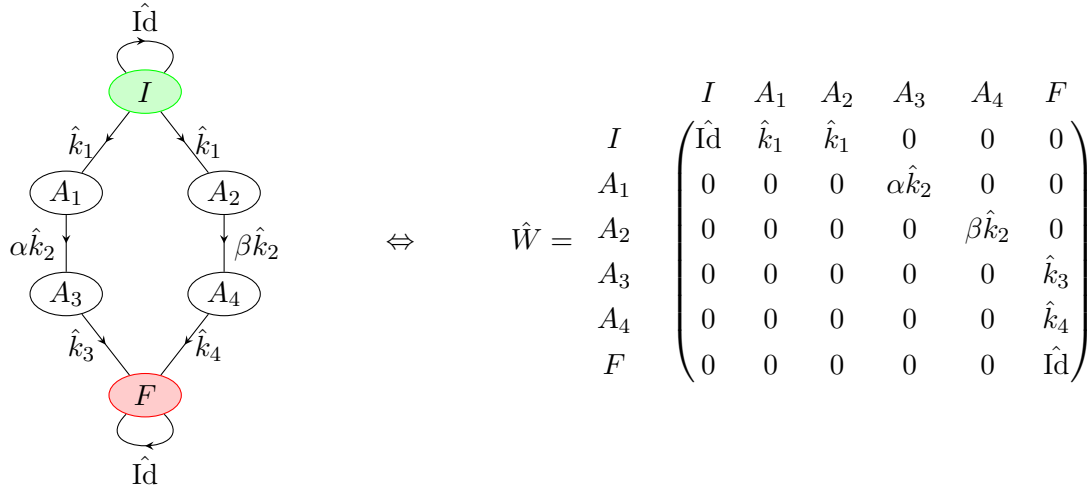
Note that not all compressions presented here are used in our current implementation, because sums of different operators on a single edge are not supported, yet. We present them, nevertheless, in this general way in order to show possible enhancements and as a guideline to the manual construction of FSMs.

### Deparallelization

The idea behind the deparallelization [McC07] is to combine columns in the MPO site tensor, or analogously in the state-transition table, which only differ by a constant factor in every element and store these factors in a transfer matrix  $T$ , which is afterwards multiplied onto the next site. The numerical deparallelization acts purely on the MPO site tensors and is therefore naturally capable of handling site-dependent parameters. In the case of the symbolic deparallelization, which is applied on FSMs, the site-dependent parameters are only handled correctly if they are not moved away from their corresponding operator or the distance is unique and stored within the parameters.

<sup>8</sup>In this case, it is important to obtain a lossless compression, because it is possible that the resulting MPO will be applied very often, which can lead to an accumulating error.

The procedure and the differences between the numeric and the symbolic realization can best be described by comparing graphs and state-transition tables of a concrete example. Note that we consider the operator-valued state-transition table as direct representation of the MPO site tensor and assume site-independent parameters for this example in order to be able to directly apply the transfer matrix from the left, leading always to square matrices. Consider the uncompressed FSM in its maximally branched form with its graph representation and its state-transition table given in Fig. 7.9.



**Figure 7.9:** (Adopted from Fig. 7 in [PKM17]) Initial maximally-branched example FSM for the subsequent compression via deparallelization. (Left) the graph representation, (right) the state-transition table.

The columns  $A_1$  and  $A_2$  are equal and hence can be merged. Simultaneously, the edges from node  $I$  to node  $A_1$  and to node  $A_2$  are identical and therefore the nodes  $A_1$  and  $A_2$  can be merged. The resulting FSM is given in Fig. 7.10.

At this point, the columns  $A_3$  and  $A_4$  differ only by a factor  $\beta/\alpha$  and can be compressed by the numerical deparallelization method. The transfer matrix includes the two parameters  $\alpha$  and  $\beta$  on the same row, resulting in a sum in the MPO site tensor. In order to obtain the same compression within the FSM, we first need to move the parameters to their corresponding next edge ( $\alpha \hat{k}_2 \otimes \hat{k}_3 \rightarrow \hat{k}_2 \otimes \alpha^{[-1]} \hat{k}_3$  and  $\beta \hat{k}_2 \otimes \hat{k}_4 \rightarrow \hat{k}_2 \otimes \beta^{[-1]} \hat{k}_4$ )<sup>9</sup>. Afterwards, the two  $\hat{k}_2$  edges can be joined, leading to a sum on the last edge as depicted in Fig. 7.11.

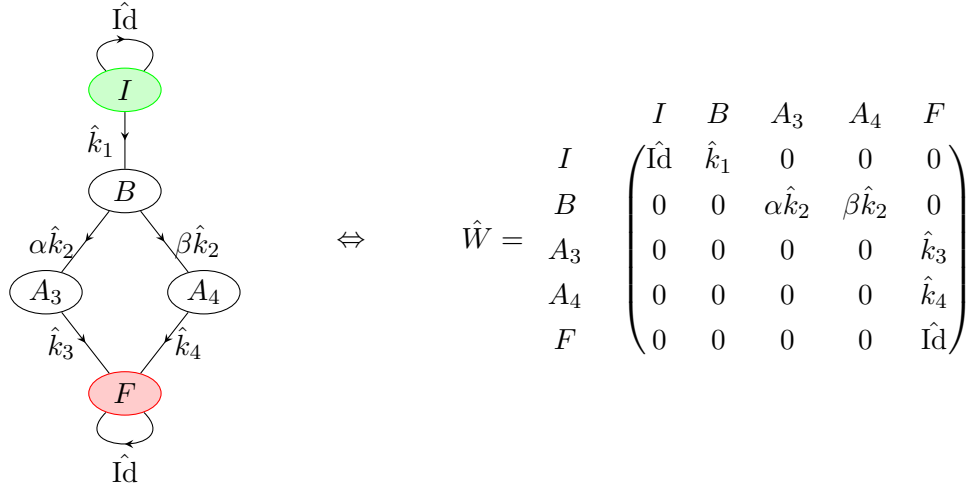
Note that the symbolic deparallelization is fully automatized and lossless. A row-wise deparallelization corresponds with merging equal edges starting from the final node and can be applied subsequently.

## Delinearization

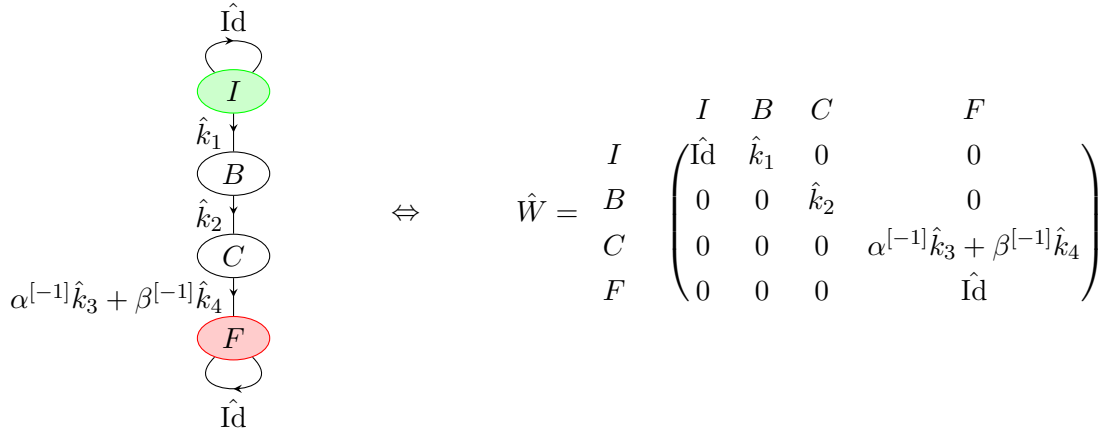
The delinearization exploits linear dependencies between columns in the reshaped MPO site tensors. As this method heavily depends on sums of operators, we present it here only for completeness and as an impulse for further development.

The idea is, again, best illustrated by an example. Therefore, consider the MPO represented by the FSM in Fig. 7.12.

<sup>9</sup>Here we introduce an offset in superscripted squared brackets that indicates to which site the parameter belongs. This becomes important when the parameters are site-dependent.

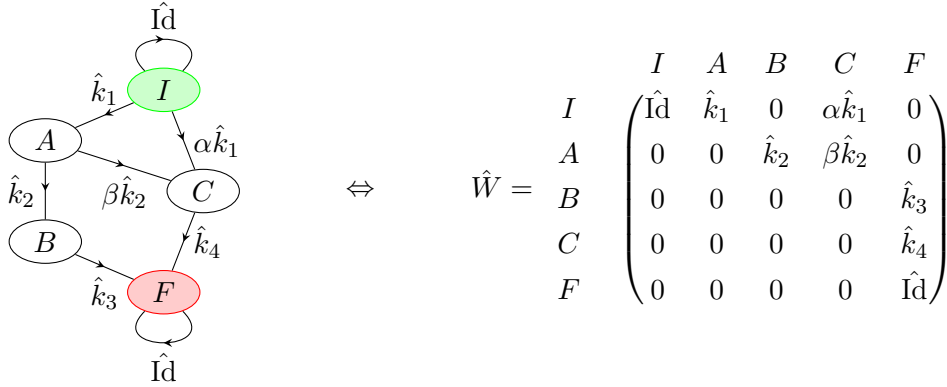


**Figure 7.10:** (Adopted from Fig. 7 in [PKM17]) Intermediate state of the compression of the example FSM. Again, (left) the graph representation, (right) the state-transition table. The identical edges from node  $I$  to  $A_1$  and to  $A_2$  are merged into one edge to the new node  $B$ . This graphical compression is identical to a first step of the deparallelization, which also would return the compressed MPO site tensor after applying the transfer matrix.

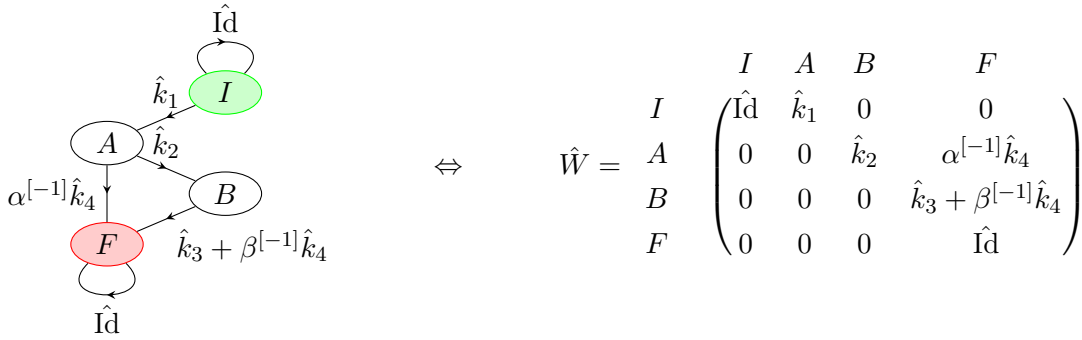


**Figure 7.11:** Fully compressed FSM. Again, (left) the graph representation, (right) the state-transition table.

It is not obvious how to compress this FSM from the graph representation, but it can be read off the state-transition table that column  $C$  is the sum of columns  $A$  and  $B$  with prefactors  $\alpha$  and  $\beta$ . In order to utilize this insight, it is necessary to find corresponding prefactors for the transition from  $A$  to  $F$  and from  $B$  to  $F$ , so that discarding the row and the column denoted with  $C$  is compensated. In [HMS17] this is done numerically via an algorithm that involves a QR decomposition of a matrix that includes all columns that can be used to construct the column at hand. In this context, a column can be used to construct another column if it does not have a finite element in a row in which the column at hand has a zero. This condition is employed in order to prevent errors due to numerical precision that could arise by subtracting nearly equal values. An example for the resulting compressed FSM is shown in Fig. 7.13.



**Figure 7.12:** Initial example FSM that is not further compressible via deparallelization for the subsequent compression via delinearization. (Left) the graph representation, (right) the state-transition table. The linear dependence of column  $C$  from the columns  $A$  and  $B$  can be read off directly from the state-transition table.



**Figure 7.13:** Fully compressed FSM. (Left) the graph representation, (right) the state-transition table.

## Precision of the FSM Operations | 7.4

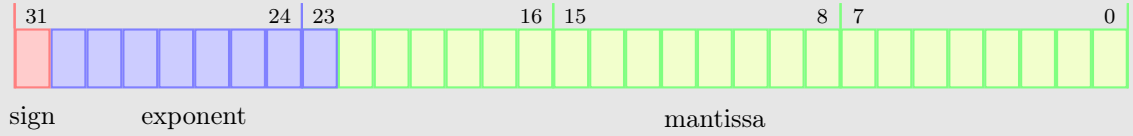
An illustrative example demonstrating the advantages of the graph arithmetic and the symbolic FSM compression is to construct two different expressions for the variance of the Hamiltonian  $\hat{H}$ , where the goal is to avoid catastrophic cancellation as best as possible while keeping calculations as cheap as possible. The variance can be used as a control parameter in numerical simulations to test whether a state  $|\psi\rangle$  is close to an eigenstate of  $\hat{H}$ . This is of particular interest in a high precision ground-state search. A naïve evaluation is obtained by directly calculating the expectation values in

$$\text{var}(\hat{H}) = \langle \psi | \hat{H}^2 | \psi \rangle - \langle \psi | \hat{H} | \psi \rangle^2 \geq 0, \quad (7.25)$$

which eventually vanishes if  $|\psi\rangle$  is an exact eigenstate, and the calculations are performed exactly. However, numerics is done with finite-precision floating-point arithmetic [IEE08]. This inevitably leads to catastrophic cancellation if large numbers need to be subtracted from each other, as it is the case in Eq. (7.25), because both terms scale as  $L^2$ . In order to apprehend the ramifications of this, we need to take a closer look at how finite-precision arithmetic actually works.

### Floating-point arithmetic

A floating-point value  $x = s \cdot m \cdot 2^e$  is stored in 3 parts. The first part that consists of one bit contains the sign  $s$  of the value. The second part is the exponent  $e$ , and the third part is the mantissa  $m$ . This structure is shown in Fig. 7.14 for a single-precision number.



**Figure 7.14:** Graphical representation of the bit alignment of a single-precision floating-point value  $x$ . It consists of a sign  $s$ , an exponent  $e$ , and a mantissa  $m$ , to represent  $x = s \cdot m \cdot 2^e$ .

The IEEE<sup>a</sup> standard defines how to store normal numbers, which leads to a hidden 1 in front of the mantissa.<sup>b</sup> Note that the most common floating-point types are single-precision (32bit,  $\sim 7$ -8 decimals) and double-precision (64bit,  $\sim 15$ -16 decimals); nevertheless other formats are also specified.

Additionally to the storing specifications, the standard also requires arithmetic operations (addition, subtraction, multiplication, division, and square-root) with exactly rounded results. This means that the result needs to equal the rounded result of a non-finite-precision operation. Nevertheless, errors will grow with every subsequent operation.

<sup>a</sup>Institute of Electrical and Electronics Engineers.

<sup>b</sup>Subnormal numbers are also defined, but in practice rarely encountered.

Every (exact) number  $z$  is numerically represented by a floating-point number  $x$ , which is usually exact up to an order of magnitude  $p$  so that we can mimic limited numerical precision by replacing  $z \rightarrow x(1 + \epsilon \times 10^{-p})$  with a random variable  $\epsilon \in (-1, 1)$ . Let  $z_{1,2}$  be two (exact) numbers and  $x_{1,2}$  their representations with equal numerical precision  $p$  and  $0 < z_1 - z_2 = 10^{-\delta}$  the exact difference. In finite-precision arithmetics, we then obtain

$$z_1 - z_2 \approx x_1(1 + \epsilon_1 \times 10^{-p}) - x_2(1 + \epsilon_2 \times 10^{-p}) \quad (7.26)$$

$$= 10^{-\delta} + \epsilon_1 \times 10^{-(p+\delta)} + x_2(\epsilon_2 - \epsilon_1) \times 10^{-p}. \quad (7.27)$$

For  $\delta > 0$ , the second term cannot be represented due to finite precision. In our case, the values for  $z_{1,2}$  are obtained from expectation values of operators acting on the full system. Hence, if we estimate them by their leading-order contribution  $z_{1,2} \sim L^q$  with magnitude  $q$ , with  $\gamma \equiv \log_{10}(L)$ , and  $\epsilon \equiv \epsilon_2 - \epsilon_1$ , we obtain

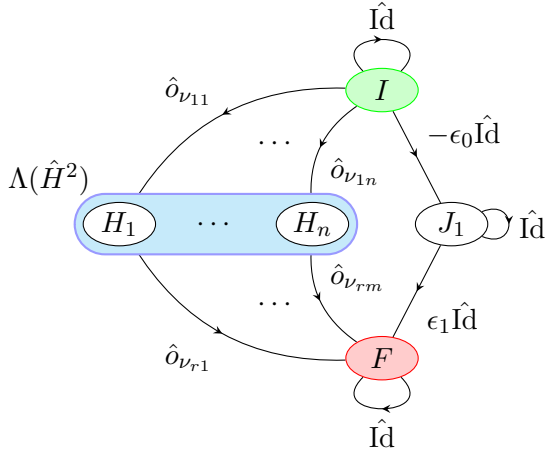
$$z_1 - z_2 \approx 10^{-\delta} + \epsilon \times 10^{-(p-\gamma \cdot q)}, \quad (7.28)$$

where  $\epsilon$  is a random variable of order  $\pm 1$ . It follows that we require  $\delta < p - \gamma \cdot q$  in order to calculate the difference reliably. In case of the variance, i.e.,  $q = 2$ , with double-precision arithmetics,  $p = p_{\text{num}} = 16$ , the naïve evaluation,  $\delta < 16 - 2\gamma$ , yields an upper bound of a maximally possible precision of  $10^{-12}$  for a lattice with  $L = 100$  sites. However, the numerical precision in general is not only bound by the exact numerical precision  $p_{\text{num}}$ , but it is also subject to increasing round-off errors due to preceding calculations. Thus, in actual calculations, we have an effective precision  $p(L, \chi, \dots) \leq p_{\text{num}}$  that depends on simulation parameters. Returning to

Eq. (7.25), we realize that the calculated variance is not only bounded, but may even become negative (as  $\epsilon$  in Eq. (7.28) can be negative).

The problem can be addressed by a minimization of  $\gamma \cdot q$ , for instance by constructing an MPO representation for the operator  $\hat{V} = (\hat{H} - \langle \psi | \hat{H} | \psi \rangle)^2$  (e.g., by distributing the expectation value  $\langle \psi | \hat{H} | \psi \rangle$  over the lattice sites). Unfortunately, this comes at the cost of having constructed an operator that is dependent on an expectation value. Hence, its numeric MPO representation has to be rebuilt for every new state  $|\psi\rangle$ , which requires an efficient way of obtaining MPO representations of the square of an operator.

(a)



(b)

$$\hat{W}_j^{\sigma_j, \sigma'_j} = \hat{H}^2 \begin{pmatrix} I & \hat{H}^2 & J_1 & F \\ I & \hat{I}\hat{d} & -\epsilon_0 & 0 \\ \hat{H}^2 & 0 & 0 & \hat{I}\hat{d} \\ J_1 & & \hat{I}\hat{d} & \epsilon_1 \\ F & & & \hat{I}\hat{d} \end{pmatrix}$$

**Figure 7.15:** (Adopted from Fig. 8 in [PKM17]) (a) Graph representation  $\Lambda_1(\text{var}(\hat{H}))$ , compare Eq. (7.29). The representation of  $\hat{H}^2$  and of  $\langle \hat{H} \rangle^2$  are considered independently: For the former, we depict transitions from the initial state to the subgraph  $\Lambda(\hat{H}^2)$  and then to the final state indicated by the operators  $\hat{o}_j$  ( $\Lambda(\hat{H}^2)$  represents the graph of  $\hat{H}^2$ ). For the latter, we depict transitions from the initial state to state  $J_1$  of the FSM, and then to the final state via identity operators. Those identities carry the weights  $-\epsilon_0$  and  $\epsilon_1$ , respectively, which, after performing the multiplication, yields  $-\langle \hat{H} \rangle^2$ , see Eq. (7.29). (b) MPO site tensor obtained from the graph representation  $\Lambda_1(\text{var}(\hat{H}))$ . Matrix entries with blue background denote the entries obtained from the site-tensor representation of  $\Lambda(\hat{H}^2)$ .

To analyze both problems, we investigate two MPO representations  $\Lambda_1(\text{var}(\hat{H}))$  (see Fig. 7.15a for the corresponding graph) and  $\Lambda_2(\text{var}(\hat{H}))$ , which are obtained from the graph representations

$$\begin{aligned} \Lambda_1(\text{var}(\hat{H})) &= \Lambda(\hat{H}^2) \oplus \Lambda(-E^2 \hat{I}\hat{d}) \\ &= \Lambda(\hat{H}^2) \oplus \left[ \Lambda(-\epsilon_0 \hat{I}\hat{d}) \wedge \Lambda(\epsilon_1 \hat{I}\hat{d}) \right], \quad \text{with } \epsilon_i = \langle \hat{H} \rangle \frac{\sqrt{2}}{L-i} \end{aligned} \quad (7.29)$$

$$\begin{aligned} \Lambda_2(\text{var}(\hat{H})) &= \Lambda(\hat{H} - \langle \hat{H} \rangle) \otimes \Lambda(\hat{H} - \langle \hat{H} \rangle) \\ &= \left[ \Lambda(\hat{H}) \oplus \Lambda(\epsilon \hat{I}\hat{d}) \right]^{\otimes 2}, \quad \text{with } \epsilon = -\frac{\langle \hat{H} \rangle}{L}. \end{aligned} \quad (7.30)$$

The numerical costs for obtaining the MPO representation of the graphs  $\Lambda_{1,2}(\text{var}(\hat{H}))$  are to leading order governed by the expenses of calculating the expectation value  $\langle \hat{H} \rangle$  as long as the MPS bond dimensions are the cost-determining factor. This demonstrates a major advantage of mapping the MPO arithmetics onto the graph representations: As the expectation value  $\langle \hat{H} \rangle$ , in general, is already available, the additional numerical costs for constructing the MPO representation are not significant compared to the application of the MPO.



Next, we take a look at the numerical stability. The graph  $\Lambda_2(\text{var}(\hat{H}))$  generates the MPO representation of the operator  $\langle \hat{H} - \langle \hat{H} \rangle \rangle$  multiplied by itself. Thus, the graph of  $\hat{H}$  is expanded so that the expectation value  $\langle \hat{H} \rangle$  is equally distributed over all lattice sites with an on-site value of  $\langle \hat{H} \rangle / L$ . As operator arithmetics is represented exactly by means of the constructed graph  $\Lambda_2(\text{var}(\hat{H}))$ , the only relevant source of catastrophic cancellation is the evaluation of  $\langle \hat{H} - \langle \hat{H} \rangle \rangle$  along the lattice that compares terms of order  $\mathcal{O}(L)$ , hence  $q = 1$ . Thus, we expect the variance to be bounded from below by  $10^{-(16-\gamma)}$ . Yet, the graph  $\Lambda_1(\text{var}(\hat{H}))$  in general also suffers from catastrophic cancellation with  $q = 2$ , as in the naïve evaluation of the variance. This is best seen by evaluating the structure of the generated matrix representation for non-vanishing tensor blocks  $W_{\tau_{j-1}\tau_j}^{n_j n'_j}$  (see Fig. 7.15b). As the latter contains the complete matrix representation of  $\hat{H}^2$ , we end up at the final site by comparing numbers of order  $\mathcal{O}(L^2)$  when performing the tensor contractions to evaluate the variance. Therefore,  $q = 2$  yields the variance to be bounded from below by  $10^{-(16-2\gamma)}$ .

In addition, graph arithmetics is exact, whereas MPO arithmetic needs a vast amount of numerical operations completed before expectation values can be calculated. Therefore, using MPO arithmetics is much more prone to collecting round-off errors, which, in drastic cases, reduces the numerical precision  $p$  by orders of magnitudes [Sch11].

### Example: Variance for S=1 Heisenberg Chains | 7.4.1

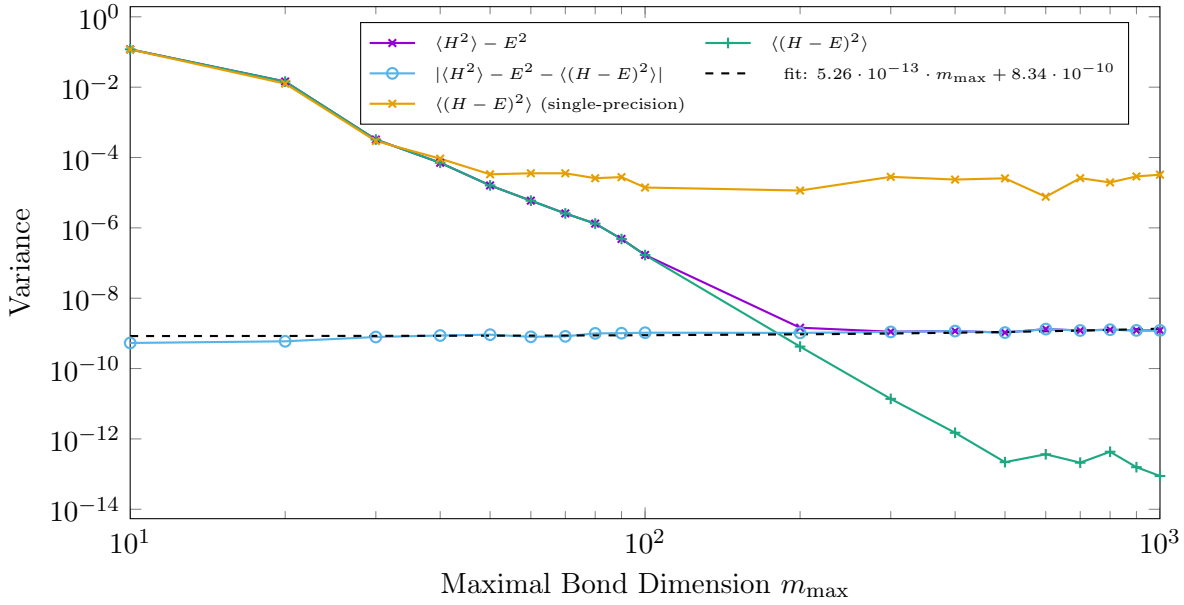
We end this chapter by testing the behavior of the variance obtained from site-tensor representations of  $\Lambda_{1,2}(\text{var}(\hat{H}_{\text{Heisenberg}}))$  for the ground state of spin-1 antiferromagnetic Heisenberg chains (see Sec. 3.2). For the ground-state search, we sweep through the system (using open boundary conditions) and optimize the site tensors via a standard Lanczos algorithm, see chapter 8.

The largest observed bond dimensions  $w_{\max}^{[1,2]}$  of the site-tensor matrices  $W_{j;w_{j-1},w_j}^{\sigma_j,\sigma'_j}$  for the MPO representations of  $\Lambda_{1,2}(\text{var}(\hat{H}_{\text{Heisenberg}}))$  are  $w_{\max}^{[1]} = 12$  and  $w_{\max}^{[2]} = 10$ , respectively. Note that these bond dimensions are generally smaller than MPS bond dimensions used during the simulations. From  $w_{\max}^{[2]} < w_{\max}^{[1]}$ , we conclude that the distribution of the constant energy term  $\langle \hat{H}_{\text{Heisenberg}} \rangle$  over the lattice sites in  $\Lambda_2(\text{var}(\hat{H}_{\text{Heisenberg}}))$  ensures a more efficient compression of the resulting MPO representation. Consistently, for common MPS bond dimensions, we find that  $\Lambda_2(\text{var}(\hat{H}_{\text{Heisenberg}}))$  is evaluated faster than  $\Lambda_1(\text{var}(\hat{H}_{\text{Heisenberg}}))$ .

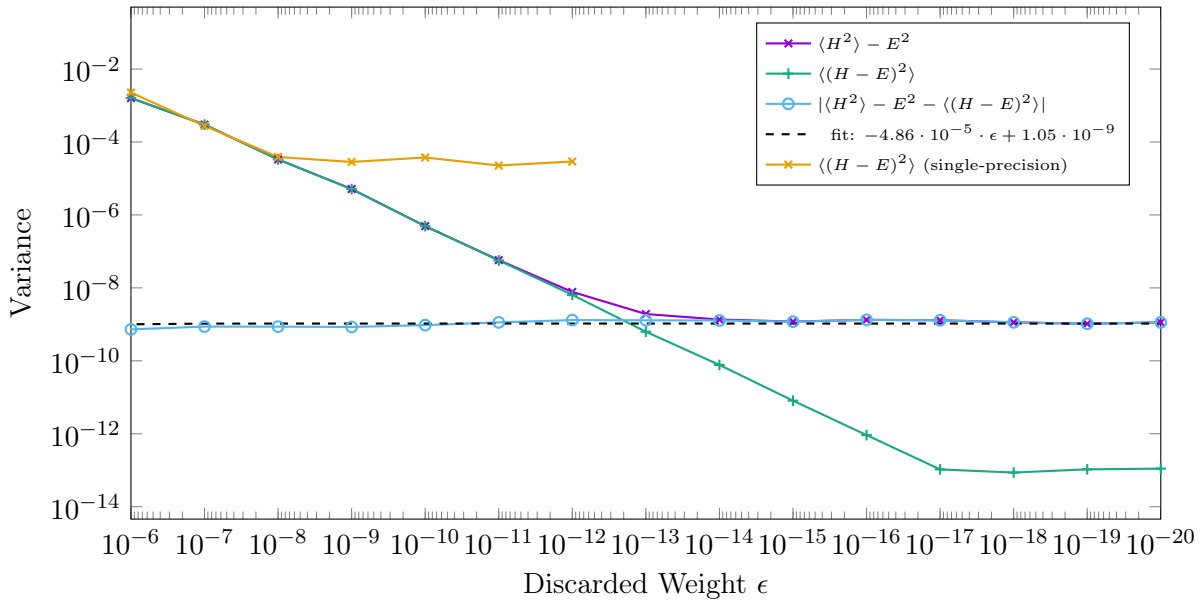
We have performed various simulations with single- and double-precision arithmetics, in which we varied either the total bond dimension per MPS site tensor  $m_{\max}$  or the discarded weight  $\epsilon$  while keeping the respective other fixed. Figures 7.16 and 7.17 show results for various values of  $m_{\max}$  and  $\epsilon$ . An important criterion for consistency of the calculations is the independence of the threshold at which catastrophic cancellation sets in. We find that varying both parameters  $m_{\max}$  and  $\epsilon$  yields a constant value of

$$\Delta \equiv |\langle H^2 \rangle - E^2 - \langle (H - E)^2 \rangle| \simeq 10^{-9}, \quad (7.31)$$

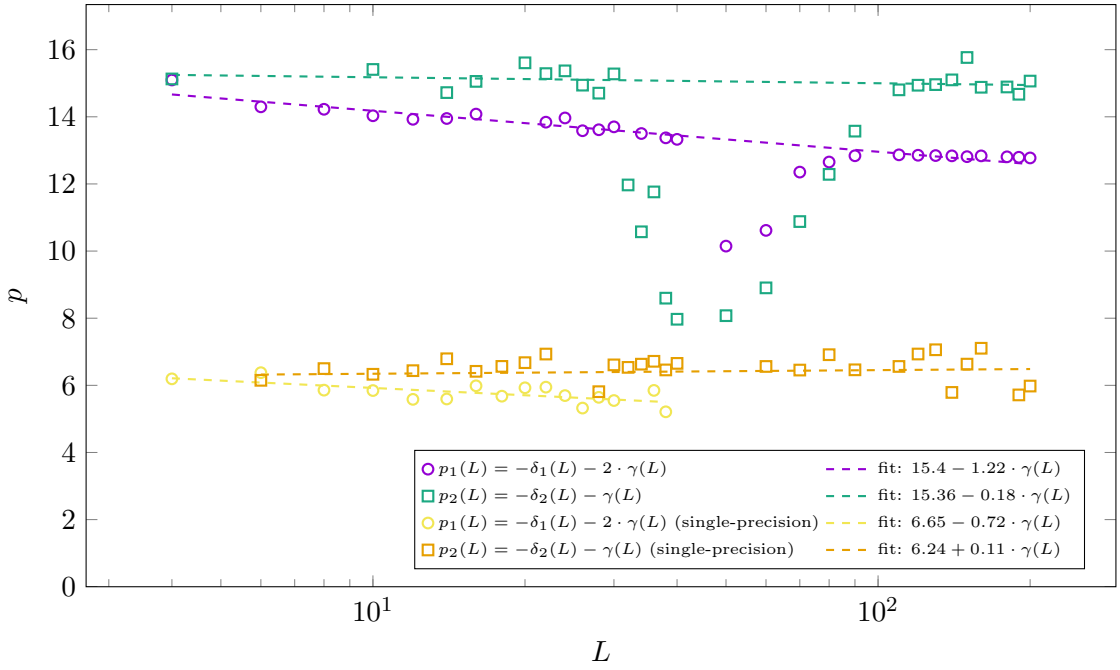
which corresponds to the value at which the graph representation  $\Lambda_1(\text{var}(\hat{H}_{\text{Heisenberg}}))$  saturates. Employing Eq. (7.28), these results suggest for the actual numerical precision an ansatz of the form  $p = p_{\text{num}} - p_r(L)$  with a correction  $p_r(L)$ , which to first order depends only on the lattice size  $L$ . Repeating the calculations for  $S = 1$  Heisenberg chains with various system sizes, we can thus extract  $p_r$  from Eq. (7.28) by estimating  $\delta = \log_{10}(\text{var}(\hat{H}_{\text{Heisenberg}}))$  from the saturated value for the variances. For the two graph representations we consider the estimator for the



**Figure 7.16:** (Extended version of Fig. 10a in [PKM17]) Variance as function of the maximum total bond dimension  $m_{\max}$  for a spin-1 Heisenberg chain with 100 sites and discarded weight  $\epsilon = 0$  in the ground state for double- and single-precision arithmetics.  $\Delta(m_{\max})$  is obtained from a linear fit of the difference  $|\langle H^2 \rangle - E^2 - \langle (H - E)^2 \rangle|$ , which yields  $\Delta(m_{\max}) = 5.26 \cdot 10^{-13} \cdot m_{\max} + 8.34 \cdot 10^{-10}$  for the case of double-precision arithmetics. For single-precision arithmetic, the evaluation of  $\Lambda_1$  does not return valid results.



