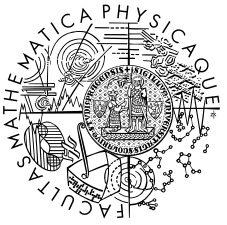


FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

DOCTORAL THESIS

Miroslav Urbanek

Modelling of Ultracold Gases in Multidimensional Optical Lattices

Department of Chemical Physics and Optics

Supervisor of the doctoral thesis:	doc. Ing. Pavel Soldán, Dr.
Study programme:	Physics
Study branch:	Quantum Optics and Optoelectronics

Prague 2017

I declare that I carried out this doctoral thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In Prague 27^{th} June 2017 Miroslav Urbanek

I would like to thank my supervisor, doc. Ing. Pavel Soldán, Dr., for his patient guidance and valuable advice throughout my entire time as his student. Furthermore, I would like to thank my parents, my brother, Tomáš Charvát, and Sahar Sajadieh for their support and encouragement.

Title: Modelling of Ultracold Gases in Multidimensional Optical Lattices

Author: Miroslav Urbanek

Department: Department of Chemical Physics and Optics

Supervisor: doc. Ing. Pavel Soldán, Dr.

Abstract:

Optical lattices are experimental devices that use laser light to confine ultracold neutral atoms to periodic spatial structures. A system of bosonic atoms in an optical lattice can be described by the Bose–Hubbard model. Although there exist powerful analytic and numerical methods to study this model in one dimension, their extensions to multiple dimensions have not been as successful yet. I present an original numerical method based on tree tensor networks to simulate time evolution in multidimensional lattice systems with a focus on the two-dimensional Bose–Hubbard model. The method is used to investigate phenomena accessible in current experiments. In particular, I have studied phase collapse and revivals, boson expansion, and many-body localization in two-dimensional optical lattices. The outcome of this work is TEBDOL — a program for modelling one-dimensional and two-dimensional lattice systems.

Keywords:

Bose–Hubbard model, multidimensional system, optical lattice, tensor network

Contents

Introduction

Advances in optical manipulation with ultracold atoms have led to the invention of optical lattices. An optical lattice is a device that uses a periodic potential, created by laser light, to confine neutral ultracold atoms in a periodic structure. All parameters of a lattice can be easily controlled by changing laser parameters. It provides a clean environment to study many-body quantum physics [1–4]. In a sense, atoms in an optical lattice resemble electrons in a crystal lattice. The geometry of an optical lattice can be easily designed to confine the system dynamics to two or one dimensions. They are therefore an excellent environment to study low-dimensional many-body quantum physics [5, 6].

Atoms in a lattice could be either fermionic or bosonic particles. The simplest nontrivial model that describes fermionic particles is the Hubbard model, and the simplest model for bosonic particles is the Bose–Hubbard model [7, 8]. Both models have two main parts. The first part describes the motion of particles in the lattice. The second part corresponds to interactions between multiple particles located at a single lattice site. An optional third term can account for an external potential.

Even though these models are fairly simple, they exhibit nontrivial properties. The Bose–Hubbard model, for example, has two distinct ground states based on its parameters. One is the Mott insulator with localized and strongly interacting particles, and the other one is the superfluid state with delocalized and weakly interacting particles. The quantum phase transition between these two regimes was predicted [9, 10] and then observed in early experiments with ultracold atoms in optical lattices [5, 11].

One of the fundamental problems in modern physics is the analysis of interacting many-body systems. Their microscopic properties are often known well enough, but predictions about energy levels or dynamics of a system are rarely available when it consists of a moderate number of particles. Analytic and numerical investigation is often restricted to small systems, or to limiting cases of weak or strong interactions. Both the Hubbard and the Bose–Hubbard are difficult to solve in a general case. Exact solutions are valid for static particles or for noninteracting particles only. Numerical studies are limited to one-dimensional systems [10] or to very small systems.

There has been a consistent effort to develop new methods of solving strongly interacting quantum systems in the last 25 years. The density matrix renormalization group method (DMRG) [12–15] provided a way to find ground states of one-dimensional lattice models. Among its early successes was the confirmation of the Haldane gap in the spin-1 Heisenberg model [13]. It was later discovered that DMRG is a variational method over a class of quantum states known as matrix product states (MPS) [16]. Expressing DMRG in the language of MPS provided a solid foundation for its generalizations. At the same time, time-evolving block decimation method (TEBD) [17, 18] emerged in the field of quantum information science. It provided a simple algorithm to simulate time evolution of one-dimensional lattice models on a classical computer. Both methods were later unified into time-dependent DMRG (tDMRG) [19, 20]. The infinite TEBD method (iTEBD) generalized the idea to translationally invariant systems [21].

These methods have been successfully applied to studies of one-dimensional

lattice models. They have confirmed experimental results and predicted new phenomena. Nowadays, methods based on MPS are routinely used to investigate properties of systems composed of tens or even hundreds of particles. Although there are some drawbacks in specific cases [22], one-dimensional many-body problem can be considered a solved problem for all practical purposes [23]. Naturally, the majority of physical systems reside in a three-dimensional space. There are also important systems that behave quite unusually when their dynamics is restricted to two dimensions. Examples are the famous quantum Hall effect or the high-temperature superconductivity. Unfortunately, the application of MPS-based methods to three-dimensional and two-dimensional systems has not been very successful yet.

One reason for the inefficiency of these methods in multidimensional settings is the so-called area law [24]. It predicts that for many systems in the ground state, entanglement between system parts grows with the size of the boundary that separates the parts. In particular, the entanglement entropy scales as $S \propto L^{D-1}$, where L is the system size and D is its dimension. In one dimension, the boundary is just a point, and S does not increase with the system size. In two dimensions, the boundary scales linearly with the system size, and in three dimensions, it scales quadratically. The area law therefore restricts the applicability of MPS-based methods in two-dimensional and three-dimensional problems, because the problem complexity rapidly increases with the problem size.

There have been, however, several attempts to overcome this issue. A progress has been made by recasting matrices in MPS as tensors, which can be generalized to multidimensional systems. Projected entangled pair states (PEPS) [25, 26] represent one such generalization to two dimensions. The method assigns a tensor to each lattice site, and takes into account correlations between all neighbouring sites. There has been a lot of improvement in PEPS theory in recent years. Unfortunately, some physical properties are very hard to calculate for a system state in this representation. The applicability of PEPS is therefore limited to small systems with small amount of entanglement currently.

PEPS are instances of tensor networks. Tensor networks can represent many states of practical importance effectively [27, 28]. Another approach based on tensor networks is the tree tensor network states method (TTNS) [29–31]. In contrast to PEPS, TTNS do not contain any network cycles. This makes them better suited as a generalization of MPS. The absence of cycles makes it possible to orthogonalize tensors in a prescribed way, which substantially simplifies the resulting equations. Yet another generalization is the multi-scale entanglement renormalization ansatz (MERA) [32, 33] that takes into account the coarse-graining corresponding to different length scales.

In this work, I report on TTNS-based methods of modelling ultracold atoms in multidimensional optical lattices. In the first part, I introduce optical lattices and review the Bose–Hubbard model which describes bosonic particles in a lattice. Then I define tensors and discuss their properties, because the method of choice is based on TTNS. Next chapter briefly explains MPS for modelling one-dimensional systems. A generalization to PEPS is introduced, and I discuss its advantages and disadvantages. Another generalization are TTNS, which better suit my goal of simulating ultracold atoms in two dimensions. I discuss the differences between PEPS and TTNS, and present an algorithm to simulate time evolution in the

two-dimensional Bose–Hubbard model with TTNS.

The second part discusses several physical problems. I first report on the results obtained with the one-dimensional algorithm. It is a study of a binary mixture of atoms in an optical lattice. Then I review performance scaling of the parallel one-dimensional algorithm. Later chapters deal with two-dimensional systems. I first compare the results obtained with my approach and with the exact diagonalization to analyze their accuracy. Then I introduce three interesting two-dimensional problems.

The first one is a study of quasimomentum revivals in a two-dimensional optical lattice. Quasimomentum revivals represent an accessible experiment that has been performed in early days of the field [34, 35]. It has been used to distinguish between the Mott insulator and the superfluid phase. In two dimensions, I studied the dependence of the revivals on the difference in tunnelling strength in the x direction and in the y direction.

The next study examines an expansion of a cloud of bosonic atoms in a two-dimensional optical lattice. It has been found that its dynamics differs substantially in a one-dimensional and in a two-dimensional lattice [36]. In one dimension, the expansion is ballistic for strongly interacting atoms. However, the interactions suppress the expansion in two dimensions. The core of the atomic cloud stays located in the lattice centre even after long evolution times.

The last studied problem is the many-body localization in two dimensions. Most of the physical systems thermalize, that is they reach thermal equilibrium after evolving for some time [37–39]. It has been found that some interacting systems do not exhibit this behaviour. They stay in a quasi-steady state that keeps a memory of the initial state. This typically happens in systems influenced by strong disorder potentials. An experiment in two-dimensional optical lattice confirmed the onset of localization at a critical disorder strength [40]. The lack of usable simulation algorithms for two-dimensional models made it difficult to compare the experimental findings with theory. With a simulation of a smaller, but similar system, I was able to confirm the onset of many-body localization numerically.

The last chapter briefly discusses three-dimensional systems and systems with nontrivial topologies. A detailed example of MPS compression and implementation details about TEBDOL are in Appendices.

The main output of my research is TEBDOL [41], a software package for simulating time evolution in optical lattices. It is based on tensor networks and supports both one-dimensional and two-dimensional models. It uses a generalized TEBD algorithm to calculate the evolution of a system state. TEBDOL takes into account particle number conservation to reduce the problem complexity. It shows good performance on hard problems.

Numerical results presented in this work were calculated on supercomputers Anselm and Salomon operated by IT4Innovations [42]. This work was supported by The Ministry of Education, Youth, and Sports from the Large Infrastructures for Research, Experimental Development, and Innovations project "IT4Innovations National Supercomputing Center – LM2015070". I further acknowledge support of the Czech Science Foundation – GAČR (Grant No. P209/15-10267S).

1. Optical lattices

Optical lattices are devices that use laser light to trap atoms in a periodic structure. They employ the AC-Stark effect, that is the influence of alternating electric field on energy levels of a neutral atom. The effective potential depends on the frequency of electric field oscillations. If the frequency is close but not equal to the transition frequency between the ground state and the excited state ω_{eg} in the two-level atomic model, the optical potential is given by [3, 43]

$$V_0 = \frac{3\pi c^2 \Gamma_{eg}}{2\omega_{eg}^3} \frac{I}{\Delta}, \tag{1.1}$$

where c is the speed of light, Γ_{eg} is the natural linewidth of the transition line, $\Delta = \omega - \omega_{eg}$ is the detuning of the laser frequency ω and the transition frequency ω_{eg}, and I is the electric field intensity. The model assumes $|\Delta| \ll \omega_0$ and $|\Delta| \gg \Gamma$. For a given atomic transition, the optical potential V_0 is proportional to the intensity I and inversely proportional to the detuning Δ. It is attractive for red detuning ($\Delta < 0$) and repulsive for blue detuning ($\Delta > 0$).

A scheme of optical lattice operation is shown in Figure 1.1 [4, 44]. A laser beam with a frequency ω is reflected from a mirror. The two counter-propagating beams interfere and create a standing electromagnetic wave. The spatial dependency of the electric field is of the form

$$\varepsilon(x) = \varepsilon_0 \sin(kx), \tag{1.2}$$

with $k = 2\pi/\lambda$, and λ being the laser wavelength. The corresponding light intensity is then

$$I(x) = I_0 \sin^2(kx). \tag{1.3}$$

This setup leads to a potential (1.1) given by

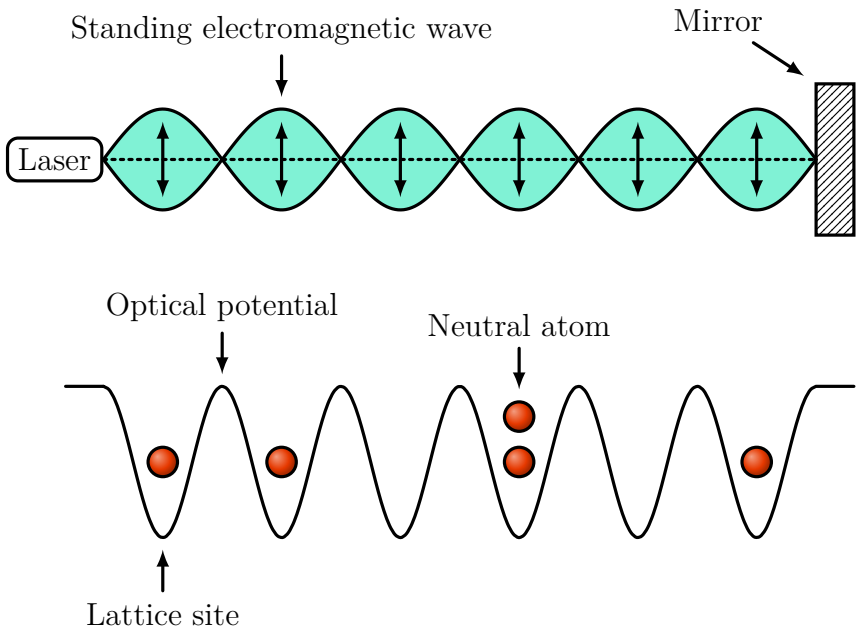
$$V(x) = V_0 \sin^2(kx). \tag{1.4}$$

The optical potential $V(x)$ therefore depends on the position in space. Atoms stay close to the potential minima if the potential is strong enough. By using lasers in all three spatial directions, a three-dimensional optical lattice can be created.

An important property of optical lattices is a possibility to create low dimensional systems. All lasers can be controlled independently, therefore the depth of potential wells can be different in the x, y and z direction. If the potential is very strong in the y and z directions and weak in the x direction, the movement of atoms in the y and z directions is restricted. Such setup results in atoms confined to parallel one-dimensional cylinders. Each cylinder is effectively an independent one-dimensional optical lattice. Similarly, if the potential is strong in the z direction and weak in the x and the y direction, the movement of atoms in the z direction is suppressed.

Experiments in optical lattices are usually performed with alkali atoms. Their transition frequencies are conveniently located in the frequency range of contemporary lasers. An atom is composed of protons, neutrons, and electrons. They are all spin-1/2 particles. A neutral atom is characterized by the numbers of its protons Z and by the number of its neutrons N. If there is an even number of

Figure 1.1: Operation of an optical lattice. A standing electromagnetic wave created by two counter-propagating laser beams gives rise to a periodic optical potential for neutral atoms. Atoms are forced to occupy sites at potential minima if the potential is strong enough.



constituent particles, the atom has an integer total spin, and it subsequently acts as a boson. Otherwise, it acts as a fermion. Because a neutral atom contains the same number of protons and electrons, the difference between these two cases is determined by N. Atoms with an even N are bosonic particles, and atoms with an odd N are fermionic particles. Examples of bosonic species are ^{7}Li, ^{23}Na, ^{39}K, ^{41}K, ^{85}Rb, ^{87}Rb, and ^{133}Cs. Fermionic species include ^{6}Li and ^{40}K.

Ultracold atoms confined to optical lattices resemble in many ways electrons confined to crystal lattices in solids. The differences are that optical lattices have larger spacing, they are almost defects-free, and all their parameters can be controlled by changing laser parameters. The experimental progress has led to possibility of addressing even single atoms in optical lattices [45]. They serve as advanced devices for studying strongly interacting many-body quantum systems in a controlled environment.

1.1 Bose–Hubbard model

A system of bosonic atoms in an optical lattice can be described by the Bose–Hubbard model [7, 8]. It is a conceptually simple model that in most cases explains observed behaviour very well. Atoms in a lattice exhibit two main phenomena — they tunnel between sites and they interact with each other.

Atoms are confined to potential wells, but they can tunnel between them. The tunnelling probability decreases with increasing lattice depth. It is strongest between adjacent sites and decreases quickly with distance. The Bose–Hubbard

model takes into account tunnelling between nearest-neighbouring sites only.

Neutral atoms interact with each other. The distinctive feature of optical lattices is that the particle separation is much larger than their interaction range. The two-body interactions become dominant in this regime. Particles interact mostly due to the van der Waals forces. Dominant interactions are interactions between atoms occupying the same site. Atomic gas temperature is typically in the nanokelvin range [46], thus the kinetic energy of atoms is very low. It is therefore sufficient to consider only s-wave scattering under these conditions. The scattering length can be increased by employing Feshbach resonances [4].

The last considered effect is that atoms can be influenced by another potential besides the periodic optical potential. A typical example is a harmonic confining potential due to an optical trap. Another example is a random disorder potential used in certain experiments.

Taking into account all these effects leads to the Bose–Hubbard model. Its Hamiltonian is given by

$$\hat{H} = -J \sum_{\langle \mathbf{i},\mathbf{j} \rangle} \hat{a}_{\mathbf{i}}^{\dagger} \hat{a}_{\mathbf{j}} + \frac{U}{2} \sum_{\mathbf{i}} \hat{n}_{\mathbf{i}} \left(\hat{n}_{\mathbf{i}} - 1 \right) + \sum_{\mathbf{i}} V_{\mathbf{i}} \hat{n}_{\mathbf{i}}, \tag{1.5}$$

where $\hat{a}_{\mathbf{i}}^{\dagger}$ is the creation operator at the site with coordinates $\mathbf{i} = (i_x, i_y, i_z)$, $\hat{a}_{\mathbf{i}}$ is the annihilation operator at $\mathbf{i}$, $\hat{n}_{\mathbf{i}} = \hat{a}_{\mathbf{i}}^{\dagger} \hat{a}_{\mathbf{i}}$ is the particle number operator at $\mathbf{i}$, J is the tunnelling strength, U is the on-site interaction strength, and $V_{\mathbf{i}}$ is the external potential. Angle brackets denote sum over each pair of adjacent sites. The first part of the Hamiltonian corresponds to particle tunnelling, the second part corresponds to particle interactions, and the third part describes the external potential. The model assumes bosonic particles, therefore the creation and annihilation operators satisfy bosonic commutation relations

$$\begin{aligned} [\hat{a}_{\mathbf{i}}, \hat{a}_{\mathbf{j}}] &= \left[\hat{a}_{\mathbf{i}}^{\dagger}, \hat{a}_{\mathbf{j}}^{\dagger} \right] = 0, \\ \left[\hat{a}_{\mathbf{i}}, \hat{a}_{\mathbf{j}}^{\dagger} \right] &= \delta_{\mathbf{ij}}. \end{aligned} \tag{1.6}$$

The parameters J, U, and $V_{\mathbf{i}}$ can be calculated given the particle species and the lattice depth [3, 4, 44]. The calculation takes into account the band structure of the lattice, the corresponding Bloch functions, and a transformation of Bloch functions into Wannier functions. Wannier functions are a convenient set of orthogonal functions corresponding to individual lattice sites.

This work investigates mainly the two-dimensional Bose–Hubbard model. Lattice tunnelling can be set independently in the x and in the y direction in this model. Its Hamiltonian is then given by

$$\begin{aligned} \hat{H} = &- J_x \sum_{\langle i_x, i_x' \rangle} \sum_{i_y} \hat{a}_{i_x, i_y}^{\dagger} \hat{a}_{i_x', i_y} - J_y \sum_{i_x} \sum_{\langle i_y, i_y' \rangle} \hat{a}_{i_x, i_y}^{\dagger} \hat{a}_{i_x, i_y'} \\ &+ \frac{U}{2} \sum_{i_x} \sum_{i_y} \hat{n}_{i_x, i_y} \left(\hat{n}_{i_x, i_y} - 1 \right) + \sum_{i_x} \sum_{i_y} V_{i_x, i_y} \hat{n}_{i_x, i_y}, \end{aligned} \tag{1.7}$$

where J_x and J_y are the tunnelling strengths in the x and in the y direction, respectively. The angle brackets denote again a sum over adjacent sites. When $J_x = 0$ or $J_y = 0$, the model becomes effectively one-dimensional.

The Bose–Hubbard model has two quantum phases [4, 9, 11, 47]. In the limit $J \ll U$, the lattice tunnelling is suppressed, and the system is dominated by the interactions. They force atoms to occupy distinct sites. The ground state is a Mott insulator with an integer number of atoms per site n. It is a product state of localized atoms represented by a wavefunction

$$|\psi_{\mathrm{MI}}\rangle \propto \prod_{\mathbf{i}} \left(\hat{a}_{\mathbf{i}}^{\dagger}\right)^{n} |0\rangle, \tag{1.8}$$

where $|0\rangle = \bigotimes_{\mathbf{i}} |0\rangle_{\mathbf{i}}$ is the vacuum state with no particles in the lattice. The opposite limit $J \gg U$ corresponds to strong tunnelling and weak interactions. The ground state of the model is a superfluid state with each atom delocalized over the entire lattice. Its wavefunction is given by

$$|\psi_{\mathrm{SF}}\rangle \propto \left(\sum_{\mathbf{i}} \hat{a}_{\mathbf{i}}^{\dagger}\right)^{N} |0\rangle, \tag{1.9}$$

where N is the total number of particles.

1.2 Methods of solution

There exist both analytic and numerical methods to extract information about the Bose–Hubbard model. They are often limited to systems with weak or strong interactions, or to one-dimensional systems. There is no known approach that works well in all cases. In this section I list several popular approaches [4].

Analytic methods usually differentiate between weakly and strongly interacting particles. *Bogoliubov approach* is a mean-field method that can be used to solve the model in the weakly interacting limit. The *strong-coupling expansion* is a perturbative method to solve the model in the strongly interacting limit. Other analytic methods include *perturbative mean-field method* and *Gutzwiller mean-field method*. There are also several approaches limited to one-dimensional systems — *Jordan–Wigner transformation*, *bosonization*, and *Bethe ansatz*.

Examples of numerical methods are *exact diagonalization* which is limited to small systems, *quantum Monte Carlo* to study systems in equilibrium, and *phase space methods* that are closer to classical physics. Finally, there are methods based on *tensor networks*. These represent the approach explored in this work.

2. Tensors

A *tensor* is a multidimensional array of complex numbers. It is characterized by its *rank*, which is a number of its dimensions. Tensors are a generalization of scalars, vectors, and matrices. A rank-zero tensor is a complex number, a rank-one tensor is a vector, and a rank-two tensor is a matrix. Each tensor element is denoted by its *indices*, which give its position in the array.

Two kinds of important tensors are recognized in this work — dense tensors and symmetric tensors. Dense tensors are arrays that contain all tensor elements. Symmetric tensors take advantage of symmetries of the physical system to reduce the number of elements that are required to store. They only include blocks that can contain nonzero elements. I first present an overview of dense tensors and later of symmetric tensors.

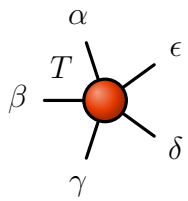
A tensor index of a dense tensor is an integer variable $\alpha \in \{1, \ldots, D_\alpha\}$, where D_α is the size of the respective dimension. A tensor can have an arbitrary number of indices. Its number of elements is the product of sizes of all its indices. Figure 2.1 shows a graphical representation of a tensor as a ball with several lines emerging from it. Each such line represents an index. Joined lines of two tensors represent a tensor contraction defined later. Several additional terms used in this work are defined as follows:

- *Conjugate index* $\alpha^\dagger$ of a tensor V is an index that allows contraction with an index α of a tensor U. There is no difference between an index and its conjugate index for dense tensors, because both of them are just the same sets of integers. However, there is a difference in the case of symmetric tensors described later. I therefore distinguish between indices and their conjugate indices for all tensor.

- *Conjugate tensor* $T^\dagger$ contains complex conjugates of all elements of a tensor T. Its indices are the conjugate indices of T.

- *Identity tensor* I is a rank-two tensor that carries a pair of conjugate indices α and $\alpha^\dagger$, and its elements are $I_{\alpha\alpha^\dagger} = \delta_{\alpha\alpha^\dagger}$, where $\delta_{\alpha\alpha^\dagger}$ is the Kronecker delta. It is equivalent to an identity matrix.

Figure 2.1: A tensor T with five indices α, β, γ, δ, and ϵ.



2.1 Tensor operations

This section explains several important tensor operations. Figure 2.2 describes these operations graphically. They are described for dense tensors. Although the details are different for symmetric tensors, the abstract notation is equivalent for both tensor types.

2.1.1 Index permutation

Index permutation changes the order of tensor indices. This operation takes a tensor and a permutation as inputs. It outputs a new tensor with correspondingly reordered indices. The tensor elements are reordered as well. Index permutation is analogous to matrix transposition.