**Figure 7.17:** (Extended version of Fig. 10b in [PKM17]) Variance as function of the discarded weight  $\epsilon$  for a spin-1 Heisenberg chain with 100 sites and  $m_{\max} = 5000$  in the ground state for single- and double-precision calculations.  $\Delta(\chi_{\max})$  is obtained from a linear fit of the difference  $|\langle H^2 \rangle - E^2 - \langle (H - E)^2 \rangle|$ , which yields  $\Delta(\epsilon) = -4.86 \cdot 10^{-5} \cdot \epsilon + 1.05 \cdot 10^{-9}$  in the case of double-precision arithmetics. Note that in the case of single-precision arithmetics the calculations for  $\Lambda_1$  do not return valid results. The same holds true for the results of  $\Lambda_2$  for discarded weights  $\epsilon < 10^{-12}$ .



**Figure 7.18:** (Extended version of Fig. 11 in [PKM17]) Numerical precision calculated from Eq. (7.28) for graph representations  $\Lambda_{1,2}(\text{var}(\hat{H}_{\text{Heisenberg}}))$  evaluated for various lattice sizes. The dashed lines are linear fits that illustrate the average dependence on the lattice size of the contribution  $p_r(L)$  of the effective numerical precision in the tensor-network contractions,  $p(L) = p_{\text{num}} - p_r(L)$ . Note that for intermediate system sizes the ground-state search did not converge comparably, hence the values are not used for the fits.

actual numerical precision

$$p_{1,2}(L) = -\delta_{1,2}(L) - \gamma(L)q_{1,2}. \quad (7.32)$$

For both representations of the variance, we perform a linear fit obtaining

$$p_1 = 15.4 - 1.22 \cdot \gamma(L) \quad (7.33)$$

$$p_2 = 15.36 - 0.18 \cdot \gamma(L), \quad (7.34)$$

for the double-precision calculations and

$$p_1 = 6.65 - 0.72 \cdot \gamma(L) \quad (7.35)$$

$$p_2 = 6.24 + 0.11 \cdot \gamma(L), \quad (7.36)$$

for the single-precision calculations shown in Fig. 7.18. We emphasize that the observed behavior is consistent with the ansatz above for constant single precision  $p_{\text{num, single}} \approx 8$  and double precision  $p_{\text{num, double}} \approx 16$  and residual numeric precision  $p_r(L) \approx \gamma(L) = \log_{10}(L)$ . We hence find that, aside from catastrophic cancellation, the dominating contribution to the loss of numerical precision is proportional to the lattice size, which can be associated with inevitable rounding errors generating an error  $\mathcal{O}(10^{-p_{\text{num}}})$  per lattice site.

We showed that FSM arithmetics are lossless and rounding errors only occur due to the numerical application of the generated MPO on the MPS. The methods introduced in Ref. [HMS17] obtain an additional rounding error due to the numerical evaluation of MPO sums and products. Hence, the FSM arithmetics should be favored in case of a new implementation.



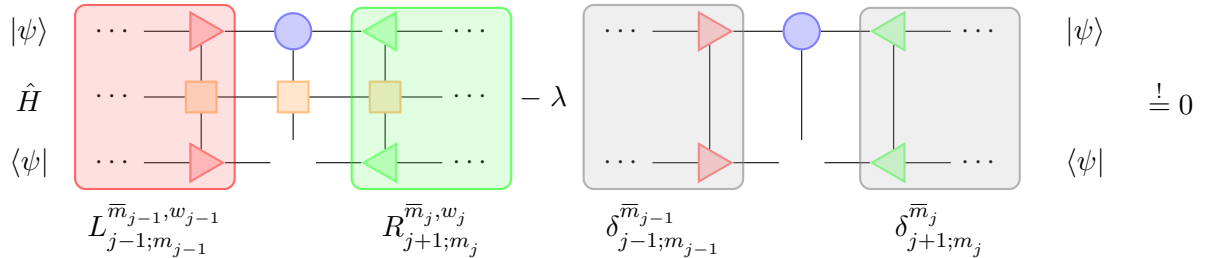
Here we review the variational ground-state search for matrix-product states (MPSs), i.e., we show how to variationally find the ground state for a given Hamiltonian  $\hat{H}$ , as described in Ref. [Sch11]. The main idea is again to use the variational principle and find the state  $|\psi\rangle$  that minimizes the energy,

$$E_0 = \min_{|\psi\rangle} \frac{\langle \psi | \hat{H} | \psi \rangle}{\langle \psi | \psi \rangle} . \quad (8.1)$$

This problem can be expressed by using the Lagrange formalism by rewriting it as minimization of  $\langle \psi | \hat{H} | \psi \rangle$  under the constraint that the norm of  $|\psi\rangle$  equals one. Introducing the Lagrange multiplier  $\lambda$  we get the Lagrange function,

$$\langle \psi | \hat{H} | \psi \rangle - \lambda \langle \psi | \psi \rangle , \quad (8.2)$$

which needs to be extremized. Following the route of the density-matrix renormalization group (DMRG), we optimize  $|\psi\rangle$  one MPS site tensor at a time. In the following, we assume that we always work in a mixed canonical representation with an active site  $j$ .

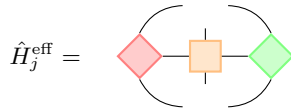


**Figure 8.1:** Summary of the partial derivatives Eq. (8.3). The tensors  $L_{j-1}$  and  $R_{j+1}$  are given in Eqs. (8.5) and (8.6).

We have to differentiate Eq. (8.2) with respect to all coefficients  $\bar{M}_{j; \bar{m}_{j-1}, \bar{m}_j}^{\sigma_l}$  and set all these differentiations to zero, as depicted in Fig. 8.1,

$$0 \stackrel{!}{=} \frac{\partial L}{\partial \bar{M}_{j; \bar{m}_{j-1}, \bar{m}_j}^{\sigma_L}} = \frac{\partial \langle \psi | \hat{H} | \psi \rangle}{\partial \bar{M}_{j; \bar{m}_{j-1}, \bar{m}_j}^{\sigma_L}} - \lambda \frac{\partial \langle \psi | \psi \rangle}{\partial \bar{M}_{j; \bar{m}_{j-1}, \bar{m}_j}^{\sigma_L}} \quad (8.3)$$

$$= \sum_{\substack{\sigma'_j, \\ \bar{m}_{j-1}, \bar{m}_j}} \sum_{w_{j-1}, w_j} \underbrace{L_{j-1}^{\bar{m}_{j-1}, w_{j-1}} W_{j; w_{j-1}, w_j}^{\sigma_j, \sigma'_j} R_{j+1}^{\bar{m}_j, w_j} M_{j; \bar{m}_{j-1}, \bar{m}_j}^{\sigma'_j}}_{\hat{H}_j^{\text{eff}}} - \lambda \underbrace{M_{j; \bar{m}_{j-1}, \bar{m}_j}^{\sigma_j}}_{|\psi_j^{\text{eff}}\rangle} , \quad (8.4)$$



in which  $L_{j-1}$  and  $R_{j+1}$  are defined only by one state  $|\psi\rangle$  (instead of two),

$$L_{j-1; m_{j-1}}^{\bar{m}_{j-1}, w_{j-1}} = \sum_{\substack{\sigma_{j-1}, \sigma'_{j-1}, \\ m_{j-2}, w_{j-2}}} \bar{A}_{j-1; \bar{m}_{j-2}, \bar{m}_{j-1}}^{\sigma'_{j-1}} L_{j-2; m_{j-2}}^{\bar{m}_{j-2}, w_{j-2}} W_{j-1; w_{j-2}, w_{j-1}}^{\sigma_{j-1}, \sigma'_{j-1}} A_{j-1; m_{j-2}, m_{j-1}}^{\sigma_{j-1}} \quad (8.5)$$

$$R_{j+1; m_j}^{\bar{m}_j, w_j} = \sum_{\substack{\sigma_{j+1}, \sigma'_{j+1}, \\ w_{j+1}, m_{j+1}}} W_{j+1; w_j, w_{j+1}}^{\sigma_{j+1}, \sigma'_{j+1}} B_{j+1; m_j, m_{j+1}}^{\sigma_{j+1}} R_{j+2; m_{j+1}}^{\bar{m}_{j+1}, w_{j+1}} \bar{B}_{j+1; \bar{m}_j, \bar{m}_{j+1}}^{\sigma'_{j+1}}. \quad (8.6)$$

This leads to a simple eigenvalue problem,

$$\hat{H}_j^{\text{eff}} |\psi_j^{\text{eff}}\rangle - \lambda |\psi_j^{\text{eff}}\rangle = 0, \quad (8.7)$$

which can be solved exactly or approximately by, e.g., the Lanczos algorithm (see Sec. 4.2). The smallest eigenvalue of this problem is the ground-state energy and the  $|\psi_j^{\text{eff}}\rangle$  is the new site tensor  $M_j$ .

In order to optimize a random starting state  $|\psi\rangle$  iteratively to represent the ground state we proceed as follows:

First, we bring  $|\psi\rangle$  into right-canonical form, without loss of generality, and create all  $R_j$  tensors, the MPS tensors take the form  $M_1 B_2 \cdots B_L$ . In this bipartition, we solve the local eigenvalue problem and store the first eigenvector back into the first site tensor. Note that  $L_0$  and  $R_{L+1}$  are dummy tensors and therefore equal to 1. Afterwards, the active site is moved to the right and a new  $L_1$  tensor is built. This procedure is repeated back and forth through the system until the energy, which is at every bipartition given by the smallest eigenvalue, converges.

In this section, we introduce two time-evolution schemata that are both based on the idea to find an approximated representation of the time-evolution-step operator  $\hat{U}(\mathcal{T})$  in order to construct the full time-evolution operator

$$\hat{U}(\tau) = e^{-\frac{i}{\hbar}\hat{H}\tau} = \lim_{N \rightarrow \infty} \left( e^{-\frac{i}{\hbar}\hat{H}\mathcal{T}} \right)^N = \lim_{N \rightarrow \infty} \left( 1 + i\mathcal{T}\hat{H} + 1/2 \left( i\mathcal{T}\hat{H} \right)^2 + \dots \right)^N \quad (9.1)$$

with  $\mathcal{T} = \tau/N$ . To keep the notation short we set  $\hbar = 1$  in the following. After obtaining this time-evolution-step operator it is applied, using the already introduced methods, until the desired time  $\tau$  is reached. Note that between every time-evolution step a truncation can be performed if necessary. There are also other time-evolution methods, e.g., Krylov methods [GR06, DWH<sup>+</sup>12] and the time-dependent variational principle [HCO<sup>+</sup>11, HLO<sup>+</sup>16], but here we refrain from explaining them as they are not used in this thesis. Instead, we briefly explain the Trotter time evolution and the MPO  $\hat{W}^{\text{II}}$  method.

## Trotter Time Evolution | 9.1

The Trotter time-evolution approach [WF04, VGRC04], also known as time-evolution block-decimation (TEBD) [Vid03, Vid04, DKSV04], is based on the idea of decomposing Eq. (9.1) into parts for which the exponential can be calculated in a feasible time. This approximative decomposition is done via the Baker-Campbell-Hausdorff formula. Consider a Hamiltonian  $\hat{H} = \sum_j \hat{h}_{j,j+1}$  that only consists of nearest-neighbor terms as, e.g., the Heisenberg chain, then Eq. (9.1) decomposes to

$$\begin{aligned} e^{-i\hat{H}\mathcal{T}} &= e^{-i\hat{H}_{\text{even}}\mathcal{T}} e^{-i\hat{H}_{\text{odd}}\mathcal{T}} e^{[\hat{H}_{\text{even}}, \hat{H}_{\text{odd}}]\mathcal{T}^2} \\ &\approx e^{-i\hat{H}_{\text{even}}\mathcal{T}} e^{-i\hat{H}_{\text{odd}}\mathcal{T}} \equiv \hat{U}^{\text{TEBD}_1}(\mathcal{T}), \end{aligned} \quad (9.2)$$

with

$$\hat{H}_{\text{even}} = \sum_{j \text{ even}} \hat{h}_{j,j+1} \quad \hat{H}_{\text{odd}} = \sum_{j \text{ odd}} \hat{h}_{j,j+1}. \quad (9.3)$$

Due to the fact that all terms of  $\hat{H}_{\text{even}}$  as well as all terms of  $\hat{H}_{\text{odd}}$  commute with each other, they can be exponentiated individually,

$$e^{-i\hat{H}_{\text{even}}\mathcal{T}} = \prod_{j \text{ even}} e^{-i\hat{h}_{j,j+1}\mathcal{T}} \quad e^{-i\hat{H}_{\text{odd}}\mathcal{T}} = \prod_{j \text{ odd}} e^{-i\hat{h}_{j,j+1}\mathcal{T}}. \quad (9.4)$$

Thus, in order to create the corresponding matrix-product operator (MPO) only matrices of dimension  $d \times d$  need to be exponentiated. The error per time step, due to discarding the

commutator in Eq. (9.2), is of second order in the time step  $O(\mathcal{T}^2)$  for a fixed decomposition, i.e., a constant commutator.

Moreover, this method is not restricted to nearest-neighbor terms, but can be used if the operator

$$\hat{H} = \sum_{\alpha} \sum_{j=\alpha}^{L/r_{\max}} \underbrace{\sum_{\nu_1 \dots \nu_n} f_{\nu_1 \dots \nu_n}^{j'} \cdot \bigotimes_{l=j'}^{j'+r_{\max}} \hat{k}_{\nu_l}^{(l)}}_{\hat{\delta}^{(j')}} \quad \text{with } j' = (1 + r_{\max})j - r_{\max} \quad (9.5)$$

can be grouped in a way that  $\hat{\delta}^{(j')}$  can be exponentiated, i.e., is Hermitian and sufficiently small, and  $\hat{\delta}^{(j')}$  commute for different  $j'$ , i.e.,  $[\hat{\delta}^{j'}, \hat{\delta}^{j'}] = 0$ . For a detailed description of the indices in Eq. (9.5) see Sec. 7.1 and note that here the identity is part of  $\mathcal{K}$ . In summary, we need to group the sum into as many sums of operators corresponding to the longest distance captured by the operator. All other operators, which act only in the corresponding part of the system, can be incorporated into those operator sums.

Starting from this decomposition, the time-evolution operator can be constructed analogously to the nearest-neighbor example,

$$\hat{U}^{\text{TEBD}_1}(\mathcal{T}) = \prod_{\alpha}^{r_{\max}} e^{-i\hat{H}_{\alpha}\mathcal{T}}. \quad (9.6)$$

Higher order decompositions can be obtained by further decomposing the exponential and result in smaller errors. As example, we show the second-order decomposition,

$$\hat{U}^{\text{TEBD}_2}(\mathcal{T}) = \prod_{\alpha}^{r_{\max}} e^{-i\hat{H}_{\alpha}\mathcal{T}/2} \cdot \prod_{\alpha}^{r_{\max}} e^{-i\hat{H}_{r_{\max}-\alpha}\mathcal{T}/2}, \quad (9.7)$$

with an error in  $O(\mathcal{T}^3)$  per time step and the fourth-order decomposition,

$$\hat{U}^{\text{TEBD}_4}(\mathcal{T}) = \hat{U}^{\text{TEBD}_2}(\mathcal{T}_1)\hat{U}^{\text{TEBD}_2}(\mathcal{T}_1)\hat{U}^{\text{TEBD}_2}(\mathcal{T}_2)\hat{U}^{\text{TEBD}_2}(\mathcal{T}_1)\hat{U}^{\text{TEBD}_2}(\mathcal{T}_1), \quad (9.8)$$

with  $\mathcal{T}_1 = \frac{\mathcal{T}}{4-4^{1/3}}$ ,  $\mathcal{T}_2 = (1 - 4\mathcal{T}_1)\mathcal{T}$ , and an error in  $O(\mathcal{T}^5)$  per time step [Suz76, Suz91].

If the previous approach to handle long-ranged interactions is not feasible, e.g., because there are only a few, but very long-ranged terms, swap gates [SW10] can be used to move the corresponding sites close to each other in order to perform the time evolution. Afterwards, the swap gates are applied in opposite order to move the sites back to their origin. By choosing a clever order of the terms of the time evolution, series of swap gates can be omitted.

## MPO $\hat{W}^{\text{I/II}}$ | 9.2

In contrast to TEBD, in which the truncation of the full time-evolution operator Eq. (9.1) is performed on a certain order of  $\tau$ , an alternative, proposed in Ref. [ZMK<sup>+</sup>15], truncates the number of operators that act on a single site. Here, we summarize the construction instructions first for the more descriptive MPO  $\hat{W}^{\text{I}}$  and afterwards also for the more precise and usually preferred MPO  $\hat{W}^{\text{II}}$ . Note that we denote general (approximated) time-evolution operator by  $\hat{U}$  and only MPO representations by  $\hat{W}$ .



Consider a Hamiltonian that is divided into all its summands

$$\hat{H} = \sum_{\vec{j}_i} \hat{H}_{\vec{j}_i} \quad (9.9)$$

in which every vector  $\vec{j}$  consists of the site indices of the operators in ascending order. In order to keep the notation compact, we define that such a vector  $\vec{j}$  is  $n$ -smaller ( $\stackrel{n}{<}$ ) than a vector  $\vec{j}'$  if maximally the largest  $n$  entries in  $\vec{j}$  overlap with the smallest  $n$  entries of  $\vec{j}'$ .

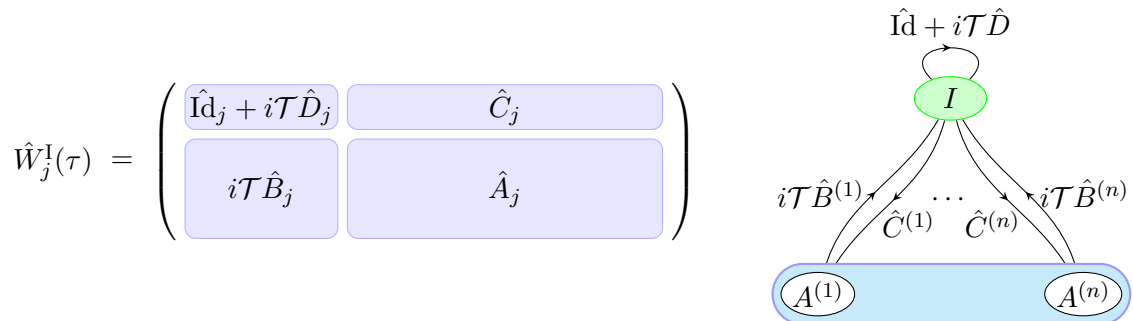
The idea for the time-evolution operator  $\hat{W}^{\text{I}}$  is to truncate the full time-evolution operator Eq. (9.1) and discard all terms in which more than one operator is applied per site. The corresponding time-evolution operator is given by

$$\hat{U}^{\text{I}}(\mathcal{T}) = \hat{\text{Id}} + i\mathcal{T} \sum_{\vec{j}} \hat{H}_{\vec{j}} + (i\mathcal{T})^2 \sum_{\vec{j} \stackrel{0}{<} \vec{j}'} \hat{H}_{\vec{j}} \hat{H}_{\vec{j}'} + (i\mathcal{T})^3 \sum_{\vec{j} \stackrel{0}{<} \vec{j}' \stackrel{0}{<} \vec{j}''} \hat{H}_{\vec{j}} \hat{H}_{\vec{j}'} \hat{H}_{\vec{j}''} + \dots, \quad (9.10)$$

in which furthermore the division by the factorial is discarded in order to obtain a simple finite-state machine (FSM) representation. This additional approximation is permitted, because the error is nevertheless at order  $\mathcal{T}^2$ . In order to understand how to construct an FSM representation of  $\hat{U}^{\text{I}}$ , consider a site tensor of a Hamiltonian in its state-transition-table representation,

$$\hat{W}_j = \begin{pmatrix} \hat{\text{Id}}_j & \hat{C}_j & \hat{D}_j \\ & \hat{A}_j & \hat{B}_j \\ & & \hat{\text{Id}}_j \end{pmatrix}. \quad (9.11)$$

We identify four blocks in this representation: The start and the end of interactions,  $\hat{C}_j$  and  $\hat{B}_j$ , respectively, the intermediate parts of interactions  $\hat{A}_j$  and the local part  $\hat{D}_j$ . As the sizes of the boxes already suggest,  $\hat{A}_j$  is an operator-valued matrix with dimension  $n_{j-1} \times n_j$ ,  $\hat{B}_j$  is a vector of dimension  $n_{j-1}$ ,  $\hat{C}_j$  is a row vector of dimension  $n_j$ , and  $\hat{D}_j$  is an operator-valued scalar.  $n_j$  denotes the number of different non-local interaction terms in the Hamiltonian and  $n_{\text{max}}$  is the maximal number of local interaction terms. The bond tensors of the time-evolution operator and the corresponding FSM are shown in Fig. 9.1.



**Figure 9.1:** (Left) Block structure of the MPO  $\hat{W}_j^{\text{I}}$  and (right) the corresponding FSM. Note that the nodes  $A^{(i)}$  represent arbitrary long-range interactions and hence arbitrary FSMs.

Discarding all terms with multiple local operators is arguably not the best approximation of Eq. (9.1). For that reason,  $\hat{W}^{\text{II}}$  is introduced, which captures all terms with an overlap of

maximally one. Unfortunately, there cannot be a closed representation in terms of FSMs nor MPOs for an approximation that includes all those terms. This can be best illustrated with a small example, namely a Hamiltonian that only consists of local operators,

$$\hat{H} = \sum_j \hat{S}_j^z, \quad (9.12)$$

for which we would like to represent the time-evolution operator

$$\hat{U}^{\text{II}} = \hat{\text{Id}} + i\mathcal{T} \sum_{\vec{j}} \hat{H}_{\vec{j}} + \frac{(i\mathcal{T})^2}{2} \sum_{\vec{j}^1 < \vec{j}'} \hat{H}_{\vec{j}^1} \hat{H}_{\vec{j}'} + \frac{(i\mathcal{T})^3}{6} \sum_{\vec{j}^1 < \vec{j}' < \vec{j}''} \hat{H}_{\vec{j}^1} \hat{H}_{\vec{j}'} \hat{H}_{\vec{j}''} + \dots \quad (9.13)$$

For every power  $\mathcal{T}^l$  a local operator string  $(\hat{S}_j^z)^l$  arises and therefore an infinite number of edges would be necessary to represent Eq. (9.13) exactly with an FSM and hence an MPO with infinite bond dimension  $w$ .

In order to nevertheless represent Eq. (9.13) as an MPO with site tensors

$$\hat{W}_j^{\text{II};\sigma_j,\sigma'_j}(\tau) = \left( \begin{array}{cc} \boxed{W_{j;\mathcal{D}}^{\text{II};\sigma_j,\sigma'_j}} & \boxed{W_{j;\mathcal{C}}^{\text{II};\sigma_j,\sigma'_j}} \\ \boxed{W_{j;\mathcal{B}}^{\text{II};\sigma_j,\sigma'_j}} & \boxed{W_{j;\mathcal{A}}^{\text{II};\sigma_j,\sigma'_j}} \end{array} \right), \quad (9.14)$$

a matrix exponential, without any friendly properties, needs to be solved. The derivation of the matrix exponential can be found in Ref. [ZMK<sup>+</sup>15] and in Ref. [PKS<sup>+</sup>19]. In summary,

$$\hat{W}_{j;\mathcal{S};i,i'}^{\text{II};\sigma_j,\sigma'_j} = \langle \sigma_j, \sigma'_j | \begin{pmatrix} \delta_{\mathcal{S},\mathcal{D}} \\ \delta_{\mathcal{S},\mathcal{C}} \\ \delta_{\mathcal{S},\mathcal{B}} \\ \delta_{\mathcal{S},\mathcal{A}} \end{pmatrix}^t \exp \left( \begin{pmatrix} i\mathcal{T}\hat{D}_{1,1} & 0 & 0 & 0 \\ \hat{C}_{1,i'} & i\mathcal{T}\hat{D}_{1,1} & 0 & 0 \\ i\mathcal{T}\hat{B}_{i,1} & 0 & i\mathcal{T}\hat{D}_{1,1} & 0 \\ \hat{A}_{i,i'} & i\mathcal{T}\hat{B}_{i,1} & \hat{C}_{1,i'} & i\mathcal{T}\hat{D}_{1,1} \end{pmatrix} \right) \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} | \sigma_j, \sigma'_j \rangle, \quad (9.15)$$

with  $\mathcal{S} \in \{\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}\}$  needs to be solved for all entries of  $\hat{W}_j^{\text{II};\sigma_j,\sigma'_j}(\tau)$ . Note that the matrix exponential only needs to be solved for  $\mathcal{A}_{i,i'}$  and can be reused afterwards. Hence, the complexity of creating the MPO  $\hat{W}^{\text{II}}$  is of the order of  $d^3 n_{\text{max}}^2$ , which is small compared to the usual complexity of the tensor contractions, which scale with the bond dimension of the matrix-product state (MPS). It is therefore always beneficial to use  $\hat{W}^{\text{II}}$  instead of  $\hat{W}^{\text{I}}$ , because the final MPO bond dimension is identical but the approximation is more accurate.

A universal quantum computer is a very powerful tool compared to a classical computer as it is supposed to solve some NP hard problems in polynomial time [Pre12]. Based on the method development introduced in part II, we present here a universal quantum-computer simulator (QCS). Since it is simulated on a classical computer, it obviously cannot exploit the quantum supremacy. Nevertheless, it is a useful tool in the process of developing quantum algorithms. Furthermore, due to its customizability (see Sec. 10.3), the QCS is also a very versatile tool for arbitrary quantum-state manipulation. One example is the creation of an infinite-temperature state for Hubbard-like models, which is explained in Sec. 10.4.4. Other examples, which are not captured in this thesis, are the preparation of initial states, and the measurements and exhaustive analysis on stored quantum states.

At the time of writing, several QCSs already exist, some of them publicly available. An extensive summary over the most prominent ones is given in [LaR18]. Furthermore, a large collection of quantum algorithms, with implementations for the IBM quantum experience, is summarized in [CEP<sup>+</sup>18].

Maybe the most intuitively programmable is the QCS that is associated with the IBM quantum experience and their quantum computer. The programming language used within their framework is called OpenQASM [CBSG17]. It offers all commands to run universal quantum-computer codes, i.e., the possibility to apply at least a set of universal quantum gates, measurements, and some program control structures. Furthermore, a graphical programming tool is available online, but it does not support all features of OpenQASM.

In order to gain more flexibility, another abstraction layer called QISKit can be used to obtain the necessary OpenQASM code. QISKit offers an interface to different back-ends including the real IBM quantum-computer hardware. The included, exact simulation back-ends are limited by the number of qubits. This is due to representation of the state in the full basis, which is extremely memory consuming, or in a vector basis, which is still very memory consuming. The same holds true for Microsoft's LIQUi|> [WS14] and their Quantum Development Kit [SGT<sup>+</sup>18]. In the last two decades, the idea to build a QCS based on matrix-product states (MPSs) was already formulated [Vid03] and applied, e.g., to Shor's factorization algorithm [DHH17].

Nevertheless, we present the benefits of such an implementation here. Our QCS represents the state as (possibly truncated) matrix-product states, see chapter 5. Therefore, not the number of qubits but the increase in entanglement due to application of gates is the limiting factor. Because MPSs can, in principle, represent every state if no truncation is used, our simulator scales never worse in memory usage than those working in the full basis. Furthermore, we obtain good estimations of the errors that occur in our simulation, namely the truncation error and the gate application error. Note that the discussion whether or not large entanglement entropy is the necessary ingredient for performance growth due to quantum computation is still unclear [JL03, VdN13].

Another important difference is that we can measure the entanglement entropy without any further numerical costs. Hence, we can predict when a quantum algorithm does not need a quantum computer and therefore makes no use of the quantum supremacy. In order to show the

inverse statement, one would need to prove that the entanglement grows in all bases. This has been shown to be a NP-complete problem [usc01].

In this chapter, we use the computational basis, which is widely used in quantum information, with the translation from the physical basis

$$|0\rangle = |\uparrow\rangle \quad |1\rangle = |\downarrow\rangle. \quad (10.1)$$

$|\uparrow\rangle$  and  $|\downarrow\rangle$  denote the spin-1/2  $S^z$  basis vectors. Equalities between operators and matrices refer to the matrix representation in the  $S^z$  basis.

This chapter starts with an overview of the most common quantum gates and the definition of a set of universal quantum gates. Afterwards, we explain how our QCS measures an observable in contrast to a quantum computer. Then, the steps we took to extend our MPS code to a fully programmable universal QCS are presented. We complement the chapter by giving some small example applications of the QCS.

## Universal Quantum Gates | 10.1

In this section, we introduce the quantum gates that are available in our QCS. Most of them are in one way or the other present in all quantum computer environments that are programmable by gates. The provided information is consistent with those given in [NC04, IBM].

Additionally, we introduce three non-unitary helper operators. These, together with the identity, are used to construct all the following gates by means of finite-state machines (FSMs) (see chapter 7). As the helper operators are not unitary, they cannot be considered as quantum gates.

To achieve a set of universal quantum gates, several combinations are known to be sufficient. For example, the Toffoli and the Hadamard gate [Shi03] or the CNOT, the Hadamard, and the  $R_{\pi/4}^Z$  gate [NC04]. Subsequently, the CNOT and the  ${}^3U(\lambda, \phi, \Theta)$  gate are also sufficient, because of the versatility of the latter.

### Non-Unitary Helper Operators | 10.1.1

The operators in this section are the spin-raising operator  $\hat{\mathbf{S}}^+$ , the spin-lowering operators  $\hat{\mathbf{S}}^-$ , and the spin operator  $\hat{\mathbf{S}}^z$ ,

$$\hat{\mathbf{S}}_j^+ = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}_j, \quad \hat{\mathbf{S}}_j^- = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}_j, \quad \hat{\mathbf{S}}_j^z = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}_j. \quad (10.2)$$

They are implemented internally in the QCS and together with the identity operator  $\hat{\mathbf{I}}\mathbf{d}$  (see Eq. (10.3)) they are the building blocks for all operators.

### Single-Site Gates | 10.1.2

All single-site quantum gates can be represented by unitary complex two-by-two matrices. In the following, we present the pictorial and the matrix representation of the most common single-site quantum gates, accompanied by a brief explanation of their action.

In our implementation, all single-site gates presented here can be applied on a single site  $j$  by executing one of the following lines:

```
app GATENAME [j];
apply GATENAME [j];
```

### Identity



**Figure 10.1:** Pictorial representation of the identity gate Eq. (10.3), which is a simple quantum wire.

The identity operator  $\hat{\mathbf{I}}\mathbf{d}$  represents the trivial operation, i.e., it leaves the qubit unchanged. Nevertheless, it is very useful in order to construct other operators, e.g., the operators  $\hat{\mathbf{O}}$  and  $\hat{\mathbf{I}}$  in Eq. (10.12). Furthermore, it is one of the four internally implemented operators.

The pictorial representation is given in Fig. 10.1 and shows that the identity has one more very important role in every quantum-gate code: The role of the quantum wire. The matrix representation at qubit  $j$  is

$$\hat{\mathbf{I}}\mathbf{d}_j = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}_j. \quad (10.3)$$

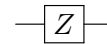
In our implementation, the identity is implicitly applied on all qubits if no other gate is specified. It is therefore never actually necessary to write it explicitly, even though it is possible.

### Z

The  $\hat{\mathbf{Z}}$  operator is derived directly from the internally implemented  $\hat{\mathbf{S}}^z$  operator. It equals the Pauli matrix  $\sigma_z$  and is called a phase-flip. This gate is a special case of the phase-shift operator Eq. (10.8) with  $\lambda = \pi$ .

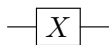
The pictorial representation is given by a “Z” in a box (see Fig. 10.2) and the matrix representation at qubit  $j$  is

$$\hat{\mathbf{Z}}_j = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}_j. \quad (10.4)$$



**Figure 10.2:** Pictorial representation of the Z gate Eq. (10.4).

### X



**Figure 10.3:** Pictorial representation of the X gate Eq. (10.5).

The  $\hat{\mathbf{X}}$  operator is the quantum version of a classical NOT gate. It turns a  $|0\rangle$  to a  $|1\rangle$  and vice versa and is also called a bit-flip. The pictorial representation is analogous to the representation of the Z gate and given in Fig. 10.3. In our implementation,  $\hat{\mathbf{X}}$  is built by the sum of  $\hat{\mathbf{S}}^-$  and  $\hat{\mathbf{S}}^+$  and is given by the Pauli matrix  $\sigma_x$ ,

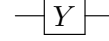
$$\hat{\mathbf{X}}_j = \hat{\mathbf{S}}_j^- + \hat{\mathbf{S}}_j^+ = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}_j. \quad (10.5)$$

## Y

Also, the Pauli matrix  $\sigma_y$  is used as a quantum gate. This gate is denoted by Y and maps  $|0\rangle$  to  $i|1\rangle$  and  $|1\rangle$  to  $-i|0\rangle$ .

The pictorial representation is also analogous to the representation of the previous gates and given in Fig. 10.4. In our implementation,  $\hat{\mathbf{Y}}$  is built by the difference of  $\hat{\mathbf{S}}^-$  and  $\hat{\mathbf{S}}^+$  multiplied by  $i$  and is given by the Pauli matrix  $\sigma_y$ ,

$$\hat{\mathbf{Y}}_j = i \left( \hat{\mathbf{S}}_j^- - \hat{\mathbf{S}}_j^+ \right) = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}_j. \quad (10.6)$$



**Figure 10.4:** Pictorial representation of the Y gate Eq. (10.6).

## Hadamard



**Figure 10.5:** Pictorial representation of the Hadamard gate Eq. (10.7).

The Hadamard gate is used to create a superposition of the states  $|0\rangle$  and  $|1\rangle$ . To be more precise, it maps  $|0\rangle$  to  $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$  and  $|1\rangle$  to  $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$ .

The pictorial representation is given in Fig. 10.5. In our implementation,  $\hat{\mathbf{H}}$  is built by a sum of the internally implemented operators Eq. (10.2) (see also Fig. 10.15 for the corresponding FSM),

$$\hat{\mathbf{H}}_j = \frac{1}{\sqrt{2}} \left( \hat{\mathbf{S}}_j^+ + \hat{\mathbf{S}}_j^- + 2\hat{\mathbf{S}}_j^z \right) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}_j. \quad (10.7)$$

## Parametrized Single-Site Gates | 10.1.3

In order to use a quantum computer in a universal way, it is beneficial to be able to use input data additionally to the already user-defined algorithm. This can be done by using parametrized gates that will act depending on an input parameter. In the following, we review the three parametrized single-site gates as they are used in the IBM quantum experience [IBM]. In the end, we also show that the first two ones can be represented by the last one. However, using the less complicated gates is beneficial to the computation time, in the simulator and in the real quantum computer.

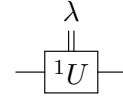
Note that parameterized gates always need additional inputs. Within our QCS, this means that the call is given by

```
app GATENAME [j, PARAMETER, ...];
apply GATENAME [j, PARAMETER, ...];
```

${}^1\mathbf{U}$ 

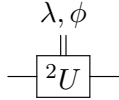
The  ${}^1\hat{\mathbf{U}}$  phase-shift operator expects one classical (real) parameter  $\lambda$ . It rotates the state  $|1\rangle$  by the angle  $\lambda$  around the  $z$  axis. In the case of  $\lambda = \pi$  we obtain the  $\hat{\mathbf{Z}}$  operator Eq. (10.4). The rotation operator  $\hat{\mathbf{R}}^z(\lambda)$  is closely related to the phase-shift operator by  $\hat{\mathbf{R}}^z(\lambda) = e^{-i\lambda/2} \cdot {}^1\hat{\mathbf{U}}$ .

The pictorial representation is given by a “ ${}^1U$ ” in a box with an additional classical wire (double line) (see Fig. 10.6). Its matrix representation is



**Figure 10.6:** Pictorial representation of the  ${}^1U$  gate, including its input parameter  $\lambda$ , see Eq. (10.8).

$${}^1\hat{\mathbf{U}}_j(\lambda) = (1 - e^{i\lambda}) \hat{\mathbf{S}}_j^z + \frac{(1 + e^{i\lambda})}{2} \hat{\mathbf{I}}_d_j = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{pmatrix}_j. \quad (10.8)$$

 ${}^2\mathbf{U}$ 

**Figure 10.7:** Pictorial representation of the  ${}^2U$  gate, including its input parameters  $\lambda$  and  $\phi$ , see Eq. (10.9).

The second operator  ${}^2\hat{\mathbf{U}}$  expects two classical (real) parameters  $\lambda$  and  $\phi$ . It can be understood as a generalized Hadamard gate, representing it for  $\lambda = \pi, \phi = 0$ .

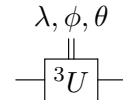
The pictorial representation is given by a “ ${}^2U$ ” in a box (see Fig. 10.7) and its matrix representation is given by

$${}^2\hat{\mathbf{U}}_j(\lambda, \phi) = \frac{1}{\sqrt{2}} \left[ e^{i\phi} \hat{\mathbf{S}}_j^- - e^{i\lambda} \hat{\mathbf{S}}_j^+ + (1 - e^{i(\lambda+\phi)}) \hat{\mathbf{S}}_j^z + \frac{(1 + e^{i(\lambda+\phi)})}{2} \hat{\mathbf{I}}_d_j \right] = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -e^{i\lambda} \\ e^{i\phi} & e^{i(\lambda+\phi)} \end{pmatrix}_j. \quad (10.9)$$

 ${}^3\mathbf{U}$ 

The third operator  ${}^3\hat{\mathbf{U}}$  expects three classical (real) parameters  $\lambda$ ,  $\phi$ , and  $\theta$ . It can be understood as the general unitary gate. All previously described gates can be represented by the  ${}^3U$  gate.

The pictorial representation is given by a “ ${}^3U$ ” in a box (see Fig. 10.8) and the matrix representation is given by



**Figure 10.8:** Pictorial representation of the  ${}^3U$  gate, including its input parameters  $\lambda$ ,  $\phi$ , and  $\theta$ , see Eq. (10.10).

$${}^3\hat{\mathbf{U}}_j(\lambda, \phi, \theta) = \left[ \sin(\theta/2) \left( e^{i\phi} \hat{\mathbf{S}}_j^- - e^{i\lambda} \hat{\mathbf{S}}_j^+ \right) + \cos(\theta/2) \left( (1 - e^{i(\lambda+\phi)}) \hat{\mathbf{S}}_j^z + \frac{(1 + e^{i(\lambda+\phi)})}{2} \hat{\mathbf{I}}_d_j \right) \right] \quad (10.10)$$

$$= \begin{pmatrix} \cos(\theta/2) & -e^{i\lambda} \sin(\theta/2) \\ e^{i\phi} \sin(\theta/2) & e^{i(\lambda+\phi)} \cos(\theta/2) \end{pmatrix}_j. \quad (10.11)$$

## Multi-Site Gates | 10.1.4

A multi-site gate is a gate that acts on more than one qubit. Such gates are essential in order to generate entangled qubits. Most of them are controlled gates, which apply a single-site gate on a qubit only if the other one is in the state  $|1\rangle$ . Otherwise, the identity is applied.

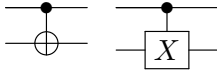
In order to use a compact notation, we introduce the non-unitary operators

$$\hat{\mathbf{o}}_j = \frac{1}{2}\hat{\mathbf{I}}_j + \hat{\mathbf{S}}_j^z = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \hat{\mathbf{i}}_j = \frac{1}{2}\hat{\mathbf{I}}_j - \hat{\mathbf{S}}_j^z = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \quad (10.12)$$

which check whether the qubit  $j$  is  $|0\rangle$  or  $|1\rangle$ , respectively. If the condition is met, these operators act as identity. Otherwise, the state is annihilated.

As these gates have different numbers of parameters, the usage within our QCS is presented for every gate. Note that we only show the non-abbreviated version while the abbreviated version also works and is used analogous to the single-site gates. Furthermore, the matrix representations for the multi-site operators depend on the choice of the basis. We choose the natural (binary) basis,  $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$  for all two-site operators.

## CNOT and cX



The CNOT gate, which acts on two qubits, is equivalent to a controlled X gate. This means that the target qubit  $t$  is flipped only if the control qubit  $c$  is  $|1\rangle$  and stays untouched otherwise.

**Figure 10.9:** Pictorial representation of the CNOT[0,1] and the cX gate Eq. (10.13).

$$\widehat{\text{CNOT}}_{c,t} = \widehat{\text{cX}}_{c,t} = \hat{\mathbf{i}}_c \otimes \hat{\mathbf{S}}_t^x + \hat{\mathbf{o}}_c \otimes \hat{\mathbf{I}}_t \cong \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (10.13)$$

The CNOT gate needs one control qubit  $c$  and one target qubit  $t$ . These are denoted by `control` and `target`, respectively. Thus, it is called by the following line of code:

```
apply CNOT[control, target];
```

Note that in contrast to real quantum computers, in which the connectivity is a challenge for itself, every combination of `control` and `target`, except `control = target`, is allowed in our simulator.

## Toffoli

The Toffoli gate is an expansion of the CNOT gate, i.e., it has two control qubits ( $c_1, c_2$ ).

This gate is capable of representing classical gates (AND, XOR, NOT, ...) [Tof80]. A quantum computer that has this quantum gate implemented is therefore capable of simulating all classical programs. Its representation is



**Figure 10.10:** Pictorial representation of the Toffoli gate Eq. (10.14).



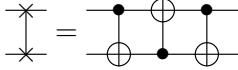
$$\mathbf{Toffoli}_{c_1, c_2, t} = \hat{\mathbf{1}}_{c_1} \otimes \hat{\mathbf{1}}_{c_2} \otimes \hat{\mathbf{S}}_t^x + (\hat{\mathbf{1}}_{c_1} \otimes \hat{\mathbf{0}}_{c_2} + \hat{\mathbf{0}}_{c_1} \otimes \hat{\mathbf{1}}_{c_2} + \hat{\mathbf{0}}_{c_1} \otimes \hat{\mathbf{0}}_{c_2}) \otimes \hat{\mathbf{I}}_t. \quad (10.14)$$

The Toffoli gate needs two control qubits  $c_1, c_2$  and one target qubit  $t$ . These are denoted by `control1`, `control2`, and `target`, respectively, i.e., it is called by

```
apply Toffoli[control1, control2, target];
```

Note that this gate is classically universal but it is not universal for quantum computing. Another single-qubit gate that creates a superposition needs to be added [Shi03].

### Swap



**Figure 10.11:** Pictorial representation of the swap gate Eq. (10.15) and its representation in terms of three CNOT gates [Got98].

The swap gate does no actual computation. It is mainly needed because in a real quantum computer not all qubits are connected in the same way, i.e., the CNOT cannot act on arbitrary gates. To overcome this connection problem, the swap gate can be used to move qubits.

In our implementation, the swap gate can be used to reduce the entanglement. For example, if a qubit should be entangled with a qubit on the other side of the system, it might be better to first move the qubits close to each other.

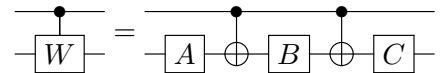
$$\begin{aligned} \widehat{\text{swap}}_{q_1, q_2} &= \underbrace{\hat{\mathbf{1}}_{q_1} \otimes \hat{\mathbf{1}}_{q_2} + \hat{\mathbf{0}}_{q_1} \otimes \hat{\mathbf{0}}_{q_2}}_{=\frac{1}{2}\hat{\mathbf{I}}_{q_1} \otimes \hat{\mathbf{I}}_{q_2} + 2\hat{\mathbf{S}}_{q_1}^z \otimes \hat{\mathbf{S}}_{q_2}^z} + \hat{\mathbf{S}}_{q_1}^+ \otimes \hat{\mathbf{S}}_{q_2}^- + \hat{\mathbf{S}}_{q_1}^- \otimes \hat{\mathbf{S}}_{q_2}^+ \cong \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned} \quad (10.15)$$

The swap gate needs two qubits  $q_1, q_2$ . These are denoted by `qubit1` and `qubit2`, respectively. In our implementation the code reads

```
apply swap[qubit1, qubit2];
```

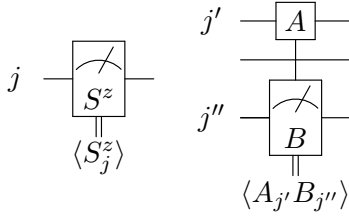
### Arbitrary Controlled Gate cW

In [BBC<sup>+</sup>95] it is shown that an arbitrary controlled gate cW can be constructed from three single-site gates and two CNOTs if the single-site gate  $W$  can be represented by a special unitary matrix. In this decomposition, the gates  $A$ ,  $B$ , and  $C$  consist of up to two rotations. These rotations can be done by the  ${}^iU$  gates.



**Figure 10.12:** Pictorial representation of the decomposition of an arbitrary controlled gate cW into three single site gates and two CNOT gates [BBC<sup>+</sup>95].

## Measurements | 10.2



**Figure 10.13:** Pictorial representation of a single-qubit measurement (left,  $\langle S_j^z \rangle$ ) and a two-qubit measurement (right,  $\langle A_{j'} B_{j''} \rangle$ ).

Within our QCS, we are capable of measuring every observable that can be built by the operators given in Sec. 10.1. Furthermore, new observables, e.g., correlation functions, can be built in terms of FSMs, see chapter 7. A simple single-site measurement is depicted on the left side of Fig. 10.13, whereas the right side shows a two-site correlation function.

The syntax for these measurements is given by

```
measure GATENAME [PARAMETERS] ;
```

where the `PARAMETERS` depend on the chosen gate `GATENAME`.

The keyword `measure` can be shortened to `meas` and `mea`. A special role is reserved for gates that only need a single parameter (the site they shall act on). There, every legal value (0 to system size minus one) is used if no parameter is defined, i.e., the observable is measured for every qubit.

In order to print the measurements to the standard output or to write them into files created in the directory `observables` the following code is used.

```
perform print [observable, GATENAME] ;
perform write [observable, GATENAME] ;
```

Again, the `GATENAME` defines the gate from which the observable is derived and the keyword `perform` can be shortened to `perf` and `per`. Furthermore, the keyword `print` can be shortened to `p`, `write` to `w`, and `observable` to `obs`.

With our QCS, we directly calculate the expectation value of an observable without destroying or changing the current quantum state. It is therefore possible to determine the exact expectation values of an arbitrary number of observables at every point within the quantum algorithm. Considering this complete access to all information, this simulator can also be used as a quantum-algorithm debugger.

In contrast, it is only possible to measure a single realization of the system with a quantum computer, i.e., the outcome is always one of the eigenvalues of the operator and after the measurement the system is in the corresponding eigenstate of the operator. For example, if the spin in  $z$ -direction of a qubit

$$|\psi\rangle = \sqrt{2/3}|0\rangle + \sqrt{1/3}|1\rangle \quad (10.16)$$

is measured, the result of our QCS is the expectation value  $\langle \hat{S}^z \rangle = 1/6$  and an unchanged quantum state. In contrast, a quantum computer obtains

$$\hat{S}^z |\psi\rangle = \begin{cases} -1/2 |1\rangle & \text{in } 1/3 \text{ of the realizations} \\ 1/2 |0\rangle & \text{in } 2/3 \text{ of the realizations} \end{cases} \quad (10.17)$$

In order to get the expectation value of the operator from a quantum computer, the calculation needs to be repeated several times, depending on the demanded precision, i.e., infinitely often for an exact result,

$$\langle \hat{S}^z \rangle_\psi = \lim_{n \rightarrow \infty} \frac{\sum_n \langle \psi_n | \hat{S}^z | \psi_n \rangle}{n}, \quad (10.18)$$

with the realizations  $|\psi_n\rangle$ . To obtain this behavior, i.e., a single realization, within our QCS, a projection onto an eigenstate of the measured operator is necessary after the measurement. The resulting eigenstate can in our case be deduced from the expectation value of the observable  $\langle \hat{S}^z \rangle$  in combination with a random number  $r \in [0, 1]$ ,

$$\hat{S}^z |\psi\rangle = \begin{cases} -1/2 |1\rangle & \text{if } \langle \hat{S}^z \rangle + 1/2 \geq r \\ 1/2 |0\rangle & \text{otherwise} \end{cases} . \quad (10.19)$$

The corresponding projection operators are  $\hat{\mathbf{O}}_j$  and  $\hat{\mathbf{I}}_j$  as defined in Eq. (10.12). Note that this projection in general changes further qubits as well. If we consider a two-qubit state  $|\phi\rangle = \sqrt{1/2}(|0_00_1\rangle + |1_01_1\rangle)$ , the resulting states after projecting the first qubit and a sequential normalization are

$$\hat{\mathbf{O}}_0 |\phi\rangle = |0_00_1\rangle \quad \hat{\mathbf{I}}_0 |\phi\rangle = |1_01_1\rangle . \quad (10.20)$$

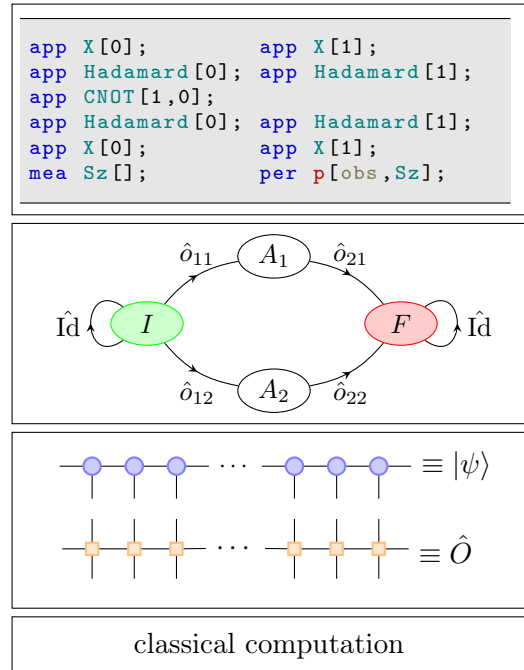
Therefore, due to the nature of quantum mechanics, the second – entangled – qubit changed after applying an operator that was only different from the identity at the first qubit. In the next section, we present the implementation of our QCS and complete the description of the command set.

## Implementation | 10.3

Our implementation is structured in several layers that are sketched in Fig. 10.14. This structure, except the top layer, is already used in the MPSs based ground-state search and time-evolution software. Due to this connection, further improvements in either of these programs will often also improve the others. Furthermore, this model allows us to use the QCS very flexibly because we also have access to all already implemented features, e.g., ground-state search and different bases. Note that states, stored as MPSs, can be saved and loaded in all programs within our framework.

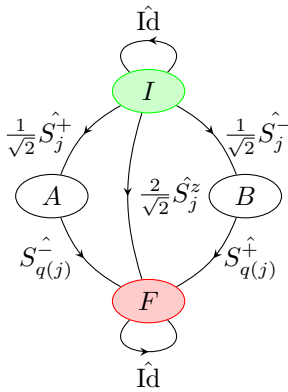
In the following, the layers are described.

**Classical-Computer Layer** The lowest layer represents the classical computer. Our implementation is written in modern C++14/17 and consists of several interacting and related classes. This layer realizes the program flow, the input and output file handling, and the representation of the higher layers onto the hardware. The most expensive operations consist of tensor algebra, which is mapped onto linear-algebra operations and handled by the linear-algebra library SciPAL [Kra13].



**Figure 10.14:** Sketch of the layer model of the QCS. Bottom to top: classical-computer layer, tensor-network layer, finite-state-machine/gate-definition layer, and quantum-code layer.

**Tensor-Network Layer** The next layer consists of the numerical representation of quantum states and operators in terms of tensor networks, namely MPSs and matrix-product operators (MPOs), and all operations with those objects. Therefore, the different algorithms, e.g., the application of an MPO onto an MPS and the time evolutions are implemented directly in the abstract form of tensor networks. The benefit of this abstraction is that, e.g, tensor contractions are implemented and tested in a general way only a few times (for several specific cases), but used very often in different cases. This way the necessary amount of code is reduced and the testability is increased dramatically.



**Figure 10.15:** Graphical representation of the FSM of the Hadamard operator  $\hat{H}_j$ , see Sec. 10.1.2. Operators act only on qubits denoted by the lower index and are zero otherwise.

**Finite-State-Machine Layer** This layer is another abstraction that represents MPOs, e.g., the gates, as  $U(1)$ -invariant FSMs (see chapter 7). Because we want to exploit the  $U(1)$  invariance, i.e., the conservation of the spin quantum number ( $\sum_j \langle S_j^z \rangle = 0$ ), in our implementation but several gates presented in Sec. 10.1 change the total spin of the system an additional auxiliary qubit is connected to every qubit containing the opposite spin. Hence, this  $U(1)$  invariance is achieved by implementing the gates as operators acting on both the system and the auxiliary qubit. The resulting FSM therefore needs an operator with the opposite spin change on the corresponding auxiliary qubit  $q(j)$  for every spin-changing operator on a system qubit  $j$ . In Fig. 10.15 this is shown for the Hadamard operator (see Sec. 10.1.2). Note that the  $\hat{S}_j^z$  operator does not change the spin and therefore does not need a corresponding operator on the auxiliary qubit.

The expansion of the system size decreases the performance of the simulator, but the benefits due to the reduction of the bond dimension usually overcompensates this effect. All these considerations are completely wrapped within this layer and

only need to be considered once new gates are created. Due to this layer and the consequent use of FSMs, it is easy to add new gates and observables, not only for universal quantum computing but also for arbitrary quantum-state manipulations. This is even more the case due to the possibility to choose the basis of the quantum state that should be altered.

**Quantum-Code Layer** The highest layer is the quantum source code with which the final quantum programs are written. Beside the FSMs defining the quantum gates in the previous layer, this layer is the only part that needed to be added to our framework in order to build up a QCS.

Additionally to the already described measurements, our incomplete<sup>1</sup> QASM [Ope, CBSG17] dialect offers the possibility to obtain further information on the current state. To this end, the `perform` instruction understands additional keywords in order to print the corresponding values into the terminal or write them in a file. Those keywords are listed in the following accompanied by an example code after every description.

- `discarded_weight, dw` With this keyword, the discarded weight per bond will be returned. Note that in this case all qubits – system and auxiliary alike – are considered. This is very

<sup>1</sup>Currently (build-hash: fe7158f6d10b2ecd10b2ecdbb98bb42e43b7e38f16a0997), the capability to store classical values in variables and using them for conditional jumps is not implemented.

important information, as it is the truncation error that is made within the MPS framework. Beside this way to obtain the discarded weight, the sum over all discarded weights is stored after every operation for every simulation.

```
per p[discarded_weight];           per w[discarded_weight];
```

- `n_states`, `ns` The number of states reveals the bond dimension at every bond. Again, note that here all qubits are considered. In combination with the discarded weight this parameter is important to pinpoint unexpected behavior in the simulation to a certain qubit.

```
per p[n_states];                   per w[n_states];
```

- `singular_value_spectrum`, `sv_spectrum`, `sv_spec` The singular-value spectrum offers an even deeper look into the underlying MPS and MPO. It is possible to obtain this spectrum either for the current `wavefunction`, then the spectrum right before the last truncation took place is returned, or for any gate `GATENAME`, then the spectrum of this operator is returned. In the second case, all parameters for the gate need to be specified after the `GATENAME`. The results are stored in a file, which has the last parameter as filename. Within this file, every column represents the singular values of a single bond in ascending order. Note that every column has trailing zeros up to the maximal amount of singular values.

```
per p[sv_spec, wavefunction];     per w[sv_spec, wavefunction];
per p[sv_spec, GATENAME, ...];    per w[sv_spec, GATENAME, ...];
```

- `wavefunction`, `wfcn` With this command it is possible to write the current MPS into a file named `sim/wavefunction/IDENTIFIER`. This file can later be used as MPS anywhere within our framework. Note that it is also technically possible to `print` an MPS onto the terminal, but because the data is printed binary it is probably not desirable.

```
// per p[wfcn, IDENTIFIER];       per w[wfcn, IDENTIFIER];
```

In this line, another important feature is introduced, namely the possibility to write comments. Note that a comment always needs to start with `//` and ends with the first semicolon.

- `overlap`, `ov` and `overlaps`, `ovs` In order to calculate the overlap between two MPSs, we need to introduce the new keyword `calculate`, `calc`, or `c`. It is used as another operation that can be performed. Currently, only overlap calculations are implemented. This operation needs at least two parameters, first a specifier for the data type – `double` or `complex` – of the MPSs files that are specified by the following parameters. If only a single file is passed, the overlap with the current state of the quantum-computer simulator is calculated.

All calculated overlaps are stored in an internal array, which can be used analogous to `discarded_weight` and `n_states`.

```
perform calculate[overlap, complex, FILENAME1, FILENAME2];
per p[overlaps];                               per w[overlaps];
```

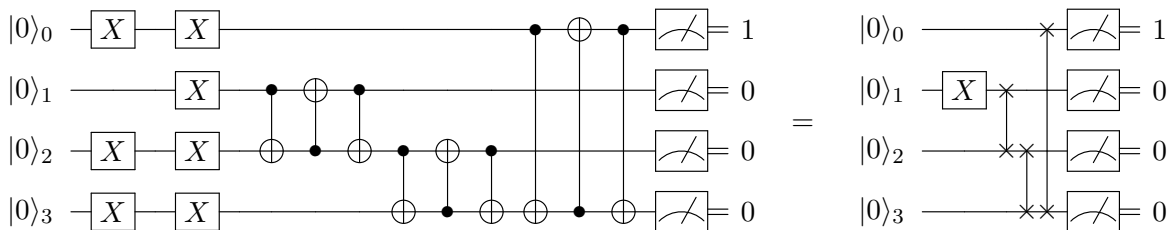
## Illustrative Applications | 10.4

In this section, we present examples of how to use the QCS in the framework of the universal gate model and examples of how to use the software for arbitrary quantum-state manipulation.

### Modular Multiplication Needs No Entanglement? | 10.4.1

In this example, we use a quantum algorithm from [MS12], which calculates the modular multiplication  $f_7(x) = 7x \pmod{15}$ . Instead of using the swap gates directly, we refer to the formulation used in the full user guide of the IBM quantum experience, in which experiment-compatible CNOT gates are used. As described there, the modular multiplication is one key ingredient for Shor's factorization algorithm [Sho97].

In the following, we show the gate model and the source code for our QCS for  $x = 13 = |1101\rangle$  that returns  $|0001\rangle = 1$ .



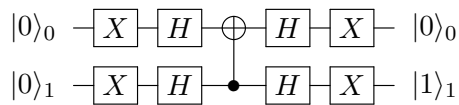
**Figure 10.16:** (Left) gate model of the modular multiplication  $f_7(x)$  with  $x = 13$  as described in the main text. The first column flips the qubits to represent the value  $x$ . The second column flips all qubits. Hence, both columns could be compressed to a single spin-flip on qubit 1. Furthermore, the CNOT gates can be represented by SWAP gates. This compressed program is depicted on the right side.

```
// Initiate qubits with x = 13 = |1101>;
app X[0]; app X[2]; app X[3];
// Do Multi(7,x)Mod15;
app X[0]; app X[1]; app X[2]; app X[3];
app CNOT[1,2]; app CNOT[2,1]; app CNOT[1,2];
app CNOT[2,3]; app CNOT[3,2]; app CNOT[2,3];
app CNOT[0,3]; app CNOT[3,0]; app CNOT[0,3];
// Measure result;
meas Sz[];
perf write[obs,Sz];
```

Because this is a quantum formulation of a classical algorithm, no entanglement is created, and it can easily be computed on a classical computer. This is consistent with [Shi03], which states that X and CNOT are not sufficient for universal quantum computing.

### (Flipped) CNOT Can Change the Control Qubit | 10.4.2

Another small example shows the difference between classical and quantum computation in a very compact way. Consider the following program in which first both qubits are flipped and afterwards a Hadamard gate is applied on both qubits, creating superpositions. Then, a CNOT gate is applied and in the end the inverse of the first two steps is applied.



```

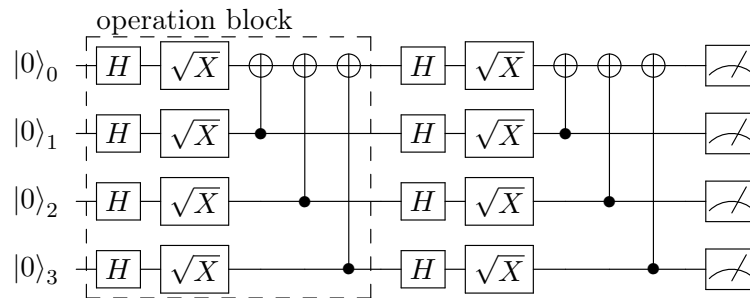
app X[0];           app X[1];
app Hadamard[0];   app Hadamard[1];
app CNOT[1,0];
app Hadamard[0];   app Hadamard[1];
app X[0];           app X[1];
    
```

**Figure 10.17:** Gate model of the code on the right-hand side.

If we look at this program from a purely classical point of view, the only qubit that can change is the target qubit of the CNOT, because the control qubit is not changed by the CNOT and the local gates are self-adjoint. Counter-intuitively, due to the quantum properties of the calculation, the control qubit of the final state is flipped while the target qubit is unchanged. This observation fits together perfectly with the fact that the control and the target qubit of a CNOT gate are flipped by surrounding Hadamard gates [CEP<sup>+</sup>18].

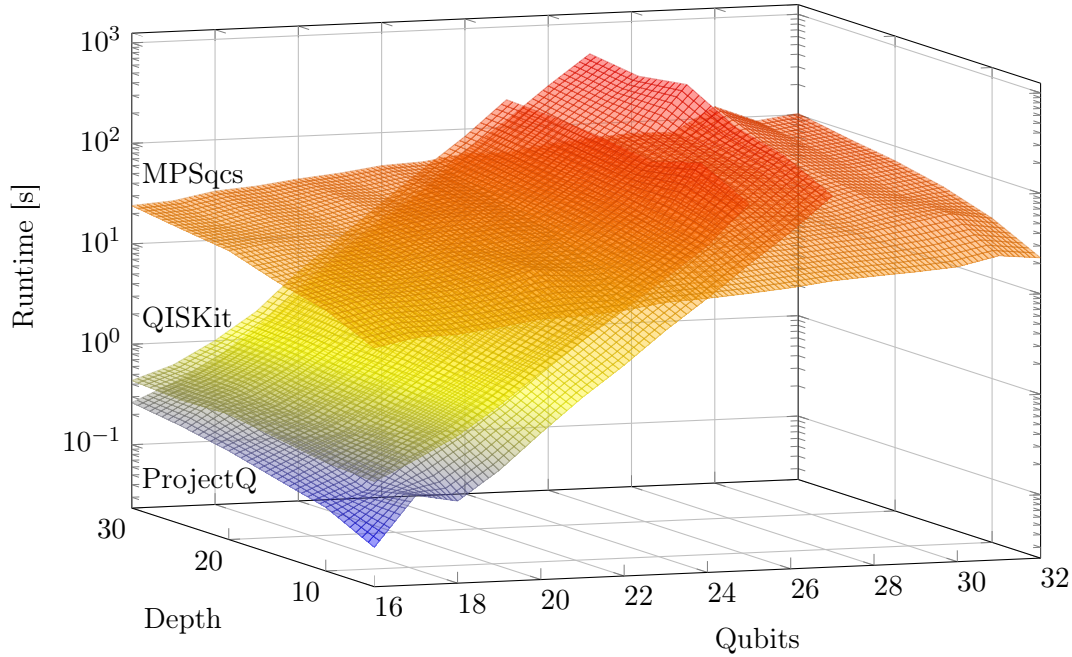
### Benchmark | 10.4.3

One way to benchmark QCS is presented in Ref. [LaR18]. The idea is to build a simply scalable algorithm that uses a variable number of qubits, but does not necessarily need to have a useful result. In Fig. 10.18 the algorithm is depicted for four qubits and a depth of two, i.e., two iterations. In order to scale the number of qubits, more CNOT operations need to be included in addition to the additional qubits. The depth is then varied by repeating the block as often as desired.



**Figure 10.18:** Exemplarily gate model of the circuit that was used for the benchmark. Here for four qubits and a depth of two and with a single operation block highlighted. The algorithm and the graphical representation is adopted from Ref. [LaR18].





**Figure 10.19:** Benchmark results comparing our results (MPSQcs) with runtimes from Ref. [LaR18] (QISKit and ProjectQ) for different numbers of qubits and different depths. As expected, the maximal number of qubits treatable in a decent amount of time is very limited in simulation approaches that do not restrict the Hilbert space. With the MPSQcs the runtimes increase much more slowly with the number of qubits, but faster with the depth of the algorithm.

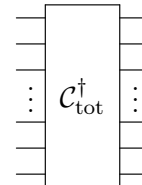
In order to obtain comparable runtimes, we benchmarked our QCS on hardware that is nearly identical to the one in [LaR18], the only difference is the doubled amount of memory.<sup>2</sup> The result of this benchmark is shown in Fig. 10.19. The fact that the runtimes of QISKit and ProjectQ scale exponentially with the number of qubits is due to their representation of the quantum system. Both simulators store the full state without limiting the available Hilbert space, whereas we only store an MPS and hence the runtimes are less affected by the number of qubits. This behavior would be drastically different if the algorithm directly entangles all qubits, because the bond dimension of the MPS would grow exponentially in this case.

## Create Canonical Infinite-Temperature States

### 10.4.4

As an example, in which the QCS is not used in the framework of universal quantum gates, we review the creation of a canonical infinite-temperature state from Sec. 5.4.2.

Most importantly, we need to change the basis in order to work with two spin-1/2 fermions on a single site instead of one. Furthermore, we need to take into account that we want to describe a mixed state and therefore need to double our system size. Then we can use the multi-site gate, which is – except the index labels – equal to the operator Eq. (5.50),



**Figure 10.20:** Pictorial representation of the multi-site gate  $\mathcal{C}_{\text{tot}}^\dagger$  Eq. (5.50).

<sup>2</sup>Hardware specification: Dell XPS 13 (9370), Intel i7-8550U, 16GB RAM, Ubuntu 16.04 LTS.



$$\hat{C}_{\text{tot}}^\dagger = \sum_{j=0}^{L/2-1} \left[ \hat{c}_{\uparrow,2j}^\dagger \otimes \hat{c}_{\downarrow,2j+1}^\dagger + \hat{c}_{\downarrow,2j}^\dagger \otimes \hat{c}_{\uparrow,2j+1}^\dagger \right], \quad (10.21)$$

with  $c_{\sigma,j}^\dagger$  being the fermionic creation operators with spin  $\sigma = \{\uparrow, \downarrow\}$  on qubit  $j$ . The corresponding pictorial representation (Fig. 10.20) spans over the entire system as the operator is not factorized and acts on every qubit.

In order to create a canonical infinite-temperature with a particle density  $\bar{n} = \frac{N}{L}$ , the operator  $C_{\text{tot}}^\dagger$  must be applied  $2N$  times.

The corresponding code is accordingly rather simple:

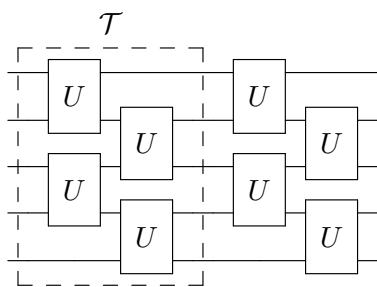
```
app C_dagger_tot [] ;
...
app C_dagger_tot [] ;
```

## Implementing Trotter Time Evolution with Two Site Gates

### 10.4.5

In order to implement the Trotter time evolution as described in Sec. 9.1, a modifiable two-site gate is necessary. This was realized via FSMs (see chapter 7 and App. A.4).

As the time-evolution operator has to obey the quantum-number conservation, not all entries need to be considered nor would the implementation of the FSM evaluation accept them. Nevertheless, the full two site operator `full_two_site_operator[...]` expects  $2d^4 + 1$  arguments, with  $d$  being the local Hilbert-space dimension. Those arguments are the first site  $j$  the operator should act on, and the real and imaginary parts of all entries of the bond matrix. This choice was made in order to gain a consistent operator also in the universal quantum gate model, in which the quantum-number conservation is realized via an auxiliary space and does not need to be obeyed by the gates. The graph representation of the full two site operator for the spin- $1/2$  basis and the Hubbard basis are presented in Apps. A.4.1 and A.4.2, respectively.



**Figure 10.21:** Pictorial representation of two Trotter time evolution steps. Here the simplest version, the first order Trotter decomposition, is shown for clarity.