2.1.2 Index fusion

Index fusion combines several indices into one. It produces a new tensor with a changed set of indices. The selected indices are fused into a single index, and all other indices are copied from the original tensor unaltered. The size of the fused index is a product of the sizes of all its constituent indices. The elements of the new tensor are rearranged elements of the original tensor. Index fusion is analogous to reshaping a matrix into a vector.

2.1.3 Index split

Index split is a reverse operation to index fusion. A split is not unique, as there are several ways how to factorize an index size. TEBDOL [41] uses this operation internally for tensor decomposition and tensor contraction. Index split is analogous to reshaping a vector into a matrix.

2.1.4 Decomposition

Tensor decomposition is an operation that takes a single tensor T and a partial list of its indices I and decomposes it into tensors U and V. It is analogous to the singular value decomposition (SVD) for matrices [48]. I lists all indices that will be transferred to U. All other indices of T are denoted J. There are two variants of the algorithm based on the unitarity requirements. The outline of the algorithm is as follows:

1. If the leftmost indices of T are not exactly the indices in I, then permute the indices of T to make them such.
2. Reshape T into a matrix $\widetilde{T}$, where all indices in I combined form the row index of $\widetilde{T}$ and all indices in J combined form the column index of $\widetilde{T}$.
3. Calculate the SVD of $\widetilde{T}$ to obtain matrices $\widetilde{U}$, $\widetilde{S}$, $\widetilde{V}^\dagger$.
4. Depending on the unitarity requirements, multiply $\widetilde{U}$ with $\widetilde{S}$ and keep $\widetilde{V}^\dagger$ unaltered, or keep $\widetilde{U}$ unaltered and multiply $\widetilde{S}$ with $\widetilde{V}^\dagger$. The resulting pair of matrices is denoted $\bar{U}$ and $\bar{V}^\dagger$. In the first case, the matrix $\bar{V}^\dagger$ is a unitary matrix. In the second case, the matrix $\bar{U}$ is a unitary matrix.

5. Reshape $\bar{U}$ into a tensor U so that the row index of $\bar{U}$ splits into indices in I. Column index of $\bar{U}$ becomes the last index of U.

6. Reshape $\bar{V}^\dagger$ into a tensor V so that the row index of $\bar{V}^\dagger$ splits into indices in J. Column index of $\bar{V}^\dagger$ becomes the first index of V.

2.1.5 Contraction

Tensor contraction is an operation that combines tensors U and V into a tensor T. It is a generalization of matrix multiplication. It takes a list of indices I_U of U and a list of indices I_V of V as an argument. Both lists must have the same length, and it is required that each index in I_U is a conjugate of a corresponding index in I_V. The contraction is performed over each pair of conjugate indices in I_U and I_V. Let J_U be all indices of U not included in I_U, and let J_V be all indices of V not included in I_V. The outline of the algorithm is as follows:

1. If the rightmost indices of U are not exactly the indices in I_U, then permute the indices of U to make them such.

2. If the leftmost indices of V are not exactly the indices in I_V, then permute the indices of V to make them such.

3. Reshape U into a matrix $\widetilde{U}$, where all indices in I_U combined form the column index of $\widetilde{U}$ and all indices in J_U combined form the row index of $\widetilde{U}$.

4. Reshape V into a matrix $\widetilde{V}$, where all indices in I_V combined form the row index of $\widetilde{V}$ and all indices in J_V combined form the column index of $\widetilde{V}$.

5. Multiply the matrices $\widetilde{U}$ and $\widetilde{V}$ to obtain a matrix $\widetilde{T}$.

6. Reshape $\widetilde{T}$ into a tensor. Split the row index into indices in J_U and split the column index into indices in J_V to obtain a tensor T.

2.2 Symmetric tensors

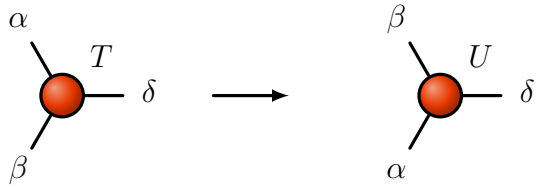
The Bose–Hubbard Hamiltonian is invariant under a global $U(1)$ transformation. This symmetry results in the conservation of the total number of particles in the system. TEBDOL takes advantage of it by using a symmetric tensor representation [16, 49–51]. Symmetric tensors ensure conservation of the total number of particles in the system and reduce computational costs. This representation differs from the dense tensors described earlier in the following aspects:

- A symmetric tensor is composed of blocks characterized by a list of particle numbers corresponding to all its indices.

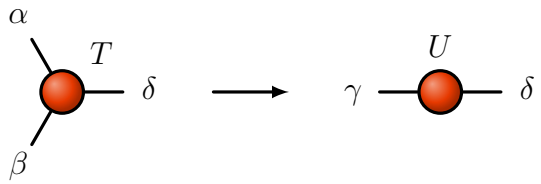
- If the sum of particle numbers characterizing a block is nonzero, all tensor elements in the block vanish.

- TEBDOL stores only those blocks that can contain nonzero elements, that is only the blocks for which the sum of their particle numbers is zero.

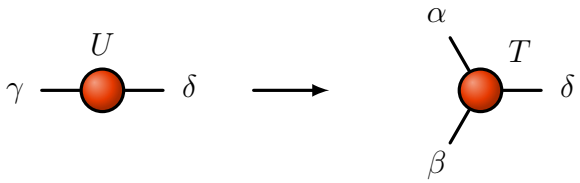
Figure 2.2: Tensor operations.



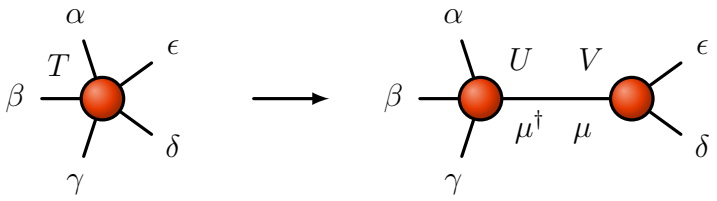
(a) (*Index permutation*) A new tensor U has the same elements and indices as the original tensor T, but the order of its indices is permuted.



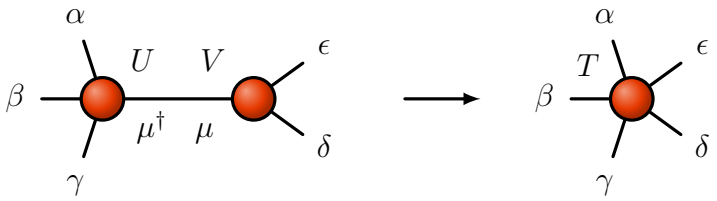
(b) (*Index fusion*) Indices α and β of a tensor T are fused together to create an index γ of a new tensor U. The size of γ is a product of the sizes of α and β.



(c) (*Index split*) Index γ of a tensor U is split into indices α and β of a new tensor T. A split is not unique. Generally, this operation is performed only on indices obtained from an index fusion to restore the original indices.



(d) (*Tensor decomposition*) A tensor T with indices α, β, γ, δ, and ϵ is decomposed into two new tensors U and V. The partition of the original set of indices between U and V can be arbitrary. U and V are connected by a new pair of conjugate indices $\mu^\dagger$ and μ.



(e) (*Tensor contraction*) Two tensors U and V are contracted into a new tensor T. The contracting indices $\mu^\dagger$ and μ must be a pair of conjugate indices. All other indices of U and V are preserved in T.

- A tensor index is a compound structure for symmetric tensors. It is a list of segments, where each segment consists of a particle number and a dimension.
- Each nonvanishing block is given by a list of segments corresponding to the tensor indices. The sum of the particle numbers in these segments must be zero. The dimensions of the block are the dimensions of the segments. A block is represented by a dense tensor.
- Index conjugation changes the signs of the particle numbers of each segment but preserves the segment dimensions. An index and its conjugate index can be contracted with each other.
- Tensor operations operate on blocks. Each block is treated separately to obtain the desired result.

3. Matrix product states

Study of strongly correlated many-body systems is hard because dimensions of their Hilbert spaces are huge. Only a few models can be solved analytically, and exact numerical solutions are typically restricted to small system sizes and small numbers of particles. Several methods have been therefore developed to obtain approximate solutions. One of the most successful methods, especially for one-dimensional systems, is the density matrix renormalization group method (DMRG) [12, 13, 15]. Its original goal was to find ground states of quantum chains. Later, it was extended to simulate time evolution (tDMRG) [19, 20], as well as to model other quantum systems. It is based on approximating reduced density matrices of system parts. The algorithm truncates the density matrices, and it retains only the most relevant states of the system.

Matrix product states (MPS) [16, 52, 53] are a class of quantum states originally developed for analytic description of quantum systems. It was found that DMRG can be conveniently formulated in the language of MPS [54, 55]. In fact, DMRG variationally optimizes the wave function in the MPS form. The decomposition into the MPS form is sometimes referred to as the tensor-train decomposition (TT) in mathematics [56].

In this section, I will introduce basic ideas of MPS for simulating time evolution in one-dimensional optical lattices. MPS is inherently a one-dimensional method. It has been applied to multidimensional systems as well, in a sense that a system had been first mapped to one dimension and then solved with the already well-understood algorithm [57, 58]. However, this approach was less successful for multidimensional problems. In later chapters, I will discuss extensions of MPS to higher-dimensional systems that better exploit system geometries.

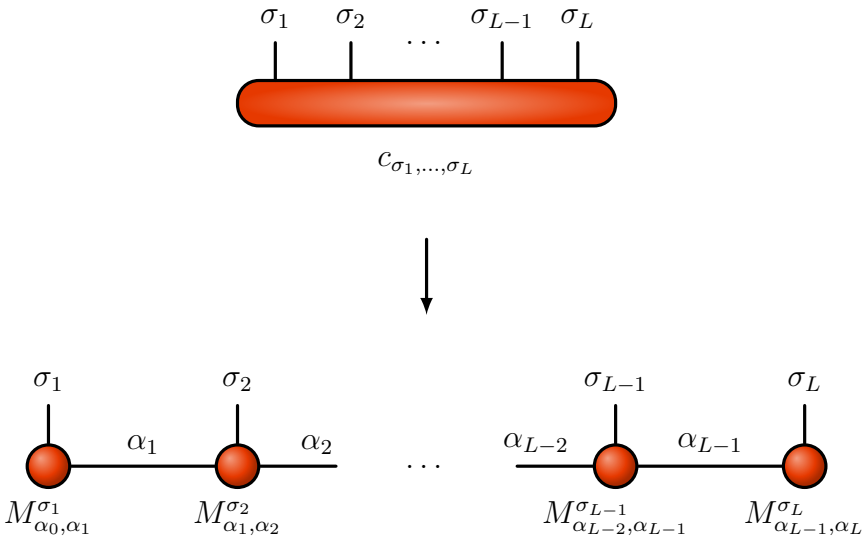
I assume a model of a quantum lattice with open boundary conditions, for example a one-dimensional Bose–Hubbard model. The lattice is composed of L sites indexed by i, with $i \in \{1, \ldots, L\}$. There are nearest-neighbour interactions only, that means that a site i interacts with sites $i-1$ and $i+1$. Local basis states of the Hilbert space at a site i are denoted by $|\sigma_i\rangle$, with $\sigma_i \in \{1, \ldots, P\}$ and P being the physical dimension. In principle, there can be a different P at each site, but I assume that the physical dimension is identical everywhere. The wavefunction of the full system is given by

$$|\psi\rangle = \sum_{\sigma_1,\ldots,\sigma_L} c_{\sigma_1,\ldots,\sigma_L} |\sigma_1, \ldots, \sigma_L\rangle, \tag{3.1}$$

where $|\sigma_1, \ldots, \sigma_L\rangle$ are the vectors of the product basis, and $c_{\sigma_1,\ldots,\sigma_L}$ are the state coefficients. The number of coefficient $c_{\sigma_1,\ldots,\sigma_L}$ grows exponentially with L. Even for lattices with tens of sites, the problem of storing and working with this many coefficients becomes quickly intractable. The goal is therefore to find a method to reduce the number of working parameters, and still be able to extract good approximations of all coefficients $c_{\sigma_1,\ldots,\sigma_L}$.

It helps to consider all coefficients $c_{\sigma_1,\ldots,\sigma_L}$ as a single complex tensor with L indices. The basic idea of MPS is to decompose this tensor into a set of tensors $M^{\sigma_i}_{\alpha_{i-1},\alpha_i}$, so that there is one tensor for each site (Figure 3.1). Each tensor $M^{\sigma_i}_{\alpha_{i-1},\alpha_i}$ has three indices, a *physical index* σ_i, and *virtual indices* α_{i-1} and α_i. Additionally, I consider each tensor $M^{\sigma_i}_{\alpha_{i-1},\alpha_i}$ as a set of matrices, where α_{i-1} is the column index

Figure 3.1: Decomposition of a tensor $c_{\sigma_1,\ldots,\sigma_L}$ into a set of tensors $M^{\sigma_i}_{\alpha_{i-1},\alpha_i}$ corresponding to each lattice site. Each tensor is represented by a geometric shape. Lines emerging from a shape represent tensor indices. Lines connecting tensors represent tensor contractions. Note that the dimension of indices α_0 and α_L is one, because they are at the edges of the lattice. They are omitted from the figure. In the matrix notation, $M^{\sigma_1}_{\alpha_0,\alpha_1}$ is a row vectors for each σ_1, and $M^{\sigma_L}_{\alpha_{L-1},\alpha_L}$ is a column vector for each σ_L.



and α_i is the row index of a matrix M^{σ_i}. In the following, the virtual indices are dropped to simplify the notation.

The coefficients $c_{\sigma_1,\ldots,\sigma_L}$ are obtained by *contracting* the decomposition. In the present case, a contraction is equivalent to multiplication of all matrices M^{σ_i},

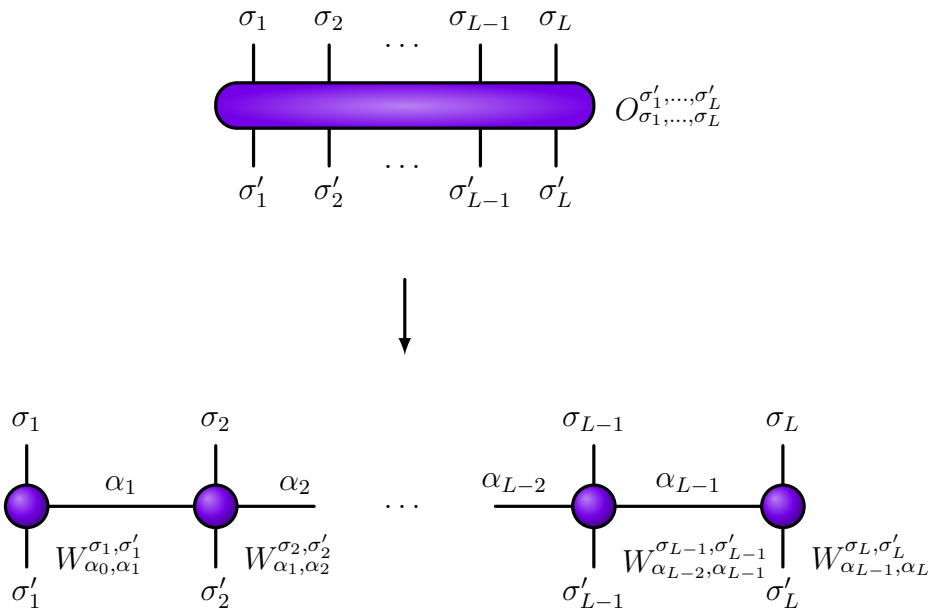
$$c_{\sigma_1,\ldots,\sigma_L} = M^{\sigma_1} M^{\sigma_2} \ldots M^{\sigma_{L-1}} M^{\sigma_L}. \tag{3.2}$$

This also explains the origin of the name "matrix product states".

I am not going to describe the details of the procedure for obtaining the decomposition, as it has been discussed extensively, see for example [16]. The idea is to reshape the tensor $c_{\sigma_1,\ldots,\sigma_L}$ into a matrix and to decompose the matrix using the singular value decomposition (SVD) [48]. The important point is that to obtain the exact decomposition of $c_{\sigma_1,\ldots,\sigma_L}$, the dimensions of indices α_i grow exponentially with L. Such decomposition would not help in reducing the complexity of the problem. Therefore, the procedure truncates the maximum dimension of α_i to D. It turns out that for many states of practical importance, such as the ground states of systems with gapped Hamiltonians, a decomposition into an MPS with a fixed D can be a very good approximation of the exact decomposition [59]. The total number of elements in $c_{\sigma_1,\ldots,\sigma_L}$ is P^L. In contrast, the total number of elements in an MPS is bounded by LPD^2, which is linear in L and P.

To find a ground state of a Hamiltonian, one typically starts with a random MPS with a fixed D. Its tensor elements are variationally optimized until the convergence is reached. The method is not guaranteed to find the true global minimum, but works very well in practice (see also Appendix A). To calculate the

Figure 3.2: Decomposition of an operator $O^{\sigma'_1,\ldots,\sigma'_L}_{\sigma_1,\ldots,\sigma_L}$ into a set of tensors $W^{\sigma_i,\sigma'_i}_{\alpha_{i-1},\alpha_i}$ corresponding to each lattice site.



time evolution of a state, one typically starts from a simple product state that can be easily decomposed into an MPS. If the MPS is a good approximation of $c_{\sigma_1,\ldots,\sigma_L}$, the problem becomes manageable at least for a short evolution interval.

Similarly to the decomposition of a wavefunction into an MPS, an operator can be decomposed into a matrix product operator (MPO) [16, 52, 60–62]. In this case, each site tensor has two physical indices (Figure 3.2). In contrast to the decomposition of a state into an MPS, the operators are usually constructed explicitly in the MPO form. In order to calculate the expectation value of an MPO $\hat{O}$ for an MPS ψ, one contracts ψ with $\hat{O}$ and with the complex conjugate of ψ (Figure 3.3).

Figure 3.3: Calculation of the expectation value of an operator $\hat{O}$ in the MPO form for a state ψ in the MPS form.

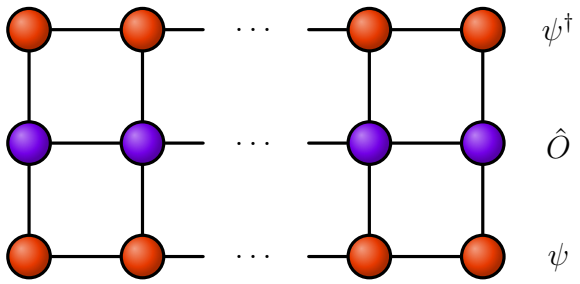
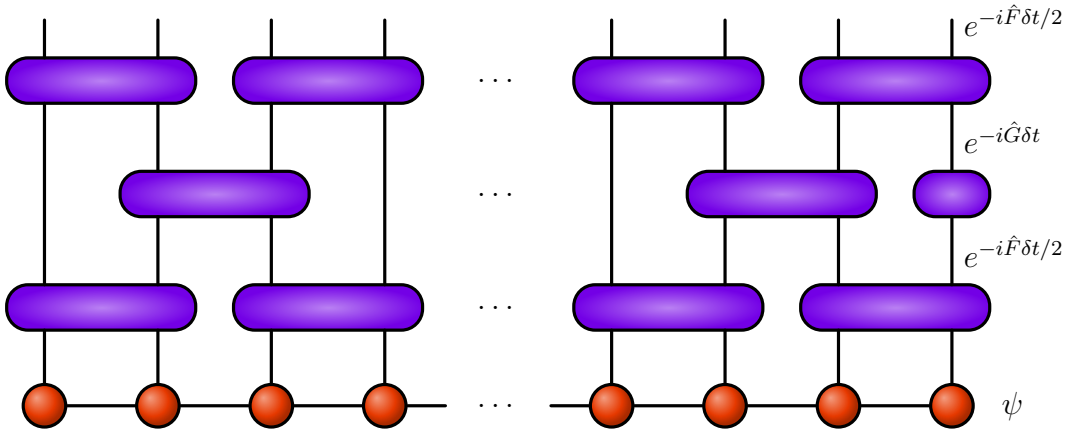


Figure 3.4: One time step of the second-order time-evolution algorithm for MPS. Operators $\exp(-i\hat{F}\delta t/2)$ and $\exp(-i\hat{G}\delta t/2)$ are products of single-site and double-site commuting exponential operators, so the terms in each of them can be applied sequentially. After application of each term, the resulting tensor is decomposed into an MPS with a fixed bond dimension D.



3.1 Time evolution

I focus on simulation of nonequilibrium dynamics in this work. My method is based on the Suzuki–Trotter decomposition of the propagator [16]. The terms in the Hamiltonian of a model with nearest-neighbour interactions can be separated into two groups, $\hat{H} = \hat{F} + \hat{G}$, such that all terms in $\hat{F}$ commute with each other, and all terms in $\hat{G}$ commute with each other as well. The Suzuki–Trotter decomposition [63–68] then gives an approximation of the system propagator. The first-order approximation is given by

$$e^{-i\hat{H}\delta t} = e^{-i\hat{F}\delta t}e^{-i\hat{G}\delta t} + \mathcal{O}(\delta t^2), \tag{3.3}$$

and the second-order approximation is given by

$$e^{-i\hat{H}\delta t} = e^{-i\hat{F}\delta t/2}e^{-i\hat{G}\delta t}e^{-i\hat{F}\delta t/2} + \mathcal{O}(\delta t^3), \tag{3.4}$$

where δt is a small time step. The evolution is calculated by sequentially applying the expansion on the right-hand side to an MPS. The total error can be controlled by choosing a sufficiently small δt. Individual terms in $\hat{F}$ and $\hat{G}$ are either single-site operators or double-site operators [18]. Because the individual terms commute with each other, the exponential operator is a product of exponentials of individual terms. The procedure for one time step is illustrated in Figure 3.4.

In summary, MPS is a representation of a quantum state that reduces the complexity of finding the ground state of a lattice Hamiltonian or simulating its time evolution. In the latter case, the propagator is approximated by sets of single-site and two-site exponential operators. In contrast to the full propagator, these individual operators can be calculated easily. The operators are applied in a prescribed way. Resulting tensors are decomposed into single-site tensors in each step, so the MPS form is maintained during the whole calculation. The application of exponential operators leads to an increase of bond dimensions. The virtual indices are therefore truncated to a fixed dimension D.

4. Projected entangled pair states

Projected entangled pair states (PEPS) [25, 26] are a class of tensor networks designed to represent a pure state of a two-dimensional system. They are a generalization of MPS to two dimensions. Each site is represented by a single tensor. The distinguishing property of PEPS is that each site is connected to all its adjacent sites in the network. PEPS are important for theoretical description of two-dimensional systems, but it is difficult to obtain a good numerical performance with them. Additionally, certain assumptions that simplify working with MPS are not valid for PEPS. I therefore did not use PEPS for calculations and instead used tree tensor networks described later. This chapter serves as a short review of PEPS with emphasis on their properties that increase the computational complexity of operations on PEPS.

I consider a system of M lattice sites. Each site is equipped with a P-dimensional local Hilbert space and spanned by orthonormal basis vectors $\{|\sigma_i\rangle\}$, with $\sigma_i \in \{1, \ldots, d\}$. The tensor network consists of a set of tensors with one tensor corresponding to each lattice site. Each tensor carries a physical index σ_i, which enumerates basis states $|\sigma_i\rangle$, and several virtual indices that describe its relations to its neighbouring sites. The number of virtual indices varies with the lattice geometry and with the tensor position in the lattice. A coordination number c denotes the number of sites neighbouring a particular site. The coordination number in the bulk of a rectangular lattice is four. Each bulk tensor therefore carries four virtual indices. The coordination number is smaller for sites at lattice boundaries. Boundary tensors therefore carry fewer indices than tensors in the bulk. I consider rectangular lattices with open boundary conditions, so tensors in the bulk have four virtual indices, tensors at the lattice boundary that are not in the lattice corners have three virtual indices, and tensors corresponding to lattice corners have two virtual indices.

An example of a PEPS network is depicted in Figure 4.1. The maximum dimension of all virtual indices is called the bond dimension D. The number of elements of each tensor is bounded by dD^c. Tensor networks with higher bond dimensions could represent a physical system more faithfully at a cost of higher number of variational parameters of each tensor. Similarly to MPS, D has to be exponentially large in the system size to cover the full Hilbert space of the system. This is usually not necessary, because a good approximation of a state can be obtained with a relatively small D.