The last piece that needs to be implemented is the Trotter decomposition itself in order to create the quantum-computer-simulator script with the corresponding arguments. As this is only linear algebra with small matrix dimensions, it can be realized easily (an example implementation is given in App. B.2). In Fig. 10.21 a pictorial representation of the simplest version, the first order Trotter decomposition, is shown. In this case, the time evolution operator is decomposed into two site operators that act on even and on odd bonds. Because all bond operators on even (odd) bonds commute, they can be applied at the same time. Higher-order Trotter decompositions need more gates, but the error due to the decomposition scales with  $\mathcal{O}(\mathcal{T}^{\text{order}+1})$  and hence less time steps are needed.

As explained in Sec. 9.1, this scheme works best for nearest-neighbor interactions unless swap gates are used. In order to expand the given implementation of the Trotter time evolution with swap gates, only the script below needs to be modified.

In the rare case that the bond matrix can be diagonalized symbolically, a smaller operator can be constructed. This was done for the anisotropic spin-1/2 Heisenberg chain by the `trotter_heisenberg[i, J, Jz, dt]` operator. A simple time-evolution step with first-order Trotter decomposition is shown in the following code.

```
app X[5];

meas Sz[];
perf write[obs, Sz];

app trotter_heisenberg[0, 1, 1, 0.01];
app trotter_heisenberg[2, 1, 1, 0.01];
app trotter_heisenberg[4, 1, 1, 0.01];
app trotter_heisenberg[1, 1, 1, 0.01];
app trotter_heisenberg[3, 1, 1, 0.01];
meas Sz[];
perf write[obs, Sz];
```

The corresponding two site gate is given in Eq. (10.22).

$$\text{trotter\_heisenberg}_i(J, Jz, dt) = e_{12} \hat{\mathbf{S}}_i^+ \otimes \hat{\mathbf{S}}_{i+1}^- + e_{21} \hat{\mathbf{S}}_i^- \otimes \hat{\mathbf{S}}_{i+1}^+ \quad (10.22)$$

$$+ e_{00} \hat{\mathbf{0}}_i \hat{\mathbf{0}}_{i+1} + e_{11} \hat{\mathbf{0}}_i \hat{\mathbf{1}}_{i+1} + e_{22} \hat{\mathbf{1}}_i \hat{\mathbf{0}}_{i+1} + e_{33} \hat{\mathbf{1}}_i \hat{\mathbf{1}}_{i+1} \quad (10.23)$$

$$= \begin{pmatrix} e_{00} & 0 & 0 & 0 \\ 0 & e_{11} & e_{12} & 0 \\ 0 & e_{21} & e_{22} & 0 \\ 0 & 0 & 0 & e_{33} \end{pmatrix} \quad (10.24)$$

with

$$e_{00} = e_{33} = \exp\left(-i \frac{Jz}{4} dt\right) \quad (10.25)$$

$$e_{11} = e_{22} = \frac{1}{2} \left[ \exp\left(-i \left(-\frac{J}{2} - \frac{Jz}{4}\right) dt\right) + \exp\left(-i \left(\frac{J}{2} - \frac{Jz}{4}\right) dt\right) \right] \quad (10.26)$$

$$e_{12} = e_{21} = \frac{1}{2} \left[ -\exp\left(-i \left(-\frac{J}{2} - \frac{Jz}{4}\right) dt\right) + \exp\left(-i \left(\frac{J}{2} - \frac{Jz}{4}\right) dt\right) \right] \quad (10.27)$$

These parameters were obtained via mathematica, see App. B.1.

Photoexcitation Dynamics  
in 1D PCMO-like Model  
Systems





The main content of this chapter is published as a regular article in Physical Review B [KRS<sup>+</sup>18]. The parts of the manuscript that are used here were written primarily by me and revised together with my coauthors, hence most parts of the following chapter are directly adopted from the manuscript of Ref. [KRS<sup>+</sup>18].

In this chapter, the dynamics of an one-dimensional (1D) PCMO-like model system (see Sec. 3.3) after a single-polaron photoexcitation (see Sec. 3.4.1), is investigated. We focus our study on the effect of varying the Hund's coupling  $\Delta$  and the interaction strength  $U$  and always compare to the values given by table 3.1. First, the dynamics of the local densities are considered. Afterwards the time evolution of the non-interacting bands and their populations as indicators for the strength of the scattering between the bands, and for time scales emerging in the course of the time evolution are explored.

All results in this chapter were obtained with the methods described in part II, and in particular Sec. 9.2, at quarter filling.

## Local Density Dynamics After a Photoexcitation

## 11.1

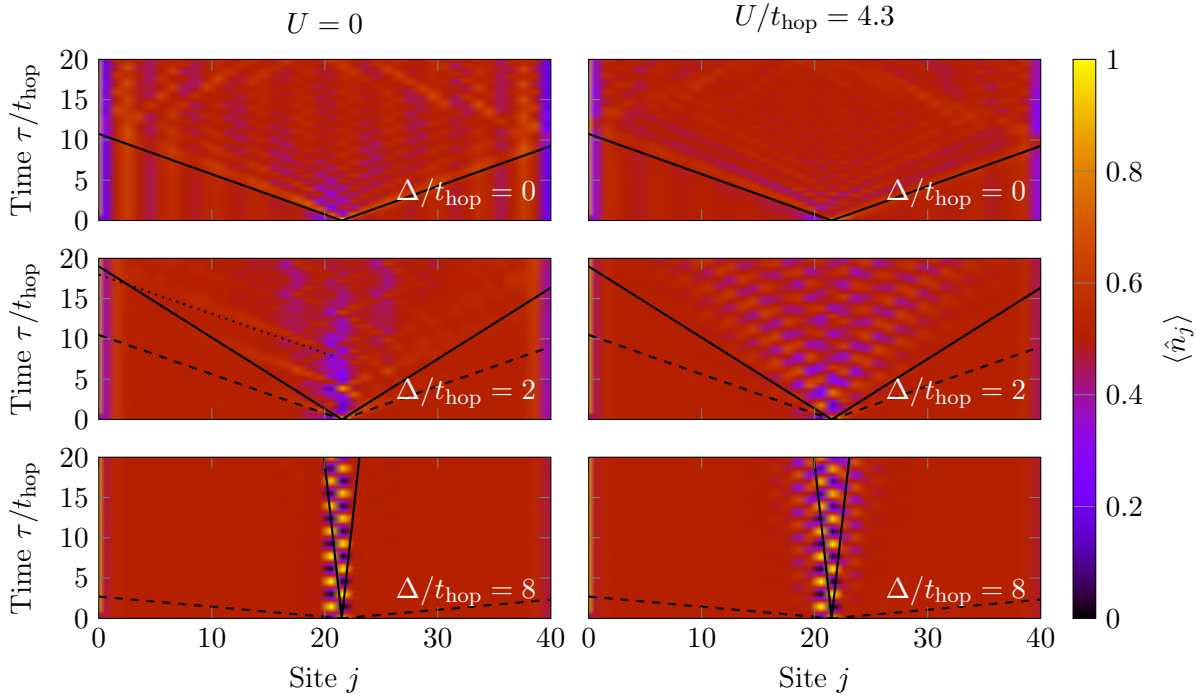
In Fig. 11.1, we show results for the time evolution of the local densities  $\langle \hat{n}_j \rangle$  following an excitation Eq. (3.35). We display results for  $\Delta/t_{\text{hop}} = 0, 2, 8$  and compare the non-interacting cases  $U = 0$  (left panels) to the interacting cases  $U/t_{\text{hop}} = 4.3$  (right panels). The case with  $\Delta/t_{\text{hop}} = 2$  and  $U/t_{\text{hop}} = 4.3$  corresponds roughly to the values of table 3.1, which represents the values for the hypothetical 1D praseodymium-calcium-manganite (PCMO).

Let us start the discussion with the behavior at  $\Delta = 0$ . In the ground state, we observe Friedel-like density oscillations caused by the open boundary conditions used [BBFN98, WAS02]. They are typical for the Luttinger liquid phase [Gia04] realized in the Hubbard chain at this filling [EFG<sup>+</sup>05]. These Friedel-like oscillations are stable and do not change over time.

Furthermore, we see that the local excitation created at the center of the system spreads through the lattice with constant maximum speed. This light cone behavior is captured by the Lieb-Robinson bounds [LR72], which state that in non-relativistic quantum lattice systems with a short-ranged Hamiltonian information spreads with a finite maximal velocity or slower.

### Non-Interacting Case

In this case, for  $U = 0$  and  $\Delta = 0$ , the maximal group velocity allowed by the band structure Eq. (3.30) is the Fermi velocity  $v_F = 2 \frac{t_{\text{hop}} a}{\hbar}$ . In the units used ( $a = t_{\text{hop}} = \hbar = 1$ ), this leads to a slope of 2 in the light cone, which is what is seen in Fig. 11.1 for  $\Delta = U = 0$ . For  $U/t_{\text{hop}} > 0$  and  $\Delta = 0$ , the velocity gets modified by the interaction, but as expected from Luttinger liquid theory [Gia04], the system will always show ballistic motion of the excitation, i.e., it will propagate with a constant maximal velocity through the system.



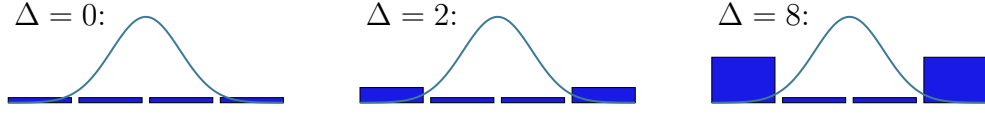
**Figure 11.1:** (Adopted from Fig. 7 in [KRS<sup>+</sup>18]) Time evolution of the local density  $\langle \hat{n}_j \rangle$  following an excitation by applying operator Eq. (3.35) at the center of the system. The panels show tDMRG results for different values of  $\Delta/t_{\text{hop}}$  for chains with  $L = 40$  lattice sites. Left side:  $U = 0$ ; right side:  $U/t_{\text{hop}} = 4.3$ . The solid lines indicate the maximal group velocity of the excited electrons obtained from the non-interacting band structure Eq. (3.30), assuming that one electron gets excited from the first to the second band. The dotted and dashed lines indicate the phase velocity (Eq. (11.3)) at the  $k$ -value with the maximal group velocity (Eq. (11.1)), as discussed in the text.

For finite values of  $\Delta$ , the Friedel-like oscillations disappear. This is expected, since for any finite value of  $\Delta$  a band gap is formed so that the Fermi surface vanishes, and with it the Luttinger liquid phase and the Friedel-like oscillations.

By increasing the value of  $\Delta/t_{\text{hop}}$ , the velocity of the spread of the excitation decreases. For  $U = 0$ , this is expected from the single-particle band structure Eq. (3.30), in which the bands become flatter with increasing  $\Delta/t_{\text{hop}}$ , which also reduces the maximal group velocity.

For the times shown  $\tau/t_{\text{hop}} \leq 20$  (corresponding to  $\sim 23$  fs using the values of table 3.1), for  $\Delta/t_{\text{hop}} = 8$  the speed of the excitation is close to zero, since the group velocities obtained from the band structure are very small already (e.g., the maximal group velocity for an electron excited to the second band is  $v \approx 0.08 \frac{t_{\text{hop}} a}{\hbar}$ ). At the site of the excitation, the dipole-like density oscillations become clearly weaker with time for  $\Delta/t_{\text{hop}} = 2$  as the energy is transferred to the neighboring sites. For the largest Hund's splitting shown,  $\Delta/t_{\text{hop}} = 8$ , the dipole oscillations remain concentrated on the central site on the time scale shown.

Whereas for  $\Delta = 0$  the non-interacting electrons move with the expected Fermi velocity  $v_F = 2 \frac{t_{\text{hop}} a}{\hbar}$ , for the intermediate value  $\Delta/t_{\text{hop}} = 2$ , an additional structure emerges, which is apparently caused by the presence of both, dipole-like oscillations of the electron on the excited Zener polaron and the relatively small tunneling barrier between the polarons. When the electron reaches the boundary between two Zener polarons, it gets partially reflected, but can also partially tunnel to the next polaron (see Fig. 11.2).



**Figure 11.2:** Sketch of an excitation of a polaron and its impact on neighboring polarons for different values of the Hund's coupling  $\Delta$ . Depending on the depth of the potential well, the excitation has less impact on the neighboring polarons.

This happens again for both, the transmitted as well as the reflected part of the electron, when they reach the border to the next polaron, and so on. The result is the intricate pattern seen in Fig. 11.1, in which the excited electron seems to spread through the system in a ping-pong or billiard-like manner for  $U = 0$  and  $\Delta/t_{\text{hop}} = 2$ .

However, now a further effect comes into play, which leads to linear structures with a slope substantially larger than the maximal group velocity allowed by the band structure. A similar situation was discussed in Ref. [CDC<sup>+</sup>18] in the context of interacting Mott insulators: the spread of information through the lattice is governed by the Lieb-Robinson velocity, which can be estimated here as the maximal group velocity determined by

$$v_{g,\nu} = \partial \epsilon_{\nu}(k) / \partial k, \quad (11.1)$$

with  $\nu$  the band, to which the electron is excited. However, as described in Ref. [CDC<sup>+</sup>18, DVS18], if the existence of quasiparticles is assumed, it is possible to express a connected correlation function within the light cone and in its vicinity as

$$G(R, \tau) \propto \frac{\mathcal{F}(k_{\text{sp}})}{(|\nabla_k^2 \epsilon_{k_{\text{sp}}}| \tau)^{D/2}} \cos \left( k_{\text{sp}} R - 2E_{k_{\text{sp},\nu} \tau + \frac{\pi}{4}} \right), \quad (11.2)$$

with  $\mathcal{F}(k_{\text{sp}})$  encoding the correlation function and its effect on the initial and the quasiparticle states,  $k_{\text{sp}}$  denoting the momentum with a stationary phase, at which the maximal group velocity  $v_{g,\nu}$  is realized, and  $\epsilon_{k_{\text{sp},\nu}$  denoting the corresponding energy. Eq. (11.2) does not only define a linear correlation edge with velocity  $2v_{g,\nu}$ , but due to the cosine term, also additional maxima that propagate with  $2v_{p,\nu}$ , which is twice the phase velocity

$$v_{p,\nu} = \epsilon_{\nu}(k_{\text{sp}}) / k_{\text{sp}}. \quad (11.3)$$

Assume the local density described by a square root of a correlation function, then the considerations above lead to a phase velocity that can be substantially larger than the maximal group velocity. This corresponds to what is seen in Fig. 11.1 for  $U = 0$  and  $\Delta/t_{\text{hop}} = 2$ : The excitation causes linear structures, whose slope is in excellent agreement with the maximal phase velocity obtained from the band  $\nu = 2$  in Eq. (3.30). However, the structure is seen to be strong only as long as it is within or close to the light cone, which is obtained from the maximal group velocity determined from  $\epsilon_2(k)$  in Eq. (3.30). As soon as they reach the border of the light cone, their amplitude decays quickly, so that they do not contribute to the spread of information through the lattice. This happens with the group velocity, as expected from the Lieb-Robinson theorem.

### Interacting Case

In the presence of a repulsive  $U$ , it is unclear whether the ballistic transport will prevail, or if the inter-particle scattering might change its speed, e.g., inhibiting transport by slowing down the

spreading of the excitation, or enhancing transport by increasing its velocity. Also, it is possible that transport at finite  $U$  could change its nature from ballistic to diffusive.

The right side of Fig. 11.1 shows results for  $U/t_{\text{hop}} = 4.3$ . For  $\Delta = 0$ , as discussed above, the ballistic motion prevails, as expected for a Luttinger liquid. At finite  $\Delta/t_{\text{hop}}$ , however, the behavior changes significantly when comparing to the corresponding  $U = 0$  cases: At  $\Delta/t_{\text{hop}} = 2$ , the ping-pong like structure disappears and is replaced by a more diffuse looking behavior. This is captured by the following scenario: Due to the rather strong interaction, the electron scatters as soon as it tunnels to the neighboring Zener polaron, since there the electron is of opposite spin, so that the Hubbard term comes into play. This scattering induces on the one hand a dipole oscillation also on this Zener polaron, and on the other hand a partial tunneling of the electron of opposite spin to the neighboring lattice site. There, the mechanism repeats, and again a dipole-like oscillation also on this Zener polaron is excited, and partial tunneling of the electron with opposite spin direction to the further Zener polaron is induced, and so on. The resulting picture is a sequence of dipole oscillations formed on each Zener polaron, with an amplitude decreasing the further one moves away from the site of the excitation. This sequence of dipole oscillations seems to replace the ping-pong pattern observed at  $U = 0$ .

It is difficult to judge whether the motion of the original excitation through the system remains ballistic, or if it might change nature. However, the strongest features are deep inside the light cone prescribed by the group velocity of the non-interacting system and seem to move with a smaller velocity, or in a diffusive manner.

Also at large  $\Delta/t_{\text{hop}} = 8$ , the effect of a finite value of  $U$  is significant: While at  $U = 0$ , on the time scales shown, there was essentially no spread of the excitation to the neighboring sites, with finite interaction the dynamics is clearly composed of the dipole oscillation on the excited polaron, plus additional dipole oscillations on the close lying neighboring polarons. Again, it is difficult to conclude whether transport might be diffusive or ballistic. We leave this aspect for future research.

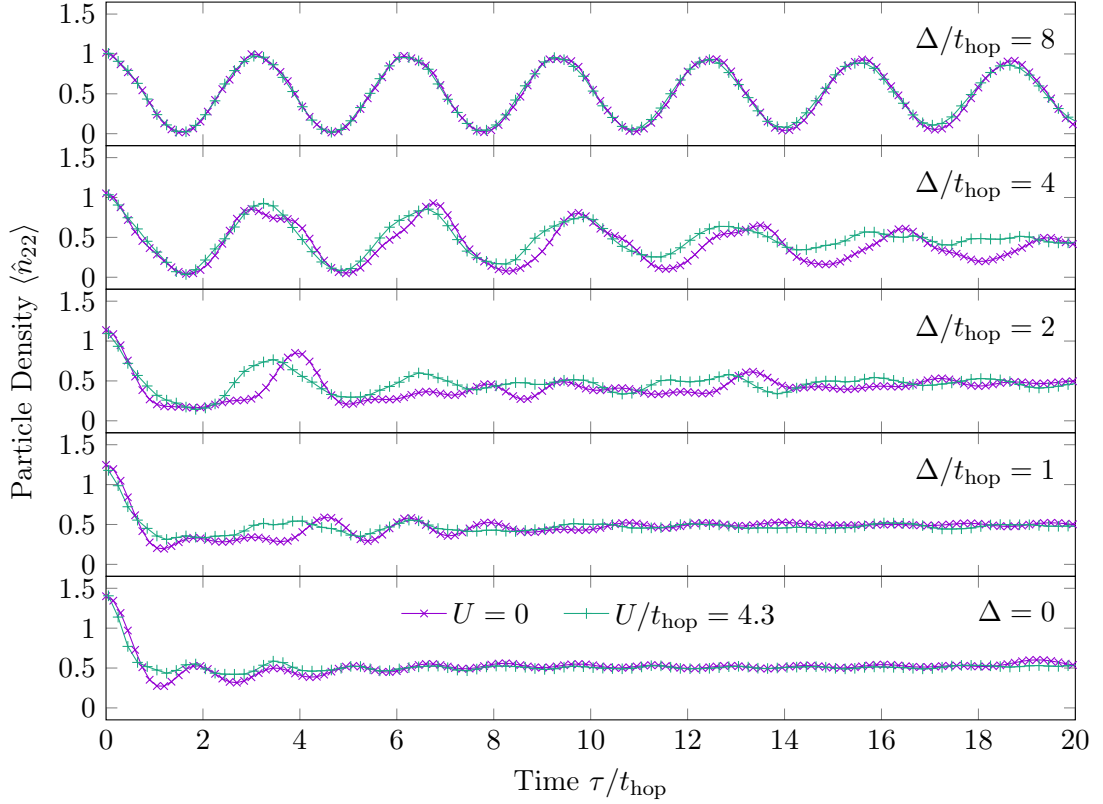
We complement this discussion by considering the time evolution of the local density  $\langle \hat{n}_j \rangle$  on the excited polaron in more detail.

In Fig. 11.3, we show results at  $U/t_{\text{hop}} = 0$  and 4.3 for the different values of  $\Delta/t_{\text{hop}}$  indicated there. In contrast to the different behavior seen in Fig. 11.1 when comparing the results for  $U = 0$  to the ones for  $U/t_{\text{hop}} = 4.3$ , in all cases shown and on the time scale displayed, the time evolution on the site of the excitation is qualitatively similar with and without interaction. On the time scale shown, three different types of behavior seem to exist: For  $\Delta/t_{\text{hop}} = 8$  the value of the local density shows a coherent oscillation for all times shown  $\tau/t_{\text{hop}} \leq 20$  (corresponding to  $\sim 23$  fs using the values of table 3.1). The amplitude of this oscillation decays only slowly. The reason for this slow decay seems to be based in the fact that the construction of Zener polarons act essentially as potential well for the excitation (see Figs. 11.1 and 11.2). As the group velocity for the excitation moving away from this place is so small in this case, the dipole oscillations decay only slowly. For the local density, the effect of  $U$  is to weakly dampen its oscillation.

In the other extreme case displayed ( $\Delta = 0$ ), the coherent oscillation of the local density is completely suppressed, and the value of the local density drops very quickly to the equilibrium value 0.5 and then shows only tiny oscillations around this value. This drop happens on a time scale  $\tau < 10\hbar/t_{\text{hop}}$ , corresponding to  $\sim 11$  fs using the parameters of table 3.1. The reason for this is that the excitation moves freely through the system. This is also true at finite  $U$ , where the system is in a Luttinger liquid phase [Gia04].

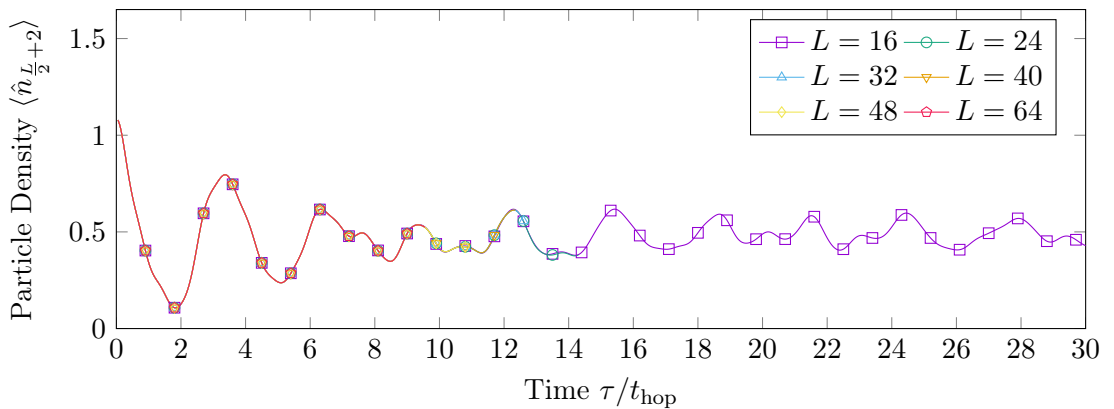
For intermediate values of  $\Delta/t_{\text{hop}}$ , the time evolution of the local density on this time scale  $\lesssim 30$  fs reflects both aspects: At short times, coherent oscillations are seen, which are indicative





**Figure 11.3:** (Adopted from Fig. 8 in [KRS<sup>+</sup>18]) Time evolution of the local density  $\langle \hat{n}_{i+1} \rangle$  for a system with  $L = 40$  lattice sites at the right site of the Zener polaron at which operator Eq. (3.35) was applied to,  $i = L/2 + 1$ . Green:  $U = 0$ ; purple:  $U/t_{\text{hop}} = 4.3$ . The lines for  $\Delta/t_{\text{hop}} = 8$  show a fit using a function of the form  $f(x) = \frac{1}{4} (\cos(ax) + \cos(bx)) + \frac{1}{2}$ .

for the dipole oscillation of the excited electron, while at later times the local density relaxes to its equilibrium value of 0.5, since the excitation then is spreading through the system.



**Figure 11.4:** (Adopted from Fig. 8 in [KRS<sup>+</sup>18]) Time evolution of the local density  $\langle \hat{n}_{j+1} \rangle$  after the operator Eq. (3.35) was applied to  $j = L/2 + 1$  for  $\Delta/t_{\text{hop}} = 2.3$  and  $U/t_{\text{hop}} = 4.3$ , which is close to the parameters of table 3.1. The plot compares tDMRG results for systems with  $L = 16$  (violet boxes),  $L = 24$  (green circles),  $L = 32$  (blue up-pointing triangles),  $L = 40$  (orange down-pointing triangles),  $L = 48$  (yellow diamonds), and  $L = 64$  (red pentagons). The results displayed are obtained with matrix-product state (MPS) matrix dimension  $m_{\text{max}} = 5000$ .

The amplitudes of the oscillations around the equilibrium value are larger than for  $\Delta = 0$  and do not depend on the system size, as can be seen in Fig. 11.4, so that finite size effects seem to be excluded as cause for this behavior.

## Electronic Momentum-Distribution Function | 11.2

In this section, we present the time evolution of the momentum distribution at short times, which we obtain from the dynamics of the electronic one-particle reduced density matrix

$$\varrho_{\sigma,j,j'}(\tau) = \langle \hat{c}_{\sigma,j}^\dagger \hat{c}_{\sigma,j'} \rangle(\tau). \quad (11.4)$$

The time evolution of the momentum distribution is obtained by Fourier-transforming the one-particle reduced density matrix by projecting onto the four bands of the non-interacting system. The momentum distribution of each band  $\nu \in \mathbb{B} = \{1, 2, 3, 4\}$  is obtained by the corresponding transformation of the creation and annihilation operators, leading to

$$n_{\sigma,\nu}^{\text{el}}(k, \tau) = \sum_{\substack{l,l' \in \mathbb{Z} \\ j,j' \in \mathbb{B}}} e^{i2\pi k(l-l')} T_{\sigma,\nu,j}^*(k) \varrho_{\sigma,4l+j,4l'+j'}(\tau) T_{\sigma,\nu,j'}(k), \quad (11.5)$$

where the tensors  $T_{\sigma,\nu,j}(k)$  are unitary and hold the eigenvectors of the Hamiltonian of a single unit cell, as derived in detail in [Bie16, KRS<sup>+</sup>18].

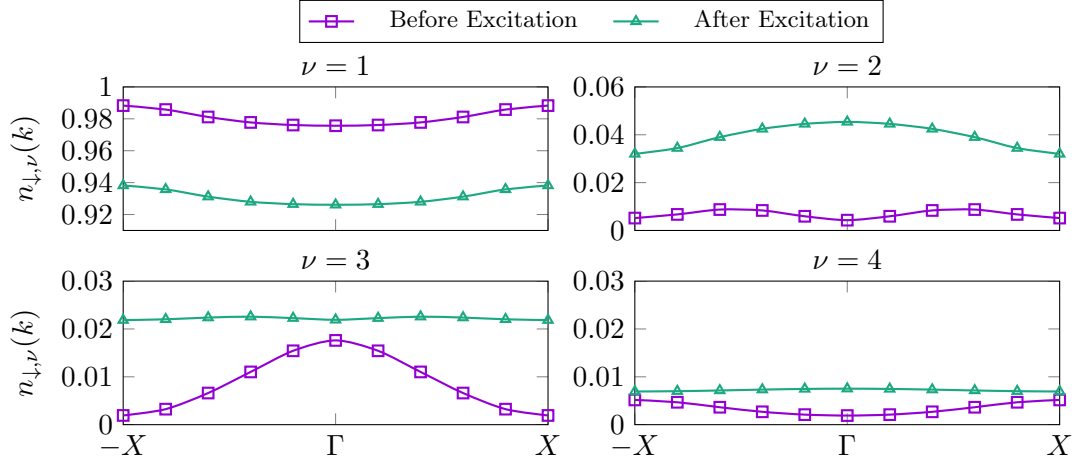
### Momentum-Distribution Function of the Ground State and the Excited State

In Fig. 11.5, we compare the momentum distribution of the ground state with the one obtained directly after the excitation. The systems are excited by applying operator Eq. (3.35) to the central site of the system. Note that the excitation affects predominantly one spin direction, which is due to the spin polarization of the polaron on which the excitation takes place. Hence, we show only the results for spin down.

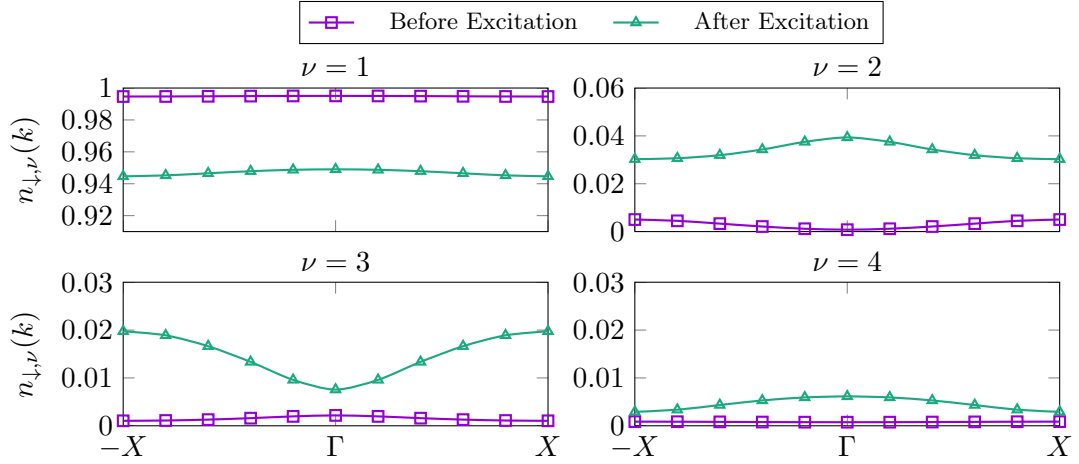
Let us first discuss the momentum distribution of the ground state. We expect at quarter filling that the first band  $\nu = 1$  is highest populated, and the population of the higher bands is negligibly small but finite since  $U/t_{\text{hop}} > 0$ . Note that at  $U/t_{\text{hop}} = 4.3$  (s. Fig. 11.5a) the populations are slightly inverted, so that the momentum distribution at  $k = 0$  is somewhat smaller than at finite  $k$ . We associate this effect to the projection onto the non-interacting band structure. In Fig. 11.5b, we show the momentum distribution function for the non-interacting case. The difference to our expectation decreases but there is still a small occupation of the higher bands in the ground state. We attribute these to the choice of the boundary conditions as the expectations match perfectly in Fig. 11.5c, where we present results for the same parameters as in Fig. 11.5b, but with periodic boundary conditions.

The photoexcitation, as modeled in Eq. (3.35), moves particles from the lowest band to the higher ones. As we model it as strongly localized in real space, the excitation here transfers all possible momenta in contrast to light, for which the momentum transfer is negligible. For  $\Delta/t_{\text{hop}} = 2.3$ , the second and third band get a higher population, whereas the one of the fourth band remains very small. For the largest value of the Hund's splitting,  $\Delta/t_{\text{hop}} = 8$  treated in the previous section, the most affected band is the second one; the population of the two highest bands remains very small. Hence, the lowest band  $\nu = 1$  is highest populated in the ground state and remains highest populated also after the excitation in all cases treated here.

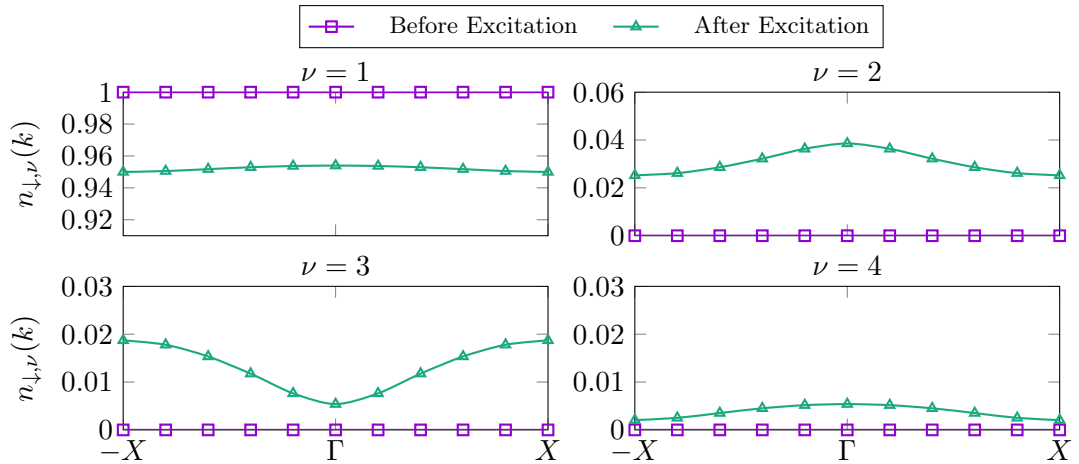
(a) Interacting case, open boundary conditions



(b) Non interacting case, open boundary conditions



(c) Non interacting case, periodic boundary conditions



**Figure 11.5:** (Adopted from Figs. 10, 15, and 16 in [KRS<sup>+</sup>18]) Momentum distribution for a system with  $L = 40$  and  $\Delta/t_{\text{hop}} = 2.3$  before (magenta) and just after (green) the photoexcitation by applying operator Eq. (3.35) at the center of the system. (a)  $U/t_{\text{hop}} = 4.3$  and (b)  $U/t_{\text{hop}} = 0.0$  with open boundary conditions. (c)  $U/t_{\text{hop}} = 0.0$  with periodic boundary conditions.

Note that the effect of the excitation is independent of the boundary conditions used. We see that for  $\Delta/t_{\text{hop}} = 2.3$  at  $U = 0$  particles are excited from the lowest band to all higher bands. The resulting distributions show a peak at the  $\Gamma$ -point in the first, second, and fourth band, while in the third band a minimum is obtained. We attribute the differences to this behavior visible in Fig. 11.5a to the effect of a finite value of  $U/t_{\text{hop}}$ .

### Momentum Distribution as Function of Time

Due to the finite value of  $U/t_{\text{hop}}$ , we expect the electrons to scatter so that the population of the four bands changes in time. In Figs. 11.6a and 11.6b, we show the time evolution of the populations of each of the four bands with time for  $\Delta/t_{\text{hop}} = 2.3$  and  $\Delta/t_{\text{hop}} = 8$ , respectively. Additionally, we show the population of the thermal state, which is obtained in Sec. 11.3, as horizontal lines.

Clearly, scattering between the bands takes place. In contrast to the time evolution of the local densities treated in Sec. 11.1, the band populations in Fig. 11.6 are indicative for bulk behavior and hence are better suitable to look for time scales on which the excitation evolves.

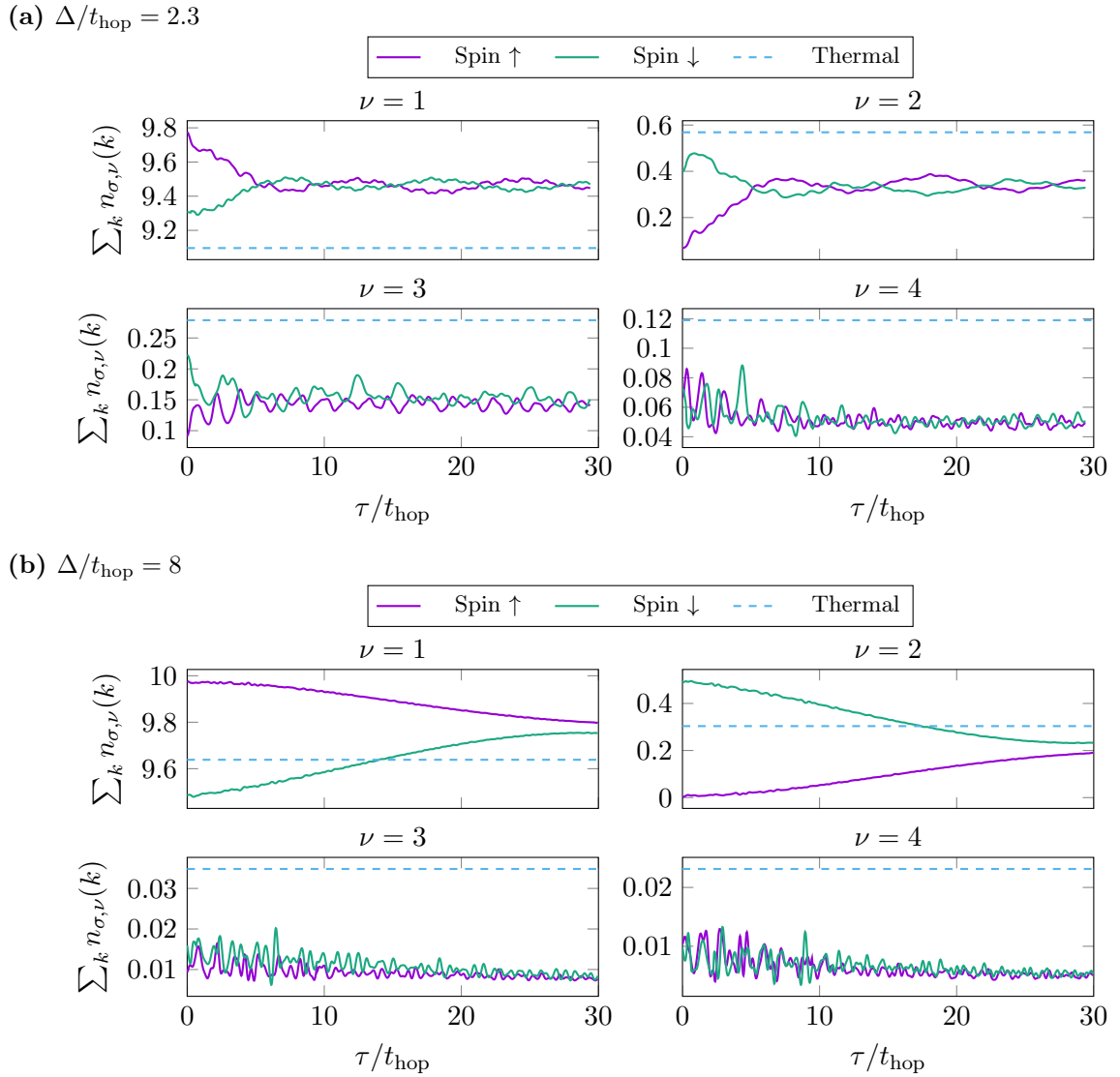
As can be seen in Fig. 11.6a for  $\Delta/t_{\text{hop}} = 2.3$ , the populations of the first and second band seem to relax to a stationary value of  $\sim 9.45$  and  $\sim 0.35$  on a time scale of  $\sim 5t_{\text{hop}}$  (corresponding to  $\sim 6$  fs using the parameters of table 3.1). The populations of both spin directions relax to the same value and afterwards show rather small oscillations around these values. Similar behavior is also seen in the third and fourth band.

For  $\Delta/t_{\text{hop}} = 8$ , instead, relaxation happens only at a time  $\tau > 30t_{\text{hop}}$ . The first two bands seem to reach a population of  $\sim 9.8$  and  $\sim 0.2$ , respectively. The third and fourth band have very small populations. The population of both spin directions seems to relax to the same value, even though at  $\tau = 0$  they significantly differ.

Since the spin moment in each band is determined by the difference of the populations of both spin directions, Fig. 11.6 shows that its value is close to zero after a time scale, which is larger when increasing  $\Delta$ , for a fixed value of  $U/t_{\text{hop}}$ . This can be expected, since the larger the value of  $\Delta/t_{\text{hop}}$ , the smaller the tunneling probability between neighboring polarons (s. Fig. 11.2), and hence also the probability of two electrons with opposite spin to scatter with each other.

As seen in Fig. 11.6, the spin moment inside the bands seems to relax on a short time scale  $\lesssim 50$  fs. Furthermore, the tDMRG results indicate that the band occupations of the first band seem to relax to expectation values, which agree with the thermal expectation values up to a few percent. Particle number conservation then leads to a difference of the band occupations in the other bands of similar absolute magnitude. This discrepancy can be due to the choice of boundary conditions and finite size effects, so that the results seem to be in good agreement with the corresponding thermal state. These results indicate that the relaxation time increases with the value of  $\Delta/t_{\text{hop}}$ .

However, it is still possible that further aspects can become important for the lifetimes of the excitations. The question arises, if one can make a quantitative prediction for the lifetime of the excitation in the presence of  $U$  and  $\Delta$  also in cases that are not amenable to the tDMRG. As much longer times are barely accessible to the tDMRG, we therefore refer to the linearized-Boltzmann-equation treatment in [Bie16] and [KRS<sup>+</sup>18], which is suitable to extract lifetimes of the excitations. Comparing the results of this treatment to the tDMRG results, we find comparable time scales.



**Figure 11.6:** (Adopted from Figs. 11 and 12 in [KRS<sup>+</sup>18]) Time evolution of the population of each band following the photoexcitation for  $U/t_{\text{hop}} = 4.3$  and (a)  $\Delta/t_{\text{hop}} = 2.3$  and (b)  $\Delta/t_{\text{hop}} = 8$ . Additionally, the population of the thermal state is given by the horizontal lines.

## Final Temperature of the Excited State | 11.3

We use techniques introduced in Sec. 5.4.1 to compute the properties of the equilibrium state at finite temperatures. In particular, we deploy an entangler Hamiltonian and an imaginary-time evolution. In our MPS implementation, we need to formulate the long-range-interaction Hamiltonian Eq. (5.47) as a finite-state machine (FSM) (see chapter 7). In order to do so, we need to rewrite the projector Eq. (5.49), because it is not possible to evaluate the absolute value

$E_{\text{exc}}/t_{\text{hop}}$	$U/t_{\text{hop}} = 0$	$U/t_{\text{hop}} = 4.3$
$\Delta/t_{\text{hop}} = 2.3$	-47.302	-45.354
$\Delta/t_{\text{hop}} = 8$	-100.90	-100.55

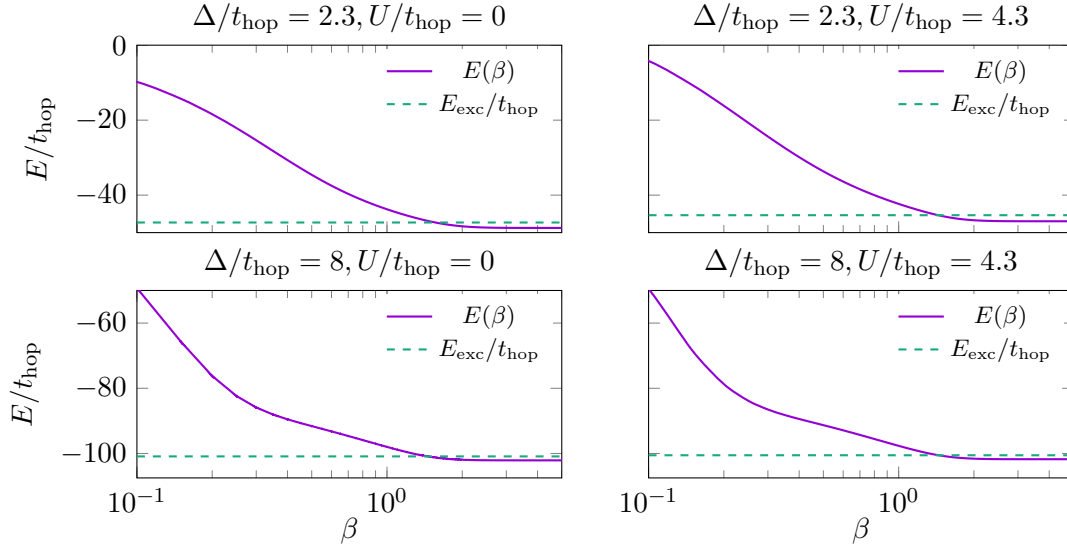
**Table 11.1:** (Published as Tab. IV in [KRS<sup>+</sup>18]) Total energy of the excited states for the given values of  $U/t_{\text{hop}}$  and  $\Delta/t_{\text{hop}}$  for a system with  $L = 40$ .

of a sum of operators in the framework of FSMs. This leads to

$$\begin{aligned}
\hat{P}_j^\sigma &= |1 - \hat{n}_{\bar{\sigma},p(j)} - \hat{n}_{\sigma,q(j)}| = (\hat{P}_j^\sigma)^2 \\
&= 1 - 2\hat{n}_{\bar{\sigma},p(j)} - 2\hat{n}_{\sigma,q(j)} + \hat{n}_{\bar{\sigma},p(j)}\hat{n}_{\sigma,q(j)} + \hat{n}_{\sigma,q(j)}\hat{n}_{\bar{\sigma},p(j)} + \hat{n}_{\bar{\sigma},p(j)}^2 + \hat{n}_{\sigma,q(j)}^2 \\
&= 1 - \hat{n}_{\bar{\sigma},p(j)} - \hat{n}_{\sigma,q(j)} + 2\hat{n}_{\bar{\sigma},p(j)}\hat{n}_{\sigma,q(j)}.
\end{aligned} \tag{11.6}$$

The corresponding FSM for the entangler is presented in App. A.1.

In Fig. 11.7 the imaginary time evolution starting from an infinite temperature state obtained as ground state of Eq. (5.47) is shown for different values of  $\Delta/t_{\text{hop}}$  and  $U/t_{\text{hop}}$ .



**Figure 11.7:** (Adopted from Fig. 17 in [KRS<sup>+</sup>18]) Total energy of systems with  $L = 40$ ,  $\Delta/t_{\text{hop}} = 2.3, 8$ , and  $U/t_{\text{hop}} = 0, 8$  as a function of the inverse temperature  $\beta$ . The results are obtained by an imaginary-time evolution starting from the ground state of Eq. (5.47), which is a suitable state with  $\beta = 0$  in the physical space. The dashed horizontal lines indicate the total energy after the excitation, which is obtained by computing the expectation value  $E_{\text{exc}} = \langle \psi(\tau) | \hat{H} | \psi(\tau) \rangle = \text{const.}$ , where  $\hat{H}$  is the Hamiltonian Eq. (3.27) and  $|\psi(\tau)\rangle$  the state after the excitation at time  $\tau$ . The intersection of the finite-temperature results and  $E_{\text{exc}}$  indicates the value of  $\beta$ , which can be attributed to the energy of the excitation.

In thermal equilibrium, the value of  $E(\beta)$  shown in Fig. 11.7 and of the energy of the excited state  $E_{\text{exc}} = \langle \psi(\tau) | \hat{H} | \psi(\tau) \rangle = \text{const.}$  is the same for the Hamiltonian Eq. (3.27). Hence, the value of  $\beta$  for which  $E(\beta) = E_{\text{exc}}$  corresponds to the temperature of the system after equilibration. The values for  $E_{\text{exc}}$  are displayed in table 11.1, the corresponding values of  $\beta$  are shown in table 11.2.

$\beta$	$U/t_{\text{hop}} = 0$	$U/t_{\text{hop}} = 4.3$
$\Delta/t_{\text{hop}} = 2.3$	1.56	1.43
$\Delta/t_{\text{hop}} = 8$	1.45	1.44

**Table 11.2:** (Published as Tab. V in [KRS<sup>+</sup>18]) Inverse temperature  $\beta$  at which  $E(\beta) = E_{\text{exc}}$  for the given values of  $U/t_{\text{hop}}$  and  $\Delta/t_{\text{hop}}$  for a system with  $L = 40$ .

## Estimation of the Energy Density of the Excitation | 11.4

We estimate the energy density of the excitation by considering the difference of the energy of the excited state to the ground state,  $E_{\text{exc}} - E_0$  and dividing it by the length of the finite system used in our simulations. In this approach, we assume that the finite system considered represents a typical part of the lattice, which is excited by the incoming light, so that the energy density of the finite system would correspond to the one of an infinite lattice.

Furthermore, we assume that the intensity  $I$  of the incoming light amounts to the same energy density. We hence obtain

$$I = \frac{E_{\text{groundstate}} - E_{\text{excited state}}}{a L \tau}. \quad (11.7)$$

We use  $a = 3.818 \text{ \AA}$  as value for the Mn-Mn distance, see [JKŠ<sup>+</sup>85]. The duration of the light pulse is estimated to be  $\tau = 1 \text{ fs}$ . The values for the ground-state energies are given in table 11.3 and are obtained via DMRG for chains with  $L = 40$  sites.

$E_{\text{gs}}/t_{\text{hop}}$	$U/t_{\text{hop}} = 0$	$U/t_{\text{hop}} = 4.3$
$\Delta/t_{\text{hop}} = 2.3$	-48.753	-46.973
$\Delta/t_{\text{hop}} = 8$	-102.11	-101.76

**Table 11.3:** (Published as Tab. VI in [KRS<sup>+</sup>18]) Total energy of the ground states for the given values of  $U/t_{\text{hop}}$  and  $\Delta/t_{\text{hop}}$  for a system with  $L = 40$ .

This leads to an intensity of  $\sim 10^8 \text{ W/mm}$ . In a pump-probe setup, this would be the intensity of the pump laser in case of perfect absorption of the pump pulse. This value hence serves as a lower bound for the intensity needed to reproduce a scenario similar to the one discussed in this chapter. As the intensity of lasers with ultrashort light pulses can reach  $\sim 10 \text{ TW/mm}$ , the estimate shows that similar investigations are within reach of typical pump-probe setups.

In the next chapter, we investigate the same model, but a totally different type of excitation. In order to model a “more realistic” representation of a photoexcitation, we apply a Peierls substitution, as discussed in Sec. 3.4.1.





## Broad, Continuous, and Intensive Excitation

12

The main content of this chapter is online available as a preprint [KPM18]. The parts of the manuscript that are used here were written primarily by me and revised together with my coauthors, hence most of the following chapter is directly adopted from the manuscript of Ref. [KPM18]. Parts that are not primarily written by me are marked with an additional quote.

In this chapter, we describe the behavior of the hypothetical one-dimensional (1D) praseodymium-calcium-manganite (PCMO) model (see Sec. 3.3) after several types of photoexcitations that are modeled by the Peierls substitution (see Sec. 3.4.1). We focus on the formation of a metastable charge-density pattern due to a spin-selective Peierls pulse and possible decay channels.

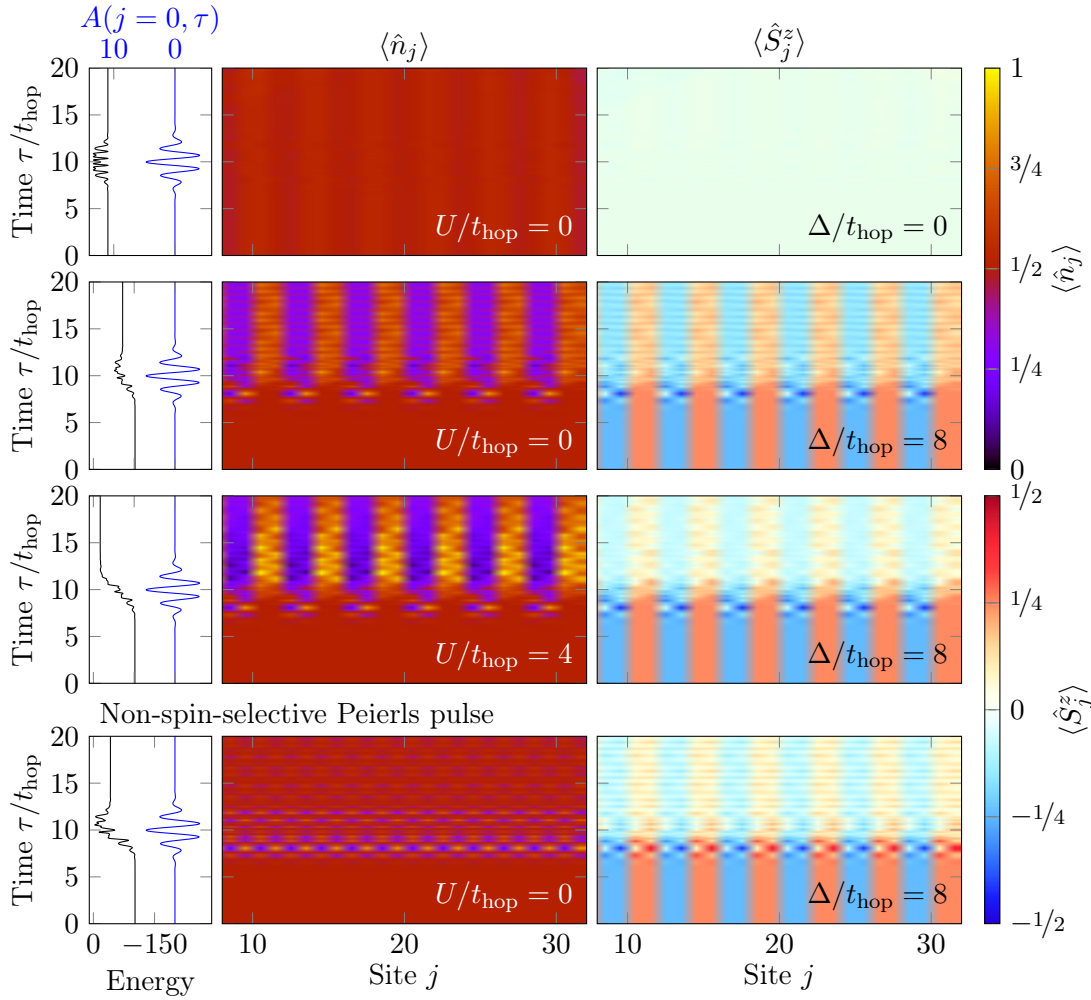
All results in this chapter were obtained with the methods described in part II, and in particular Sec. 9.2.

### Periodic Charge-Density Patterns Through Spin-Selective Photoexcitation

12.1

We first discuss the time evolution of the total energy of the system and of local observables in real space, in particular the particle and spin densities  $\langle \hat{n}_j \rangle(\tau)$  and  $\langle \hat{S}_j^z \rangle(\tau)$ , respectively. In Fig. 12.1, the first two rows display results at  $U/t_{\text{hop}} = 0$ , and  $\Delta/t_{\text{hop}} = 0$  and 8, which show the generic behavior when applying a spin-selective photoexcitation. In the top panel of Fig. 12.1 ( $U/t_{\text{hop}} = \Delta/t_{\text{hop}} = 0$ ), the ground state exhibits Friedel-like oscillations in the particle density [Gia04, BBFN98] but its overall time evolution is only weakly affected by the pulse and in particular there is no enhanced charge ordering. Furthermore, we note that there is no visible energy absorption after the pulse has passed. We attribute these observations to our choice of parameters, for which the photoexcitation has approximately no site dependence throughout the system, so that no significant change in the local observables can be expected. For finite values of  $\Delta$ , a gap opens, so that the Friedel-like oscillations are strongly suppressed, leading to a constant charge density in the bulk of the system. In contrast, the ground state shows a periodic pattern in the local spin densities  $\langle S_j^z \rangle$ , which follows the magnetic microstructure. A finite amount of energy is absorbed by the system from the pulse, leading to a highly excited state. The values of the local observables are significantly modified and remain time-dependent also after the pulse has left the system. One of the main results of our work is that a periodic pattern in the charge density is induced, which follows the periodicity of the magnetic microstructure, and is very stable on the time scales treated here, also at finite values of  $U$ , as shown in the third panel of Fig. 12.1.

The formation of this CDW-like pattern occurs together with the weakening of the spin pattern. In the bottom panel of Fig. 12.1, we present the time evolution with the photoexcitation coupling to both spin directions ( $\Delta/t_{\text{hop}} = 8$ ). Even though a significant energy absorption takes place, no stable pattern is obtained. In this case, we also observe a weakening of the spin pattern. However, this weakening does not suffice to create the CDW-like pattern; the spin-polarized excitation is



**Figure 12.1:** (Published in [KPM18]) Time evolution of system Eq. (3.36) with  $L = 40$  sites from tDMRG at quarter filling induced by a pump pulse as discussed in the text. First column: total energy of the system (black) and the modulation of the vector potential (blue). Second column: particle density  $\langle \hat{n}_j \rangle(\tau)$  in the bulk (sites 8-32). Third column: local magnetizations  $\langle \hat{S}_j^z \rangle(\tau)$ , also only in the bulk. The color bars on the right indicate the values for  $\langle \hat{n}_j \rangle(\tau)$  and  $\langle \hat{S}_j^z \rangle(\tau)$ , respectively. The top row shows results for an excitation acting only on spin-down particles in the absence of a magnetic structure,  $\Delta = 0$ , and without interaction,  $U/t_{\text{hop}} = 0$ . The second row shows results for the same excitation, but with  $\Delta/t_{\text{hop}} = 8$  and  $U/t_{\text{hop}} = 0$ . In contrast, the third row shows results for the same excitation and also  $\Delta/t_{\text{hop}} = 8$  but  $U/t_{\text{hop}} = 4$ . The bottom row shows results for an excitation acting on both spin directions for  $\Delta/t_{\text{hop}} = 8$  and  $U/t_{\text{hop}} = 0$ .

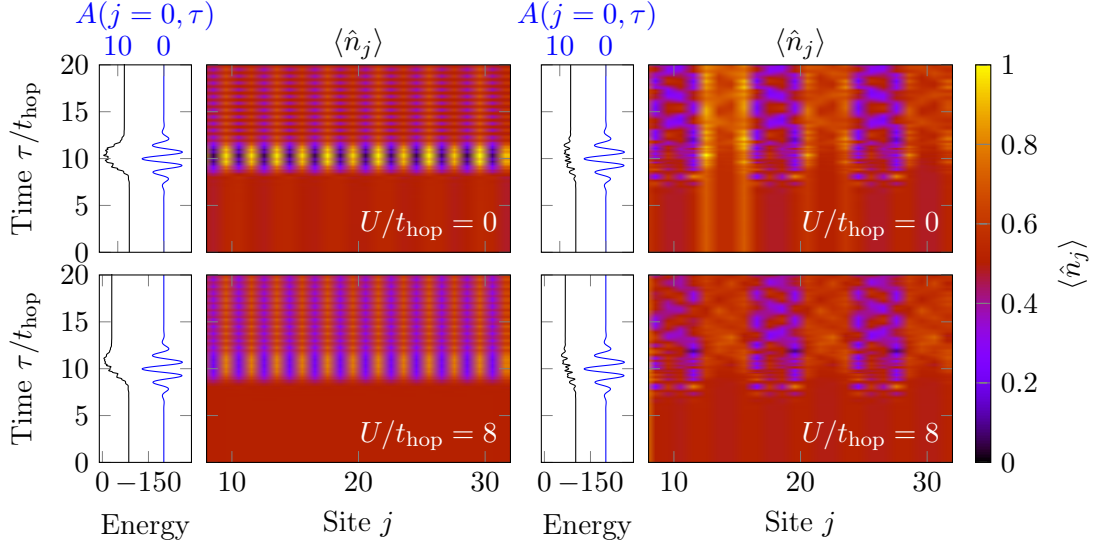
essential. Similar behavior is also obtained when changing the magnetic microstructure, e.g., by using a larger unit cell, as discussed next.

## CDW-Pattern for Various Unit Cells and Fillings

### 12.1.1

As an example that a charge-density wave (CDW) pattern also occurs for other unit cells of the magnetic microstructure we show in Fig. 12.2 the particle density for a system in which  $\Delta_j$  has

a periodicity of 8 instead of 4, i.e.,  $\Delta_j = (\Delta, \Delta, \Delta, \Delta, -\Delta, -\Delta, -\Delta, -\Delta)$ . All other parameters are similar to the main results, i.e.,  $\Delta/t_{\text{hop}} = 8$ ,  $L = 40$ , and  $U/t_{\text{hop}} = 0$  and 8.



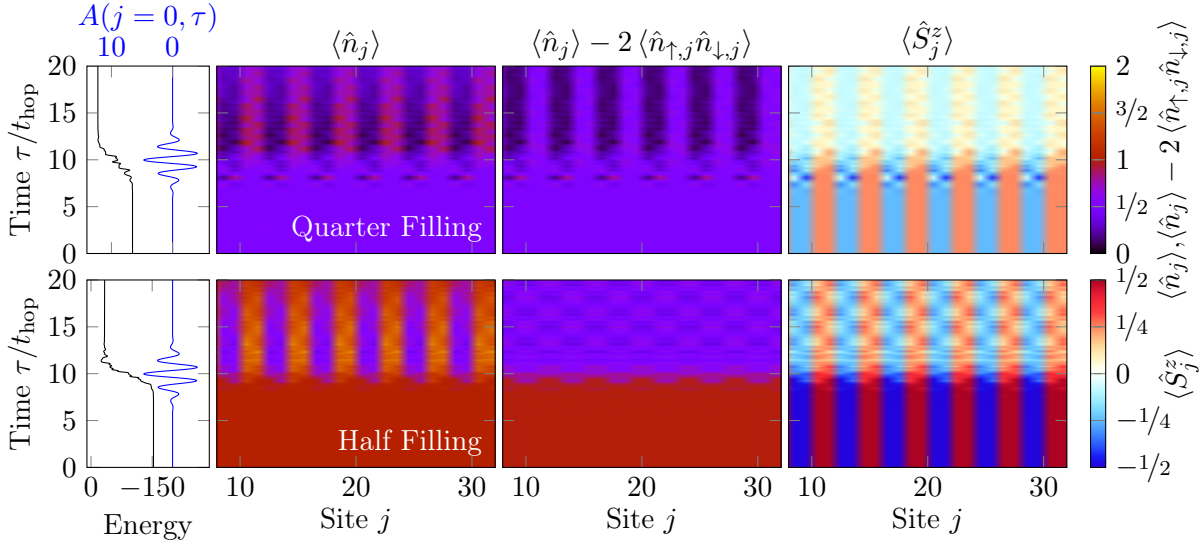
**Figure 12.2:** (Published in [KPM18]) Time evolution of system Eq. (3.36) with  $L = 40$  sites from tDMRG with differently large unit cells induced by a spin-selective pump pulse. (Left) 2 sites, i.e.,  $\Delta/t_{\text{hop}} = \{8, -8\}$  and (right) 8 sites, i.e.,  $\Delta/t_{\text{hop}} = \{8, 8, 8, 8, -8, -8, -8, -8\}$ . For both cases: (top)  $U/t_{\text{hop}} = 0$  and (bottom)  $U/t_{\text{hop}} = 8$ . First column: total energy of the system (black) and the modulation of the vector potential (blue). Second column: particle density  $\langle \hat{N}_j \rangle(\tau)$  in the bulk (sites 8-32). In the case the 8-site unit cell with  $U/t_{\text{hop}} = 8$ , the discarded weight  $\epsilon$  grows faster than in the non-interacting case and reaches  $\epsilon \sim 7 \cdot 10^{-6}$  at the end of the simulation.

Again, a stable pattern occurs in the particle density. In this case, the pattern is more pronounced at the edges of the magnetic domains.

A further magnetic pattern is a Néel-type single-site staggered magnetic structure. Here, we also obtain a CDW pattern as shown in Fig. 12.2, for non-interacting fermions as well as at finite  $U/t_{\text{hop}}$ . At half filling, all excited particles will lead to doublons, and as discussed in Sec. 12.3.2, this leads to a fast decay of the CDW pattern. We complement this discussion by comparing our results at  $U/t_{\text{hop}} = 4$  and  $\Delta/t_{\text{hop}} = 8$  obtained at quarter filling with the ones at half filling, shown in Fig. 12.3. As can be seen, on the time scale displayed, at both values of the filling a CDW pattern is obtained. However, due to the exclusive formation of doublons at half filling, the CDW is less stable in this case and will decay soon.

## Mechanism for the Formation of the CDW | 12.2

Due to the spin-selective excitation, particles of only one spin-direction are moved to the neighboring dimer with opposite local magnetic field. In this way, a new state is obtained, which is at higher energy, has a weaker spin pattern, and a density modulation. This is also seen in the analytical solution of the non-interacting case, which is derived in [KPM18] for the scenario of an infinitesimally short Peierls pulse (a “kick”) and whose main features we discuss in the following: In the limit  $\frac{2t_{\text{hop}}}{\Delta} \rightarrow 0$ , the ground state of each fermion species exhibits a CDW pattern with strong localization in the occupied dimers, thus minimizing the contribution of the magnetic microstructure to the total energy in the specific spin direction. Because a unit cell consists of



**Figure 12.3:** (Extend version of Fig. S3 in [KPM18]) Time evolution of system Eq. (3.36) with  $L = 40$  sites from tDMRG with  $U/t_{\text{hop}} = 4$  and  $\Delta/t_{\text{hop}} = 8$  at (top) quarter filling and (bottom) half filling induced by a spin-selective pump pulse. First column: total energy of the system (black) and the modulation of the vector potential (blue). Second column: particle density  $\langle \hat{n}_j \rangle(\tau)$  in the bulk (sites 8-32). Third column: doublon-purified particle density  $\langle \hat{n}_j \rangle(\tau) - 2 \langle \hat{n}_{\uparrow,j} \hat{n}_{\downarrow,j} \rangle(\tau)$ , also only in the bulk. Fourth column: local magnetizations  $\langle \hat{S}_j^z \rangle(\tau)$ , also only in the bulk. The upper color bar on the right indicates the values for  $\langle \hat{n}_j \rangle(\tau)$  and  $\langle \hat{n}_j \rangle(\tau) - 2 \langle \hat{n}_{\uparrow,j} \hat{n}_{\downarrow,j} \rangle(\tau)$ . The lower color bar on the right indicates the values for  $\langle \hat{S}_j^z \rangle(\tau)$ . Note that the maximum of the color bar for the particle densities is doubled compared to Fig. 12.1 to capture the results at half filling.

four sites, the dispersion has four bands, which are all separated by a gap (see Sec. 3.3). The instantaneous action of the spin-selective Peierls pulse on the ground state can be modeled by applying the time-independent perturbation

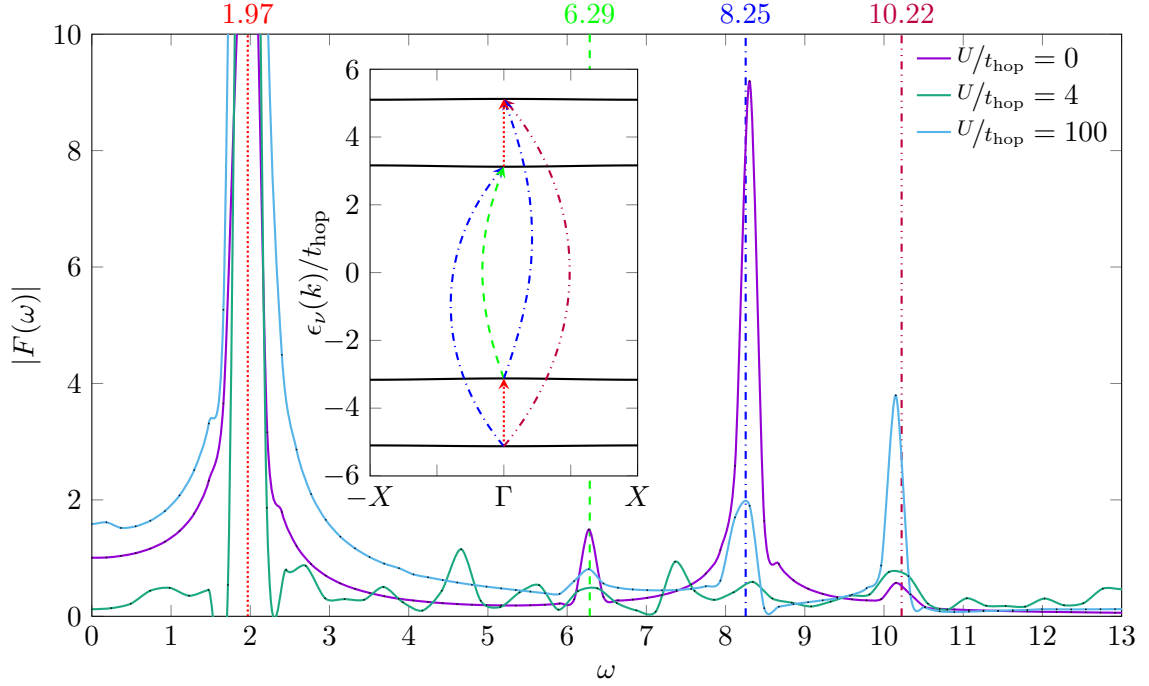
$$\hat{T}_\phi = t_{\text{hop}} \sum_i \left( e^{i\phi} \hat{c}_{\downarrow,i}^\dagger \hat{c}_{\downarrow,i+1} + e^{-i\phi} \hat{c}_{\downarrow,i+1}^\dagger \hat{c}_{\downarrow,i} \right) \quad (12.1)$$

at time  $\tau = \tau_0$ . We focus on the situation at quarter filling  $\langle \hat{N} \rangle = L/2$ , at which the lowest of the four bands is completely filled. In this case, any phase  $\phi \neq 0$  leads to a scattering of particles from the lowest band to excited states, which can occur in any of the remaining three bands, where the maximal mixing is obtained at  $\phi = k\frac{\pi}{2}$  with integer  $k$ .