The state vector in the product basis is given by

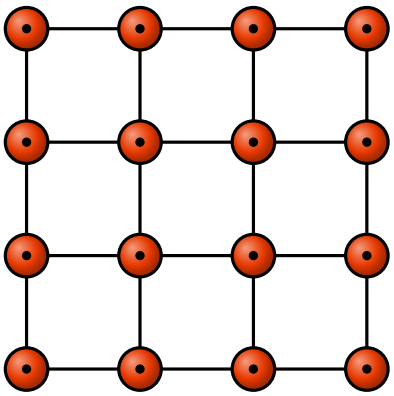
$$|\psi\rangle = \sum_{\sigma_1,\ldots,\sigma_M} c_{\sigma_1,\ldots,\sigma_M} |\sigma_1, \ldots, \sigma_M\rangle, \tag{4.1}$$

with coefficients [25]

$$c_{\sigma_1,\ldots,\sigma_M} = \mathcal{F}(T_1^{\sigma_1}, \ldots, T_M^{\sigma_M}). \tag{4.2}$$

where $T_i^{\sigma_i}$ are lattice tensors, and $\mathcal{F}(T_1^{\sigma_1}, \ldots, T_n^{\sigma_n})$ represents a contraction of all virtual indices of the network for a fixed set of physical indices $\{\sigma_1, \ldots, \sigma_n\}$.

Figure 4.1: A rectangular 4×4 PEPS with open boundary conditions. A ball represents a tensor, lines emerging from a tensor represent tensor indices, and connections represent a contractions of tensors over the respective indices. Physical indices are represented by dots, that is they are lines perpendicular to the lattice plane.



4.1 Contractions

A calculation of the norm or a calculation of the expectation value of an observable for a network in the PEPS form requires to evaluate a scalar product of two networks. The scalar product is a contraction of two PEPS with connected physical indices (Figure 4.2). In contrast to MPS, the contraction of the whole tensor network of a scalar product is a #P-Hard problem [69]. The time required to perform this calculation scales exponentially with the number of the lattice sites for any order in which the contractions are performed. There are no known classical or quantum algorithms for solving this problem utilizing resources that scale polynomially in the number of sites.

While the exact calculation of a scalar product is often unfeasible, there exist faster approximate methods to obtain the results with a good accuracy. In the following, I review several of them.

4.1.1 Sandwich contraction

Sandwich contraction [25] is an algorithm to approximate network contractions required to calculate the norm $\|\psi\|$ of a state ψ or the expectation value of an operator O. A boundary row of the state and operator tensors is identified with an MPS. The neighbouring row is identified with an MPO. The contraction then proceeds as a one-dimensional contraction of the MPS with the MPO. The bonds of the resulting MPS are truncated and the result is used as a starting MPS for the contraction with the next row of tensors. The same procedure is performed for the opposite row of tensors. The algorithm consecutively contracts rows from both sides. Finally, a pair of MPS from both sides are contracted with each other.

Figure 4.2: A scalar product of two 4×4 PEPS networks. The physical indices of tensors corresponding to a single site are contracted with each other. Finally, all virtual indices are contracted to obtain a scalar.

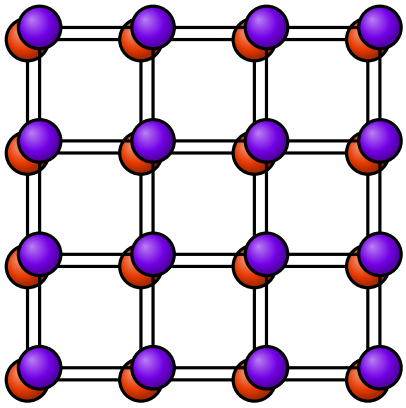
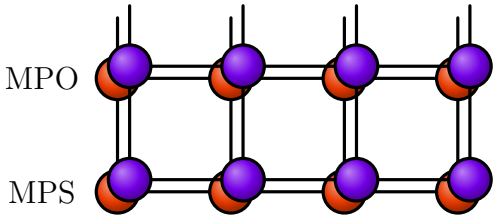


Figure 4.3: Sandwich contraction for calculating a scalar product of two PEPS networks. Each pair of tensors corresponding to a single site is considered a single tensor in an MPS or an MPO. The first row of tensors is identified with an MPS and the next row is identified with an MPO. The contraction proceeds as a contraction of an MPS and an MPO. It is necessary to truncate the bond dimension of the resulting MPS. The contraction proceeds both from the first row and from the last row. Finally, a scalar product is obtained as a contraction of two MPS networks.



4.1.2 Simple update

Simple update [70, 71] is a method to update a pair of site tensors in a PEPS network during a time-evolution simulation. It is a generalization of the TEBD algorithm from MPS to PEPS. A state is represented by Γ-tensors corresponding to lattice sites and connected by diagonal λ-matrices. A pair of sites being updated is called a *system* and the rest of the network is called its *environment*. Generally, it is necessary to fully contract the environment to correctly truncate the bond in the system after applying an operator to a tensor pair. Instead of exactly calculating or approximating the environment of a site, this method *assumes* that the environment is orthogonal, that is similar to a canonical form of an MPS. In the case of MPS, a canonical form has convenient orthogonality properties [16]. In the case of PEPS, no such properties are guaranteed. It is thus unexpected that this method could produce good results. Its advantages are its simplicity and great performance. Networks with large bond dimensions D can be analyzed easily. However, its theoretical shortcomings make it unsuitable for proper calculations.

4.1.3 Single layer

Single layer [71, 72] is a method similar to sandwich contraction with the difference that the contractions are performed only in a single layer of the sandwich tensor network. After every step, the physical indices must be traced out to create a purification MPO [60]. The aggregate dimension of physical indices would grow exponentially without this step. Single layer contraction has a better scaling than sandwich contraction, but it is less precise.

4.1.4 Cluster contraction

This method of calculating PEPS environment interpolates between sandwich contraction and simple update [71, 73]. Simple update assumes separable and local environment. Cluster contraction assumes that the environment correlations decay quickly with distance. Only a limited number of rows is taken into account to approximate the environment. The number of rows can be chosen arbitrarily. If no rows are chosen, the method resembles simple update. If all rows are chosen, the method resembles sandwich contraction.

4.2 Practical aspects

The contraction methods above are either inaccurate or computationally hard. If there are large correlations between distant sites, the whole environment should be taken into account in calculating any expectation values. Only sandwich contraction provides a correct method to do this. The bond dimension in PEPS is quite limited by this requirement. In contrast to MPS, where bond dimensions are typically in the order of 10^3, bond dimensions in PEPS are mostly in the order of 10^1. Additionally, a contraction is required not only for calculating a scalar product of two networks, but also for performing every bond truncations during time-evolution steps. The corresponding equations do not simplify as in the case of MPS. This makes PEPS a very complicated representation to work with.

I implemented sandwich contraction but the achieved bond dimensions and performance were very low. Further investigation revealed that the problem was the nonexistence of states similar to canonical states in MPS. Canonical states are tensor networks with prescribed orthonormalization conditions [16]. The impossibility to construct such states in PEPS is caused by cycles in the network, that is by multiple paths between any pair of tensors. I therefore decided to use tree tensor networks states described in the next chapter instead. In contrast to PEPS, they do not contain any cycles, and it is possible to orthonormalize them in a similar way as MPS.

5. Tree tensor network states

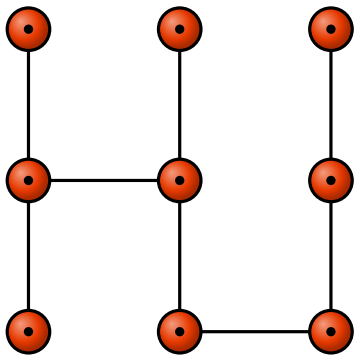
Tree tensor network states (TTNS) [29–31] are another generalization of MPS that is different from PEPS. Every bulk tensor in an MPS carries one physical index and two virtual indices. A boundary tensor in an MPS carries one physical index and one virtual index. In contrast, a tensor in a TTNS can have any number of virtual indices. Tensors are interconnected and form a graph. There can be multiple paths between a pair of tensors in a general tensor network. In that case the network contains a *cycle*. A condition that distinguishes TTNS from other tensor networks is that TTNS do not contain any cycles (Figure 5.1). The nonexistence of cycles brings an important advantage over PEPS. Similarly to MPS, it is then possible to orthogonalize tensors in a TTNS in order to obtain a canonical form, which will be defined in Section 5.1.

The number of elements in a dense tensor T is a product of all its index dimensions,

$$N_T = \prod_{\alpha \in A} D_\alpha, \tag{5.1}$$

where A is the set of its indices and D_α is the dimension of an index α. It grows exponentially with the number of indices for fixed index dimensions. Although there are fewer elements in symmetric tensors compared to dense tensors, the scaling of the total tensor size is similar. It is therefore advantageous to keep the number of virtual indices limited. A tensor network with at most two virtual indices is a one-dimensional MPS. TEBDOL [41] uses at most three virtual indices in the two-dimensional algorithm. I have also tested a version with at most four virtual indices, but it is very computationally challenging to perform operations on large tensors with four virtual indices. The number of virtual indices was therefore limited to three.

Figure 5.1: A TTNS is an interconnected set of tensors that does not contain any cycles. Each tensor carries a physical index denoted by a black dot and virtual indices connecting it to other tensors.



5.1 Canonical states

An important property of MPS is the existence of canonical states. A canonical state is a tensor network with well-defined orthogonalization of its tensors. Certain network operations are substantially simpler for a network in a canonical state. In particular, truncation of tensor dimensions, application of local operators, and calculation of expectation values of local observables are all straightforward operations if a network is in a canonical state.

Its definition starts with the definition of a *semi-unitary* tensor. Let α be an index of a tensor T and β_j, where $j \in \{1, \ldots, n\}$, all its other indices. A tensor is semi-unitary if

$$\sum_{\{\beta_1,\beta_1^\dagger\},\ldots,\{\beta_n,\beta_n^\dagger\}} T^\dagger_{\alpha^\dagger,\beta_1^\dagger,\ldots,\beta_n^\dagger} T_{\alpha,\beta_1,\ldots,\beta_n} = I_{\alpha^\dagger,\alpha}, \tag{5.2}$$

where $T^\dagger$ is the conjugate tensor of T, I is a unit tensor, and $\{\beta_1, \beta_1^\dagger\}$ denotes a contraction over a pair of conjugated indices β_j and $\beta_j^\dagger$. A semi-unitary tensor is analogous to a semi-unitary matrix. The contraction runs over multiple indices instead of a single index in the case of matrix multiplication. Semi-unitary tensors can be obtained from a tensor decomposition (Figure 5.2).

A tensor in a TTNS contains a single physical index and multiple virtual indices. A *leaf* is a tensor that carries only a single virtual index. A *branch* is a subset of TTNS that is connected to the rest of the TTNS with a single virtual index, that is all its other virtual indices are connected to tensors inside the branch (Figure 5.3). A leaf is also a branch. For models with open boundary conditions, a tensor can carry one, two or three virtual indices. Leaves have one virtual index, tensors in a branch that are not leaves have two virtual indices, and tensors connecting branches have three virtual indices. A *conjugate branch* is a branch composed of conjugate tensors of the original branch and with the identical connection structure.

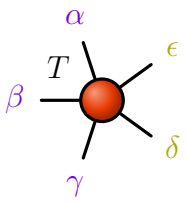
A branch is in the *canonical form*, if the contraction of this branch with its conjugate over all corresponding physical indices gives an identity tensor (Figure 5.4). To bring a leaf tensor T into the canonical form, T is first decomposed into U and $\widetilde{V}^\dagger$, where U is a semi-unitary tensor. Then $\widetilde{V}^\dagger$ is contracted to the neighbouring tensor N, and N is replaced with the resulting tensor (Figure 5.5).

A branch is transformed into the canonical form recursively (Figure 5.6). Let T be the tensor that is connected to the rest of the network. First, all other branches connected to T are brought into their canonical forms. The procedure is then similar as with leaf tensors — T is decomposed into U and $\widetilde{V}^\dagger$, and $\widetilde{V}^\dagger$ is contracted to the neighbouring tensor.

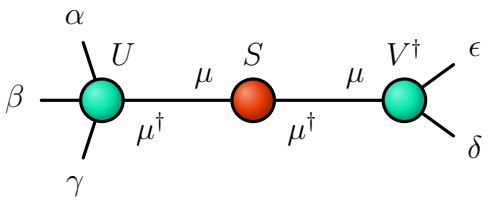
The transformation into a canonical form therefore starts from leaf tensors and brings all tensors in the branch into semi-unitary tensors. The semi-unitarity of branch tensors leads to the required condition that the contraction of the tensors with their conjugates over their physical indices gives an identity tensor.

Tensors in any TTNS can be divided into two sets called a *system* S a an *environment* E. What tensors form S or what tensors form E varies during the calculation. Generally, S is a part of the TTNS being operated on. The definition of a system is further restricted to ensure its compactness. It is required that there exists a path connecting any two tensors in S, and all tensors in this path also belong to S.

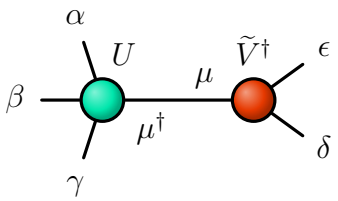
Figure 5.2: A tensor T can be decomposed into two or three new tensors. The resulting tensors have orthogonalization properties similar to the properties of semi-unitary matrices obtained in a compact matrix SVD.



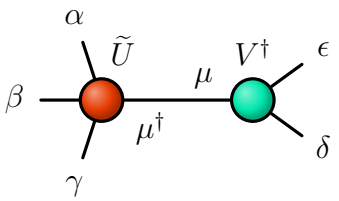
(a) First, the indices of T are separated into two groups. In this example, indices α, β, and γ belong to the first group, and indices δ and ϵ to the second group. All indices in a group become a part of a single tensor after the decomposition.



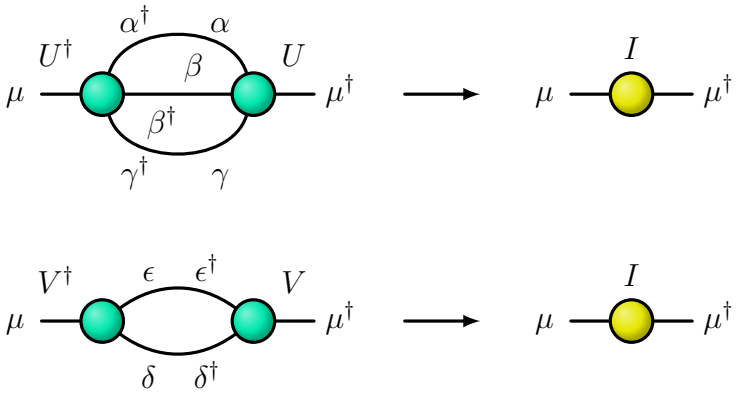
(b) T is decomposed into tensors U, S, and $V^\dagger$. This operation is analogous to SVD. The indices in the first group and in the second group belong to U and $V^\dagger$, respectively. Indices μ and $\mu^\dagger$ connect the tensors together.



(c) Contracting S and $V^\dagger$ into a tensor $\tilde{V}^\dagger$ produces a decomposition of T into a semi-unitary tensor U and a tensor $\tilde{V}^\dagger$.

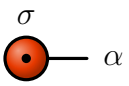


(d) Contracting U and S into a tensor $\tilde{U}$ produces a decomposition of T into a tensor $\tilde{U}$ and a semi-unitary tensor $V^\dagger$.

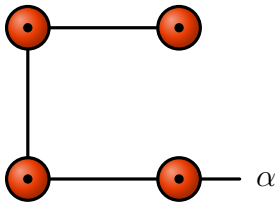


(e) Contracting semi-unitary tensors with their conjugates over the original indices results in identity tensors.

Figure 5.3: A leaf and a branch of a TTNS.



(a) A leaf tensor carries a single physical index σ and a single virtual index α.



(b) A branch is connected to the rest of a network with a single virtual index α.

Figure 5.4: Contraction of a canonical branch with its conjugate produces an identity tensor.

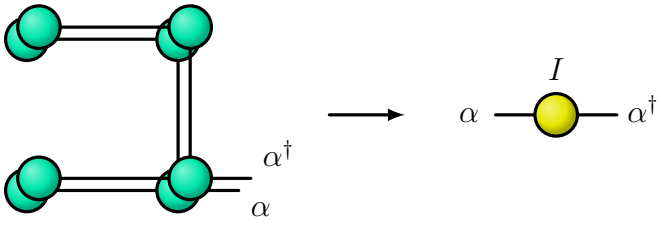


Figure 5.5: Transformation of a leaf T into the canonical form. N is a neighbouring tensor, U and $\widetilde{V}^\dagger$ come from the decomposition of T, and $\widetilde{N}$ is the product of contracting $\widetilde{V}^\dagger$ with N. U is a semi-unitary tensor.

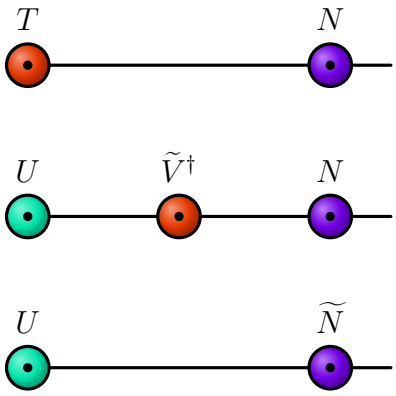
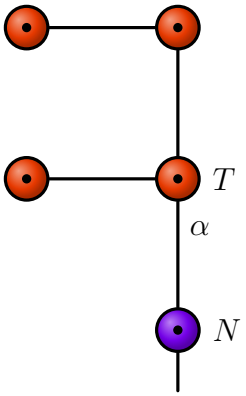
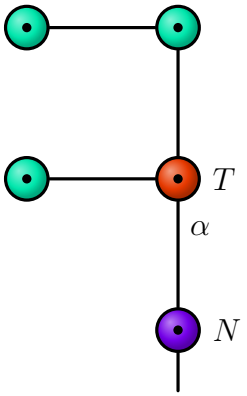


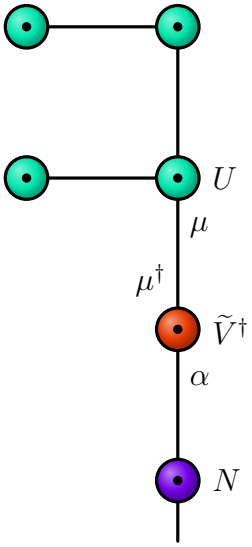
Figure 5.6: Transformation of a branch into the canonical form.



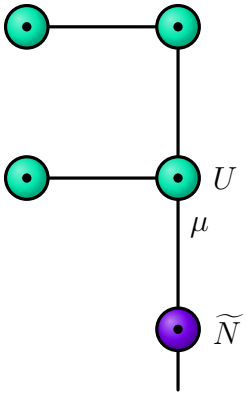
(a) A branch is connected by an index α to the rest of the system. α belongs to a branch tensor T. A neighbouring tensor N is not a part of the branch.



(b) First, all other branches connected to T are transformed into their canonical forms using the procedure described here recursively.

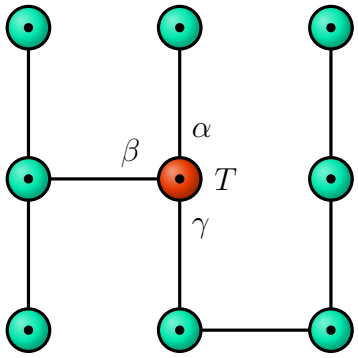


(c) T is decomposed into U and $\widetilde{V}^\dagger$, where U is a semi-unitary tensor.



(d) $\widetilde{V}^\dagger$ is contracted to the neighbouring tensor N obtaining a new tensor $\widetilde{N}$. All branch tensors are now semi-unitary and the branch is in the canonical form. The branch is connected to the rest of the network by the index μ.

Figure 5.7: A TTNS in a canonical state. The environment consists of three branches connected to a single system tensor T by indices α, β, and γ. All three branches are in their canonical forms.



S and E are connected with a set of contractions C. TTNS do not contain any cycles, so each contraction in C connects a separate branch of E. A TTNS is in a *canonical state*, if all branches of the environment are in their canonical forms (Figure 5.7). Each branch has exactly one canonical form, but a canonical state of the whole TTNS depends on its separation into S and E.

It is natural to generalize canonical states from MPS to multidimensional tensor networks. However, canonical states cannot be defined for networks that contain cycles, for example for PEPS. Nonexistence of canonical states makes local operations challenging. Even a calculation of the expectation value of a local observable requires the contraction of the full network. This is known to be a hard operation for PEPS [69]. The existence of canonical states in TTNS is the main argument in favour of TTNS over PEPS.

5.2 Network topology

Tree tensor networks are instances of graphs. A *network topology* is the graph structure of a TTNS. Besides the requirement that TTNS do not contain any cycles, their topologies can be arbitrary. The *default topology* is a prescribed initial topology with the following structure. Tensors in each column are interconnected and the columns are connected through the tensors in the bottom row (Figure 5.8). The left-bottom tensor is considered as a system S and the rest of the network as an environment E. It is further required that the network is in the canonical state.

To define the default topology more precisely, I consider a rectangular two-dimensional lattice with L_x sites in x direction and L_y sites in y direction. Let (i_x, i_y), where $i_x \in \{1, \ldots, L_x\}$ and $i_y \in \{1, \ldots, L_y\}$, be coordinates of a single lattice site. Each lattice site corresponds to a single tensor in a tensor network. The tensors carry one physical index and one to three virtual indices. All tensors are connected in the y direction. A tensor at (i_x, i_y) is connected to a tensor at $(i_x, i_y - 1)$ if $i_y > 1$, and to a tensor at $(i_x, i_y + 1)$ if $i_y < L_y$. Additionally, tensor in the bottom row are connected in the x direction. A tensor at $(i_x, 1)$ is connected to a tensor at $(i_x - 1, 1)$ if $i_x > 1$ and to a tensor at $(i_x + 1, 1)$ if $i_x < L_x$.

Figure 5.8: Default topology of a TTNS. The network is in the canonical form with respect to the left-bottom tensor.

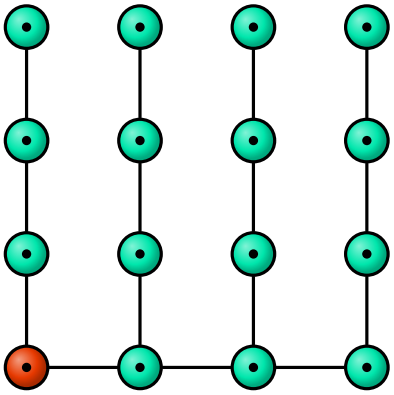
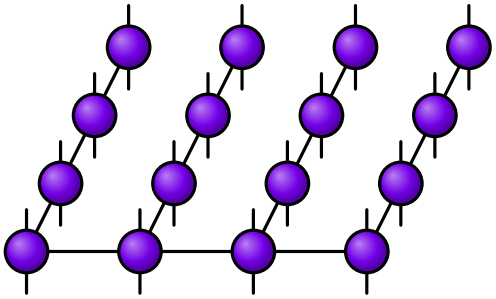


Figure 5.9: Each tensor in a TTNO has two physical indices and up to three virtual indices. All TTNO have the default topology.



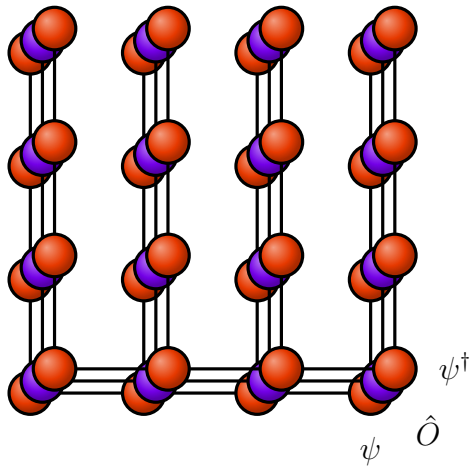
The advantage of the default network topology is that the network distance between any two sites is less than $L_x + 2L_y - 3$. Such TTNS can better capture correlations between distant sites.