In the non-interacting case, the time evolution in the quenched fermionic species is dominated by oscillations with frequencies given by the energy differences of the four bands. However, due to the absence of any interaction between both spin directions, the state of the particles of the other species remains unaltered. In this way, the CDW pattern in the ground state of the unaffected species becomes visible. Without interactions, the band populations are conserved quantities. Thus, excitations cannot relax back to the lowest band, and the CDW pattern remains stable up to arbitrary times.

In Fig. 12.4, we further elucidate this scenario by considering the frequencies of the density oscillations on a dimer, which we obtain by first subtracting the double occupancies (see Sec. 12.3.2),

$$\langle (\hat{S}_j^z)^2 - (\hat{S}_{j+1}^z)^2 \rangle = \langle \hat{n}_j - \hat{n}_{j+1} - 2(\hat{n}_{\uparrow,j} \hat{n}_{\downarrow,j} - \hat{n}_{\uparrow,j+1} \hat{n}_{\downarrow,j+1}) \rangle, \quad (12.2)$$



**Figure 12.4:** (Published in [KPM18]) Fourier-transformed dynamics of Eq. (12.2) with  $j$  and  $j + 1$  on a dimer in the center of the system, for different values of  $U$  with  $\Delta/t_{\text{hop}} = 8$ . The vertical lines correspond to the average band gap between the first and the second band (red dotted), the first and the third band (blue, dash-dotted), the second and the third band (green, dashed), and the first and the fourth band (wine red, dash-dot-dotted). The inset shows the band structure in the non-interacting case (also for  $\Delta/t_{\text{hop}} = 8$ ). The transitions between the bands, corresponding to the vertical lines in the main figure, are shown with the same styles at  $k = \Gamma$ .

and then Fourier-transforming the result. The non-interacting band structure at  $\Delta/t_{\text{hop}} = 8$  is shown in the inset, and the results are compared to those at finite  $U$ . The oscillation frequencies of the particle density within a dimer and the associated averaged band gaps are marked and show excellent agreement. This analysis for the non-interacting case can now be used as a starting point to investigate the behavior in the interacting case.

## Effect of Interactions | 12.3

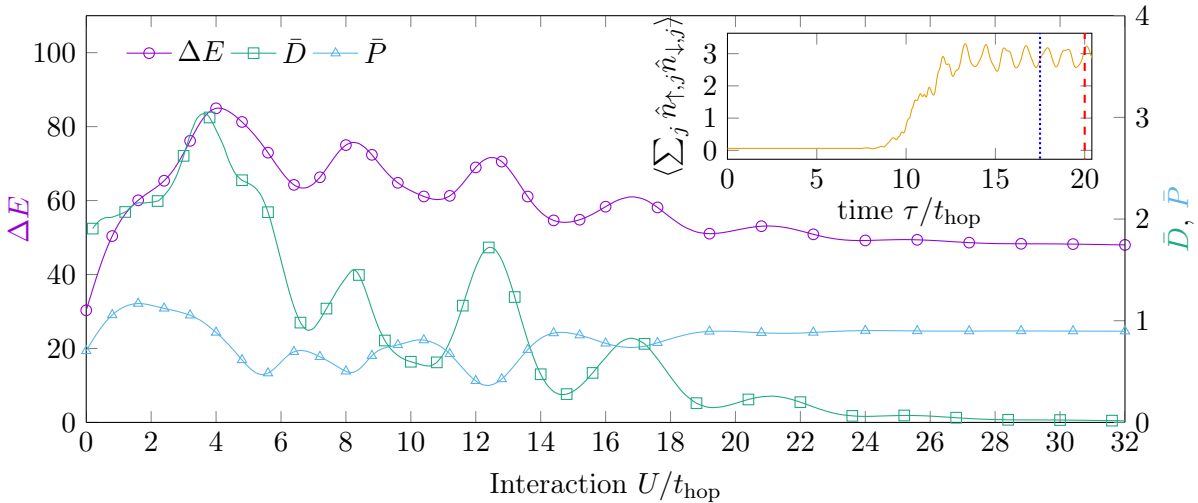
The main effect of a finite Hubbard interaction is to induce scattering between the two fermion species and thereby to transfer energy between them, opening a decay channel for the CDW pattern of the fermion species that is not excited by the pulse. However, the Hubbard term also enforces the formation of local moments with finite  $\hat{S}_j^z$ , which lower the energy in the staggered field and in this way stabilize the periodic pattern for each fermion species. Thus, localization of single fermions within the dimers is enforced by the repulsion. Therefore, we expect the description in terms of the non-interacting system to give at least a qualitative understanding of the dynamics. Indeed, at large  $U$  a mean-field decoupling in the spins ( $\hat{S}_j^z = \langle \hat{S}_j^z \rangle + \delta \hat{S}_j^z$ ) shows that the Zeeman interaction is shifted according to  $\Delta \rightarrow \tilde{\Delta} = \frac{1}{2}(\Delta + 4U)$ , while a strong on-site potential  $\propto U$  localizes the fermions and correlates the motion between the two species (see Ref. [KPM18]). Thus, in the strong coupling limit, the single-particle dynamics for the excited

dimers is also dominated by the frequencies of the non-interacting system, indicating a strong localization of single particles on the dimers and hence a stabilization of the CDW pattern. From Fig. 12.4 we see that in the regime of intermediate interaction  $U \approx \Delta$  there are more decay channels for single-particle excitations. However, there is still the dominant contribution at  $\omega \approx 2$ , i.e., the low-energy excitation of the non-interacting single-particle description. In order to better understand the connection between the two limits, we consider the mean-field decoupling for  $t_{\text{hop}} \ll U$  in more detail. Within this limit, a Peierls pulse redistributes the amplitude of the local moments  $(\hat{S}_i^z)^2$  over the dimers. In Ref. [KPM18] it is shown that the mean-field Hamiltonian essentially resembles a Heisenberg  $XX$  chain with a strong, staggered magnetic field  $\tilde{\Delta}$ . Thus, relaxation of the local moments after the quench is suppressed with  $\tilde{\Delta}$ . The corresponding observable can be written in terms of the local particle densities via

$$(\hat{S}_j^z)^2 = \frac{1}{4}(\hat{n}_{\uparrow,j} - \hat{n}_{\downarrow,j})^2 \propto \hat{n}_j - 2\hat{n}_{\uparrow,j}\hat{n}_{\downarrow,j}. \quad (12.3)$$

### CDW Amplitude versus Interaction Strength | 12.3.1

Since the states obtained after the excitation are to a good approximation invariant under translation by one unit cell at all instances of time, the total number of particles in one unit cell can be considered to be conserved, so that we can identify the doublon density  $\hat{n}_{\uparrow,j}\hat{n}_{\downarrow,j}$  and its dynamics as the dominating decay channel. Subtracting the doublon density from the local density, we expect to obtain the single-particle dynamics. Indeed, in Fig. 12.4 we see that on the time scales reached by our simulations, the doublon-purified density follows the single-particle dynamics for any value of the interaction strength. The question arises how interaction effects during the pulse may correlate the fermion species, thereby reducing the amplitude of the CDW pattern.



**Figure 12.5:** (Extended version of Fig. 3 in [KPM18]) Absorbed energy  $\Delta E = |E_0 - E_{\tau/t_{\text{hop}}=20}|$  (purple circles) at time  $\tau/t_{\text{hop}} = 20$  as function of the interaction strength  $U/t_{\text{hop}}$ . Total number of doublons  $\bar{D}$  (see Eq. (12.5)) (green squares) averaged from time  $\tau_1 = 17.5$  to  $\tau_2 = 20$  ( $\tau_{12} = \tau_2 - \tau_1$ ), and amplitude of the CDW in the center of the system  $\bar{P}$  (see Eq. (12.4)) (blue triangles, see text), also averaged over time as function of the interaction strength  $U/t_{\text{hop}}$ . The strength of the magnetic microstructure is  $\Delta/t_{\text{hop}} = 8$  and the Peierls pulse is spin selective.



In Fig. 12.5, we show tDMRG results for the averaged amplitude of the CDW pattern,

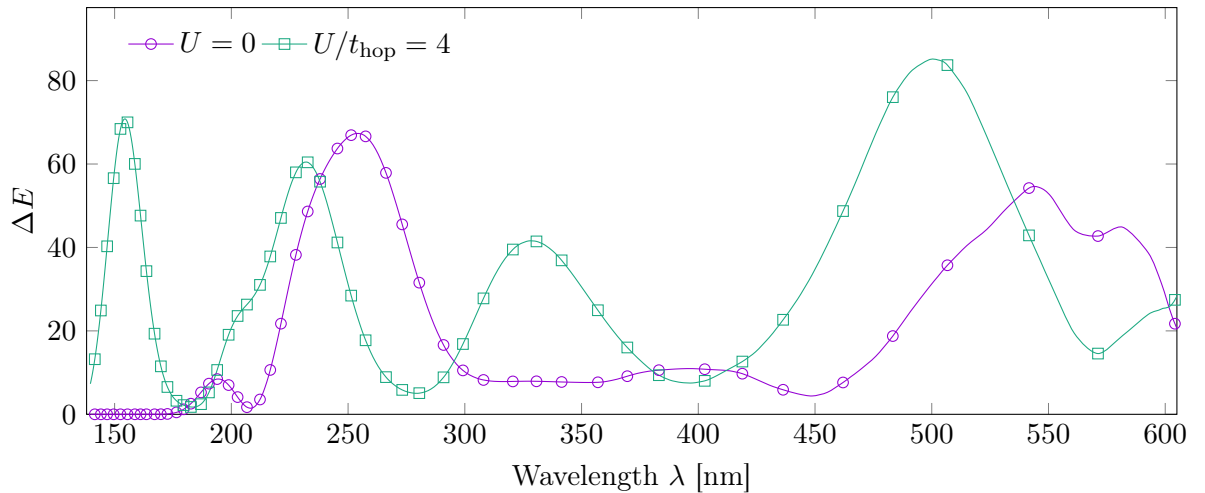
$$\bar{P} = \int_{\tau_1}^{\tau_2} \frac{d\tau}{\tau_{12}} |\langle \hat{n}_{L/2} \rangle(\tau) + \langle \hat{n}_{L/2+1} \rangle(\tau) - \langle \hat{n}_{L/2+2} \rangle(\tau) - \langle \hat{n}_{L/2+3} \rangle(\tau)|, \quad (12.4)$$

within a unit cell, the average doublon density

$$\bar{D} = \int_{\tau_1}^{\tau_2} \frac{d\tau}{\tau_{12}} \sum_j \langle \hat{n}_{\uparrow,j} \hat{n}_{\downarrow,j} \rangle(\tau), \quad (12.5)$$

and the absorbed energy for  $\Delta/t_{\text{hop}} = 8$  as function of  $U$ . In order to get values for the doublon density  $\bar{D}$  and the CDW pattern amplitude  $\bar{P}$  after the pulse, which can be related to each other, we perform a time average in the window  $\tau_1 = 17.5$  to  $\tau_2 = 20$ , which is sufficiently late after the pulse. An example for the full dynamics of the doublon population and the time window for the average is displayed in the inset of Fig. 12.5 for  $U/t_{\text{hop}} = 4$  and  $\Delta/t_{\text{hop}} = 8$ .

The average doublon density  $\bar{D}$  follows the behavior of the absorbed energy. We attribute this to the Hubbard term that adds an energy contribution proportional to the double occupancy. Note that these oscillations do not depend on  $\Delta$  but vary with the wavelength  $\lambda$  of the incoming light. In Fig. 12.6 the amount of absorbed energy  $\Delta E$  as a function of the wavelength  $\lambda$  is shown



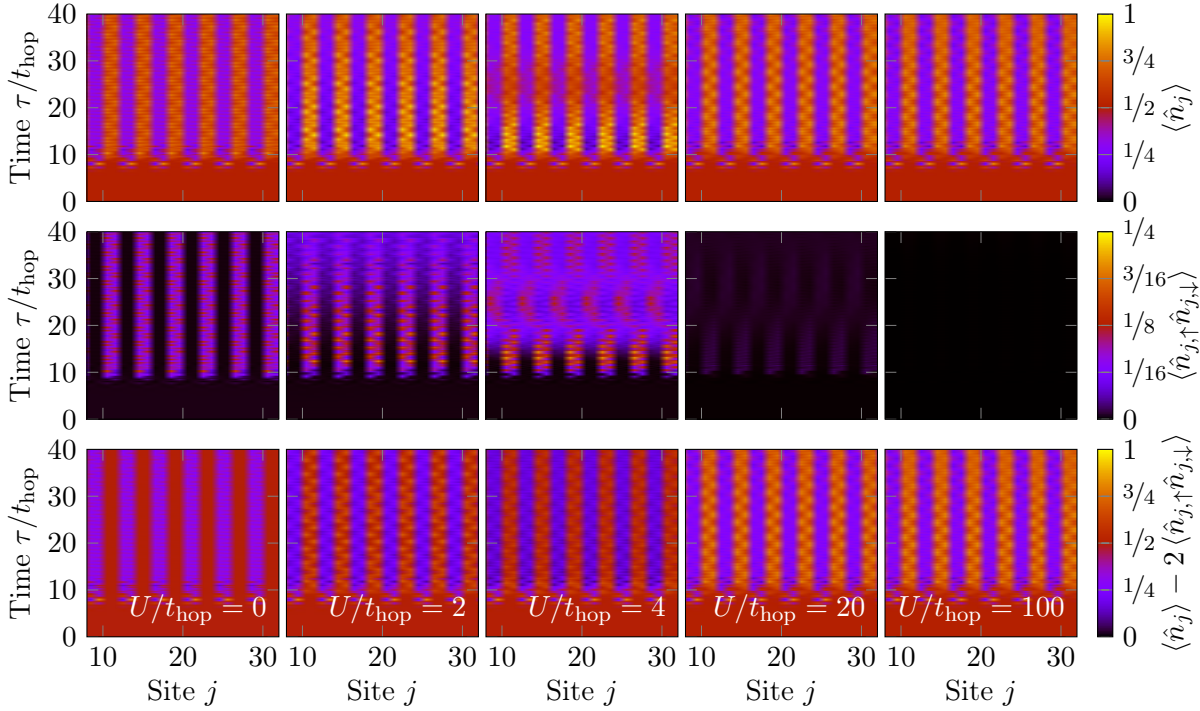
**Figure 12.6:** (Published in [KPM18]) Absorbed energy  $\Delta E = |E_0 - E_{t=20}|$  as function of the wavelength  $\lambda$ . Here we used the parameters  $\Delta/t_{\text{hop}} = 8$ ,  $U/t_{\text{hop}} = 0, 4$  with quarter filling. The Peierls phase is only acting on the spin down direction. Most of the calculations in this manuscript are performed at  $\lambda \sim 500$  nm, so that a substantial amount of energy is absorbed. Note that for clarity not all computed data points are shown, furthermore a spline interpolation (over all computed data points) is used as guide for the eye.

for two interaction strengths  $U/t_{\text{hop}} = 0$  and 4. A non-trivial dependence of the absorption from the parameters of the system is obtained. However, at the wavelength of our choice (see table 3.3)  $\lambda \approx 500$  nm, a significant energy absorption in both cases takes place, so that we expect the effects to be representative for all wavelengths at which absorption takes place. A full scan of the  $\lambda - U$  dependence of the absorbed energy would be helpful for further studies as the  $U$  dependence differs considerably for other wavelengths.

In contrast, the amplitude of the CDW pattern is minimal at the maxima of the absorbed energy. Note that a doublon consists of two particles of each spin direction and has energy  $U + \Delta$ , irrespective of its position. Thus, by a second-order process, at any finite  $U$ , doublons

can move throughout the magnetic microstructure at no energy cost. Therefore, we expect that in the long time limit this yields a homogeneous background particle density. We hence have seen that the motion of doublons is one mechanism for the decay of the CDW pattern in the presence of interactions. However, the doublons themselves cannot decay further, since their energy is the same irrespective of their position, and there is no dissipation channel for their energy. Therefore, we conclude that the motion of the doublons is the dominant decay channel for the induced pattern. Hence, at low enough fillings, the single-particle picture can be restored and we find that the major decay channel for the CDW pattern is due to the delocalization of the doublons. Consequently, creating fewer doublons is beneficial for the strength of the CDW pattern. In contrast, at half filling all the absorbed energy is used to form doublons, so that the CDW pattern will vanish on the delocalization time scale of the doublons [HP12].

### Doublon-Purified Particle Densities | 12.3.2



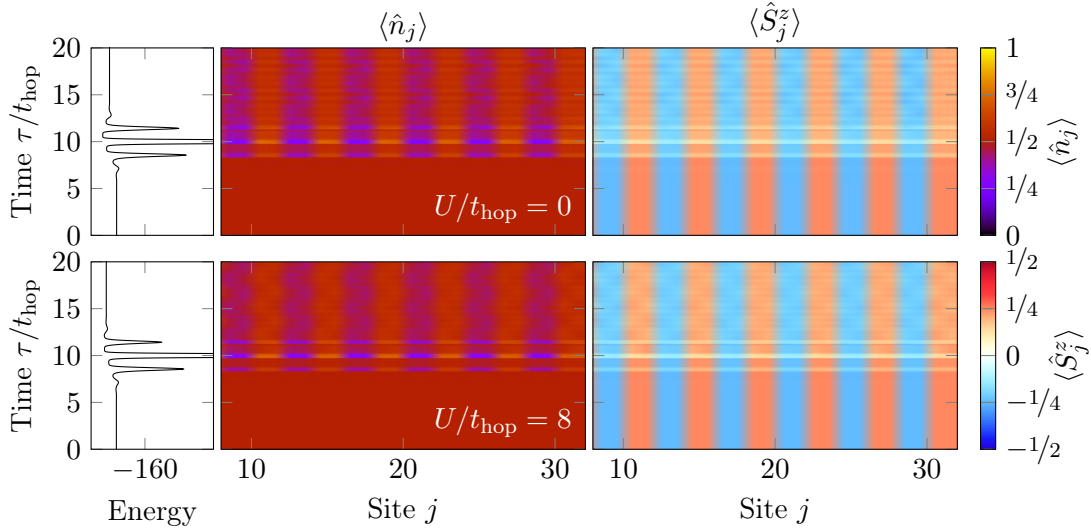
**Figure 12.7:** (Published in [KPM18]) Local particle density (top), local double occupancy (middle), and the local, doublon-cleaned particle density (bottom) for systems with  $\Delta/t_{\text{hop}} = 8$ ,  $L = 40$ , and  $U/t_{\text{hop}} = 0, 2, 4, 20$ , and  $100$ . In the case of small but finite  $U$ , the CDW seems to decay in the local particle-density plots. Considering the double occupancy plots, it becomes clear that this decay is only due to the creation of the space invariant background. Hence, in the doublon-cleaned particle density the decay is absent. Note that for  $U/t_{\text{hop}} = 0$  the double occupancy cannot move away from the dimer to which the spin-down particles were excited and that at large  $U$  nearly no double occupancies were created in the first place. In order to obtain longer times, the bond dimension  $m$  is increased by a factor of 10, i.e.,  $m = 5000$ , to reach a discarded weight  $\epsilon < 10^{-7}$ .

As discussed above, the observed CDW pattern decays through doublon delocalization. Here, we consider the doublon densities, their behavior in time, and the resultant charge densities



after removing the doublon part in more detail. In Fig. 12.7, we show the long-time behavior of the particle density, the double occupancy, and the particle density after subtracting the double occupancy for different values of the interaction  $U/t_{\text{hop}} = 0, 2, 4, 20,$  and  $100$ . The first observation is that at  $U/t_{\text{hop}} = 0$  all affected spin-down particles create double occupancies as the particle density on the particle-rich dimers stays constant and the CDW is only visible due to the particle-poor dimers in the doublon-purified particle density. At finite interaction, this is no longer the case and the amplitude of the CDW is increased. On the other hand, the creation of double occupancies is suppressed with further increasing the value of the interaction  $U/t_{\text{hop}}$ ; a superposition of particles each located on one of the sites of a dimer is preferred instead. Hence, subtracting the doublon contribution from the charge density is insightful in the presence of interactions, as the double occupancy will eventually spread equally over the whole system, because the magnetic microstructure does not constitute a barrier for its motion and it also does not decay. Therefore, only the density without the double occupancy is indicative for the long time behavior of the CDW pattern.

## Possible Experimental Realizations | 12.4



**Figure 12.8:** (Published in [KPM18]) Periodic modulation of the lattice of only one species of particles also leads to a CDW pattern. (Left) Energy of systems with  $\Delta/t_{\text{hop}} = 8$  and (top)  $U/t_{\text{hop}} = 0$  and (bottom)  $U/t_{\text{hop}} = 8$ . (Center) Particle density for these systems. (Right) Spin density for the systems above. In the interacting case ( $U/t_{\text{hop}} = 8$ ), the discarded weight  $\epsilon$  grows rapidly and reaches a value of  $\epsilon \sim 10^{-5}$  at the end of the simulation.

**Periodic Modulation of the Lattice in Ultra-Cold-Gas Experiments** Especially important for those experiments is that the presented considerations are expected to hold even if the excitation is not completely spin-polarized. In ultra-cold-gas experiments, the underlying magnetic (or ionic) pattern can be realized by super lattices [PPT<sup>+</sup>03, SSAJP06, TCF<sup>+</sup>08, ALB<sup>+</sup>07, MDU<sup>+</sup>15, AAN<sup>+</sup>13, AAN<sup>+</sup>11, YDS<sup>+</sup>17, LSZ<sup>+</sup>15, SLCB16]. The spin-selective photoexcitation in this case can be obtained by shaking the lattice of only one species [KIMG06, GTU<sup>+</sup>11, SGJ<sup>+</sup>10, SPA<sup>+</sup>10]. We have seen that this leads to similar behavior, which can be investigated using quantum-gas microscopes [BGP<sup>+</sup>09, PHM<sup>+</sup>15, CNO<sup>+</sup>15, OBH<sup>+</sup>15, GB17, BHS<sup>+</sup>16]. In

Fig. 12.8 we present our results, in which the spin-selective photoexcitation is emulated by a periodic modulation of the lattice of only one fermionic species. This can be realized in experiments with ultra-cold quantum gases on optical lattices. As can be seen, the CDW pattern emerges also in this setup, indicating that the details of the excitation are not crucial, as long as it is acting on only one fermionic species. As in the case of a Peierls pulse, at finite  $U$  the absorbed energy is larger than at  $U/t_{\text{hop}} = 0$ .

In this thesis, we developed methods in order to numerically simulate different approaches for photoexcitations in strongly-correlated interacting electron systems. The many-body quantum states in those simulations are represented by matrix-product states (MPSs) (see chapter 5) and operators are represented by matrix-product operators (MPOs) (see chapter 6), which are created by finite-state machines (FSMs) (see chapter 7). Inspired by praseodymium-calcium-manganite (PCMO) we employed a hypothetical one-dimensional (1D) model that is a variant of the Hubbard model (see Secs. 3.1 and 3.3).

In the first part of this conclusion, we summarize the work of this thesis. Afterwards, we discuss perspectives for future investigations. Both parts are divided in a method-development and a photo-excitation section.

### Method development

Due to the representation of operators by FSMs, we found in Sec. 7.2 that arithmetic operations of sums of operator products can be performed exactly by manipulating their graphs. Furthermore, we showed in Sec. 7.3 that subsequent lossless compressions can be executed exactly. Both claims, which follow from construction, are still underpinned by an example in which we showed that the variance can be determined up to a chosen (machine) precision (see Sec. 7.4.1).

In chapter 10, we presented a quantum-computer simulator (QCS) that is fully programmable in terms of universal quantum gates. All available gates are described and several examples concerning the usage are given. The development of the QCS was inspired by the easy creation of operators due to the usage of FSMs and the convenient representation of the qubits in terms of MPSs. Furthermore, this QCS is not limited by the exponential growth of the Hilbert space due to the number of qubits as QCSs that are based on exact diagonalization. The limitation rather results from entanglement, which may eventually accumulate due to gate applications. The QCS was used for several MPS manipulations in day-to-day life, e.g., the creation of a canonical infinite-temperature state.

### Photoexcitations

The first type of photoexcitation investigated in chapter 11 is an excitation of a single dimer in the center of a dimerized system at its ground state. The excitation produces a dipole oscillation on the dimer, which can be described as an electron-hole pair. Without Hund's splitting, i.e.,  $\Delta = 0$ , the expected light cone was observed. This holds true with and without interactions. For the latter, but with intermediate Hund's splitting, i.e.,  $\Delta/t_{\text{hop}} = 2$ , an additional linear structure with a slope corresponding to the phase velocity emerges in the local density. The maximal velocity for information transport is the group velocity and, in fact, the linear structure

decays quickly as soon as it reaches the border of the light cone whereby it obeys the Lieb-Robinson bound. This behavior has been observed before [CDC<sup>+</sup>18], however, here it occurs in a local observable in a system with magnetic microstructure. In the case of large Hund's splitting, we found that the excitation mainly behaved like a particle in a box. Nevertheless, due to the combination of the tunneling effects and the repulsion, we obtained an energy transport from the excited dimer to its neighbors in the interacting case.

In order to investigate the bulk properties, we determined the population of the four bands as functions of time and compared them to their expectation values in thermal equilibrium (see Sec. 11.2). We found that the occupation of the bands relaxed closely to the thermal values, but differently quickly depending on the value of the Hund's splitting. The relaxation times were found to be in good agreement with values found by linearized Boltzmann equations in [KRS<sup>+</sup>18, Bie16]. We furthermore compared the momentum distribution before and after the excitation and found (in full agreement with our predictions) that momentum at all  $k$  values is transferred by the photoexcitation, which is local in real space.

In order to obtain a more realistic photoexcitation, we next used a wave packet, modeled semi-classically by the Peierls substitution (see Sec. 3.4.1), as excitation. In this case, a very large amplitude was necessary to obtain an excitation that persists after the wave packet has left the system, which would have led to a dielectric breakdown of the material. As we were interested in an intact solid state, we introduced spin-selective excitations with smaller amplitudes instead. With this setup, we found in chapter 12 that a long-living charge-density-wave-like pattern emerges from the initial spin-density-wave-like state for a finite value of the Hund's splitting. In the non-interacting case, this behavior follows from the construction of the initial ground state and the remaining oscillations can be identified from the band structure. In order to find the long-time behavior of the emerged structure in the case of finite interactions, it is beneficial to remove the doublon density from the particle density, as shown in Sec. 12.3. There, we argue that the spreading of the doublons is the main decay channel for the charge-density wave (CDW). By removing the doublon density, we expect to obtain the CDW amplitude of the long time limit shortly after the wave packet has left the system.

Concerning experimental realizations, spin-selective excitations might be realizable by polarized light. The CDW should then be visible in angle-resolved photoemission spectroscopy (ARPES) experiments. Furthermore, it is possible to realize a magnetic pattern in ultracold-gases experiments, and simulate spin-selective excitations by shaking only the lattice of one species. We showed in Sec. 12.4 that a similarly long-living charge-density-like pattern emerges also in this case. In materials, however, additional effects like the dynamics of the spin structure and phonons will eventually destroy the pattern. We expect this to happen on the Frank-Condon time scale, and therefore expect a lifetime of  $\sim$  ps for the pattern to persist in this context.

## Perspective | 13.2

In the following sections, we lay out perspectives, applications, and further possible investigations following the results of this thesis.

### Method Development and Quantum Information

As usual, the development is always nearly finished, which means there is still a lot of work left to be done: Starting from implementing the Trotter time-evolution scheme with swap gates

[SW10], over applying parallelization schemes [SW13], ending with including higher-precision linear-algebra packages<sup>1</sup>. Besides these to-dos and the incessant wish for treating higher-dimensional systems, the FSM representation of MPOs enables us to have more flexibility in the exact creation of operators. Furthermore, a better understanding of further compression schemes can be developed on the graph level, which might even lead to lossy, but usefully compressed MPOs. The concept of FSMs can also be generalized to higher-dimensional tensor networks, e.g., PEPS [VMC08]. In this case, the transition amplitudes get additional degrees of freedom ('colors'), corresponding to either transversal or longitudinal auxiliary indices.

The development of the QCS is only the first step into the quantum-information world. As a next step, it is necessary to show that there are quantum algorithms with low entanglement, and also that it is possible to obtain valid results from simulations in which the state is truncated. In order to test this, we need to implement the commonly known algorithms and compare them to other simulators and real experiments. If our QCS passes those tests, another question arises: How do we simulate typical error sources? Is it enough to introduce a temperature and include a probability for the success of every gate application? Or do we need to actually simulate the radio pulses that are used to modify the qubits, which are realized by Josephson junctions? Either way, the capability to perform measurements with subsequent projections will become important soon. Furthermore, a fully interactive version of the QCS is possible and desirable.

In order to simulate the relaxation process of a (light) excitation more realistically it would be beneficial to be able to treat phonons and therefore bosons in the simulations. Additionally, if simulations at very low temperatures are required, an imaginary-time evolution becomes necessary that conserves quantum-numbers separately, both in the auxiliary part and in the physical part.

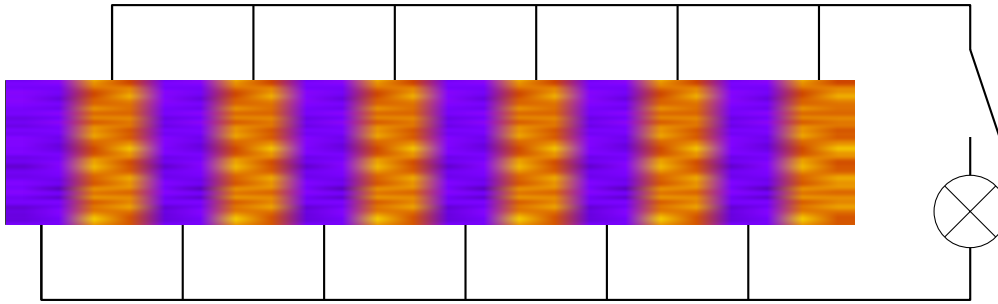
### Photoexcitations

As already mentioned, investigating the photoexcitations in a hypothetical two-dimensional PCMO model would lead to an even better understanding of the relaxation processes. A feasible step in that direction would be a ladder, in which the zig-zag chains of a CE structure [WK55] are represented by weakly coupled chains. Another change in the geometry of the model would be the change from open-boundary conditions (OBC) to periodic-boundary conditions (PBC), which can be combined with the ladder into a cylindric model. It would also be helpful to understand simpler excitations than the Peierls substitution, while still having an excitation that is local in  $k$ -space. This could be realized by a global real-space excitation, e.g., a variant of Eq. (3.34).

Independent of the applied excitation, further observables would give additional insight into the properties of the system afterwards. Foremost, the time-dependent optical conductivity, e.g., measured with an additional probe pulse, again modeled via the Peierls substitution, could give answers to questions like: Is there optically driven superconductivity? Does the conductivity change at all due to the excitation?

A more immediate future investigation is to check our findings for further fillings between  $1/4$  and  $1/2$ . Also, a more complete overview of the absorption spectrum for different combinations of interaction strengths and wavelengths could be helpful for experiments.

<sup>1</sup>For example, <http://mplapack.sourceforge.net/>.



**Figure 13.1:** Schematic 2D model with CDW connected to a switch and a load. If the switch is closed, the excited electrons will move through the load and could therefore relax into the less occupied parts of the system. Note that this is not a realistic calculation, but only a schematic for an idea.

Another question, which leads in the direction of information storage, is: Can we reset the CDW back to a spin-density wave (SDW)? The simplest approach is exciting the system with another Peierls pulse that excites the opposite spin species. This would destroy the CDW but would also decrease the amplitude of the SDW instead of increasing it. As we have increased the energy in the system, it would be necessary to decrease the energy with some kind of action. One approach would be to design a load and a switch and connect it to the system as depicted in Fig. 13.1. Of course, we assume that the job of the load will after some time be done by phonons and thermal effects, which are neither preventable nor controllable. Nevertheless, a controllable load would, in this scenario, lead to a very fast, optically driven, volatile memory.

# Appendices







In this appendix, we present the most important finite-state machines (FSMs) used in this thesis.

```

subsection hubbard_entangler
set edges
    = Id; c_up_hub; c_dagger_up_hub; c_down_hub; c_dagger_down_hub; \
      n_up_hub; n_down_hub; n_up_down_hub; n_hub; s_hub
set final_states
    = F
set initial_states
    = I
set model
    = I:Id:I; F:Id:F; \
\
I:phys*c_dagger_up_hub:I_L_D_11_down;          I_L_D_11_down:c_dagger_down_hub:IM_D_down; \
I:neg_phys*n_up_hub*c_dagger_up_hub:I_L_D_N1_down;  I_L_D_N1_down:c_dagger_down_hub:IM_D_down; \
I:neg_phys*c_dagger_up_hub:I_L_D_1N_down;          I_L_D_1N_down:n_down_hub*c_dagger_down_hub:IM_D_down; \
I:2_phys*n_up_hub*c_dagger_up_hub:I_L_D_NN_down;    I_L_D_NN_down:n_down_hub*c_dagger_down_hub:IM_D_down; \
\
I:phys*c_dagger_down_hub:I_L_D_11_up;            I_L_D_11_up:c_dagger_up_hub:IM_D_up; \
I:neg_phys*n_down_hub*c_dagger_down_hub:I_L_D_N1_up;  I_L_D_N1_up:c_dagger_up_hub:IM_D_up; \
I:neg_phys*c_dagger_down_hub:I_L_D_1N_up;          I_L_D_1N_up:n_up_hub*c_dagger_up_hub:IM_D_up; \
I:2_phys*n_down_hub*c_dagger_down_hub:I_L_D_NN_up;    I_L_D_NN_up:n_up_hub*c_dagger_up_hub:IM_D_up; \
\
IM_D_down:phys*c_up_hub:F_L_11_down;             F_L_11_down:c_down_hub:F; \
IM_D_down:neg_phys*c_up_hub*n_up_hub:F_L_N1_down;  F_L_N1_down:c_down_hub:F; \
IM_D_down:neg_phys*c_up_hub:F_L_1N_down;          F_L_1N_down:c_down_hub*n_down_hub:F; \
IM_D_down:2_phys*c_up_hub*n_up_hub:F_L_NN_down;    F_L_NN_down:c_down_hub*n_down_hub:F; \
\
IM_D_up:phys*c_down_hub:F_L_11_up;               F_L_11_up:c_up_hub:F; \
IM_D_up:neg_phys*c_down_hub*n_down_hub:F_L_N1_up;  F_L_N1_up:c_up_hub:F; \
IM_D_up:neg_phys*c_down_hub:F_L_1N_up;           F_L_1N_up:c_up_hub*n_up_hub:F; \
IM_D_up:2_phys*c_down_hub*n_down_hub:F_L_NN_up;    F_L_NN_up:c_up_hub*n_up_hub:F; \
\
I:phys*c_up_hub:I_L_11_down;                     I_L_11_down:c_down_hub:IM_down; \
I:neg_phys*c_up_hub*n_up_hub:I_L_N1_down;        I_L_N1_down:c_down_hub:IM_down; \
I:neg_phys*c_up_hub:I_L_1N_down;                 I_L_1N_down:c_down_hub*n_down_hub:IM_down; \
I:2_phys*c_up_hub*n_up_hub:I_L_NN_down;          I_L_NN_down:c_down_hub*n_down_hub:IM_down; \
\
I:phys*c_down_hub:I_L_11_up;                     I_L_11_up:c_up_hub:IM_up; \
I:neg_phys*c_down_hub*n_down_hub:I_L_N1_up;      I_L_N1_up:c_up_hub:IM_up; \
I:neg_phys*c_down_hub:I_L_1N_up;                 I_L_1N_up:c_up_hub*n_up_hub:IM_up; \
I:2_phys*c_down_hub*n_down_hub:I_L_NN_up;        I_L_NN_up:c_up_hub*n_up_hub:IM_up; \
\
IM_down:phys*c_dagger_up_hub:F_L_D_11_down;      F_L_D_11_down:c_dagger_down_hub:F; \
IM_down:neg_phys*n_up_hub*c_dagger_up_hub:F_L_D_N1_down;  F_L_D_N1_down:c_dagger_down_hub:F; \
IM_down:neg_phys*c_dagger_up_hub:F_L_D_1N_down;    F_L_D_1N_down:n_down_hub*c_dagger_down_hub:F; \
IM_down:2_phys*n_up_hub*c_dagger_up_hub:F_L_D_NN_down;  F_L_D_NN_down:n_down_hub*c_dagger_down_hub:F; \
\
IM_up:phys*c_dagger_down_hub:F_L_D_11_up;        F_L_D_11_up:c_dagger_up_hub:F; \
IM_up:neg_phys*n_down_hub*c_dagger_down_hub:F_L_D_N1_up;  F_L_D_N1_up:c_dagger_up_hub:F; \
IM_up:neg_phys*c_dagger_down_hub:F_L_D_1N_up;      F_L_D_1N_up:n_up_hub*c_dagger_up_hub:F; \
IM_up:2_phys*n_down_hub*c_dagger_down_hub:F_L_D_NN_up;  F_L_D_NN_up:n_up_hub*c_dagger_up_hub:F; \
\
set nodes
    = IM_D_down:Id:IM_D_down; IM_D_up:Id:IM_D_up; IM_down:Id:IM_down; IM_up:Id:IM_up
    = I; F; IM_D_down; IM_D_up; IM_down; IM_up; \
I_L_D_11_down; I_L_D_1N_down; I_L_D_N1_down; I_L_D_NN_down; I_L_D_11_up; I_L_D_1N_up; I_L_D_N1_up; I_L_D_NN_up; \
I_L_11_down; I_L_1N_down; I_L_N1_down; I_L_NN_down; I_L_11_up; I_L_1N_up; I_L_N1_up; I_L_NN_up; \
F_L_11_down; F_L_1N_down; F_L_N1_down; F_L_NN_down; F_L_11_up; F_L_1N_up; F_L_N1_up; F_L_NN_up; \
F_L_D_11_down; F_L_D_1N_down; F_L_D_N1_down; F_L_D_NN_down; F_L_D_11_up; F_L_D_1N_up; F_L_D_N1_up; F_L_D_NN_up
set print_ignore
    = Id
set weight_functions
    = phys: {even(position)}; \
      neg_phys: {(-1.0) * even(position)}; \
      2_phys: {2.0 * even(position)}
end

```

## Peierls FSM A.2

```

subsection hubbard_with_peierls
set edges = Id; c_up_hub; c_dagger_up_hub; c_down_hub; c_dagger_down_hub; \
n_up_hub; n_down_hub; n_up_down_hub; n_hub; s_hub
set final_states = F
set initial_states = I
set model = I:Id:I; F:Id:F; \
I:U*n_up_down_hub:F; \
I:t1up*c_up_hub:T_11; T_11:c_dagger_up_hub:F; \
I:t2up*c_dagger_up_hub:T_12; T_12:c_up_hub:F; \
I:t1down*c_down_hub:T_21; T_21:c_dagger_down_hub:F; \
I:t2down*c_dagger_down_hub:T_22; T_22:c_down_hub:F; \
I:even*b_z_up*n_up_hub:B11; B11:Id:F; \
I:even*Id:B12; B12:b_z_up*n_up_hub:F; \
I:even*b_z_down*n_down_hub:B21; B21:Id:F; \
I:even*Id:B22; B22:b_z_down*n_down_hub:F
set nodes = I; F; T_11; T_12; T_21; T_22; U_1; B11; B12; B21; B22
set print_ignore = Id
set weight_functions = U: {U}; even: {even(position)}; \
t1up: {t * exp((imag_i * e_el * a_lattice * alpha_up) / (2.0 * hbar) * e0 / (k0 * c) * \
if((em_field==0), 0.0, 1.0) * \
if((em_field==1), (sin(a_lattice * position * k0 - c * k0 * time) \
+ sin(a_lattice * (position + 1) * k0 - c * k0 * time)), 1.0) * \
if((em_field==2), (exp(-(a_lattice * position - c * (time - t0))^2/(sigma^2) \
* sin(a_lattice * position * k0 - c * k0 * time) \
+ exp(-(a_lattice * (position + 1) - c * (time - t0))^2/(sigma^2) \
* sin(a_lattice * (position + 1) * k0 - c * k0 * time) ), 1.0) \
)); \
t2up: {-1.0 * t * exp(-(imag_i * e_el * a_lattice * alpha_up) / (2.0 * hbar) * e0 / (k0 * c) * \
if((em_field==0), 0.0, 1.0) * \
if((em_field==1), (sin(a_lattice * position * k0 - c * k0 * time) \
+ sin(a_lattice * (position + 1) * k0 - c * k0 * time)), 1.0) * \
if((em_field==2), (exp(-(a_lattice * position - c * (time - t0))^2/(sigma^2) \
* sin(a_lattice * position * k0 - c * k0 * time) \
+ exp(-(a_lattice * (position + 1) - c * (time - t0))^2/(sigma^2) \
* sin(a_lattice * (position + 1) * k0 - c * k0 * time) ), 1.0) \
)); \
t1down: {t * exp((imag_i * e_el * a_lattice * alpha_down) / (2.0 * hbar) * e0 / (k0 * c) * \
if((em_field==0), 0.0, 1.0) * \
if((em_field==1), (sin(a_lattice * position * k0 - c * k0 * time) \
+ sin(a_lattice * (position + 1) * k0 - c * k0 * time)), 1.0) * \
if((em_field==2), (exp(-(a_lattice * position - c * (time - t0))^2/(sigma^2) \
* sin(a_lattice * position * k0 - c * k0 * time) \
+ exp(-(a_lattice * (position + 1) - c * (time - t0))^2/(sigma^2) \
* sin(a_lattice * (position + 1) * k0 - c * k0 * time) ), 1.0) \
)); \
t2down: {-1.0 * t * exp(-(imag_i * e_el * a_lattice * alpha_down) / (2.0 * hbar) * e0 / (k0 * c) * \
if((em_field==0), 0.0, 1.0) * \
if((em_field==1), (sin(a_lattice * position * k0 - c * k0 * time) \
+ sin(a_lattice * (position + 1) * k0 - c * k0 * time)), 1.0) * \
if((em_field==2), (exp(-(a_lattice * position - c * (time - t0))^2/(sigma^2) \
* sin(a_lattice * position * k0 - c * k0 * time) \
+ exp(-(a_lattice * (position + 1) - c * (time - t0))^2/(sigma^2) \
* sin(a_lattice * (position + 1) * k0 - c * k0 * time) ), 1.0) \
)); \
b_z_up: {0.5 * (b_z + if((zeeman_coupling==1), \
if((em_field==1), beta_up * unit_factor * (e0 / c) \
* cos(a_lattice * position * k0 - c * k0 * time), 0.0) + \
if((em_field==2), beta_up * unit_factor * (e0 / c) * cos(a_lattice * position * k0 \
- c * k0 * time) * exp(-(position * a_lattice \
- c * (time-t0))^2/(sigma^2)), 0.0), 0.0) )}; \
b_z_down: {-0.5 * (b_z + if((zeeman_coupling==1), \
if((em_field==1), beta_down * unit_factor * (e0 / c) \
* cos(a_lattice * position * k0 - c * k0 * time), 0.0) + \
if((em_field==2), beta_down * unit_factor * (e0 / c) \
* cos(a_lattice * position * k0 - c * k0 * time) \
* exp(-(position * a_lattice - c * (time-t0))^2/(sigma^2)), 0.0), \
0.0) ) }
end

```

## Heisenberg Trotter Gate FSM A.3

```

subsection trotter_heisenberg
set edges = Id; Sz; Splus; Sminus
set final_states = F
set initial_states = I
set model = I:Id:I; F:Id:F; \

```

```

I:at_i*e12*Splus:A1; A1:Sminus:A2; A2:Sminus:A3; A3:Splus:F; \
I:at_i*e21*Sminus:B1; B1:Splus:E2; E2:Splus:B3; B3:Sminus:F; \
I:at_i*oh*e00*Id:C1; C1:Id:C2; C2:oh*Id:C3; C3:Id:F; I:at_i*e00*Sz:D1; D1:Id:C2; C2:Sz:D3; D3:Id:F; \
I:at_i*oh*e33*Id:E1; E1:Id:E2; E2:oh*Id:E3; E3:Id:F; I:at_i*e33*m1*Sz:F1; F1:Id:E2; E2:m1*Sz:D3; F3:Id:F; \
I:at_i*oh*e11*Id:G1; G1:Id:G2; G2:oh*Id:G3; G3:Id:F; I:at_i*e11*Sz:H1; H1:Id:G2; G2:m1*Sz:H3; H3:Id:F; \
I:at_i*oh*e22*Id:I1; I1:Id:I2; I2:oh*Id:I3; I3:Id:F; I:at_i*e22*m1*Sz:J1; J1:Id:I2; I2:Sz:J3; J3:Id:F;
set nodes = I; F; A1; A2; A3; B1; B2; B3; C1; C2; C3; D1; D3; \
E1; E2; E3; F1; F3; G1; G2; G3; H1; H3; I1; I2; I3; J1; J3
set print_ignore = Id
set weight_functions = m1: {-1.0}; oh: {0.5}; at_i: {if( (position/2.0 == i), 1, 0)}; \
e00: {exp(-imag_i * h/4.0 * dt)}; e33: {exp(-imag_i * h/4.0 * dt)}; \
e11: {(exp(-imag_i * (-D/2.0 - h/4.0) * dt) + exp(-imag_i * (D/2.0 - h/4.0) * dt))/2.0}; \
e22: {(exp(-imag_i * (-D/2.0 - h/4.0) * dt) + exp(-imag_i * (D/2.0 - h/4.0) * dt))/2.0}; \
e12: {(-exp(-imag_i * (-D/2.0 - h/4.0) * dt) + exp(-imag_i * (D/2.0 - h/4.0) * dt))/2.0}; \
e21: {(-exp(-imag_i * (-D/2.0 - h/4.0) * dt) + exp(-imag_i * (D/2.0 - h/4.0) * dt))/2.0}; \
end

```

## Full Two Site Gates FSM | A.4

## Spin-1/2 | A.4.1

```

subsection full_twosite_op
set edges = Id; Sz; Splus; Sminus
set final_states = F
set initial_states = I
set model = I:Id:I; F:Id:F; \
\
I:at_i*oh*e00*Id:E001; E001:oh*Id:F; I:at_i*oh*e00*Id:E002; E002:Sz:F; \
I:at_i*e00*Sz:E003; E003:oh*Id:F; I:at_i*e00*Sz:E004; E004:Sz:F; \
\
I:at_i*oh*e11*Id:E111; E111:oh*Id:F; I:at_i*oh*e11*Id:E112; E112:m1*Sz:F; \
I:at_i*e11*Sz:E113; E113:oh*Id:F; I:at_i*e11*Sz:E114; E114:m1*Sz:F; \
\
I:at_i*e12*Splus:E121; E121:Sminus:F; I:at_i*e21*Sminus:E211; E211:Splus:F; \
\
I:at_i*oh*e22*Id:E221; E221:oh*Id:F; I:at_i*oh*e22*Id:E222; E222:Sz:F; \
I:at_i*m1*e22*Sz:E223; E223:oh*Id:F; I:at_i*m1*e22*Sz:E224; E224:Sz:F; \
\
I:at_i*oh*e33*Id:E331; E331:oh*Id:F; I:at_i*oh*e33*Id:E332; E332:m1*Sz:F; \
I:at_i*m1*e33*Sz:E333; E333:oh*Id:F; I:at_i*m1*e33*Sz:E334; E334:m1*Sz:F;
set nodes = I; F; \
E001; E002; E003; E004; \
E111; E112; E113; E114; E121; \
E211; E221; E222; E223; E224; \
E331; E332; E333; E334;
set print_ignore = Id
set weight_functions = m1: {-1.0}; oh: {0.5}; at_i: {if( (position == i), 1, 0)}; \
e00: {e00r + e00i * imag_i}; e01: {e01r + e01i * imag_i}; e02: {e02r + e02i * imag_i}; e03: {e03r + e03i * imag_i}; \
e10: {e10r + e10i * imag_i}; e11: {e11r + e11i * imag_i}; e12: {e12r + e12i * imag_i}; e13: {e13r + e13i * imag_i}; \
e20: {e20r + e20i * imag_i}; e21: {e21r + e21i * imag_i}; e22: {e22r + e22i * imag_i}; e23: {e23r + e23i * imag_i}; \
e30: {e30r + e30i * imag_i}; e31: {e31r + e31i * imag_i}; e32: {e32r + e32i * imag_i}; e33: {e33r + e33i * imag_i};
end

```

## Fermionic | A.4.2

```

subsection full_twosite_op
set edges = Id; c_dagger_up_hub; c_up_hub; c_dagger_down_hub; c_down_hub; s_hub; \
n_hub; n_up_down_hub; n_up_hub; n_down_hub
set final_states = F
set initial_states = I
set model = I:Id:I; F:Id:F; \
\
I:at_i*e00*Id:E000; E000:Id:F; I:at_i*e00*Id:E001; E001:m1*n_hub:F; I:at_i*e00*Id:E002; E002:n_up_down_hub:F; \
I:at_i*e00*m1*n_hub:E003; E003:Id:F; I:at_i*e00*m1*n_hub:E004; E004:m1*n_hub:F; \
I:at_i*e00*n_up_down_hub:E005; E005:n_up_down_hub:F; I:at_i*e00*n_up_down_hub:E006; E006:Id:F; \
I:at_i*e00*n_up_down_hub:E007; E007:m1*n_hub:F; I:at_i*e00*n_up_down_hub:E008; E008:n_up_down_hub:F; \
I:at_i*e11*Id:E110; E110:n_up_hub:F; I:at_i*e11*Id:E111; E111:m1*n_up_down_hub:F; \
I:at_i*e11*m1*n_hub:E112; E112:n_up_hub:F; I:at_i*e11*m1*n_hub:E113; E113:m1*n_up_down_hub:F; \
I:at_i*e11*n_up_down_hub:E114; E114:n_up_hub:F; I:at_i*e11*n_up_down_hub:E115; E115:m1*n_up_down_hub:F; \
I:at_i*e22*Id:E220; E220:n_down_hub:F; I:at_i*e22*Id:E221; E221:m1*n_up_down_hub:F; \
I:at_i*e22*m1*n_hub:E222; E222:n_down_hub:F; I:at_i*e22*m1*n_hub:E223; E223:m1*n_up_down_hub:F; \

```

```

I:at_i*e22*n_up_down_hub:E224; E224:n_down_hub:F; I:at_i*e22*n_up_down_hub:E225; E225:m1*n_up_down_hub:F;\
I:at_i*e33*Id:E330; E330:n_up_down_hub:F; I:at_i*e33*m1*n_hub:E331; E331:n_up_down_hub:F;\
I:at_i*e33*n_up_down_hub:E332; E332:n_up_down_hub:F; I:at_i*e44*n_up_hub:E440; E440:Id:F;\
I:at_i*e44*n_up_hub:E441; E441:m1*n_hub:F; I:at_i*e44*n_up_hub:E442; E442:n_up_down_hub:F;\
I:at_i*e44*m1*n_up_down_hub:E443; E443:Id:F; I:at_i*e44*m1*n_up_down_hub:E444; E444:m1*n_hub:F;\
I:at_i*e44*m1*n_up_down_hub:E445; E445:n_up_down_hub:F; I:at_i*e55*n_up_hub:E550; E550:n_up_down_hub:F;\
I:at_i*e55*n_up_hub:E551; E551:m1*n_up_down_hub:F; I:at_i*e55*m1*n_up_down_hub:E552; E552:n_up_down_hub:F;\
I:at_i*e55*m1*n_up_down_hub:E553; E553:m1*n_up_down_hub:F; I:at_i*e66*n_up_hub:E660; E660:n_down_hub:F;\
I:at_i*e66*n_up_hub:E661; E661:m1*n_up_down_hub:F; I:at_i*e66*m1*n_up_down_hub:E662; E662:n_down_hub:F;\
I:at_i*e66*m1*n_up_down_hub:E663; E663:m1*n_up_down_hub:F; I:at_i*e77*n_up_hub:E770; E770:n_up_down_hub:F;\
I:at_i*e77*m1*n_up_down_hub:E771; E771:n_up_down_hub:F; I:at_i*e88*n_down_hub:E880; E880:Id:F;\
I:at_i*e88*n_down_hub:E881; E881:m1*n_hub:F; I:at_i*e88*n_down_hub:E882; E882:n_up_down_hub:F;\
I:at_i*e88*m1*n_up_down_hub:E883; E883:Id:F; I:at_i*e88*m1*n_up_down_hub:E884; E884:m1*n_hub:F;\
I:at_i*e88*m1*n_up_down_hub:E885; E885:n_up_down_hub:F; I:at_i*e99*n_down_hub:E990; E990:n_up_down_hub:F;\
I:at_i*e99*n_down_hub:E991; E991:m1*n_up_down_hub:F; I:at_i*e99*m1*n_up_down_hub:E992; E992:n_up_down_hub:F;\
I:at_i*e99*m1*n_up_down_hub:E993; E993:m1*n_up_down_hub:F; I:at_i*eAA*n_down_hub:EAA0; EAA0:n_down_hub:F;\
I:at_i*eAA*n_down_hub:EAA1; EAA1:m1*n_up_down_hub:F; I:at_i*eAA*m1*n_up_down_hub:EAA2; EAA2:n_down_hub:F;\
I:at_i*eAA*m1*n_up_down_hub:EAA3; EAA3:m1*n_up_down_hub:F; I:at_i*eBB*n_down_hub:EBB0; EBB0:n_up_down_hub:F;\
I:at_i*eBB*m1*n_up_down_hub:EBB1; EBB1:n_up_down_hub:F; I:at_i*eCC*n_up_down_hub:ECC0; ECC0:Id:F;\
I:at_i*eCC*n_up_down_hub:ECC1; ECC1:m1*n_hub:F; I:at_i*eCC*n_up_down_hub:ECC2; ECC2:n_up_down_hub:F;\
I:at_i*eDD*n_up_down_hub:EDD0; EDD0:n_up_down_hub:F; I:at_i*eDD*n_up_down_hub:EDD1; EDD1:m1*n_up_down_hub:F;\
I:at_i*eEE*n_up_down_hub:EEE0; EEE0:n_down_hub:F; I:at_i*eEE*n_up_down_hub:EEE1; EEE1:m1*n_up_down_hub:F;\
I:at_i*eFF*n_up_down_hub:EFF0; EFF0:n_up_down_hub:F;\
I:at_i*m1*e14*c_up_hub*n_up_hub:E140; E140:c_dagger_up_hub*Id:F;\
I:at_i*m1*e14*c_up_hub*n_up_hub:E141; E141:m1*c_dagger_up_hub*n_down_hub:F;\
I:at_i*m1*e14*m1*c_up_hub*n_up_down_hub:E142; E142:c_dagger_up_hub*Id:F;\
I:at_i*m1*e14*m1*c_up_hub*n_up_down_hub:E143; E143:m1*c_dagger_up_hub*n_down_hub:F;\
I:at_i*e41*c_dagger_up_hub*Id:E410; E410:c_up_hub*n_up_down_hub:F;\
I:at_i*e41*c_dagger_up_hub*Id:E411; E411:m1*c_up_hub*n_up_down_hub:F;\
I:at_i*e41*m1*c_dagger_up_hub*n_down_hub:E412; E412:c_up_hub*n_up_down_hub:F;\
I:at_i*e41*m1*c_dagger_up_hub*n_down_hub:E413; E413:m1*c_up_hub*n_up_down_hub:F;\
I:at_i*m1*e28*c_down_hub*n_down_hub:E280; E280:c_dagger_down_hub*Id:F;\
I:at_i*m1*e28*c_down_hub*n_down_hub:E281; E281:m1*c_dagger_down_hub*n_up_down_hub:F;\
I:at_i*m1*e28*m1*c_down_hub*n_up_down_hub:E282; E282:c_dagger_down_hub*Id:F;\
I:at_i*m1*e28*m1*c_down_hub*n_up_down_hub:E283; E283:m1*c_dagger_down_hub*n_up_down_hub:F;\
I:at_i*e82*c_dagger_down_hub*Id:E820; E820:c_down_hub*n_down_hub:F;\
I:at_i*e82*c_dagger_down_hub*Id:E821; E821:m1*c_down_hub*n_up_down_hub:F;\
I:at_i*e82*m1*c_dagger_down_hub*n_up_down_hub:E822; E822:c_down_hub*n_down_hub:F;\
I:at_i*e82*m1*c_dagger_down_hub*n_up_down_hub:E823; E823:m1*c_down_hub*n_up_down_hub:F;\
I:at_i*m1*e36*c_up_hub*n_up_down_hub:E360; E360:c_dagger_up_hub*n_down_hub:F;\
I:at_i*m1*e36*m1*c_up_hub*n_up_down_hub:E361; E361:c_dagger_up_hub*n_down_hub:F;\
I:at_i*e63*c_dagger_up_hub*Id:E630; E630:c_up_hub*n_up_down_hub:F;\
I:at_i*e63*m1*c_dagger_up_hub*n_down_hub:E631; E631:c_up_hub*n_down_hub:F;\
I:at_i*m1*e39*c_down_hub*n_down_hub:E390; E390:c_dagger_down_hub*n_up_down_hub:F;\
I:at_i*m1*e39*m1*c_down_hub*n_up_down_hub:E391; E391:c_dagger_down_hub*n_up_down_hub:F;\
I:at_i*e93*c_dagger_down_hub*Id:E930; E930:c_down_hub*n_up_down_hub:F;\
I:at_i*e93*m1*c_dagger_down_hub*n_up_down_hub:E931; E931:c_down_hub*n_up_down_hub:F;\
I:at_i*m1*e3C*c_down_hub*c_up_hub:E3C0; E3C0:c_dagger_down_hub*c_dagger_up_hub:F;\
I:at_i*e3C*c_dagger_down_hub*c_dagger_up_hub:E3C0; E3C0:c_down_hub*c_up_hub:F;\
I:at_i*e69*c_dagger_down_hub*c_up_hub:E690; E690:c_dagger_up_hub*c_down_hub:F;\
I:at_i*e96*c_dagger_up_hub*c_down_hub:E960; E960:c_dagger_down_hub*c_up_hub:F;\
I:at_i*m1*e6C*c_down_hub*n_up_down_hub:E6C0; E6C0:c_dagger_down_hub*Id:F;\
I:at_i*m1*e6C*c_down_hub*n_up_down_hub:E6C1; E6C1:m1*c_dagger_down_hub*n_up_down_hub:F;\
I:at_i*e6C*c_dagger_down_hub*n_up_down_hub:E6C0; E6C0:c_down_hub*n_down_hub:F;\
I:at_i*e6C*c_dagger_down_hub*n_up_down_hub:E6C1; E6C1:m1*c_down_hub*n_up_down_hub:F;\
I:at_i*m1*e7D*c_down_hub*n_up_down_hub:E7D0; E7D0:c_dagger_down_hub*n_up_down_hub:F;\
I:at_i*eD7*c_dagger_down_hub*n_up_down_hub:E7D0; E7D0:c_down_hub*n_up_down_hub:F;\
I:at_i*m1*e9C*c_up_hub*n_up_down_hub:E9C0; E9C0:c_dagger_up_hub*Id:F;\
I:at_i*m1*e9C*c_up_hub*n_up_down_hub:E9C1; E9C1:m1*c_dagger_up_hub*n_down_hub:F;\
I:at_i*e9C*c_dagger_up_hub*n_down_hub:E9C0; E9C0:c_up_hub*n_up_down_hub:F;\
I:at_i*e9C*c_dagger_up_hub*n_down_hub:E9C1; E9C1:m1*c_up_hub*n_up_down_hub:F;\
I:at_i*m1*eBE*c_up_hub*n_up_down_hub:EBE0; EBE0:c_dagger_up_hub*n_down_hub:F;\
I:at_i*eEB*c_dagger_up_hub*n_down_hub:EEB0; EEB0:c_up_hub*n_up_down_hub:F;\
set nodes = I; F; \
E000; E001; E002; E003; E004; E005; E006; E007; E008; E110; E111; E112; E113; E114; E115; E220; E221; E222; E223;\
E224; E225; E330; E331; E332; E440; E441; E442; E443; E444; E445; E550; E551; E552; E553; E660; E661; E662; E663;\
E770; E771; E880; E881; E882; E883; E884; E885; E990; E991; E992; E993; EAA0; EAA1; EAA2; EAA3; EBB0; EBB1; ECC0;\
ECC1; ECC2; EDD0; EDD1; EEE0; EEE1; EFF0; E140; E141; E142; E143; E410; E411; E412; E413; E280; E281; E282; E283;\
E820; E821; E822; E823; E360; E361; E630; E631; E390; E391; E930; E931; E3C0; E3C0; E690; E960; E6C0; E6C1; E7D0; ED70; E9C0; EC90; EC91; EBE0; EEB0;\
set print_ignore = Id
set weight_functions = m1: {-1.0}; oh: {0.5}; at_i: {if( (position == i), 1, 0)}; \
e00: {e00r + e00i*imag_i}; e11: {e11r + e11i*imag_i}; e14: {e14r + e14i*imag_i}; e22: {e22r + e22i*imag_i};\
e28: {e28r + e28i*imag_i}; e33: {e33r + e33i*imag_i}; e36: {e36r + e36i*imag_i}; e39: {e39r + e39i*imag_i};\
e3C: {e3Cr + e3Ci*imag_i}; e41: {e41r + e41i*imag_i}; e44: {e44r + e44i*imag_i}; e55: {e55r + e55i*imag_i};\
e63: {e63r + e63i*imag_i}; e66: {e66r + e66i*imag_i}; e69: {e69r + e69i*imag_i}; e6C: {e6Cr + e6Ci*imag_i};\
e77: {e77r + e77i*imag_i}; e7D: {e7Dr + e7Di*imag_i}; e82: {e82r + e82i*imag_i}; e88: {e88r + e88i*imag_i};\
e93: {e93r + e93i*imag_i}; e96: {e96r + e96i*imag_i}; e99: {e99r + e99i*imag_i}; e9C: {e9Cr + e9Ci*imag_i};\
eAA: {eAAr + eAAi*imag_i}; eBB: {eBBr + eBBi*imag_i}; eBE: {eBEr + eBEi*imag_i}; eC3: {eC3r + eC3i*imag_i};\
eC6: {eC6r + eC6i*imag_i}; eC9: {eC9r + eC9i*imag_i}; eCC: {eCCr + eCCi*imag_i}; eD7: {eD7r + eD7i*imag_i};\
eDD: {eDDr + eDDi*imag_i}; eEB: {eEBr + eEBi*imag_i}; eEE: {eEEr + eEEi*imag_i}; eFF: {eFFr + eFFi*imag_i};
end

```

## Heisenberg Trotter Gate

B

In this appendix, we present some code snippets used for the Trotter-time-evolution in Sec. 10.4.5.

### Exact Heisenberg Trotter Gate

B.1

In this part of App. B, we show the Mathematica code to obtain the operator for a single Trotter-time-evolution step.

```
H = {{Jz, 0, 0, 0}, {0, -Jz, J, 0}, {0, J, -Jz, 0}, {0, 0, 0, Jz}}
d = DiagonalMatrix[Eigenvalues[H]]
d = MatrixExp[-I d dt]
U = Eigenvectors[H]
Transpose[U].d.U
```

### Numeric Heisenberg Trotter Gate

B.2

```
int main(int argc, char* argv[])
{
    typedef SciPAL::CudaComplex<double> T;
    T imag_i(0.0, 1.0);

    unsigned int systemsize = 16;
    double timestepwidth = 0.01;
    double time = 0.0;
    unsigned int timesteps = 1500;
    int initial_exc = 5;

    std::vector<T> J(systemsize);
    std::vector<T> Jz(systemsize);
    std::vector<T> hz(systemsize);

    for (unsigned int i = 0; i < systemsize; i++)
    {
        J[i] = 1.0;
        Jz[i] = 0.0;
        hz[i] = 0.0;
    }

    auto meas = [&]
    {
        std::cout << "meas_Sz[];\nperf_write[obs,Sz];" << std::endl;
    };

    auto apply_bond = [&] (double dt, unsigned int i)
```

```

{
    unsigned int n_rows = 4, n_cols = 4;
    SciPAL::Matrix<T, blas> h(n_rows, n_cols);
    SciPAL::Matrix<T, blas> D(n_rows, n_cols);
    SciPAL::Vector<double, blas> e_val(n_rows);
    SciPAL::Matrix<T, blas> tmp(n_rows, n_cols);

    T half_i = (i==0) ? 1.0 : 0.5;
    T half_ipo = ((i+1)==(systemsize-1)) ? 1.0 : 0.5;

    /// Build up bond matrix
    h(0, 0, Jz[i]*0.25 + half_i*hz[i]*0.5 + half_ipo*hz[i+1]*0.5);

    h(1, 1, -Jz[i]*0.25 + half_i*hz[i]*0.5 - half_ipo*hz[i+1]*0.5);
    h(1, 2, J[i]*0.5);

    h(2, 1, J[i]*0.5);
    h(2, 2, -Jz[i]*0.25 - half_i*hz[i]*0.5 + half_ipo*hz[i+1]*0.5);

    h(3, 3, Jz[i]*0.25 - half_i*hz[i]*0.5 - half_ipo*hz[i+1]*0.5);

    lapack::Eigen(h, e_val);

    for (unsigned d = 0; d < 4; d++)
        D(d, d, exp(-e_val[d] * dt * imag_i));

    tmp = h * D;
    D = tmp * SciPAL::adjoint(h);

    std::cout << "app_full_twosite_op[" << i << ", " << std::flush;

    for (unsigned int c = 0; c < n_cols; c++)
        for (unsigned int r = 0; r < n_rows; r++)
            std::cout << SciPAL::real(D(r, c)) << ", "
                << SciPAL::imag(D(r, c))
                << ((r==n_rows-1 && c==n_cols-1)?"");\n":", "
                << std::flush;
};

auto even = [&] (double dt)
{
    for (unsigned int i = 0; i < systemsize-1; i += 2)
    {
        apply_bond(dt, i);
    }
};

auto odd = [&] (double dt)
{
    for (unsigned int i = 1; i < systemsize-1; i += 2)
    {
        apply_bond(dt, i);
    }
};

auto first_order_trotter_step = [&] (double dt)
{

```

```
        even(dt);
        odd(dt);
        time += dt;
    };

    auto second_order_trotter_step = [&] (double dt)
    {
        even(dt*0.5);
        odd(dt);
        even(dt*0.5);
        time += dt;
    };

    auto fourth_order_trotter_step = [&] (double dt)
    {
        double A = 4.0 - std::pow(4.0, 1.0/3.0);

        even(1.0/(2.0 * A) * dt);
        odd(1.0/A * dt);
        even(1.0/A * dt);
        odd(1.0/A * dt);
        even((A - 3.0)/(2.0 * A) * dt);
        odd((A - 4.0)/A * dt);
        even((A - 3.0)/(2.0 * A) * dt);
        odd(1.0/A * dt);
        even(1.0/A * dt);
        odd(1.0/A * dt);
        even(1.0/(2.0 * A) * dt);
        time += dt;
    };

    std::cout << "app_X[" << initial_exc << "]" << std::endl;

    meas();
    for (unsigned int t = 0; t < timesteps; t++)
    {
        second_order_trotter_step(timestepwidth);

        meas();
        std::cout << std::endl;
    }
}
```





## References in Alphabetical Order of Labels

- [AAN<sup>+</sup>11] M. Aidelsburger, M. Atala, S. Nascimbène, S. Trotzky, Y.-A. Chen, and I. Bloch. Experimental realization of strong effective magnetic fields in an optical lattice. *Phys. Rev. Lett.*, 107:255301, Dec 2011.
- [AAN<sup>+</sup>13] M. Aidelsburger, M. Atala, S. Nascimbène, S. Trotzky, Y.-A. Chen, and I. Bloch. Experimental realization of strong effective magnetic fields in optical superlattice potentials. *Applied Physics B*, 113(1):1–11, Oct 2013.
- [AHRDA13] K. A. Al-Hassanieh, J. Rincón, E. Dagotto, and G. Alvarez. Wave-packet dynamics in the one-dimensional extended Hubbard model. *Phys. Rev. B*, 88:045107, Jul 2013.
- [AHRF<sup>+</sup>08] K. A. Al-Hassanieh, F. A. Reoredo, A. E. Feiguin, I. González, and E. Dagotto. Excitons in the one-dimensional Hubbard model: A real-time study. *Phys. Rev. Lett.*, 100:166403, Apr 2008.
- [AKLT87] I. Affleck, T. Kennedy, E. H. Lieb, and H. Tasaki. Rigorous results on valence-bond ground states in antiferromagnets. *Phys. Rev. Lett.*, 59:799–802, Aug 1987.
- [ALB<sup>+</sup>07] M. Anderlini, P. J. Lee, B. L. Brown, J. Sebby-Strabley, W. D. Phillips, and J. V. Porto. Controlled exchange interaction between pairs of neutral atoms in an optical lattice. *Nature*, 448:452, Jul 2007.
- [Bar16] T. Barthel. Matrix product purifications for canonical ensembles and quantum number distributions. *Phys. Rev. B*, 94:115157, Sep 2016.
- [BBC<sup>+</sup>95] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter. Elementary gates for quantum computation. *Phys. Rev. A*, 52:3457–3467, Nov 1995.
- [BBFN98] G. Bedürftig, B. Brendel, H. Frahm, and R. M. Noack. Friedel oscillations in the open Hubbard chain. *Phys. Rev. B*, 58:10225, Oct 1998.
- [BDD<sup>+</sup>00] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, editors. *Templates for the Solution of Algebraic Eigenvalue Problems: a Practical Guide*. SIAM, 2000.
- [BDS14] V. Bach and L. Delle Site. *Many-Electron Approaches in Physics, Chemistry and Mathematics: A Multidisciplinary View*. Springer, 2014.
- [BDZ08] I. Bloch, J. Dalibard, and W. Zwerger. Many-body physics with ultracold gases. *Rev. Mod. Phys.*, 80:885, Jun 2008.
- [Beb00] M. Bebendorf. Approximation of boundary element matrices. *Numerische Mathematik*, 86(4):565–589, Oct 2000.
- [Bet31] H. Bethe. Zur Theorie der Metalle. *Zeitschrift für Physik*, 71(3-4):205–226, 1931.
- [BGP<sup>+</sup>09] W. S. Bakr, J. I. Gillen, A. Peng, S. Fölling, and M. Greiner. A quantum gas microscope for detecting single atoms in a Hubbard-regime optical lattice. *Nature*, 462:74, Nov 2009.