5.3 Observables

Observables are represented by tree tensor network operators (TTNO). They have a similar structure as TTNS, but they carry two physical indices instead of a single physical index. TTNO in TEBDOL are constructed explicitly and have the default topology. All tensors are connected in the y directions and the tensor columns are connected in the first row (Figure 5.9). Expectation value of an observable O is obtained by contracting a TTNS ψ, a TTNO $\hat{O}$, and a conjugate of ψ (Figure 5.10).

For local operators acting on the system S only, all tensors that correspond to the environment are identity tensors. It is thus easy to calculate the expectation value of a local operator if a TTNS is in the canonical state with respect to S. Identity tensors in the TTNO lead to a contraction of the TTNS with its conjugate in the environment. This gives an identity tensor for each environment branch because these branches are in the canonical form. The contraction of the full

Figure 5.10: To calculate the expectation value of a TTNO $\hat{O}$ for a TTNS ψ, $\hat{O}$ is inserted between ψ and its conjugate $\psi^\dagger$, and the network is contracted. ψ, $\hat{O}$, and $\psi^\dagger$ have the default topology. To achieve good performance, the contraction should start from the leaves.



network therefore simplifies to a contraction of the system tensors only (Figure 5.11).

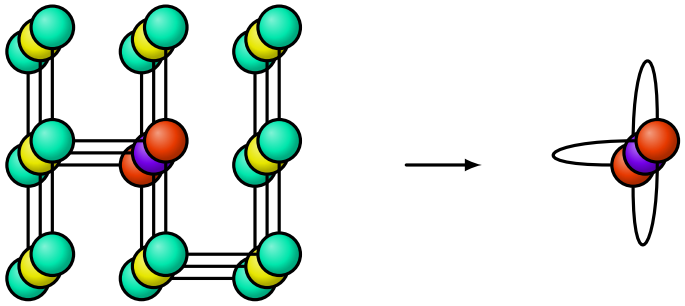
5.4 Index shifting

Not all neighbouring sites are connected directly in a tree tensor network. It is then unclear how to apply a two-site operator to a pair of such sites. The method used in TEBDOL rearranges the network into a new topology in which the respective sites are connected directly.

The basic operation to achieve this is to transfer an index from one tensor to another tensor connected to it. Let T_A and T_B be two connected tensors. T_A is connected with its index α to T_B. The task is to move an index γ from T_A to T_B. First, T_A is decomposed into tensors U and T'_A. U includes indices α and γ, and T'_A includes all other indices of T_A. T_A is then replaced by T'_A. The tensor U is contracted with T_B into T'_B which replaces T_B. The result is that the index γ becomes a part of the tensor T'_B corresponding to a different site. A shift of an index of one tensor is typically accompanied by a shift of a corresponding index of another connected tensor. This way it is possible to shift a connection from one pair of tensors to another pair of tensors (Figure 5.12).

The orthonormalization condition in the decomposition of T_A can be chosen so that either T'_A or U is semi-unitary. The choice depends on a canonical state one wants to obtain. It is also possible to truncate the dimensions of the new indices created in the decomposition. This may be necessary to keep index dimensions bounded.

Figure 5.11: Expectation value of a local operator is easy to calculate if the TTNS is in the corresponding canonical state. A local TTNO consists of a local operator acting on a system S and identity tensors acting on an environment E. If the branches of the TTNS are in their canonical forms, the contraction of branches produces identity tensors. The expectation value can be therefore calculated without contracting the full network. Instead, only a contraction of the TTNS tensors with the TTNO tensors and with the conjugate TTNS tensors that belong to S is performed.



5.5 Suzuki–Trotter expansion

The Suzuki–Trotter expansion is a method to approximate the action of exponential operators. In TEBDOL, it is used to approximate the action of a propagator. The idea is to separate terms of the Hamiltonian into groups, so that operators in each group commute among themselves. It is easy to calculate the exponential of a sum of commuting operators because it is just a product of exponentials of individual terms.

As shown in Chapter 3, the terms in one-dimensional Hamiltonians with nearest-neighbour interactions can be separated into two groups [16, 18]. Similarly, the terms of the two-dimensional Bose–Hubbard Hamiltonian can be separated into four groups, two horizontal and two vertical sets of operators (Figure 5.13), given by

$$\hat{H} = \hat{D} + \hat{E} + \hat{F} + \hat{G}. \tag{5.3}$$

These four operators include all hopping terms for the odd bonds in the x direction ($\hat{D}$), for the even bonds in the x direction ($\hat{E}$), for the odd bonds in the y direction ($\hat{F}$), and for the even bonds in the y direction ($\hat{G}$). Local terms acting on a single site only can be included in any of them. Here, they are included in $\hat{D}$ and $\hat{E}$ — local terms associated with sites with odd x coordinates are included in $\hat{D}$, and local terms associated with sites with even x coordinates are included in $\hat{E}$. Terms in each group commute among themselves.

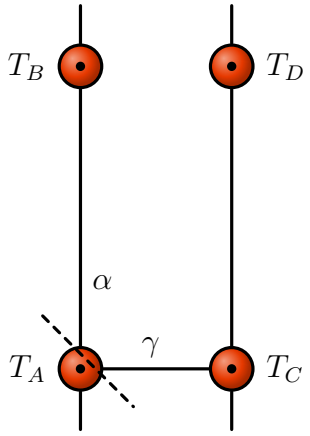
The full propagator for a small time step δt can be approximated by the Suzuki–Trotter expansion of the first order,

$$e^{-i\hat{H}\delta t} = e^{-i\hat{D}\delta t} e^{-i\hat{E}\delta t} e^{-i\hat{F}\delta t} e^{-i\hat{G}\delta t} + \mathcal{O}(\delta t^2). \tag{5.4}$$

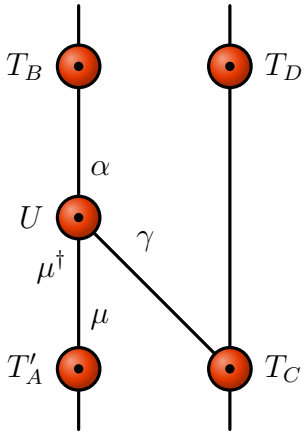
A better approximation is given by the second-order formula [74, 75]

$$e^{-i\hat{H}\delta t} = e^{-i\hat{D}\delta t/2} e^{-i\hat{E}\delta t/2} e^{-i\hat{F}\delta t/2} e^{-i\hat{G}\delta t/2} e^{-i\hat{G}\delta t/2} e^{-i\hat{F}\delta t/2} e^{-i\hat{E}\delta t/2} e^{-i\hat{D}\delta t/2} + \mathcal{O}(\delta t^3). \tag{5.5}$$

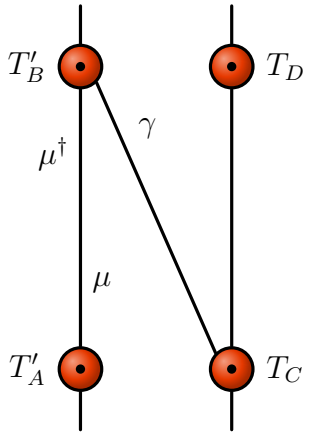
Figure 5.12: Shifting indices in a TTNS.



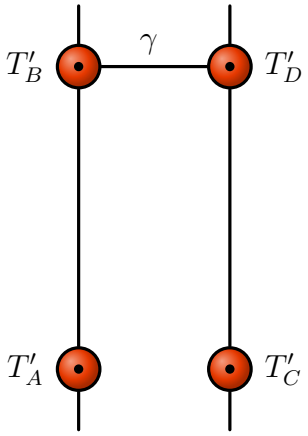
(a) Indices of T_A are separated into a group containing the virtual indices α and γ and a group containing all other indices. T_A is then decomposed with respect to this separation.



(b) The decomposition produces tensors U and T'_A. The orthonormalization can be chosen so that either U or T'_A is a semi-unitary tensor. If the TTNS is in a canonical form, it is also possible to truncate the dimensions of indices μ and $\mu^\dagger$.

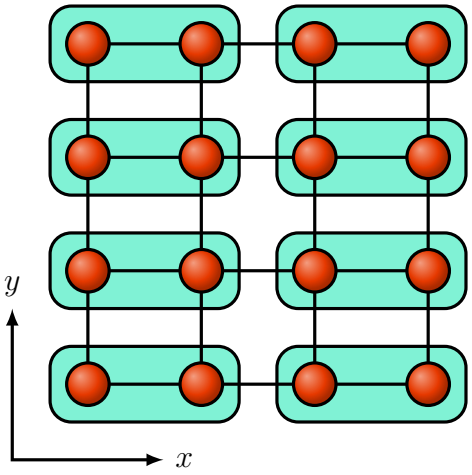


(c) Contracting U to T_B produces a tensor T'_B. This finalizes a shift of the index γ between tensors at different sites.

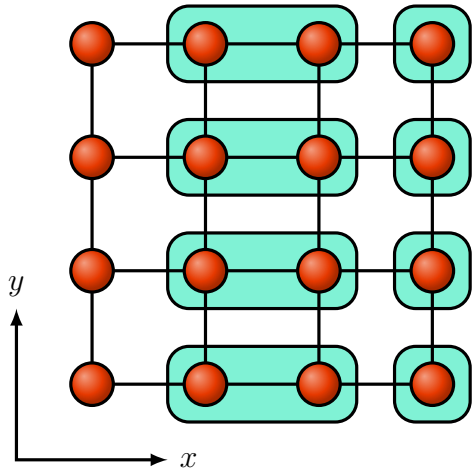


(d) The index can be shifted also from T_C to T_D. Both shifts combined produce a TTNS with a connection between the different pair of neighbouring tensors.

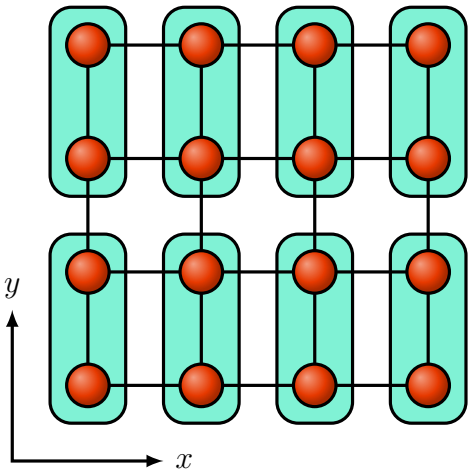
Figure 5.13: Horizontal and vertical operators. Terms in a two-dimensional Hamiltonian with nearest-neighbour interactions can be separated into four groups, $\hat{H} = \hat{D} + \hat{E} + \hat{F} + \hat{G}$. Terms in each group commute among themselves.



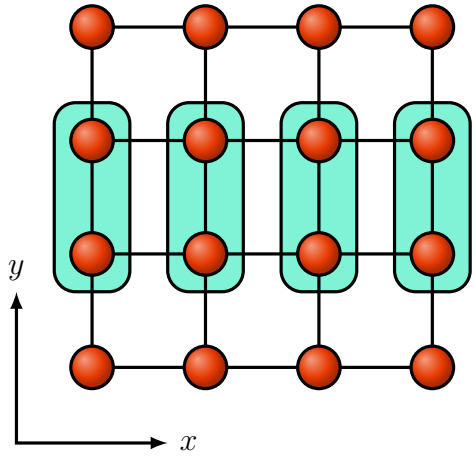
(a) Operator $\hat{D}$ includes all terms corresponding to the odd horizontal bonds. Each two-site operator also includes local terms corresponding to its left site.



(b) Operator $\hat{E}$ includes all terms corresponding to the even horizontal bonds. Again, each two-site operator also includes local terms corresponding to its left site. Local terms at the edge of the lattice are included in the single-site operators.



(c) Operator $\hat{F}$ includes all odd vertical bonds.



(d) Operator $\hat{G}$ includes all even vertical bonds.

Each term on the right-hand side is an exponential of a sum of commuting operators, $\hat{X} = \sum_j \hat{X}_j$, where $\hat{X} = \hat{D}$, $\hat{E}$, $\hat{F}$, or $\hat{G}$, and $\hat{X}_j$ are constituent terms of $\hat{X}$. Each term is therefore a product of exponentials,

$$e^{-i\hat{X}\delta t/2} = \prod_j e^{-i\hat{X}_j\delta t/2}. \tag{5.6}$$

In order to calculate the action of $\exp(-i\hat{X}\delta t/2)$ on a TTNS, one applies all operators $\exp(-i\hat{X}_j\delta t/2)$ to it. Each of them acts on a single site or on a pair of adjacent sites. The product $e^{-i\hat{G}\delta t/2}e^{-i\hat{G}\delta t/2}$ can be further simplified to $e^{-i\hat{G}\delta t}$.

5.6 Time evolution

Terms in operators $\hat{D}$, $\hat{E}$, $\hat{F}$, and $\hat{G}$ act on each pair of adjacent sites. Adjacent sites are not necessarily connected in a TTNS. It is therefore necessary to introduce a method to calculate the action of a two-site operator on any pair. The approach used in TEBDOL is based on rearranging the network. Before operating on a pair, indices are shifted through other tensors to create a connection between the pair.

5.6.1 Bonds in the y direction

Adjacent sites are connected in the y direction if a TTNS has the default topology. It is therefore not required to rearrange the network in order to apply operators $\exp(-i\hat{F}\delta t/2)$ and $\exp(-i\hat{G}\delta t/2)$. However, because the application of each two-site operator involves a truncation of the corresponding virtual indices, it is necessary to bring the TTNS into an appropriate canonical state in each step. The algorithm consists of the following steps for each term in $\exp(-i\hat{F}\delta t/2)$ and $\exp(-i\hat{G}\delta t/2)$:

1. Bring the TTNS into a canonical state, so the pair of adjacent tensors being operated on represents a system S of the canonical state.
2. Apply a double-site operator to the pair.
3. Truncate the virtual indices connecting the tensors in the pair.

A TTNS has the default topology at the beginning of a calculation. During each step it is necessary to bring it into a new canonical state. Some branches of the TTNS are already in their canonical forms, therefore it is not necessary to bring each tensor in the network into a canonical form again. Only the part that represented a system S in the previous step has to be properly reorthogonalized. This substantially decreases the task complexity.

5.6.2 Bonds in the x direction

The network has to be rearranged to apply all double-site operators in the x direction. In the following, L_x is the width of the lattice, that is the total number of sites in the x direction, and L_y is the height of the lattice, that is the total number of sites in the y direction. T_{i_x,i_y} denotes a tensor at a site with coordinates

(i_x, i_y), where $i_x \in \{1, \ldots, L_x\}$ and $i_y \in \{1, \ldots, L_y\}$. Furthermore, a *right index* is the index connecting a tensor to the adjacent tensor with a greater i_x coordinate, and analogously a *left index* is the index connecting a tensor to the adjacent tensor with a lower i_x coordinate. A *column* is a set of all pairs of tensors T_{i_x,i_y} and T_{i_x+1,i_y} for a given i_x and for all i_y.

The goal of the algorithm is to calculate the action of all operators included in $\exp(-i\hat{D}\delta t/2)$ and $\exp(-i\hat{E}\delta t/2)$, that is all two-site operators acting on bonds in the x direction, to a TTNS. The operator $\exp(-i\hat{D}\delta t/2)$ comprises operators acting on the odd bonds, whereas $\exp(-i\hat{E}\delta t/2)$ comprises operators acting on the even bonds. The algorithm is similar in both cases, so it is only described for odd bonds here.

The operators are applied sequentially to columns (Figure 5.14). The procedure starts with the first column and proceeds to the right until it reaches the last column. Because the operators act on the odd bonds only, it proceeds from the column at i_x to the column at $i_x + 2$. For each column, all operators are applied while the network is being rearranged. The algorithm uses index shifting described earlier. For all i_y the following steps are performed:

1. Bring the TTNS into a canonical state in which tensors T_{i_x,i_y} and T_{i_x+1,i_y} represent the system S of the canonical state. It is not necessary to reorthogonalize all tensors in the TTNS to do so. It is usually sufficient to bring only the tensors that were modified in the last step into the canonical state.

2. Apply the respective operator to a pair of tensors T_{i_x,i_y} and T_{i_x+1,i_y}.

3. Truncate the virtual indices connecting T_{i_x,i_y} and T_{i_x+1,i_y}.

4. If $i_y < L_y$, shift the right index of T_{i_x,i_y} to T_{i_x,i_y+1} and truncate the virtual indices connecting T_{i_x,i_y} and T_{i_x,i_y+1}.

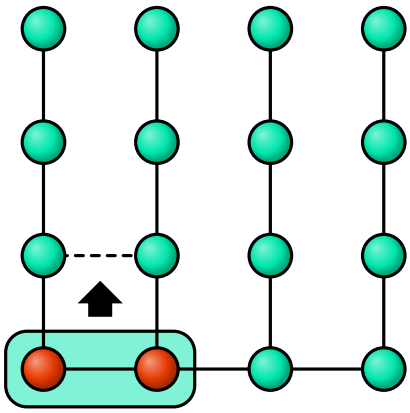
5. If $i_y < L_y$, shift the left index of T_{i_x+1,i_y} to T_{i_x+1,i_y+1} and truncate the virtual indices connecting T_{i_x+1,i_y} and T_{i_x+1,i_y+1}.

Again, each truncation step requires the TTNS to be in a canonical state. Otherwise the retained states are not the states that have the highest weight in the full density matrix of the system.

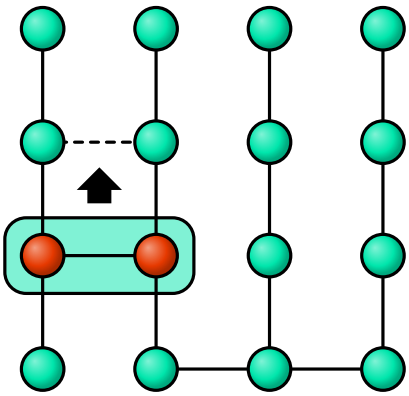
After applying operators to a column at i_x, the tensors T_{i_x+1,i_y} for all i_y are reorthogonalized. The procedure then continues with the column at $i_x + 2$. After processing all odd columns, the program starts to apply operators to the even columns. The procedure works analogously except the columns are processed from right to left. The final network topology after applying all operators in $\exp(-i\hat{D}\delta t/2)$ and $\exp(-i\hat{E}\delta t/2)$ is a topology similar to the default topology, except that columns are connected through top-row tensors with $i_y = L_y$. The horizontal operators are applied twice in the second-order Suzuki–Trotter expansion. The second application starts from this topology and reaches the default topology in the end.

After applying all operators corresponding to the odd bonds in the x direction to a TTNS, the network topology becomes essentially one-dimensional with an accompanied increase of the maximum distance between sites. During the development of the method, I experimented with several other schemes of rearranging

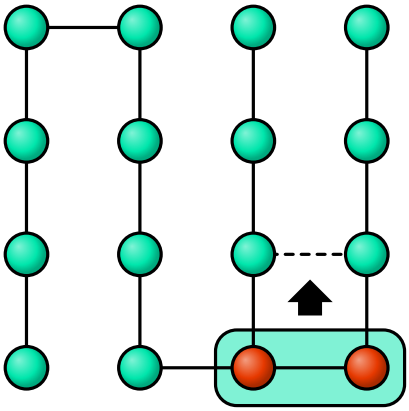
Figure 5.14: The scheme for applying time-evolution operators corresponding to the odd bonds in the x direction to a TTNS.



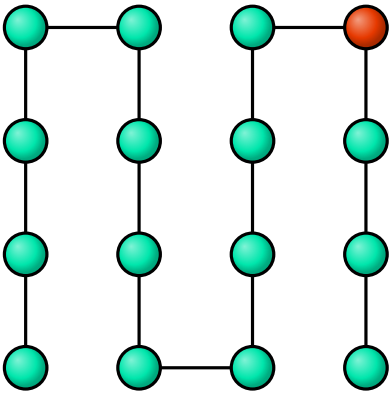
(a) The TTNS has the default topology initially. A double-site operator is first applied to the pair of tensors in the bottom-left corner of the lattice. The connection between the pair is then shifted upwards.



(b) An operator is applied to the next pair of tensors and the connection is shifted upwards again. The procedure continues for all pairs in the column.



(c) The branch with applied operators is transformed into the canonical form. The procedure continues in the next column two sites away.



(d) The network topology obtained after applying all operators corresponding to the odd bonds in the x direction. The next step is to apply operators for the even bonds. It proceeds from right to left.

the network. One scheme was to bring the network back into the default topology after applying all operators to a single column. This would keep the graph distance between any pair of tensors bounded. However, I obtained the best accuracy and also the fastest calculation times with the method described above. TEBDOL therefore uses the described method in the two-dimensional algorithm.

6. One-dimensional systems

The main topic of this thesis is modelling of multidimensional systems. The work naturally started with modelling of one-dimensional systems. They are simpler to deal with and ideal for testing various algorithms and ideas. At the same time, the physics of one-dimensional bosons is highly nontrivial [76]. I applied the one-dimensional method to several interesting problems during the course of the development. In this chapter, I report on two extensions of the usual one-dimensional approach.

The first extension is the application of TEBDOL [41] to mixtures of atoms in optical lattices. Instead of one particle species, I simulated the time evolution of an interacting ultracold gas composed of two bosonic species [77]. The second extension was the development of a parallel version of the algorithm that can be run on computer clusters [41]. This is especially important for hard problems that consume a lot of computing resources.

6.1 Binary mixtures

Two-component ultracold gases exhibit a rich set of interesting effects [78, 79]. The goal of this study was to simulate the time evolution of a binary mixture of bosonic particles in a one-dimensional optical lattice. In particular, I investigated the influence of the second species on the phase collapse and revivals of the first species. Revivals of the phase coherence [34, 80] represent a well-known effect in the Bose–Hubbard model. I specifically investigated the case of noninteracting species A. In the absence of species B, this would lead to perfect revivals. By turning on interaction between A and B, the revivals of A are influenced by the dynamics of B [44, 77].

The Hamiltonian of the binary Bose–Hubbard model is given by

$$\begin{aligned}\hat{H} = &- J_A \sum_{\langle j,k\rangle} \hat{a}_j^\dagger \hat{a}_k + \frac{U_{AA}}{2} \sum_j \hat{n}_j^A \left(\hat{n}_j^A - 1\right) \\ &- J_B \sum_{\langle j,k\rangle} \hat{b}_j^\dagger \hat{b}_k + \frac{U_{BB}}{2} \sum_j \hat{n}_j^B \left(\hat{n}_j^B - 1\right) \\ &+ U_{AB} \sum_j \hat{n}_j^A \hat{n}_j^B,\end{aligned} \tag{6.1}$$

where at a site j, $\hat{a}_j^\dagger$ and $\hat{b}_j^\dagger$ are the creation operators, $\hat{a}_j$ and $\hat{b}_j$ are the annihilation operators, and $\hat{n}_j^A = \hat{a}_j^\dagger \hat{a}_j$ and $\hat{n}_j^B = \hat{b}_j^\dagger \hat{b}_j$ are the particle number operators for particle species A and B, respectively. The parameter J_X denotes the amount of tunnelling for species X, and the parameter U_{XY} denotes the intensity of on-site repulsion between species X and Y. Only the nearest-neighbour tunnelling is considered, therefore the angle brackets denote sum over adjacent sites. The energy scale is fixed by setting $\hbar \equiv 1$.

Phase collapse and revivals in optical lattices are a result of a sudden change in lattice parameters from $J_X \gg U_{XX}$ to $J_X \ll U_{XX}$. Initially, the system is assumed to be in the ground state of the Hamiltonian (6.1) with parameters

$J_A = J_B \neq 0$ and $U_{AA} = U_{BB} = U_{AB} = 0$. This ground state is given by [77, 80]

$$|\psi\rangle = \frac{1}{\sqrt{N_A! N_A^{N_A} N_B! N_B^{N_B}}} \left(\sum_{j=1}^{L} \alpha_j \hat{a}_j^\dagger \right)^{N_A} \left(\sum_{j=1}^{L} \beta_j \hat{b}_j^\dagger \right)^{N_B} |0\rangle, \tag{6.2}$$

where L is the number of lattice sites, $|0\rangle$ is the vacuum state, and N_A and N_B are the total particle numbers for species A and B, respectively. For a lattice with open boundary conditions, the amplitudes are given by [77, 80]

$$\begin{aligned} \alpha_j &= \sqrt{\frac{2N_A}{L+1}} \sin\left(\frac{\pi}{L+1} j\right), \\ \beta_j &= \sqrt{\frac{2N_B}{L+1}} \sin\left(\frac{\pi}{L+1} j\right). \end{aligned} \tag{6.3}$$

At time $t = 0$, the parameters of the Hamiltonian (6.1) corresponding to particle species A suddenly change to the opposite set of parameters $J_A = 0$ and $U_{AA} \neq 0$. If there are no interactions between species, the quench leads to the perfect revivals for species A. Otherwise, the revivals of A are affected by the dynamics of B.