- [BHS<sup>+</sup>16] M. Boll, T. A. Hilker, G. Salomon, A. Omran, J. Nespolo, L. Pollet, I. Bloch, and C. Gross. Spin- and density-resolved microscopy of antiferromagnetic correlations in Fermi-Hubbard chains. *Science*, 353(6305):1257–1260, 2016.
- [Bie14] S. Biermann. Dynamical screening effects in correlated electron materials – A progress report on combined many-body perturbation and dynamical mean field theory: ‘GW + DMFT’. *Journal of Physics: Condensed Matter*, 26(17):173202, 2014.
- [Bie16] F. R. A. Biebl. *Thermalization in one-dimensional quantum-many-body systems*. PhD thesis, University of Göttingen, 2016.
- [BKT<sup>+</sup>17] J. Becker, T. Köhler, A. C. Tiegel, S. R. Manmana, S. Wessel, and A. Honecker. Finite-temperature dynamics and thermal intraband magnon scattering in Haldane spin-one chains. *Phys. Rev. B*, 96:060403, Aug 2017.
- [Blo05] I. Bloch. Ultracold quantum gases in optical lattices. *Nature Physics*, 1(1):23–30, 2005.
- [BR03] M. Bebendorf and S. Rjasanow. Adaptive low-rank approximation of collocation matrices. *Computing*, 70(1):1–24, Feb 2003.
- [CB08] G. M. Crosswhite and D. Bacon. Finite automata for caching in matrix product algorithms. *Phys. Rev. A*, 78:012356, Jul 2008.
- [CBSG17] A. W. Cross, L. S. Bishop, J. A. Smolin, and J. M. Gambetta. Open Quantum Assembly Language. *ArXiv e-prints*, 1707.03429, Jul 2017.
- [CDC<sup>+</sup>18] L. Cevolani, J. Despres, G. Carleo, L. Tagliacozzo, and L. Sanchez-Palencia. Universal scaling laws for correlation spreading in quantum systems with short- and long-range interactions. *Phys. Rev. B*, 98:024302, Jul 2018.
- [CDV08] G. M. Crosswhite, A. C. Doherty, and G. Vidal. Applying matrix product operators to model systems with long-range interactions. *Phys. Rev. B*, 78:035116, Jul 2008.
- [CEP<sup>+</sup>18] P. J. Coles, S. Eidenbenz, S. Pakin, A. Adedoyin, J. Ambrosiano, P. Anisimov, W. Casper, G. Chennupati, C. Coffrin, H. Djidjev, D. Gunter, S. Karra, N. Lemons, S. Lin, A. Lokhov, A. Malyzhenkov, D. Mascarenas, S. Mniszewski, B. Nadiga, D. O’Malley, D. Oyen, L. Prasad, R. Roberts, P. Romero, N. Santhi, N. Sinitsyn, P. Swart, M. Vuffray, J. Wendelberger, B. Yoon, R. Zamora, and W. Zhu. Quantum Algorithm Implementations for Beginners. *ArXiv e-prints*, 1804.03719, Apr 2018.
- [CNO<sup>+</sup>15] L. W. Cheuk, M. A. Nichols, M. Okan, T. Gersdorf, V. V. Ramasesh, W. S. Bakr, T. Lompe, and M. W. Zwierlein. Quantum-gas microscope for fermionic atoms. *Phys. Rev. Lett.*, 114:193001, May 2015.
- [CW02] J. K. Cullum and R. A. Willoughby. *Lanczos Algorithms for Large Symmetric Eigenvalue Computations*. Society for Industrial and Applied Mathematics, 2002.
- [DdSAHF<sup>+</sup>10] L. G. G. V. Dias da Silva, K. A. Al-Hassanieh, A. E. Feiguin, F. A. Reboredo, and E. Dagotto. Real-time dynamics of particle-hole excitations in Mott insulator-metal junctions. *Phys. Rev. B*, 81(12):125113, Mar 2010.
- [DHH17] A. Dang, C. D. Hill, and L. C. L. Hollenberg. Optimising matrix product state simulations of Shor’s algorithm. *ArXiv e-prints*, 1712.07311, Dec 2017.
- [DKSV04] A. J. Daley, C. Kollath, U. Schollwöck, and G. Vidal. Time-dependent density-matrix renormalization-group using adaptive effective Hilbert spaces. *J. Stat. Mech.*, 2004(04):P04005, 2004.

- [DVB<sup>+</sup>15] F. Dorfner, L. Vidmar, C. Brockett, E. Jeckelmann, and F. Heidrich-Meisner. Real-time decay of a highly excited charge carrier in the one-dimensional Holstein model. *Phys. Rev. B*, 91:104302, Mar 2015.
- [DVS18] J. Despres, L. Villa, and L. Sanchez-Palencia. Twofold correlation spreading in a strongly correlated lattice Bose gas. *ArXiv e-prints*, 1812.05029, December 2018.
- [DWa] Quantum computing demo announcement. <https://web.archive.org/web/20110718092907/http://dwave.wordpress.com/2007/01/19/quantum-computing-demo-announcement/>. Last accessed 2018-12-10.
- [DWH<sup>+</sup>12] P. E. Dargel, A. Wöllert, A. Honecker, I. P. McCulloch, U. Schollwöck, and T. Pruschke. Lanczos algorithm with matrix product states for dynamical correlation functions. *Phys. Rev. B*, 85:205119, May 2012.
- [EFG<sup>+</sup>05] F. H. L. Essler, H. Frahm, F. Göhmann, A. Klümper, and V. E. Korepin. *The One-Dimensional Hubbard Model*. Cambridge University Press, Cambridge, 2005.
- [EPR35] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, 47:777–780, May 1935.
- [Ess] F.H.L. Essler. Statistical mechanics and phase transitions. <http://www-thphys.physics.ox.ac.uk/people/FabianEssler/C6web2012/StatMech.pdf>. Last accessed 2018-12-10.
- [EW13] M. Eckstein and P. Werner. Nonequilibrium dynamical mean-field simulation of inhomogeneous systems. *Phys. Rev. B*, 88:075135, Aug 2013.
- [Fey82] R. P. Feynman. Simulating physics with computers. *Int. J. Theor. Phys.*, 21:467–488, 1982.
- [FMNR01] W. M. C. Foulkes, L. Mitas, R. J. Needs, and G. Rajagopal. Quantum Monte Carlo simulations of solids. *Rev. Mod. Phys.*, 73:33–83, Jan 2001.
- [FW15] M. T. Fishman and S. R. White. Compression of correlation matrices and an efficient method for forming matrix product states of fermionic Gaussian states. *Phys. Rev. B*, 92:075132, Aug 2015.
- [GB17] C. Gross and I. Bloch. Quantum simulations with ultracold atoms in optical lattices. *Science*, 357(6355):995–1001, 2017.
- [Geb97] F. Gebhard. *The Mott Metal-Insulator Transition. Models and Methods*. Springer, 1 edition, 1997.
- [Gel17] P. Gelß. *The Tensor-Train Format and Its Applications*. PhD thesis, Free University Berlin, 2017.
- [Gia04] T. Giamarchi. *Quantum Physics in One Dimension*. International Series of Monographs on Physics. Oxford University Press, Oxford, 2004.
- [GK92] A. Georges and G. Kotliar. Hubbard model in infinite dimensions. *Phys. Rev. B*, 45:6479–6483, Mar 1992.
- [Got98] D. Gottesman. The Heisenberg representation of quantum computers. In *Group theoretical methods in physics. Proceedings, 22nd International Colloquium, Group22, ICGTMP'98, Hobart, Australia, Jul 13-17, 1998*, pages 32–43, 1998.
- [GR06] J. J. García-Ripoll. Time evolution of matrix product states. *New Journal of Physics*, 8(12):305, 2006.
- [Gre00] W. Greiner. *Relativistic Quantum Mechanics. Wave equations*. Springer, Berlin, Heidelberg, 3 edition, 2000.

- [GTU<sup>+</sup>11] D. Greif, L. Tarruell, T. Uehlinger, R. Jördens, and T. Esslinger. Probing nearest-neighbor correlations of ultracold fermions in an optical lattice. *Phys. Rev. Lett.*, 106:145302, Apr 2011.
- [Gut63] M. C. Gutzwiller. Effect of correlation on the ferromagnetism of transition metals. *Phys. Rev. Lett.*, 10:159–162, Mar 1963.
- [Hal83] F. D. M. Haldane. Nonlinear field theory of large-spin Heisenberg antiferromagnets: Semiclassically quantized solitons of the one-dimensional easy-axis Néel state. *Phys. Rev. Lett.*, 50:1153–1156, Apr 1983.
- [Hal17] F. D. M. Haldane. Nobel lecture: Topological quantum matter. *Rev. Mod. Phys.*, 89:040502, Oct 2017.
- [Has07] M. B. Hastings. An area law for one-dimensional quantum systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(08):P08024, 2007.
- [HCO<sup>+</sup>11] J. Haegeman, J. I. Cirac, T. J. Osborne, I. Pižorn, H. Verschelde, and F. Verstraete. Time-dependent variational principle for quantum lattices. *Phys. Rev. Lett.*, 107:070601, Aug 2011.
- [hei] “Perzeptron”: Einfaches neuronales Netz mittels Quantencomputer. <https://heise.de/-4236505>. Last accessed 2018-12-05.
- [Hei28] W. Heisenberg. Zur Theorie des Ferromagnetismus. *Zeitschrift für Physik*, 49(9-10):619–636, 1928.
- [HK64] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136:B864–B871, Nov 1964.
- [HLO<sup>+</sup>16] J. Haegeman, C. Lubich, I. Oseledets, B. Vandereycken, and F. Verstraete. Unifying time evolution and optimization with matrix product states. *Phys. Rev. B*, 94:165116, Oct 2016.
- [HMS17] C. Hubig, I. P. McCulloch, and U. Schollwöck. Generic construction of efficient matrix product operators. *Phys. Rev. B*, 95:035129, Jan 2017.
- [HP12] F. Hofmann and M. Potthoff. Doublyon dynamics in the extended Fermi-Hubbard model. *Phys. Rev. B*, 85:205127, May 2012.
- [Hub63] J. Hubbard. Electron correlations in narrow energy bands. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 276(1365):238–257, 1963.
- [Hub64a] J. Hubbard. Electron correlations in narrow energy bands. ii. the degenerate band case. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 277(1369):237–259, 1964.
- [Hub64b] J. Hubbard. Electron correlations in narrow energy bands. iii. an improved solution. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 281(1386):401–419, 1964.
- [Hub65] J. Hubbard. Electron correlations in narrow energy bands. iv. the atomic representation. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 285(1403):542–560, 1965.
- [Hub17] C. Hubig. *Symmetry-protected tensor networks*. PhD thesis, Ludwig-Maximilians-Universität München, 10 2017.
- [Hå90] J. Håstad. Tensor rank is NP-complete. *Journal of Algorithms*, 11(4):644 – 654, 1990.
- [IBM] IBM Quantum Experience. <http://www.research.ibm.com/quantum>. Last accessed 2018-10-18.

- [IEC08] IEC 80000-13:2008(en), Quantities and units — Part 13: Information science and technology, Mar 2008. [Online; accessed 1. Dec. 2018].
- [IEE08] IEEE Task P754. *IEEE 754-2008, Standard for Floating-Point Arithmetic*. IEEE, New York, NY, USA, Aug 2008.
- [Ite01] <https://www.iter.org/proj/inafewlines>, Last accessed 2018-12-01.
- [Jae98] G. Jaeger. The Ehrenfest classification of phase transitions: Introduction and evolution. *Archive for History of Exact Sciences*, 53(1):51–81, May 1998.
- [JAG<sup>+</sup>11] M. W. Johnson, M. H. S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, E. M. Chapple, C. Enderud, J. P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M. C. Thom, E. Tolkacheva, C. J. S. Truncik, S. Uchaikin, J. Wang, B. Wilson, and G. Rose. Quantum annealing with manufactured spins. *Nature*, 473(7346):194, May 2011.
- [JKŠ<sup>+</sup>85] Z. Jiráček, S. Krupička, Z. Šimša, M. Dlouhá, and S. Vratislav. Neutron diffraction study of  $\text{Pr}_{1-x}\text{Ca}_x\text{MnO}_3$  perovskites. *Journal of Magnetism and Magnetic Materials*, 53:153–166, November 1985.
- [JL03] R. Jozsa and N. Linden. On the role of entanglement in quantum-computational speed-up. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 459(2036):2011–2032, 2003.
- [Jos62] B. D. Josephson. Possible new effects in superconductive tunnelling. *Physics Letters*, 1(7):251 – 253, 1962.
- [Jos74] B. D. Josephson. The discovery of tunnelling supercurrents. *Rev. Mod. Phys.*, 46:251–254, Apr 1974.
- [Kan63] J. Kanamori. Electron correlation and ferromagnetism of transition metals. *Progress of Theoretical Physics*, 30(3):275–289, 1963.
- [KIMG06] C. Kollath, A. Iucci, I. P. McCulloch, and T. Giamarchi. Modulation spectroscopy with ultracold fermions in an optical lattice. *Phys. Rev. A*, 74:041604, Oct 2006.
- [KPM18] T. Köhler, S. Paeckel, and S. R. Manmana. Charge density patterns in spin-selectively photoexcited interacting fermions. *ArXiv e-prints*, 1808.02138, Aug 2018.
- [Kra13] S. C. Kramer. *CUDA-based Scientific Computing: Tools and Selected Applications*. Niedersächsische Staats- und Universitätsbibliothek Göttingen, 2013.
- [KRS<sup>+</sup>18] T. Köhler, S. Rajpurohit, O. Schumann, S. Paeckel, F. R. A. Biebl, M. Sotoudeh, S. C. Kramer, P. E. Blöchl, S. Kehrein, and S. R. Manmana. Relaxation of photoexcitations in polaron-induced magnetic microstructures. *Phys. Rev. B*, 97:235120, Jun 2018.
- [Lan50] C. Lanczos. Electromagnetic fields and dielectric response in empirical tight-binding theory. *J. Res. Nat. Bur. Standards*, 45:255–282, 1950.
- [LaR18] R. LaRose. Overview and comparison of gate level quantum software platforms. *ArXiv e-prints*, 1807.02500, Jul 2018.
- [lbn] Berkeley Lab to Build an Advanced Quantum Computing Testbed. <http://cs.lbl.gov/news-media/news/2018/berkeley-lab-to-build-an-advanced-quantum-computing-testbed/>. Last accessed 2018-12-05.
- [LEP15] Z. Lenarčič, M. Eckstein, and P. Prelovšek. Exciton recombination in one-dimensional organic Mott insulators. *Phys. Rev. B*, 92:201104, Nov 2015.

- [Lis] List of published papers that use the IBM quantum experience. <https://quantumexperience.ng.bluemix.net/qx/community?channel=papers>. Last accessed 2018-12-10.
- [LP13] Z. Lenarčič and P. Prelovšek. Ultrafast charge recombination in a photoexcited Mott-Hubbard insulator. *Phys. Rev. Lett.*, 111:016401, Jul 2013.
- [LP14] Z. Lenarčič and P. Prelovšek. Charge recombination in undoped cuprates. *Phys. Rev. B*, 90:235136, Dec 2014.
- [LR72] E. H. Lieb and D. W. Robinson. The finite group velocity of quantum spin systems. *Commun. Math. Phys.*, 28(3):251–257, Sep 1972.
- [LSZ<sup>+</sup>15] M. Lohse, C. Schweizer, O. Zilberberg, M. Aidelsburger, and I. Bloch. A Thouless quantum pump with ultracold bosonic atoms in an optical superlattice. *Nature Physics*, 12:350, Dec 2015.
- [LW68] E. H. Lieb and F. Y. Wu. Absence of Mott transition in an exact solution of the short-range, one-band model in one dimension. *Phys. Rev. Lett.*, 20:1445–1448, Jun 1968.
- [Man80] Y. I. Manin. The computable and not computable, 1980.
- [Man10] E. Manousakis. Photovoltaic effect for narrow-gap Mott insulators. *Phys. Rev. B*, 82:125109, Sep 2010.
- [Man17] S. R. Manmana. *Quantum Magnetism, Nonequilibrium Dynamics and Quantum Simulation of Correlated Quantum Systems*. Habilitation treatise, Georg-August-Universität Göttingen, May 2017.
- [MBE15] J. H. Mentink, K. Balzer, and M. Eckstein. Ultrafast and reversible control of the exchange interaction in Mott insulators. *Nature Communications*, 6:6708, Mar 2015.
- [McC07] I. P. McCulloch. From density-matrix renormalization group to matrix product states. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(10):P10014, 2007.
- [MDU<sup>+</sup>15] M. Messer, R. Desbuquois, T. Uehlinger, G. Jotzu, S. Huber, D. Greif, and T. Esslinger. Exploring competing density order in the ionic Hubbard model with ultracold fermions. *Phys. Rev. Lett.*, 115:115303, Sep 2015.
- [Moo65] G. E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8):114–117, 1965.
- [Moo17] G. E. Moore. Gordon Moore: The man whose name means progress, the visionary engineer reflects on 50 years of Moore’s law. *IEEE Spectrum: Special Report*, 50, 2017.
- [MS12] I. L. Markov and M. Saeedi. Constant-optimized quantum circuits for modular multiplication and exponentiation. *Quantum Info. Comput.*, 12(5-6):361–394, May 2012.
- [MW66] N. D. Mermin and H. Wagner. Absence of ferromagnetism or antiferromagnetism in one- or two-dimensional isotropic Heisenberg models. *Phys. Rev. Lett.*, 17:1133–1136, Nov 1966.
- [NA16] A. Nocera and G. Alvarez. Symmetry-conserving purification of quantum states within the density matrix renormalization group. *Phys. Rev. B*, 93:045137, Jan 2016.
- [NC04] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 1 edition, Jan 2004.



- [NR09] W. Nolting and A. Ramakanth. *Quantum Theory of Magnetism*. Springer Science & Business Media, 2009.
- [OBH<sup>+</sup>15] A. Omran, M. Boll, T. A. Hilker, K. Kleinlein, G. Salomon, I. Bloch, and C. Gross. Microscopic observation of Pauli blocking in degenerate fermionic lattice gases. *Phys. Rev. Lett.*, 115:263001, Dec 2015.
- [Ope] OpenQASM. <https://github.com/IBM/qiskit-openqasm>. Last accessed 2018-10-19.
- [Ose09] I. V. Oseledets. A new tensor decomposition. *Doklady Mathematics*, 80(1):495–496, Aug 2009.
- [Pei33] R. E. Peierls. On the theory of the diamagnetism of conduction electrons. *Z. Phys.*, 80:763–791, 1933.
- [PGD13] N. Proukakis, S. Gardiner, and M. Davis, editors. *Quantum Gases: Finite Temperatures and Non-Equilibrium Dynamics*. Imperial College Press, London, 2013.
- [PHM<sup>+</sup>15] M. F. Parsons, F. Huber, A. Mazurenko, C. S. Chiu, W. Setiawan, K. Wooley-Brown, S. Blatt, and M. Greiner. Site-resolved imaging of fermionic <sup>6</sup>Li in an optical lattice. *Phys. Rev. Lett.*, 114:213002, May 2015.
- [PKM17] S. Paeckel, T. Köhler, and S. R. Manmana. Automated construction of  $U(1)$ -invariant matrix-product operators from graph representations. *SciPost Phys.*, 3:035, 2017.
- [PKS<sup>+</sup>19] S. Paeckel, T. Köhler, A. Swoboda, S. R. Manmana, U. Schollwöck, and C. Hubig. Time-evolution methods for matrix-product states. *ArXiv e-prints*, 1901.05824, Jan 2019.
- [PMS<sup>+</sup>14] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O’Brien. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.*, 5:4213, Jul 2014.
- [PPT<sup>+</sup>03] S. Peil, J. V. Porto, B. Laburthe Tolra, J. M. Obrecht, B. E. King, M. Subbotin, S. L. Rolston, and W. D. Phillips. Patterned loading of a Bose-Einstein condensate into an optical lattice. *Phys. Rev. A*, 67:051603, May 2003.
- [Pre12] J. Preskill. Quantum computing and the entanglement frontier. *ArXiv e-prints*, 1203.5813, Mar 2012.
- [PTVF07] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 3 edition, #sep# 2007.
- [PVE03] <https://www.nrel.gov/pv/assets/pdfs/pv-efficiencies-07-17-2018.pdf>, Last accessed 2018-12-03.
- [PWKH99] I. Peschel, X. Wang, M. Kaulke, and K. Hallberg, editors. *Density Matrix Renormalization - A New Numerical Method in Physics*. Springer Verlag, Berlin, 1999.
- [RDO08] M. Rigol, V. Dunjko, and M. Olshanii. Thermalization and its mechanism for generic isolated quantum systems. *Nature*, 452(7189):854, Apr 2008.
- [RO97] S. Rommer and S. Östlund. Class of ansatz wave functions for one-dimensional spin systems and their relation to the density matrix renormalization group. *Phys. Rev. B*, 55:2164–2181, Jan 1997.
- [Ros57] F. Rosenblatt. The perceptron a perceiving and recognizing automaton. *Cornell Aeronautical Laboratory, Buffalo, NY, USA, Tech. Rep*, pages 85–460, 1957.
- [RS12] M. Rigol and M. Srednicki. Alternatives to eigenstate thermalization. *Phys. Rev. Lett.*, 108:110601, Mar 2012.

- [Sac99] S. Sachdev. *Quantum Phase Transitions*. Cambridge University Press, Cambridge, 1999.
- [San10] A. W. Sandvik. Computational studies of quantum spin systems. *AIP Conference Proceedings*, 1297(1):135–338, 2010.
- [Sch26] E. Schrödinger. An undulatory theory of the mechanics of atoms and molecules. *Phys. Rev.*, 28:1049–1070, Dec 1926.
- [Sch35] E. Schrödinger. Discussion of probability relations between separated systems. *Mathematical Proceedings of the Cambridge Philosophical Society*, 31(4):555–563, 1935.
- [Sch05] U. Schollwöck. The density-matrix renormalization group. *Rev. Mod. Phys.*, 77:259–315, Apr 2005.
- [Sch11] U. Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96 – 192, Jan 2011.
- [Sch13] U. Schollwöck. Matrix product state algorithms: DMRG, TEBD and relatives. In A. Avella and F. Mancini, editors, *Strongly Correlated Systems - Numerical Methods*, chapter 3, pages 67–98. Springer, 2013.
- [Sch16] O. Schumann. Model study of light absorption in a manganite. Master’s thesis, Georg-August-Universität Göttingen, 2016.
- [SGJ<sup>+</sup>10] N. Strohmaier, D. Greif, R. Jördens, L. Tarruell, H. Moritz, T. Esslinger, R. Sensarma, D. Pekker, E. Altman, and E. Demler. Observation of elastic doublon decay in the Fermi-Hubbard model. *Phys. Rev. Lett.*, 104:080401, Feb 2010.
- [SGT<sup>+</sup>18] K. Svore, A. Geller, M. Troyer, J. Azariah, C. Granade, B. Heim, V. Kliuchnikov, M. Mykhailova, A. Paz, and M. Roetteler. Q#: Enabling scalable quantum computing and development with a high-level DSL. In *Proceedings of the Real World Domain Specific Languages Workshop 2018*, RWDSL2018, pages 7:1–7:10, New York, NY, USA, 2018. ACM.
- [Shi03] Y. Shi. Both Toffoli and controlled-NOT need little help to do universal quantum computing. *Quantum Info. Comput.*, 3(1):84–92, Jan 2003.
- [Sho94] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, Nov 1994.
- [Sho97] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.
- [SLCB16] C. Schweizer, M. Lohse, R. Citro, and I. Bloch. Spin pumping and measurement of spin currents in optical superlattices. *Phys. Rev. Lett.*, 117:170405, Oct 2016.
- [SPA<sup>+</sup>10] R. Sensarma, D. Pekker, E. Altman, E. Demler, N. Strohmaier, D. Greif, R. Jördens, L. Tarruell, H. Moritz, and T. Esslinger. Lifetime of double occupancies in the Fermi-Hubbard model. *Phys. Rev. B*, 82:224302, Dec 2010.
- [SRB<sup>+</sup>17] M. Sotoudeh, S. Rajpurohit, P. Blöchl, D. Mierwaldt, J. Norpoth, V. Roddatis, S. Mildner, B. Kressdorf, B. Iffland, and C. Jooss. Electronic structure of  $\text{Pr}_{1-x}\text{Ca}_x\text{MnO}_3$ . *Phys. Rev. B*, 95:235150, Jun 2017.
- [SSAJP06] J. Sebby-Strabley, M. Anderlini, P. S. Jessen, and J. V. Porto. Lattice of double wells for manipulating pairs of cold atoms. *Phys. Rev. A*, 73:033605, Mar 2006.
- [Suz76] M. Suzuki. Relationship between d-dimensional quantum spin systems and (d+1)-dimensional Ising systems: Equivalence, critical exponents and systematic ap-



- proximants of the partition function and spin correlations. *Progress of Theoretical Physics*, 56(5):1454–1469, 1976.
- [Suz91] M. Suzuki. General theory of fractal path integrals with applications to many-body theories and statistical physics. *Journal of Mathematical Physics*, 32:400–407, February 1991.
- [SW10] E. M. Stoudenmire and S. R. White. Minimally entangled typical thermal state algorithms. *New Journal of Physics*, 12(5):055026, 2010.
- [SW13] E. M. Stoudenmire and Steven R. White. Real-space parallel density matrix renormalization group. *Phys. Rev. B*, 87:155137, Apr 2013.
- [Tak72] M. Takahashi. One-dimensional Hubbard model at finite temperature. *Progress of Theoretical Physics*, 47(1):69–82, 1972.
- [TCF<sup>+</sup>08] S. Trotzky, P. Cheinet, S. Folling, M. Feld, U. Schnorrberger, A. M. Rey, A. Polkovnikov, E. A. Demler, M. D. Lukin, and I. Bloch. Time-resolved observation and control of superexchange interactions with ultracold atoms in optical lattices. *Science*, 319(5861):295–299, 2008.
- [Thy17] H. Thyen. The adaptive cross approximation for the dynamics of quantum many body systems using matrix product states. Master’s thesis, University of Göttingen, 2017.
- [TMGB18] F. Tacchino, C. Macchiavello, D. Gerace, and D. Bajoni. An artificial neuron implemented on an actual quantum processor. *ArXiv e-prints*, 1811.02266, Nov 2018.
- [Tof80] T. Toffoli. Reversible computing. In J. de Bakker and J. van Leeuwen, editors, *Automata, Languages and Programming*, pages 632–644, Berlin, Heidelberg, 1980. Springer Berlin Heidelberg.
- [TW05] M. Troyer and U.-J. Wiese. Computational complexity and fundamental limitations to fermionic Quantum Monte Carlo simulations. *Phys. Rev. Lett.*, 94:170201, May 2005.
- [usc01] <http://www-bcf.usc.edu/~tbrun/Course/lecture17.pdf>, Last accessed 2018-12-01.
- [VdN13] M. Van den Nest. Universal quantum computation with little entanglement. *Phys. Rev. Lett.*, 110:060504, Feb 2013.
- [VGRC04] F. Verstraete, J. J. García-Ripoll, and J. I. Cirac. Matrix product density operators: Simulation of finite-temperature and dissipative systems. *Phys. Rev. Lett.*, 93(20):207204, Nov 2004.
- [Vid03] G. Vidal. Efficient classical simulation of slightly entangled quantum computations. *Phys. Rev. Lett.*, 91:147902, Oct 2003.
- [Vid04] G. Vidal. Efficient simulation of one-dimensional quantum many-body systems. *Phys. Rev. Lett.*, 93:040502, Jul 2004.
- [VMC08] F. Verstraete, V. Murg, and J. I. Cirac. Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems. *Advances in Physics*, 57(2):143 – 224, 2008.
- [VWN80] S. H. Vosko, L. Wilk, and M. Nusair. Accurate spin-dependent electron liquid correlation energies for local spin density calculations: a critical analysis. *Canadian Journal of Physics*, 58(8):1200–1211, 1980.
- [WAS02] S. R. White, I. Affleck, and D. J. Scalapino. Friedel oscillations and charge density waves in chains and ladders. *Phys. Rev. B*, 65:165122, Apr 2002.

- [Wat] Phase diagram of water. <https://commons.wikimedia.org/w/index.php?curid=14939155>. Last accessed 2018-12-05.
- [WBC<sup>+</sup>11] S. Wall, D. Brida, S. R. Clark, H. P. Ehrke, D. Jaksch, A. Ardavan, S. Bonora, H. Uemura, Y. Takahashi, T. Hasegawa, H. Okamoto, G. Cerullo, and A. Cavalleri. Quantum interference between charge excitation paths in a solid-state Mott insulator. *Nature Physics*, 7(2):114–118, 2011.
- [WF04] S. R. White and A. E. Feiguin. Real-time evolution using the density matrix renormalization group. *Phys. Rev. Lett.*, 93:076401, Aug 2004.
- [Whi92] S. R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69(19):2863–2866, Nov 1992.
- [Whi93] S. R. White. Density-matrix algorithms for quantum renormalization groups. *Phys. Rev. B*, 48(14):10345–10356, Oct 1993.
- [Wil75] K. G. Wilson. The renormalization group: Critical phenomena and the kondo problem. *Rev. Mod. Phys.*, 47(4):773, 1975.
- [WK55] E. O. Wollan and W. C. Koehler. Neutron diffraction study of the magnetic properties of the series of perovskite-type compounds  $[(1-x)\text{La}, x\text{Ca}]\text{MnO}_3$ . *Phys. Rev.*, 100:545–563, Oct 1955.
- [WS14] D. Wecker and K. M. Svore. LIQUi|>: A Software Design Architecture and Domain-Specific Language for Quantum Computing. *ArXiv e-prints*, 1402.4467, Feb 2014.
- [YDS<sup>+</sup>17] B. Yang, H.-N. Dai, H. Sun, A. Reingruber, Z.-S. Yuan, and J.-W. Pan. Spin-dependent optical superlattice. *Phys. Rev. A*, 96:011602, Jul 2017.
- [ZMK<sup>+</sup>15] M. P. Zaletel, R. S. K. Mong, C. Karrasch, J. E. Moore, and F. Pollmann. Time-evolving a matrix product state with long-ranged interactions. *Phys. Rev. B*, 91:165112, Apr 2015.

# Acknowledgments

First of all, I like to show my gratitude to my supervisor Salvatore R. Manmana. He already supervised my master's thesis and gave me the opportunity to continue the started developments through this PhD project. He always has an open ear and a never-ending motivation to explore (also unusual) paths whereby he provided many helpful comments and ideas.

Furthermore, I like to show my deep respect and gratefulness to Thomas Pruschke († 2016), who even from the hospital, where he stayed most of the time of my PhD project, was an active, helpful, motivated, supportive, kind, and nearly unstoppable leader of his group. *You left way too early!* At this point, I also want to thank all other faculty members who stepped in, in order to overcome administrative problems and to compensate the loss of knowledge, foremost Stefan Kehrein and Peter Blöchl. I also thank Peter Blöchl for giving me access to his cluster at the TU Clausthal, where nearly all results of this thesis were produced.

Of course, I also want to thank the CRC 1073, which financed most of this project, motivated everybody to participate in conferences, workshops, and summer/autumn/winter schools, and offered a large foundation for collaborations. Although the theorists are a somewhat exotic species within this construct, the doors for discussions (even in the laboratories) were always open (not only after we demonstrated our “sonic skills”).

I want to thank my “colleague” Sebastian Paeckel for a wonderful coffee addiction, a nearly infinite amount of animated discussions, and a very balanced working environment. Furthermore, to Robert Schade, who not only was my support whenever I had a cluster problem (he co-maintained the cluster in Clausthal) but also helped me in every life situation since the beginning of our undergrad studies: *Thank you!* Of course, a special thanks is in order for my (former) office mates: Alexander Tiegel, Benjamin Lenz, Piet Dargel, Ansgar Kalz, Oliver Bodensiek, Fabian Biebl, Constantin Meyer, and Sebastian Paeckel.<sup>1</sup> Moreover, I'd like to thank the members of the Institute for Theoretical Physics, in particular the condensed-matter-theory group, and Trisha Nath, for a providing, lively, and social environment. In particular, I thank Nils Abeling for all the fun we had while writing our theses.

Last but not least I like to thank my parents who supported me and my career in every way possible. Furthermore, they and my brother taught me to think logically or at least admit it when I do not. Last but not least, I want to thank my wife, Isabel, who always believed in me, motivated, and helped me. Without her, this thesis would not have been possible.

---

<sup>1</sup>Note that despite this long list it always was at most an office for 4 persons.