The physical quantity observed in this kind of experiments is the mean number of particles with zero quasimomentum. It can by directly measured as the intensity of the central peak in time-of-flight images [34]. The density of particles with zero quasimomentum for species A is given by [80]

$$n(t) \equiv \frac{1}{L} \sum_{j=1}^{L} \sum_{k=1}^{L} \langle \hat{a}_j^\dagger \hat{a}_k \rangle_t, \tag{6.4}$$

where $\langle \hat{a}_j^\dagger \hat{a}_k \rangle_t$ are the single-particle density matrix elements

$$\langle \hat{a}_j^\dagger \hat{a}_k \rangle_t = \langle \psi | e^{i\hat{H}t} \hat{a}_j^\dagger \hat{a}_k e^{-i\hat{H}t} | \psi \rangle. \tag{6.5}$$

A normalized particle density, useful for comparing systems with unequal numbers of particles, is given by

$$\tilde{n}(t) = \frac{n(t)}{n(0)}. \tag{6.6}$$

I investigated the behaviour of the model for vanishing J_A and a small nonvanishing J_B. The system was initially in the superfluid state (6.2) with the given amplitudes (6.3). The lattice had $L = 51$ sites, $N_A = 25$ particles of species A, and $N_B = 25$ particles of species B. The simulation was performed for several sets of parameters U_{AB} and U_{BB}. The maximum bond dimension during the calculation was $D = 600$. Figure 6.1 shows dependence of $\tilde{n}$ on time for interspecies interaction strength $U_{AB} = U_{AA}$ and for several intraspecies interaction strengths U_{BB}. The revival peaks are in same positions as they would be without species B present. However, intraspecies interactions attenuate the revivals. The attenuation is the strongest for vanishing U_{BB}, and the revivals are intensified with increasing U_{BB}.

Figure 6.2 shows the dependence of the attenuation of the first and the second peak on U_{BB}. The behaviour depends on U_{AB}. For weak interspecies interactions, that is for small U_{AB}, the attenuation slightly increases for large U_{BB}. For strong

Figure 6.1: Normalized quasimomentum revivals for particle species A on a lattice with $L = 51$ sites and $N_A = N_B = 25$ particles for the interspecies interaction strength $U_{AB} = U_{AA}$ and for the intraspecies interaction strengths $U_{BB} = 0\,U_{AA}$, $1/8\,U_{AA}$, $1\,U_{AA}$, and $8\,U_{AA}$. Hopping parameters were $J_A = 0\,U_{AA}$ and $J_B = 1/16\,U_{AA}$.

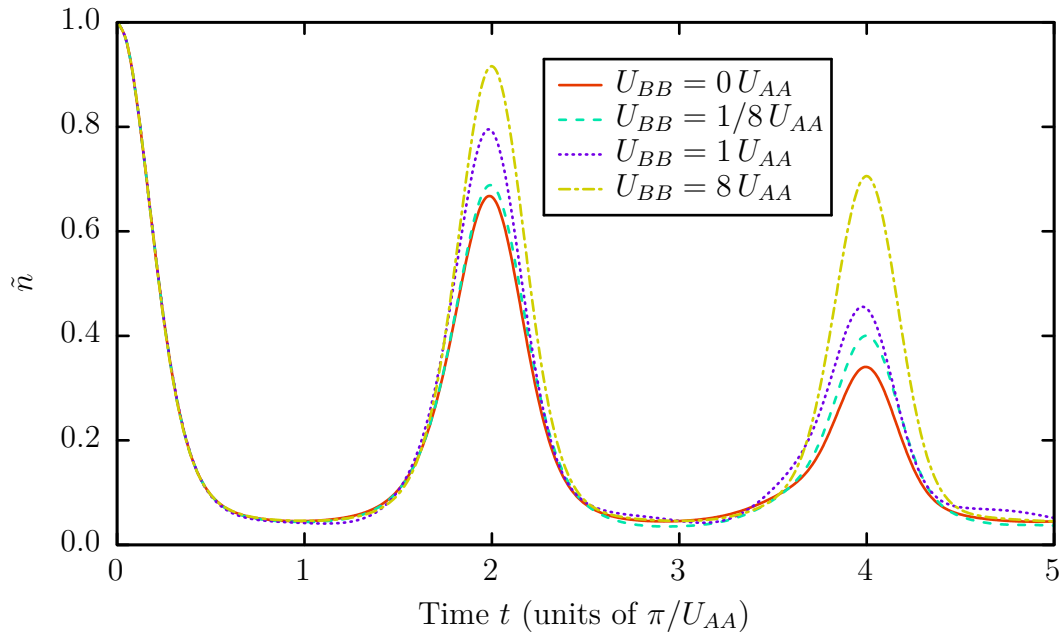
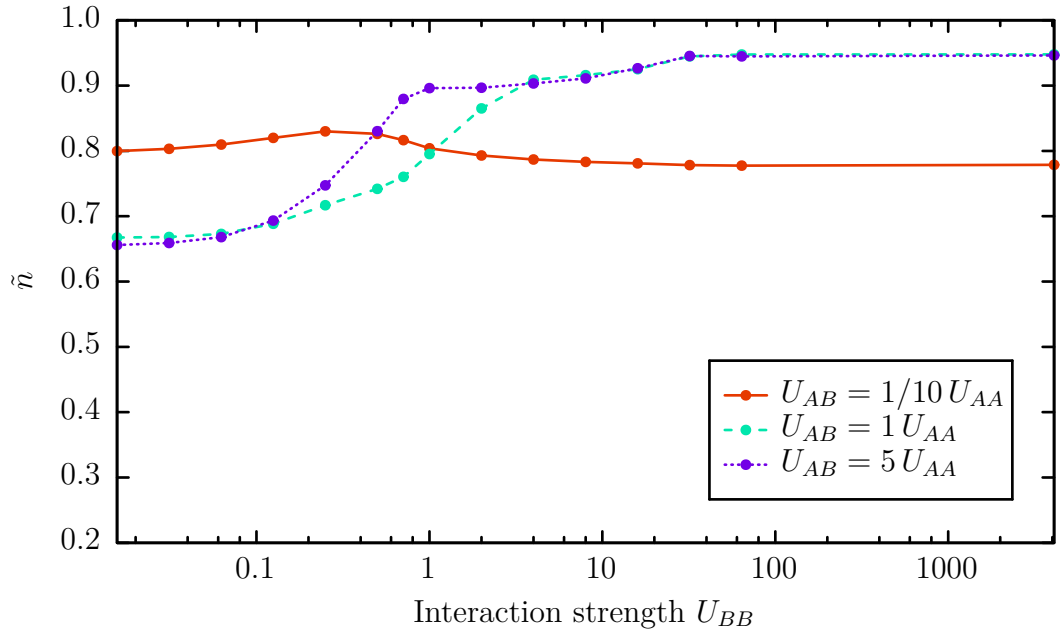
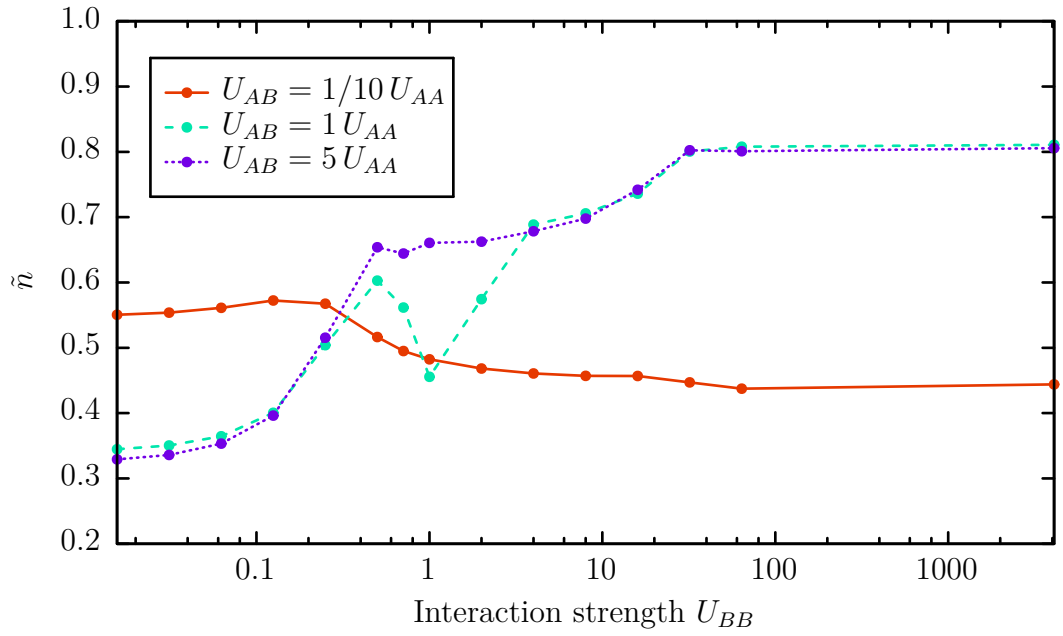


Figure 6.2: Dependence of $\tilde{n}$ at its peaks on U_{BB}. Attenuation is defined as the decrease of $\tilde{n}$ at the top of a peak in comparison with its initial value.



(a) The dependence of $\tilde{n}$ on the parameter U_{BB} for the peak at $t = 2\,\pi/U_{AA}$.



(b) The dependence of $\tilde{n}$ on the parameter U_{BB} for the peak at $t = 4\,\pi/U_{AA}$.

interspecies interactions, the attenuation decreases with increasing U_{BB}. There is also a dip in the second-peak values for $U_{AB} = U_{AA}$. It is a sign of a complex dynamics with a resonance for close values of the interaction parameters.

This section briefly summarized an investigation of a binary mixture of particles in an optical lattice. Further details are presented in article [77]. Binary Bose–Hubbard model (6.1) is a simple extension of the standard Bose–Hubbard model. Interactions between particles influence the dynamics of both species in a nontrivial way. The damping of revivals is the largest with a large interspecies interaction U_{AB} strength and a small intraspecies interaction strength U_{BB}. In this case, particles B do not repel each other and easily move in the lattice. They act as a coupling between particles A, which causes a coherence loss.

6.2 Parallelization

I implemented a parallel version of the algorithm for one-dimensional models. This section briefly summarizes my approach and presents the scaling performance of the algorithm. Full details are explained in article [41]. The parallel algorithm exhibits remarkable scalability, and allows to simulate complicated models with large bond dimensions. Its trade-off is that the accuracy of results diminishes if there are substantial truncations of virtual indices.

The basic idea is to divide the lattice into blocks, distribute the blocks among compute nodes, and perform calculations on each block separately. Each node keeps tensors that correspond to the assigned block only. The nodes holding adjacent blocks exchange tensors corresponding to block boundaries as necessary. The fundamental time-evolution operator is a two-site operator, so the minimum size of a block is two sites.

The algorithm uses a representation from the original formulation of the TEBD algorithm [17, 18]. A system state is decomposed into a set of tensors Γ carrying one physical index σ and two virtual indices, and a set of diagonal tensors Λ carrying two virtual indices. A wavefunction of the system takes the form [16]

$$|\psi\rangle = \sum_{\sigma_1,\ldots,\sigma_L} \Gamma_1^{\sigma_1}\Lambda_1\Gamma_2^{\sigma_2}\Lambda_2\ldots\Lambda_{L-1}\Gamma_L^{\sigma_L}|\sigma_1,\ldots,\sigma_L\rangle, \tag{6.7}$$

where L is the length of the lattice, $\Gamma_i^{\sigma_i}$ are tensors associated with the lattice site i, and Λ_i are tensors associated with the bond between sites i and $i+1$. The tensors utilize a matrix notation, so every $\Gamma_i^{\sigma_i}$ and Λ_i is a matrix. The virtual indices of $\Gamma_i^{\sigma_i}$ correspond to its matrix indices and the physical index is denoted by σ_i. Similarly, the virtual indices of Λ_i are identified with its matrix indices. A multiplication of adjacent matrices denotes a tensor contraction over the respective virtual indices of adjacent tensors.

The advantage of this representation is that the density matrix at a site i can be obtained just from the site tensor $\Gamma_i^{\sigma_i}$ and the diagonal bond tensors Λ_{i-1} and Λ_i. In each time-evolution step, a two-site operator is applied to a pair of adjacent tensors, and the virtual indices connecting the pair are truncated [16–18, 81]. The chosen representation makes it possible to do this independently for each pair of lattice sites. If a pair of sites is separated by a boundary between blocks, one of the tensors is first transferred to the adjacent compute node, the calculation is performed on that node, and then the tensor is transferred back to the original

node. The calculation of time evolution, which is the most time-consuming part of the algorithm, is therefore performed in parallel.

To calculate the expectation value of a local operator, it is again sufficient to work with local tensors only. For nonlocal operators, as is the quasimomentum operator discussed in the previous section, it is necessary to sequentially contract the full tensor network. The contraction starts on the node that keeps the block corresponding to one lattice edge. The block tensors are contracted, and the resulting tensor is transferred to the next node. The contraction then continues on each node sequentially. The last node produces the final results. The process is therefore serial and there is no parallel speedup. This part of the algorithm is however substantially faster than the time-evolution calculation and represents only a little hit on the total performance.

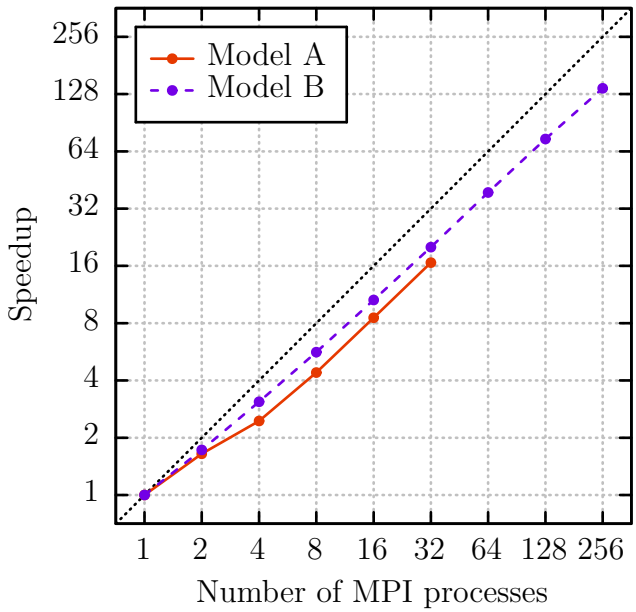
The parallel scaling is characterized by two measures — the strong and the weak scaling. Strong scaling describes dependence of the calculation time on the number of parallel processes for a fixed problem size. On the other hand, weak scaling describes dependence of the calculation time on the number of parallel processes when the problem size increases with the number of processes. Strong scaling therefore shows what speedup can be achieved for a given problem, and weak scaling shows how large problems can be solved with increased resources.

The measured scaling performance is shown in Figure 6.3. The calculation involved a simulation of time evolution of the Bose–Hubbard model after a quench that produced a quasimomentum collapse and its subsequent revivals. The strong scaling benchmark was performed for two models with $L = 65$ and with $L = 513$ lattice sites. The obtained scaling is approximately linear with a factor of about one-half of the perfect scaling. The weak scaling benchmark shows that the calculation time stays constant if the lattice size and the number of compute nodes are increased by the same factor simultaneously.

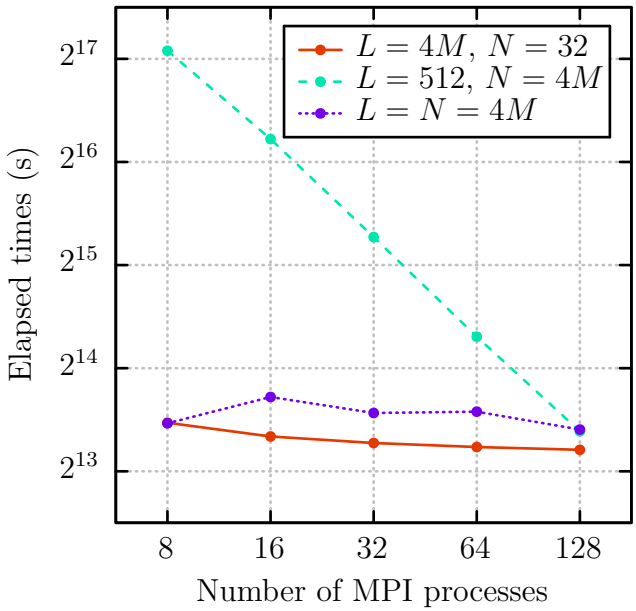
It was assumed that truncations of virtual indices can be performed for each block independently. However, this assumption is invalid if the weights of truncated states in the density matrix are too large [41, 82]. An accurate truncation has to take into account all sites in the system. In the serial version of the algorithm this is achieved by keeping the tensor network in a canonical state and by reorthogonalizing the tensors sequentially. With parallel truncations and without reorthogonalizing the tensors after each step, the truncations gradually become less accurate. This is not a problem if the truncations are small. If they are not, it is necessary to reorthogonalize the tensors after each step. This effectively cancels the parallel speedup. This point has to be taken into account if one wishes to use the parallel code. It is one of the reasons I implemented the parallel version for one-dimensional models only. However, even the serial calculation is inherently not accurate with large truncations. Therefore both the serial and the parallel version of the algorithm are valid only in the case of limited truncations.

The parallel TEBDOL uses Message Passing Interface (MPI) for communication between nodes. It has been tested on several supercomputing clusters with various MPI libraries. The code is optimized, and it could be easily extended to other quantum models besides the Bose–Hubbard model. Further details about MPS truncation and about the implementation are in Appendices A and B.

Figure 6.3: Scaling benchmarks for the parallel TEBDOL.



(a) Strong scaling benchmark for two models. Model A has $L = 65$ sites, $N = 65$ particles, and the hopping parameter of $J = U/10$. The calculation was performed with bond dimensions up to $D = 2000$. Model B has $L = 513$ lattice sites, $N = 513$ particles, the hopping parameter of $J = U/25$, and bond dimensions up to $D = 1000$.



(b) Weak scaling for three definitions of the problem size. The number of lattice sizes L, the total number of particles N, and both parameters at the same time were increased with the number of parallel processes M. The Bose–Hubbard hopping parameter was $J = 1/10$, and the maximum bond dimension was $D = 2000$.

7. Two-dimensional systems

This chapter presents an introduction to simulating the two-dimensional Bose–Hubbard model with TTNS. I discuss the model properties and the important calculation parameters. My program is called TEBDOL [41]. I also present a comparison of time evolution of a small model obtained by exact diagonalization and by TEBDOL. Several difficult systems are investigated in the following chapters, in particular the phase revivals, boson expansion, and many-body localization in two dimensions. The calculation in this chapter therefore serves as an example.

7.1 Model

In this work I deal exclusively with the Bose–Hubbard model. The Hamiltonian of a model with isotropic tunnelling ($J = J_x = J_y$) and with no external potential is given by

$$\hat{H} = -J \sum_{\langle \mathbf{i},\mathbf{j} \rangle} \hat{a}_{\mathbf{i}}^{\dagger} \hat{a}_{\mathbf{j}} + \frac{U}{2} \sum_{\mathbf{i}} \hat{n}_{\mathbf{i}} \left(\hat{n}_{\mathbf{i}} - 1 \right), \tag{7.1}$$

where $\hat{a}_{\mathbf{i}}^{\dagger}$ is the bosonic creation operator, $\hat{a}_{\mathbf{i}}$ is the bosonic annihilation operator, $\hat{n}_{\mathbf{i}} = \hat{a}_{\mathbf{i}}^{\dagger} \hat{a}_{\mathbf{i}}$ is the particle number operator, J is the hopping parameter, and U is the on-site interaction parameter. The operators are indexed by the lattice site $\mathbf{i} = (i_x, i_y)$. There are nearest-neighbour interactions only, so the angle brackets denote sum over each pair of adjacent sites. The model assumes a square lattice with open boundary conditions. In the following I set $\hbar \equiv 1$ to simplify the discussion.

7.2 Parameters

Accuracy and performance of TEBDOL depend on a set of parameters. They represent a trade-off between the calculation time and results accuracy. They are also limited by the available computer resources such as the computer memory. By choosing a suitable set of parameters, a calculation can finish in a reasonable time and achieve a good approximation to the exact result.

The number of particles at a single site is unlimited due to the bosonic commutation relations between $\hat{a}_{\mathbf{i}}$ and $\hat{a}_{\mathbf{i}}^{\dagger}$. It is necessary to limit this number in a calculation. TEBDOL uses a parameter called physical dimension P that specifies the local dimension. The possible number of particles is then $0, \ldots, P-1$. If the total number of particles N in the system is a small number, it is possible to set $P = N + 1$ to cover all states. If the number of particles is large and $U > 0$, it is sufficient to set P to a suitably large number because states with many particles at a single site have very high energy and therefore low weight in the system density matrix. This was the case in all performed calculations.

The next important parameter is the time step δt. It should satisfy $\delta t \ll J^{-1}$, so the simulation can precisely capture the evolution during a single tunnelling time. Also, the accuracy of the Suzuki–Trotter decomposition increases with

decreasing time step. On the other hand, a small time step leads to an increase in the total calculation time and to more truncations of the tensor network.

The TTNS algorithm is based on truncating off state components that do not contribute substantially to the full state, that is they have low weight in the density matrix. The truncation takes place after each application of a two-site operator and also in the process of rearranging the network. In a tensor decomposition, a tensor is reshaped into a matrix, and a singular value decomposition is performed on this matrix. Let s_k, $k \in \{1, \ldots, D_\mathrm{s}\}$ be the singular values of a particular decomposition. The singular values are real, nonnegative and sorted in nonincreasing order, that is $s_k \geq s_{k+1}$. TEBDOL uses two parameters to control the truncation, ε and D. First, the smallest singular values are truncated off so that the sum of squares of the truncated values is relatively smaller than ε. That is, TEBDOL finds the smallest D_ε for which the inequality

$$1 - \frac{\sum_{k=1}^{D_\varepsilon} s_k^2}{\sum_{k=1}^{D_\mathrm{s}} s_k^2} \leq \varepsilon \tag{7.2}$$

holds. If D_ε is still larger than D, TEBDOL keeps only the first D singular values. Parameter ε therefore controls the weight of truncated singular values, and parameter D controls the number of truncated singular values. D is usually called bond dimension.

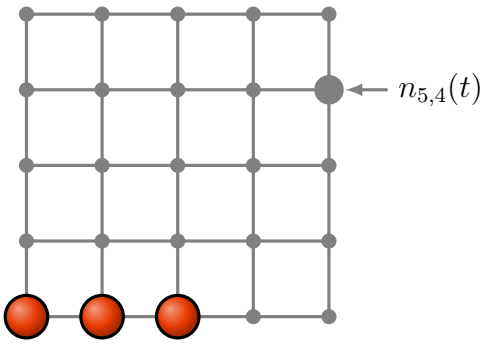
TEBDOL does not normalize the singular values after a decomposition. This leads to a decrease of the norm of a TTNS during the calculation. Therefore each measured observable has to be divided by the norm squared. This division is not explicitly stated in the expressions to keep them compact. A calculation typically starts with a TTNS with a unit norm. The decrease of norm corresponds to the weight of the states that were dropped from the full state description. It therefore serves as a measure of the accuracy of the result. The results are more accurate if the norm stays close to unity. If there is a lot of entanglement generated during the time evolution, the norm tends to decrease and approach zero quickly. The accuracy of the results is limited in this case. Parameter D has the largest influence on the norm. If the norm is too small, it could help to increase D. On the other hand, larger ε leads to significantly shorter calculation times. It also influences the norm negatively but less than D.

The parameters should be set in the following order. A physical dimension P depends on the problem being solved. A suitable time step δt is obtained from the parameters of the Bose–Hubbard model. I typically use $\delta t = 1/16\, J^{-1}$. Truncation limit ε should be set low enough to keep all important states in the TTNS, but high enough to discard very small singular values. Small singular values are often numerical artefacts, they significantly increase the calculation time, and introduce numerical instabilities. I use values in the range $10^{-7} < \varepsilon < 10^{-6}$. Finally, D should be set as large as possible considering the required computation time. I use values in the range $512 \leq D \leq 1024$ to finish a calculation in several days on a single computer.

7.3 Comparison to exact diagonalization

Exact diagonalization is a method that gives accurate time evolution of a quantum system limited only by the precision of a computing architecture. Its computation

Figure 7.1: The initial state in the example calculation is a product state of three particles located in the bottom left corner of the 5×5 lattice. The grid represents lattice sites and the balls represent particles. The measured quantity is the particle density $n_{5,4}(t)$ at the site with coordinates $(5, 4)$.



complexity grows exponentially with the system size. It is therefore limited to very small systems. I used exact diagonalization to calculate dynamics of the Bose–Hubbard model with several particles on a small lattice and compared the results to the results obtained with TEBDOL.

The system in question is the Bose–Hubbard model on a two-dimensional lattice with 5×5 sites and with three particles. The hopping parameter of the model is $J = U$. The particles are initially located at sites with coordinated $(1, 1)$, $(2, 1)$, and $(3, 1)$, and they are in a product state (Figure 7.1),

$$|\psi(t = 0)\rangle = \hat{a}^\dagger_{1,1}\hat{a}^\dagger_{2,1}\hat{a}^\dagger_{3,1}|0\rangle, \tag{7.3}$$

where $|0\rangle$ is the vacuum state with zero particles. The system then evolves in time, and because both J and U are nonzero, this evolution is complicated. The measured quantity is the expected particle number at a site with coordinates $(5, 4)$,

$$n(t) = \langle\psi(t)|\hat{n}_{5,4}|\psi(t)\rangle. \tag{7.4}$$

I implemented a simple exact diagonalization procedure with particle number conservation. The dimension of the corresponding Hilbert space is 2925, and its Hamiltonian has 26625 nonzero elements. The numerical diagonalization is a simple task that can be calculated in the order of seconds. The calculation time for TEBDOL is very short as well. The model captures all important effects, namely the particle hopping and the repulsive interactions between particles. It serves as a good tool for establishing the correctness of the algorithm and to investigate the role of the parameters in TEBDOL. The physical dimension was $P = 4$, and there was no truncation threshold, that is $\varepsilon = 0$.

The following figures compare the results from exact diagonalization and from TEBDOL. Figure 7.2 shows the particle density $n_{5,4}(t)$ obtained from exact diagonalization and from TEBDOL for three values of D. Figure 7.3 presents the same data as a difference between the TEBDOL results and the exact-diagonalization results. As expected, the results become more accurate with increasing D. However, the precise error dependence on D is complicated.

Figure 7.2: Comparison of the results calculated with exact diagonalization (ED) and with TEBDOL for three values of the maximum bond dimension D. Time step in TEBDOL was $\delta t = 1/16\, J^{-1}$. The figure shows the particle density $n_{5,4}$ at the site with coordinates $(5, 4)$. The accuracy of the results increases with D.

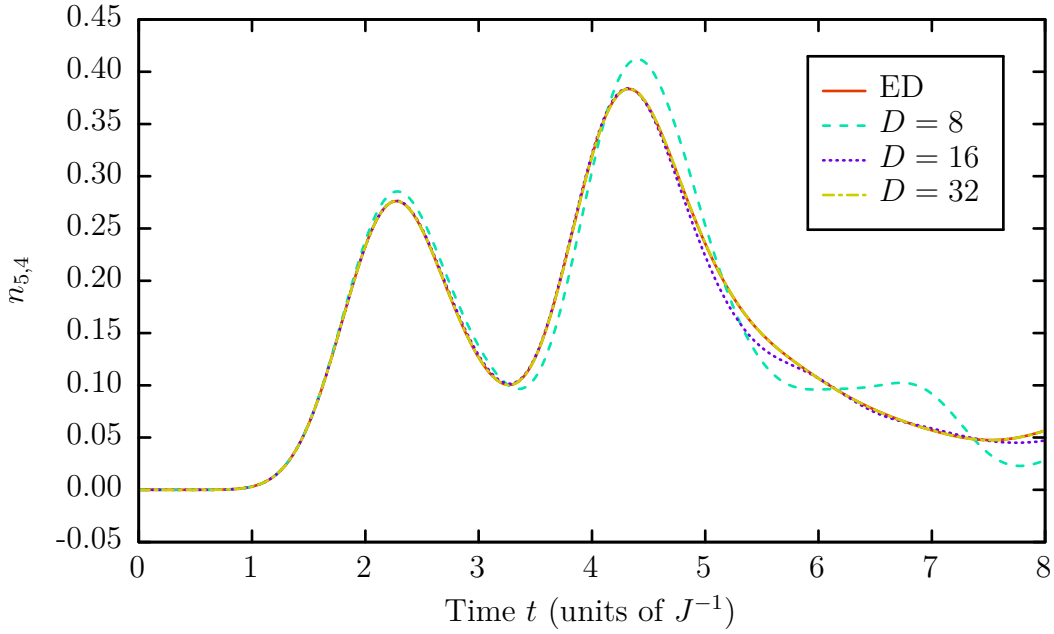


Figure 7.4 shows the error dependence on a decreasing time step δt. Again, the results become more accurate with decreasing time step. Figure 7.5 shows the same data as error ratios. In contrast to the dependence on D, the error is regular in time, that is the evolution of error has a similar shape for all three values of δt but differs in magnitude. It is important to note that due to a large value of $D = 32$, there are practically no truncations during the evolution. For a complicated calculation with more truncations, the dependence on δt would be less regular.

7.4 Convergence properties

The comparison above shows that the TEBDOL results converge to the exact values with increasing bond dimension and with decreasing time step. The exact dependence of the accuracy on the calculation parameters is complicated. It is therefore necessary to check convergence by increasing D and decreasing δt. A good measure of accuracy is the TTNS norm. If it is close to unity, then just a small portion of the important part of a TTNS was truncated away.

After the convergence in D has been established, the next task is to check convergence in δt. If the full simulated time interval is T, the number of required time steps is $M = T/\delta t$. The error is obtained by considering the second-order Suzuki–Trotter approximations at each time step,

$$e^{-i\hat{H}T} = \left(e^{-i\hat{H}\delta t}\right)^M = [U(\delta t)]^M + \mathcal{O}(M\delta t^3) = [U(\delta t)]^M + \mathcal{O}(T\delta t^2), \qquad (7.5)$$

Figure 7.3: Comparison of the results calculated with TEBDOL and with exact diagonalization for three values of the maximum bond dimension. Time step in TEBDOL was $\delta t = 1/16 J^{-1}$. Particle densities $n^T_{5,4}$ and $n^E_{5,4}$ are the densities at the site with coordinates $(5, 4)$ obtained from TEBDOL and from exact diagonalization, respectively. Their difference decreases with the increasing bond dimension D. Its absolute value is below 6×10^{-4} for $D = 32$.

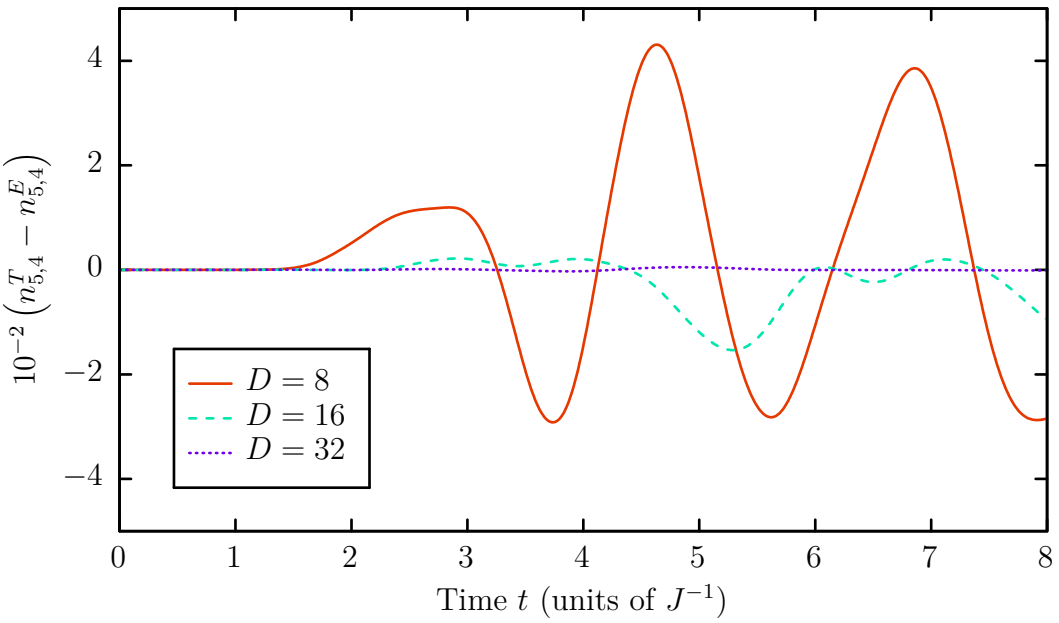


Figure 7.4: Comparison of results calculated with TEBDOL and with exact diagonalization for three values of the time step. The maximum bond dimension was $D = 32$ in the TEBDOL calculation. Particle densities $n^T_{5,4}$ and $n^E_{5,4}$ are the densities at the site with coordinates $(5, 4)$ obtained from TEBDOL and from exact diagonalization, respectively. Their difference decreases with the decreasing time step δt in TEBDOL. Its absolute value is below 4×10^{-5} for $\delta t = 1/64\, J^{-1}$.

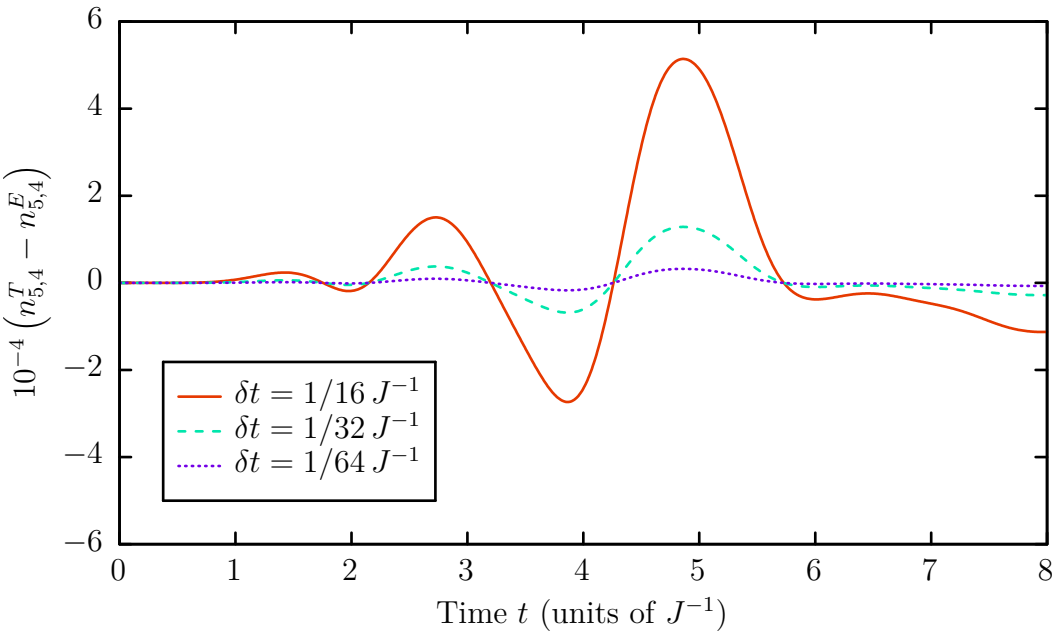
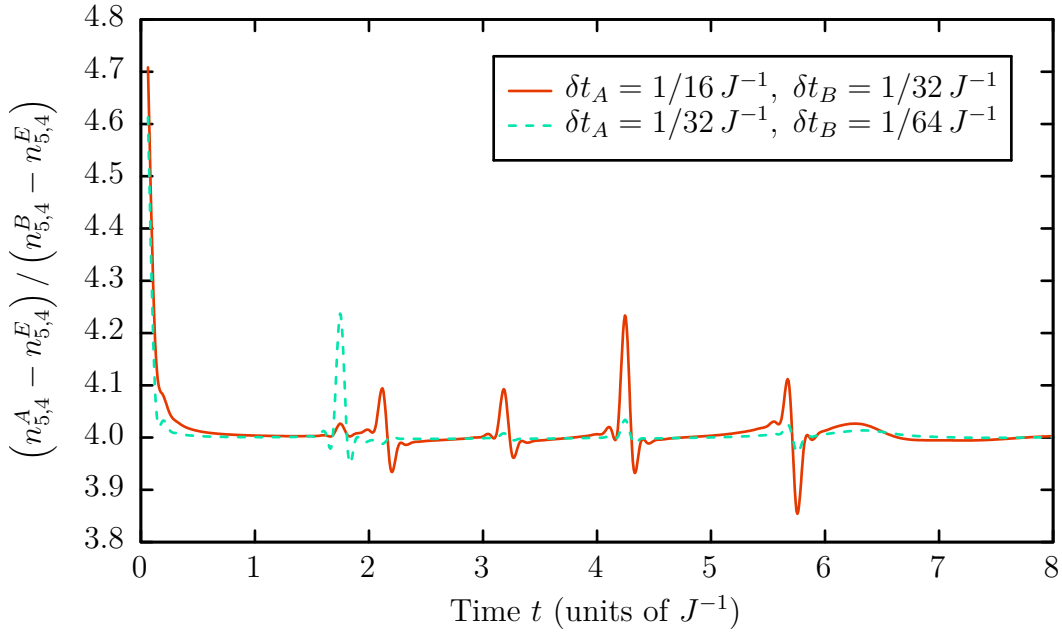


Figure 7.5: Ratio of the errors for three values of the time step. The maximum bond dimension was $D = 32$. Particle densities $n^A_{5,4}$ and $n^B_{5,4}$ were obtained from TEBDOL with time steps δt_A and δt_B, respectively. Particle density $n^E_{5,4}$ was obtained from exact diagonalization.



where $U(\delta t)$ is the exponential expansion. The error $\mathcal{O}(T\delta t^2)$ increases linearly with the length of the simulated time interval and quadratically with the time step. By doubling the time step, the error increases four times. This is the reason that the error ratio in Figure 7.5 is close to four. Deviations from the value of four are caused by the errors in the denominator being close to zero.

In summary, the convergence in δt for the second-order Suzuki–Trotter expansion is of a quadratic nature. It is possible to repeat the calculation for several decreasing values of τ and to use the quadratic dependence for $\tau \to 0$ in order to obtain an accurate result. However, this approach assumes that D is sufficiently large.

8. Phase revivals

Periodic phase collapses and revivals represent a well-known effect that takes place in a gas of ultracold atoms confined to an optical lattice [34, 77, 79, 80, 83–88]. The effect originates from a nonequilibrium evolution of the system after performing a quantum quench. It is well accessible experimentally. In the chapter on modelling one-dimensional systems, I discussed phase revivals in a binary mixture of particles in one-dimensional optical lattices. This chapter deals with single-species dynamics in a two-dimensional lattice.

8.1 Model

The system is described by the two-dimensional Bose–Hubbard model with open boundary conditions on a square lattice. The model further assumes independent hopping in the x and y directions. Its Hamiltonian is given by

$$\begin{aligned}\hat{H} = &- J_x \sum_{i_x=1}^{L_x-1} \sum_{i_y=1}^{L_y} \left(\hat{a}^\dagger_{i_x,i_y} \hat{a}_{i_x+1,i_y} + \hat{a}^\dagger_{i_x+1,i_y} \hat{a}_{i_x,i_y} \right) \\ &- J_y \sum_{i_x=1}^{L_x} \sum_{i_y=1}^{L_y-1} \left(\hat{a}^\dagger_{i_x,i_y} \hat{a}_{i_x,i_y+1} + \hat{a}^\dagger_{i_x,i_y+1} \hat{a}_{i_x,i_y} \right) \\ &+ \frac{U}{2} \sum_{i_x=1}^{L_x} \sum_{i_y=1}^{L_y} \hat{n}_{i_x,i_y} \left(\hat{n}_{i_x,i_y} - 1 \right),\end{aligned} \tag{8.1}$$

where L_x is the lattice width, L_y is the lattice height, $\hat{a}^\dagger_{i_x,i_y}$ is the creation operator at the site with coordinates (i_x, i_y), $\hat{a}_{i_x,i_y}$ is the annihilation operator at (i_x, i_y), and $\hat{n}_{i_x,i_y}$ is the particle number operator at (i_x, i_y). The model has two hopping parameters J_x, J_y, and an on-site interaction parameter U. The energy scale is defined by the parameter U and by setting $\hbar \equiv 1$.

Initially, the parameters of the Hamiltonian (8.1) are $J_x \neq 0$, $J_y \neq 0$, and $U = 0$. The system is in the ground state of the Hamiltonian. The choice of the parameters corresponds to the ground state being a superfluid state. Each particle is delocalized across the lattice. The initial state is given by [80]

$$|\psi\rangle = \frac{1}{\sqrt{N! N^N}} \left(\sum_{i_x=1}^{L_x} \sum_{i_y=1}^{L_y} \alpha_{i_x,i_y} \hat{a}^\dagger_{i_x,i_y} \right)^N |0\rangle, \tag{8.2}$$

where N is the total number of particles, and α_{i_x,i_y} is the amplitude associated with the site (i_x, i_y). The model assumes open boundary conditions, which lead to amplitudes given by [80]

$$\alpha_{i_x,i_y} = \sqrt{\frac{2N}{(L_x+1)(L_y+1)}} \sin\left(\frac{\pi}{L_x+1} i_x \right) \sin\left(\frac{\pi}{L_y+1} i_y \right). \tag{8.3}$$

Phase revivals are conveniently observed in the quasimomentum distribution. The mean number of particles with quasimomentum (q, r) is given by [80]

$$n(q, r; t) = \frac{1}{L_x L_y} \sum_{i_x=1}^{L_x} \sum_{i_y=1}^{L_y} \sum_{i'_x=1}^{L_x} \sum_{i'_y=1}^{L_y} \langle \hat{a}^\dagger_{i_x,i_y} \hat{a}_{i'_x,i'_y} \rangle_t e^{i a_{\mathrm{lat}} \left[q(i_x - i'_x) + r\left(i_y - i'_y\right) \right]}, \tag{8.4}$$

where (i_x, i_y) and (i'_x, i'_y) are positions of lattice sites, a_{lat} is the lattice constant, and $\langle \hat{a}^\dagger_{i_x,i_y} \hat{a}_{i'_x,i'_y} \rangle_t$ are the single-particle density matrix elements given by

$$\langle \hat{a}^\dagger_{i_x,i_y} \hat{a}_{i'_x,i'_y} \rangle_t = \langle \psi | e^{i\hat{H}t} \hat{a}^\dagger_{i_x,i_y} \hat{a}_{i'_x,i'_y} e^{-i\hat{H}t} | \psi \rangle. \tag{8.5}$$

Similarly to the one-dimensional case, the particle density with quasimomentum $(q, r) = (0, 0)$ is defined by

$$n(t) \equiv n(0, 0; t) = \frac{1}{L_x L_y} \sum_{i_x=1}^{L_x} \sum_{i'_x=1}^{L_x} \sum_{i_y=1}^{L_y} \sum_{i'_y=1}^{L_y} \langle \hat{a}^\dagger_{i_x,i_y} \hat{a}_{i'_x,i'_y} \rangle_t. \tag{8.6}$$

The particle density $n(t)$ corresponds to the height of the central peak in a two-dimensional time-of-flight measurement data. It depends on the total number of particles. To compare systems with unequal numbers of particles, it is convenient to define the normalized particle number

$$\tilde{n}(t) = \frac{n(t)}{n(0)}. \tag{8.7}$$

At the time $t = 0$, the parameters are suddenly changed to J_x, J_y and $U \neq 0$. Nonzero U leads to nonequilibrium time-evolution. If $J_x = J_y = 0$, the problem can be solved analytically, and the model exhibits perfect oscillations of the quasimomentum distribution. For nonzero J_x and J_y, the revivals are suppressed, and dynamics becomes complicated. The aim is to investigate the dependence of the revivals on J_x and J_y. Due to rapid entanglement growth in models with strong tunnelling, the study is limited to small values of J_x and J_y [89].

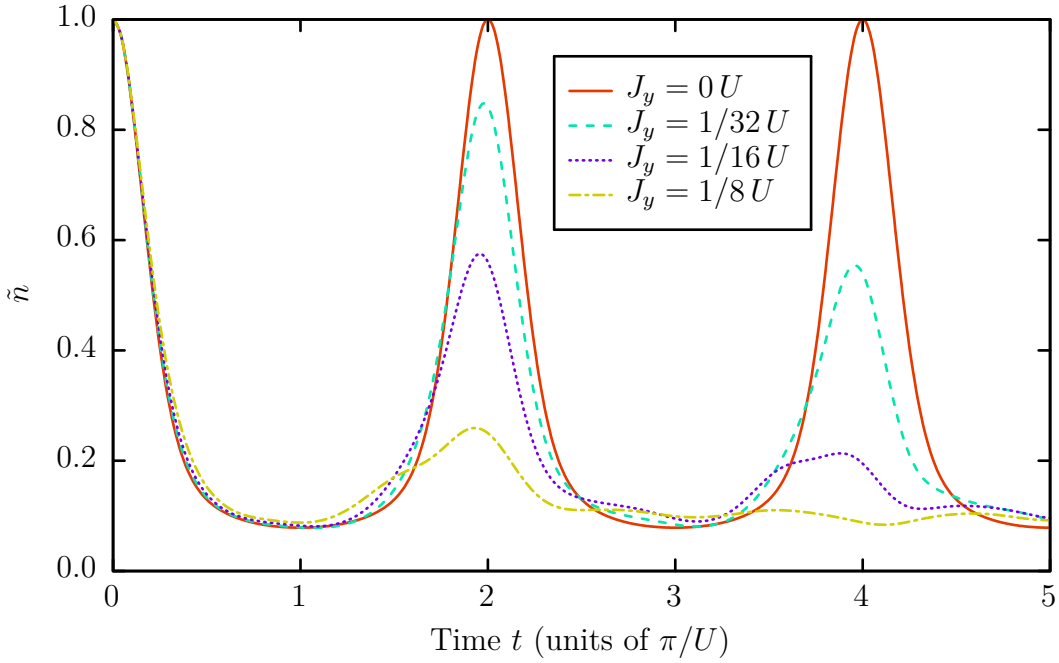
8.2 Calculation

The model was simulated on a 5×5 lattice with $N = 25$ particles. The calculation was performed for $J_x = 0\,U$, $1/32\,U$, $1/16\,U$, and $1/8\,U$, and for the same values of J_y. The case with $J_x = J_y = 0$ corresponds to the perfect revivals. The calculation parameters were chosen as follows:

- Physical dimension was set to $P = 16$. Due to a large value of U compared to J_x and J_y during the time evolution, the probability that many particles occupy a single site is very low. This value of P is sufficient to cover the most relevant part of the Hilbert space.
- The bond dimension was scaled dynamically with parameters $\varepsilon = 2^{32} \times \varepsilon_m \approx 4.77 \times 10^{-7}$ and $D = 512$, where $\varepsilon_m \approx 1.11 \times 10^{-16}$ is the double-precision machine epsilon in the IEEE 754 standard. The smallest singular values, whose sum was relatively smaller than ε, were truncated after each tensor decomposition. When there remained more than D states, TEBDOL further truncated the smallest values, so there were D singular values kept at most.
- The time step in the second-order Suzuki–Trotter approximation was set to $\delta t = 1/8\,U^{-1}$.
- The simulated time interval was $T = 16\,U^{-1}$.

The accuracy of the results was measured by the TTNS norm. The norm at the end of simulation was $\|\psi\| \approx 0.52$ for the most difficult case with $J_x = J_y = 1/8\,U$.

Figure 8.1: Normalized quasimomentum revivals on a two-dimensional 5×5 lattice with $N = 25$ particles, for horizontal hopping strength $J_x = 0\,U$, and for vertical hopping strengths $J_y = 0\,U$, $1/32\,U$, $1/16\,U$, and $1/8\,U$. The model with $J_x = J_y = 0\,U$ exhibits perfect revivals.



8.3 Results

The evolution of the central peak of the quasimomentum distribution is shown for $J_x = 0\,U$ in Figure 8.1, for $J_x = 1/32\,U$ in Figure 8.2, for $J_x = 1/16\,U$ in Figure 8.3, and for $J_x = 1/8\,U$ in Figure 8.4. Comparing to perfect revivals with $J_x = J_y = 0$, the revivals are attenuated for increasing values of J_x and J_y.

For values of both J_x and J_y below $1/16\,U$, the revivals are clearly visible. All peaks are in same positions as in the case of perfect revivals. For $J_x = 1/8\,U$, only the first peak is recognizable. The phase coherence is completely lost at later evolution times.

8.4 Conclusions

I studied quantum-phase collapse and revivals in the two-dimensional Bose–Hubbard model that occur after a quantum quench from the superfluid regime to the Mott-insulator regime. The hopping parameters in the x and in the y direction were chosen independently. The results show the influence of unequal hopping on the system dynamics. I have found that the revivals are increasingly attenuated with increasing hopping strength in both directions. The calculation serves as an extension of a problem thoroughly studied in one-dimensional numerical investigations to two-dimensional systems. Additionally, collapse and revival experiments are easy to perform in optical lattices. The results can be useful

Figure 8.2: Normalized quasimomentum revivals on a two-dimensional 5×5 lattice with $N = 25$ particles, for horizontal hopping strength $J_x = 1/32\,U$, and for vertical hopping strengths $J_y = 0\,U$, $1/32\,U$, $1/16\,U$, and $1/8\,U$.

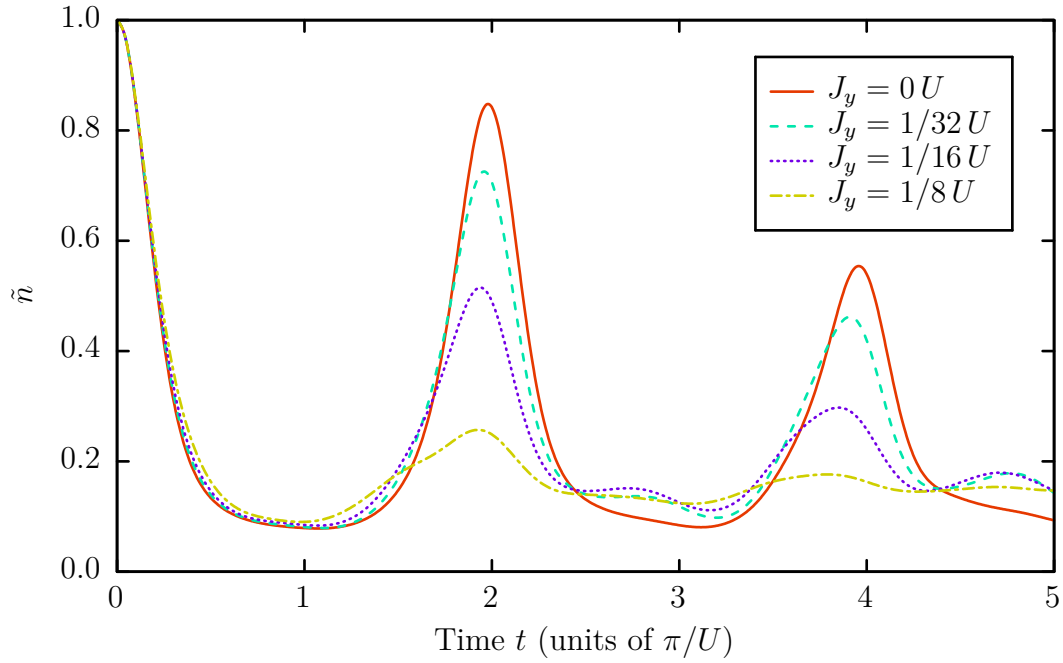


Figure 8.3: Normalized quasimomentum revivals on a two-dimensional 5×5 lattice with $N = 25$ particles, for horizontal hopping strength $J_x = 1/16\,U$, and for vertical hopping strengths $J_y = 0\,U$, $1/32\,U$, $1/16\,U$, and $1/8\,U$.

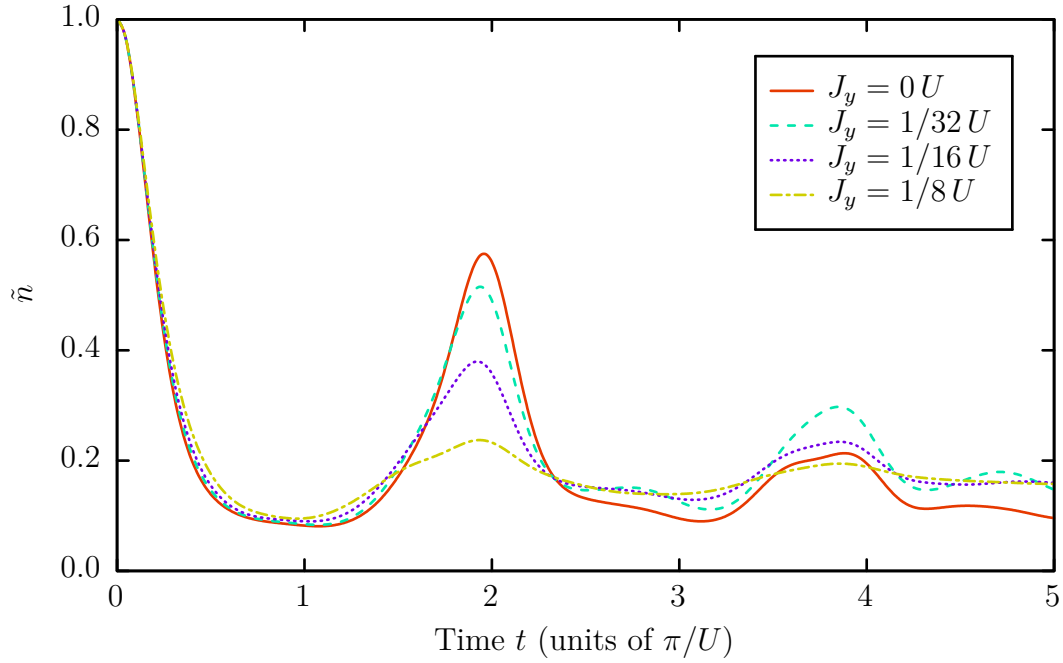
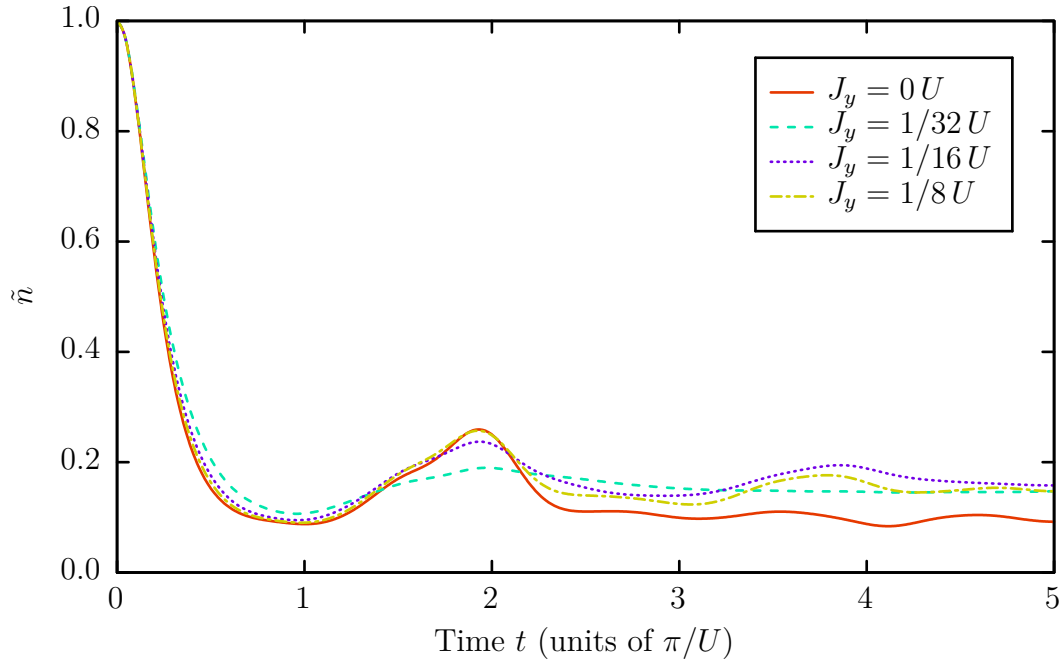


Figure 8.4: Normalized quasimomentum revivals on a two-dimensional 5×5 lattice with $N = 25$ particles, for horizontal hopping strength $J_x = 1/8\,U$, and for vertical hopping strengths $J_y = 0\,U$, $1/32\,U$, $1/16\,U$, and $1/8\,U$.



for comparing experimental and numerical data, and possibly for extracting the values of J_x/U and J_y/U from the experimental results.

A further improvement would be a model with 7×7 lattice sites and with 49 particles. Such calculation is currently achievable with TEBDOL. However, the calculation time is substantially longer than for 5×5 lattice with 25 particles.

9. Boson expansion

Boson expansion is a problem in quantum dynamics in which an initially localized cloud of ultracold atoms expands in a homogeneous optical lattice [36, 90]. The experiment starts with atoms localized in the lattice centre. There is a single atom at each occupied site with no entanglement between sites. The lattice potential is lowered, and particles start to move in the lattice. It has been observed that the velocity of cloud expansion depends on the particle interaction strength and surprisingly also on dimensionality of the system [36].

The one-dimensional problem was solved in both limiting cases, that is for noninteracting particles and for hard-core bosons [76, 91, 92]. While the behaviour for noninteracting particles in two dimensions is similar to the one-dimensional case, the dynamics of strongly interacting bosons is very different. The experiment [36] has shown that in one dimension, the expansion is ballistic, while in two dimensions the expansion is suppressed and particles tend to stay in the lattice centre. The problem therefore serves as a interesting system for studying a crossover from one to two dimensions.

The system is described by the Bose–Hubbard model. In one dimension, the expansion velocity is the largest for noninteracting particles with $U = 0$. Expansion is suppressed with increasing interaction strength, both attractive and repulsive. The minimal velocity is attained at about $U \approx 4\,J$. The expansion velocity then increases, and for large U it approaches the same value as for vanishing U [36].

The behaviour differs in two-dimensional lattices. Again, the expansion velocity is the largest for noninteracting particles. However, it decreases with growing interaction strength. For large U the velocity approaches zero. It means that the expansion of atomic cloud is severely suppressed. Atoms are localized in the initial position for very long evolution times.

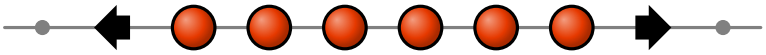
In this chapter I report on a study of boson expansion with TEBDOL [41]. The problem requires a fairly large lattice, therefore the calculation parameters were optimized for a simulation with many sites. In particular, I followed the approach from article [90] and assumed strongly interacting particles modelled by hard-core bosons. The physical dimension of hard-core bosons is $P = 2$. This value decreases the numerical complexity of the problem and leads to simulations of larger lattices and to longer evolution times.

9.1 Model

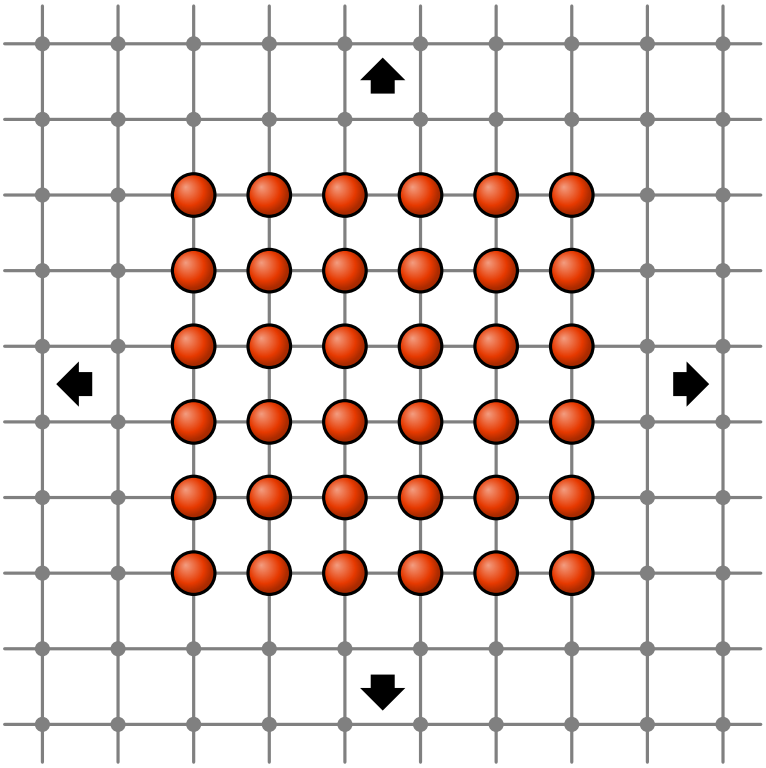
The system is described by the two-dimensional Bose–Hubbard model on a square lattice with open boundary conditions. The model assumes hard-core bosons, that is there can be only zero or one particle at a single site. Its Hamiltonian is given by

$$\begin{aligned}\hat{H} = &- J_x \sum_{i_x=1}^{L_x-1} \sum_{i_y=1}^{L_y} \left(\hat{a}^\dagger_{i_x,i_y} \hat{a}_{i_x+1,i_y} + \hat{a}^\dagger_{i_x+1,i_y} \hat{a}_{i_x,i_y} \right) \\ &- J_y \sum_{i_x=1}^{L_x} \sum_{i_y=1}^{L_y-1} \left(\hat{a}^\dagger_{i_x,i_y} \hat{a}_{i_x,i_y+1} + \hat{a}^\dagger_{i_x,i_y+1} \hat{a}_{i_x,i_y} \right),\end{aligned} \tag{9.1}$$

Figure 9.1: The initial states in the experiment of bosonic expansion are atomic clouds located in the lattice centre. The clouds are in product states with one particle per site. The grid represents the sites of an optical lattice, and the balls represent the particles in the lattice.



(a) The initial state in the one-dimensional expansion.



(b) The initial state in the two-dimensional expansion.

where L_x is the lattice width, L_y is the lattice height, $\hat{a}^\dagger_{i_x,i_y}$ is the creation operator at the site with coordinates (i_x, i_y), and $\hat{a}_{i_x,i_y}$ is the annihilation operator at (i_x, i_y). The hopping is independent in the x and y directions and given by the parameters J_x and J_y. The units are set by choosing the lattice constant $a_{\text{lat}} \equiv 1$ and the reduced Planck constant $\hbar \equiv 1$.

The experiment proceeds as follows. Initially, an atomic cloud is prepared in the centre of the lattice (Figure 9.1). The cloud is in a product state with one particle per site,

$$|\psi(t=0)\rangle = \prod_{(i_x,i_y)\in I} \hat{a}^\dagger_{i_x,i_y}|0\rangle, \tag{9.2}$$

where I is a set of indices of initially occupied sites. The lattice depth is then quickly reduced to introduce a hopping between adjacent sites. The resulting quench represent a change from $J_x = J_y = 0$ to $J_x \geq 0$ and $J_y \geq 0$. The atomic cloud then starts to expand.

The radius of the cloud in the x and in the y direction is defined by [90]

$$\begin{aligned} R_x(t) &= \sqrt{\frac{1}{N}\sum_{i_x=1}^{L_x}\sum_{i_y=1}^{L_y}\langle\hat{n}_{i_x,i_y}(t)\rangle(i_x - i_x^0)^2}, \\ R_y(t) &= \sqrt{\frac{1}{N}\sum_{i_x=1}^{L_x}\sum_{i_y=1}^{L_y}\langle\hat{n}_{i_x,i_y}(t)\rangle(i_y - i_y^0)^2}, \end{aligned} \tag{9.3}$$

where N is the total number of particles, $\langle\hat{n}_{i_x,i_y}(t)\rangle$ is the expectation value of the particle density at the site with coordinates (i_x, i_y), i_x^0 is the centre of mass in the x direction, and i_y^0 is the centre of mass in the y direction. To better extract the velocity, the initial constant part is subtracted from the squared radius to obtain

$$\begin{aligned} \tilde{R}_x(t) &= \sqrt{R_x^2(t) - R_x^2(0)}, \\ \tilde{R}_y(t) &= \sqrt{R_y^2(t) - R_y^2(0)}. \end{aligned} \tag{9.4}$$

The radial expansion velocities in the x and in the y direction are then defined by

$$\begin{aligned} v_x &= \frac{\partial\tilde{R}_x(t)}{\partial t}, \\ v_y &= \frac{\partial\tilde{R}_x(t)}{\partial t}. \end{aligned} \tag{9.5}$$

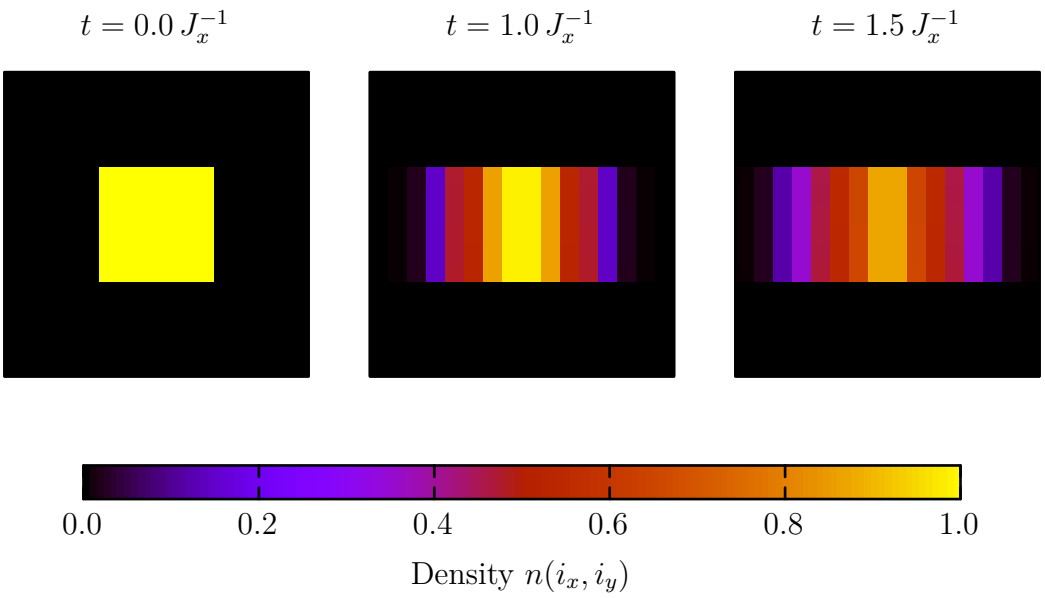
9.2 Calculation

The model was simulated on a 16×16 lattice with an initial block of 6×6 particles. The calculation was performed for a set of values of J_y in the interval $0.0\,J_x \leq J_y \leq 1.0\,J_x$. The case with $J_y = 0.0\,J_x$ is effectively a one-dimensional model. The calculation parameters were chosen as follows:

- Physical dimension was set to $P = 2$. This value corresponds to hard-core bosons with zero or one particle at a single site.
- The bond dimension was scaled dynamically with parameters $\varepsilon = 2^{32} \times \varepsilon_m \approx 4.77 \times 10^{-7}$ and $D = 512$, where $\varepsilon_m \approx 1.11 \times 10^{-16}$ is the double-precision machine epsilon in the IEEE 754 standard. The smallest singular values, whose sum was relatively smaller than ε, were truncated after each tensor decomposition. When there remained more than D states, TEBDOL further truncated the smallest values, so there were D singular values kept at most.
- The time step in the second-order Suzuki–Trotter approximation was set to $\delta t = 1/16\,J_x^{-1}$.
- The simulated time interval was $T = 1.5\,J_x^{-1}$. Even though it was possible to reach longer times, the accuracy of the results quickly decreased due to the increased entanglement in the system and larger truncations of the tensor network.

The accuracy of the results was measured by the TTNS norm. The norm at $t = 1.5\,J_x^{-1}$ was $\|\psi\| \approx 0.82$ for the most difficult case with $J_y = 1.0\,J_x$. The norm decreased rapidly for longer times, for example $\|\psi\| \approx 0.45$ for $t = 2.0\,J_x^{-1}$ and $\|\psi\| \approx 0.20$ for $t = 2.5\,J_x^{-1}$.

Figure 9.2: Expansion of a 6×6 atomic cloud on a 16×16 lattice for $J_y = 0.0\, J_x$. The figure shows the particle density at three distinct times t. There is no hopping in the y direction, therefore this case corresponds to the one-dimensional dynamics.



9.3 Results

The numerical results for one-dimensional models were presented in [36]. In [90], the numerical study was extended to small two-dimensional systems with initial blocks with sizes 2×2, 3×3, and 4×4 in a 12×12 lattice. Furthermore, the dynamics on cylinders and ladders was investigated in the latter article as well.

The results of a simulation with TEBDOL for a 6×6 block in a 16×16 lattice are presented here. The evolution of particle density is shown for $J_y = 0.0\, J_x$ in Figure 9.2, for $J_y = 0.5\, J_x$ in Figure 9.3, and for $J_y = 1.0\, J_x$ in Figure 9.4. The figures show particle density at three distinct evolution times $t = 0.0\, J_x^{-1}$, $1.0\, J_x^{-1}$, and $1.5\, J_x^{-1}$.

The results confirm that a 16×16 lattice is large enough for a simulation up to $t = 1.5\, J_x^{-1}$. The expanded cloud does not reach the lattice edges during this time interval. The model with $J_y = 0.0\, J_x$ is effectively one-dimensional. Its dynamics therefore does not depend on the y coordinate. The case with $J_y = 0.5\, J_x$ can represent a crossover from the one-dimensional dynamics to the two-dimensional dynamics. The last figure with $J_y = 1.0\, J_x$ represents a full two-dimensional model with equal hopping in both directions.

The dependence of the expansion velocities v_x and v_y on the hopping strength J_y is shown in Figure 9.5. The velocity in the one-dimensional model, obtained from an analytic solution, is given by $v_x = \sqrt{2}\, J_x$ [36]. The expansion in the x direction is suppressed with increasing J_y. The exact values of the velocities are time dependent. The results therefore change slightly if a different time interval is used for the linear velocity fit. I followed the article [90] and used the interval

Figure 9.3: Expansion of a 6×6 atomic cloud on a 16×16 lattice for $J_y = 0.5\, J_x$. The figure shows the particle density at three distinct times t. This case corresponds to a crossover from the one-dimensional to the two-dimensional dynamics.

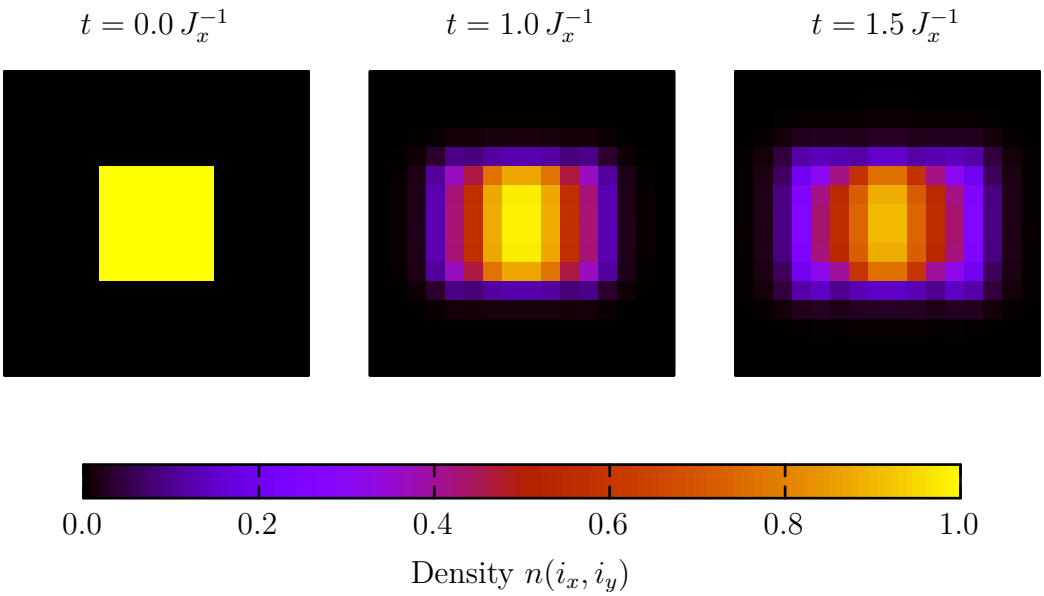


Figure 9.4: Expansion of a 6×6 atomic cloud on a 16×16 lattice for $J_y = 1.0\, J_x$. The figure shows the particle density at three distinct times t. This case corresponds to the isotropic two-dimensional dynamics.

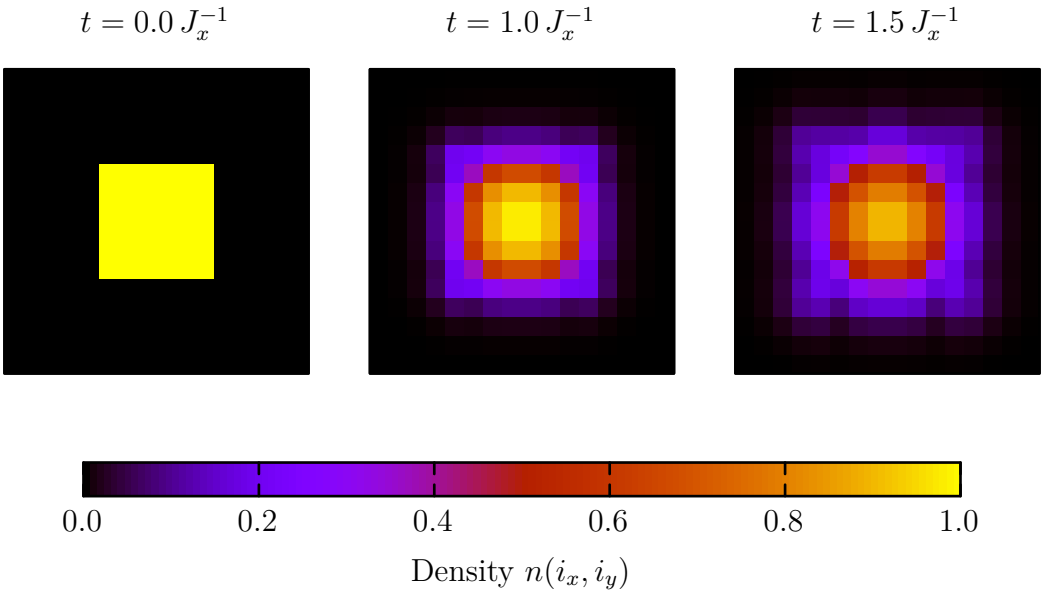
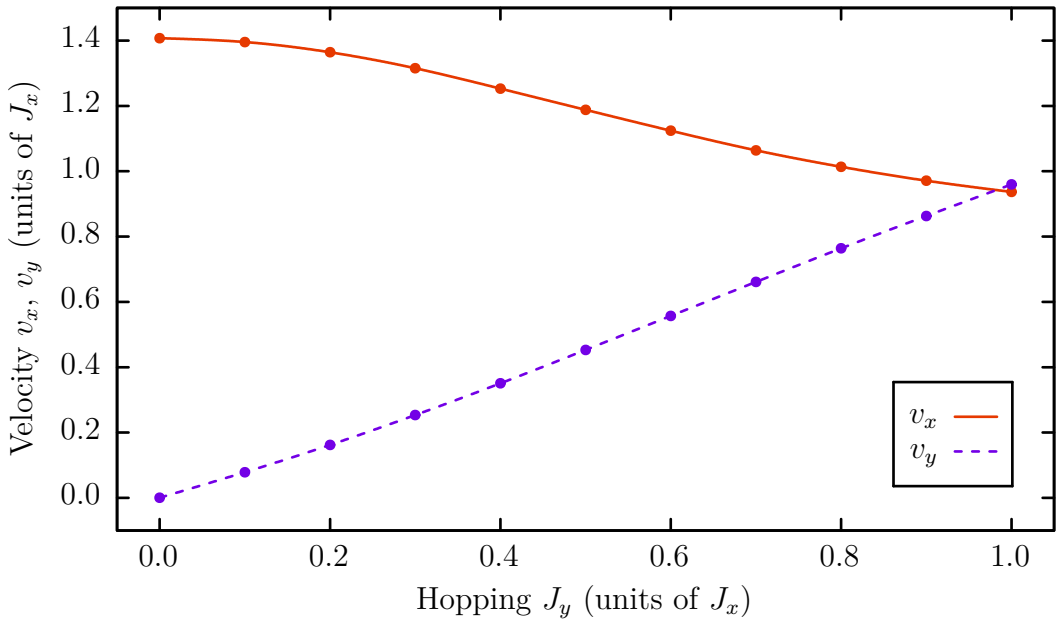


Figure 9.5: The dependence of the expansion velocities v_x and v_y on the hopping strength J_y. The case with vanishing J_y corresponds to the one-dimensional dynamics, where the analytic solution leads to the velocity $v_x = \sqrt{2}\, J_x$ [36].



$1.0 J_x^{-1} \leq t \leq 1.5 J_x^{-1}$ for the fit. The results can be therefore directly compared to the results published previously [90]. The expansion is more suppressed for a 6×6 block than for smaller blocks.

9.4 Conclusions

In this chapter I investigated the expansion of a bosonic atomic cloud in a two-dimensional optical lattice. The model assumed a large on-site interaction strength with particles represented by hard-core bosons. I studied the dependence of the cloud expansion velocity in the x and in the y direction on the hopping strength in the y direction. The results describe a continuous crossover from the one-dimensional to the two-dimensional dynamics for small J_x and J_y.

I have found that the expansion velocity was suppressed in the x direction and enhanced in the y direction with increasing hopping strength J_y. The final velocity in both directions for $J_x = J_y$ was less than 70% of the velocity in the x direction for vanishing J_y. The results confirm that the suppression takes place in the two-dimensional Bose–Hubbard model with hard-core bosons.

This effect was already observed in a calculation with a block of 4×4 particles on a 12×12 lattice [90]. I improved the calculation to a 6×6 block on a 16×16 lattice. A further improvement would be to model a block of 7×7 or 8×8 particles. It is expected that the expansion velocity would be suppressed even more in these cases, especially considering that the number of particles in the cloud bulk would be significantly larger than the number of particles at its boundary.

10. Many-body localization

Interacting physical systems tend to thermalize. They usually reach thermal equilibrium after evolving for a period of time. The properties of a system in thermal equilibrium can be described by a few physical parameters like its volume and temperature. However, not all physical systems thermalize. For example, integrable systems represent one category of systems that do not exhibit this behaviour [93].

Recently, it has been observed that some interacting systems with random disorder potential do not thermalize. This effect is similar to Anderson localization [94], but it happens in many-body systems. The term *many-body localization* (MBL) encompasses all phenomena in which a system does not thermalize due to interactions between its particles [95, 96].

Advanced experimental control of ultracold atoms has led to investigation of MBL in optical lattices [40]. They have shown that MBL indeed takes places here. For example, authors of the experiment [40] observed MBL in a two-dimensional optical lattice. They found a critical strength of the disorder potential that separates the system behaviour into thermalization and localization. The limited applicability of analytic and numerical methods to two-dimensional systems makes it hard to corroborate their findings. Additionally, the precise range of parameters that lead to localization is largely unknown.

In this section I present the results of investigation of MBL numerically using TEBDOL [41]. I have tried to reproduce results from the experiment [40], so the setup closely follows the experimental setup where applicable.

10.1 Experimental setup

The authors prepared a two-dimensional optical lattice with a lattice constant $a_{\mathrm{lat}} = 532\,\mathrm{nm}$. The harmonic potential constrained the lattice size to about 31×31 sites. They created a Mott insulator of bosonic ^{87}Rb atoms in the shape of a circle located in the centre of the trap with approximately one particle per occupied site. Right half of the atoms located at $x > 0$ was then removed using a light modulator based on a digital-mirror device (DMD). The initial state was therefore a product state of about $N = 125$ atoms localized in the left half of the lattice.

Next, the authors projected a random disorder potential onto the lattice. A new potential was generated for each of about 50 experimental realizations. Optical properties of the experiment caused that the disorder strength was not statistically independent at different sites. The authors measured a finite disorder correlation length of $0.6\,a_{\mathrm{lat}}$. Additionally, the measured disorder distribution was not symmetric with respect to the distribution mean.

The initial domain wall was then lowered by decreasing the lattice depth from $40\,E_{\mathrm{r}}$ to $12\,E_{\mathrm{r}}$, where $E_{\mathrm{r}} = h^2/8ma_{\mathrm{lat}}^2$ is the recoil energy, h is the Planck constant, and m is the atomic mass of ^{87}Rb. The system evolved for a time $t = 187\,\tau$, where $\tau = \hbar/J$ is the tunnelling time and J is the nearest-neighbour hopping strength in the Bose–Hubbard model. Afterwards, a single-site–resolved local density was measured. The measurement was parity-projected, so there was no difference between zero and two particles in the density measurement. However,

the authors expected that the occupation numbers of three or more particles are negligible. They estimated the number of doubly-occupied sites to about 2% for weak disorder and to about 9% for strong disorder.

The authors then analyzed the asymmetry in the x direction of the measured particle density. They found that for small disorder, the system evolved into a symmetric state that corresponds to thermal equilibrium. For large disorder, the particle density stayed asymmetric even for large evolution time t. Using simple double-linear fitting, the authors were able to determine the critical disorder that separates the two outcomes. The system thermalizes for smaller disorder strengths and stays localized for larger disorder strengths.

10.2 Model

The system of atoms in an optical lattice corresponding to the experiment [40] is described by a Hamiltonian

$$\hat{H} = -J \sum_{\langle \mathbf{i},\mathbf{j} \rangle} \hat{a}_{\mathbf{i}}^{\dagger} \hat{a}_{\mathbf{j}} + \frac{U}{2} \sum_{\mathbf{i}} \hat{n}_{\mathbf{i}} \left(\hat{n}_{\mathbf{i}} - 1 \right) + \sum_{\mathbf{i}} \left(V_{\mathbf{i}} + \delta_{\mathbf{i}} \right) \hat{n}_{\mathbf{i}}, \tag{10.1}$$

where $\hat{a}_{\mathbf{i}}^{\dagger}$ is the bosonic creation operator, $\hat{a}_{\mathbf{i}}$ is the bosonic annihilation operator, $\hat{n}_{\mathbf{i}} = \hat{a}_{\mathbf{i}}^{\dagger} \hat{a}_{\mathbf{i}}$ is the particle number operator, J is the hopping parameter, U is the on-site interaction parameter, $V_{\mathbf{i}}$ is the harmonic trapping potential, and $\delta_{\mathbf{i}}$ is the random disorder potential. The operators and the potentials are indexed by the lattice site $\mathbf{i} = (i_x, i_y)$. The sites form a two-dimensional rectangular lattice with L_x sites in the x direction and L_y sites in the y direction. There are nearest-neighbour interactions only, so the angle brackets denote each pair of adjacent sites. Each site in the bulk of the lattice has four adjacent sites, and sites at the lattice boundary have three or two adjacent sites due to the open boundary conditions.

The lattice size with $L_x = L_y = 31$ and $N = 125$ particles is way too large for numerical investigation. I therefore investigated smaller lattices with fewer particles, but kept other parameters as close to the experiment as possible.

10.3 Trapping potential

The harmonic potential in the experiment [40] is given by

$$V_{\mathbf{i}} = \frac{m a_{\mathrm{lat}}}{2} \left(\omega_x^2 i_x^2 + \omega_y^2 i_y^2 \right), \tag{10.2}$$

where $\omega_x = 2\pi \times 54\,\mathrm{Hz}$ and $\omega_y = 2\pi \times 60\,\mathrm{Hz}$. The origin of the coordinate system is in the lattice centre. The strength of the potential determines the average energy per particle $E_A = E_0/N$, where E_0 is the energy of the initial state. E_A encompasses both thermal and potential energy. The authors measured that $E_A = 0.28(3)\,U$ in the experiment.

Because I modelled a smaller lattice with fewer particles, the original ω_x and ω_y were too small to keep the atoms confined in the trap. On the other hand, the average energy per particle serves as a good measure of the potential strength. I

therefore fixed E_A for the initial distribution of particles and calculated ω_x and ω_y from E_A. Additionally, the same potential strength was used in both directions, that is $\omega = \omega_x = \omega_y$. In contrast to the experiment, I did not consider any thermal contribution to the energy of the initial state.

The particles were initially in a product state. They formed an approximate half-circle. There was either no particle at a site or a single particle at a site. I first calculated the dimensionless quantity

$$\widetilde{E}_0 = \sum_{i_x,i_y} n_{i_x,i_y} \left(i_x^2 + i_y^2\right), \tag{10.3}$$

where n_{i_x,i_y} is the number of particles at a site with coordinates (i_x, i_y). The harmonic potential strength for a fixed E_A was then given by

$$\frac{m a_{\mathrm{lat}} \omega^2}{2} = \frac{N E_A}{\widetilde{E}_0}. \tag{10.4}$$

I used two values of E_A in the calculations, the original value from the experiment, $E_A = 0.28\,U$, and one half of it, $E_A = 0.14\,U$. Due to the open boundary conditions, the particles were also confined by the infinite potential walls represented by the lattice edges.

10.4 Disorder potential

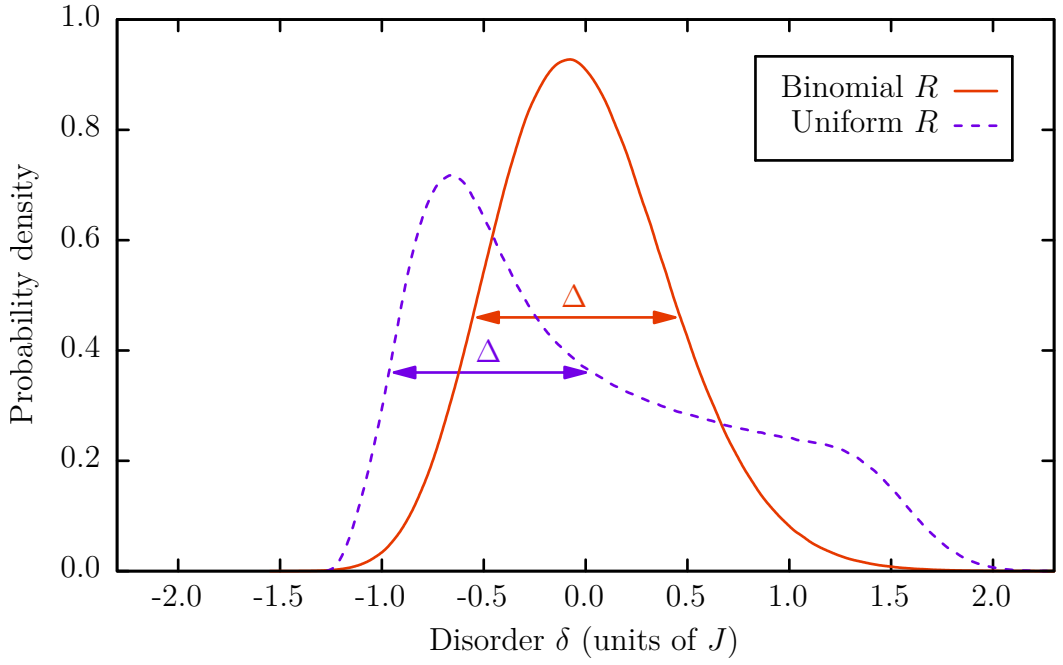
The random disorder potential in the MBL experiment [97] was created by a DMD, which converted a Gaussian laser intensity profile into a two-dimensional random intensity distribution. The authors observed correlations of the intensities at different sites. They presented a method to obtain a comparable random intensity distribution numerically [97]. First, they squared every element of a two-dimensional array of random numbers between zero and one, and then convolved the array with a normalized Gaussian. The distribution the original random numbers were taken from was not specified. I denote this distribution by R.

The authors noted that in their numerical simulation of a noninteracting system the observables depended only slightly on the exact distribution of the potential when they fixed the spatial correlations of the pattern. Additionally, they found little difference when replacing the distribution with a symmetric Gaussian. On the other hand, I wanted to create a random disorder potential as close to the potential realized in the experiment as possible. This way all possible influences of the potential correlations and the potential distribution on the dynamics of the system could be modelled. I first tried to reproduce the measured distribution by using the uniform distribution as R, but this unfortunately did not yield the expected results. I therefore analyzed the experimental setup in more detail.

The authors created the disorder potential by a DMD with 1024×768 binary pixels. They focused about 7×7 pixels on a single site. If a pixel in the DMD has equal probability of being on and off, the sum of values of 7×7 pixels corresponds to the binomial distribution

$$f(k; n, p) = \binom{n}{k} p^k (1-p)^{(n-k)}, \tag{10.5}$$

Figure 10.1: Distribution of the random disorder potential generated by the algorithm described in the text. The final distribution is obtained for two initial distributions R, the binomial distribution and the uniform distribution. When R is the binomial distribution, the final distribution models the measured disorder better. Both distributions are rescaled so that their full-width at half-maximum Δ equals one.



with $n = 49$ and $p = 1/2$. The range of the binomial distribution is a set $\{0, \ldots, n\}$, and the probability of obtaining a sample k is $f(k; n, p)$. I rescaled the samples by dividing them by n. The resulting numbers are therefore in the expected range between zero and one. Using the binomial distribution as R gives the final disorder distribution that better models distribution presented in the article [40]. Figure 10.1 compares the final distributions when R is the uniform distribution and when R is the binomial distribution. The exact algorithm used to create the random disorder potential is as follows:

1. Create a two-dimensional array A with dimensions $L_x + L_g - 1$ and $L_y + L_g - 1$, where L_g is the size of discrete Gaussian kernel. I used $L_g = 9$.

2. Assign each element of A a random number $(s/49)^2$, where s is a random sample taken from the binomial distribution $f(k; 49, 1/2)$.

3. Convolve A with a normalized two-dimensional Gaussian kernel of size $L_g \times L_g$ with a standard deviation of $0.5\, a_{\mathrm{lat}}$ to create an array B with the disorder potential [97]. The dimensions of B are L_x and L_y.

10.5 Numerical setup

The lattice size of 31×31 sites with 125 particles is too large for numerical calculations. I therefore decided to investigate MBL in a smaller system. The simulated lattice size was 6×6 sites with 6 particles. Naturally, the amount of entanglement between particles created during the time evolution in this system is limited. However, the dynamics is similar to the original system.

The initial state is a product state

$$|\psi(t=0)\rangle = \prod_{\mathbf{i}\in I} a_{\mathbf{i}}^{\dagger}|0\rangle, \tag{10.6}$$

where $I = \{(2,3),(2,4),(3,2),(3,3),(3,4),(3,5)\}$ is a set of initially filled site coordinates, and the coordinate $(1,1)$ corresponds to the site in the bottom left corner of the lattice. There is a single particle at each site in I. Figure 10.2a shows the distribution of particles in the initial state.

The time evolution is governed by the Hamiltonian (10.1). The energy scale is determined by J, and the hopping parameter is $U = 24.4\,J$. The harmonic potential $V_{\mathbf{i}}$ is specified by the energy per particle E_A as described earlier. The disorder distribution is rescaled so that its full-width at half-maximum (FWHM) is Δ. For $\Delta = 0\,J$ there is no disorder and the dynamics is constrained by the trapping potential only. In accordance with the experiment, Δ was varied from $\Delta = 0\,J$ to $\Delta = 18\,J$. The expected behaviour is that for small Δ the system thermalizes, atoms delocalize in the trap, and the final state is spatially symmetric (Figure 10.2b). For large Δ, the evolution is restricted, and there are still traces of the original asymmetry even after long evolution times (Figure 10.2c).

There are several measures of the density asymmetry. To follow the experiment, I examined the density imbalance I and the density deviation δn. Density imbalance is a normalized difference between the total density in the left and the right part of the lattice. Density deviation is a difference between the density distribution and a thermalized density distribution.

To define the density imbalance, the particle density in the left part of the lattice, N_L, and the particle density in the right part of the lattice, N_R, are first defined by

$$\begin{aligned} N_L &= \sum_{\mathbf{i}\in L} \langle\psi(t)|\hat{n}_{\mathbf{i}}|\psi(t)\rangle, \\ N_R &= \sum_{\mathbf{i}\in R} \langle\psi(t)|\hat{n}_{\mathbf{i}}|\psi(t)\rangle, \end{aligned} \tag{10.7}$$

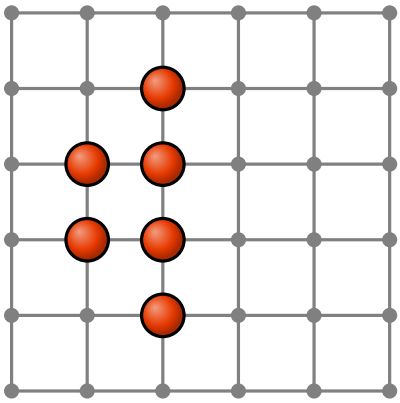
where L and R are the sets of site coordinates for the left part and for the right part of the lattice, respectively. The density imbalance is then given by [40]

$$I(L,R) = \frac{N_L - N_R}{N_R + N_R}. \tag{10.8}$$

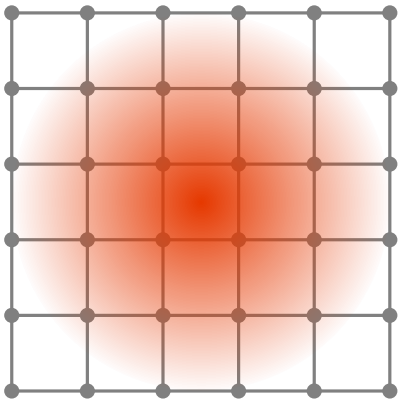
The set of sites used for calculating the imbalance was restricted in the experiment, where the authors took into account only a central five-sites-wide strip in the y direction. I take into account a two-site-wide strip,

$$\begin{aligned} L_2 &= \{(x,y) \mid x = \{1,2,3\},\ y = \{3,4\}\}, \\ R_2 &= \{(x,y) \mid x = \{4,5,6\},\ y = \{3,4\}\}, \end{aligned} \tag{10.9}$$

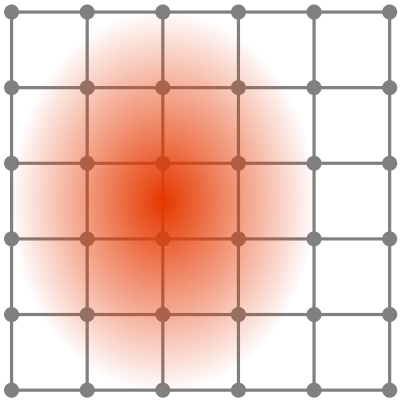
Figure 10.2: An outline of the MBL calculation in a 6×6 lattice with $N = 6$ particles. The lattice potential is a sum of a harmonic trapping potential and a random disorder potential.



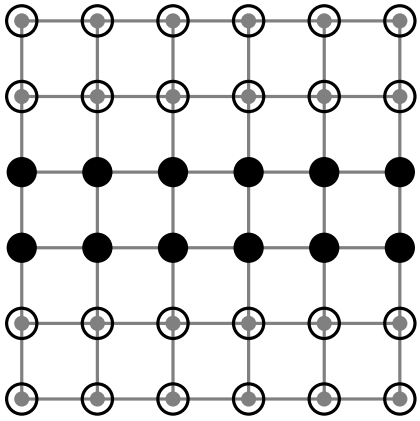
(a) The initial state is a product state of 6 particles in a shape of an approximate half-circle. There is one particle at each occupied site.



(b) If the particle density distribution evolves into a circularly symmetric shape, it is assumed that the system has thermalized.



(c) If the particle density distribution shows traces of asymmetry even for large evolution times, it is assumed that the system has not thermalized.



(d) Asymmetry measures are calculated either by taking into account all sites in the y direction or just the central two-site-wide strip.

and define the density $I_2 = I(L_2, R_2)$. I also analyze the asymmetry calculated from a full width of six sites,

$$\begin{aligned} L_6 &= \{(x,y) \mid x = \{1,2,3\},\ y = \{1,\ldots,6\}\}\,, \\ R_6 &= \{(x,y) \mid x = \{4,5,6\},\ y = \{1,\ldots,6\}\}\,, \end{aligned} \tag{10.10}$$

and define the imbalance $I_6 = I(L_6, R_6)$. The two sets of sites are shown in Figure 10.2d).

The density deviation is given by [40]

$$\delta n(Y) = \sqrt{\sum_{i_x} \left[n_{i_x}(0;Y) - n_{i_x}(\Delta;Y)\right]^2}, \tag{10.11}$$

where

$$n_{i_x}(\Delta;Y) = \frac{1}{|Y|} \sum_{i_y \in Y} n_{i_x,i_y}(\Delta) \tag{10.12}$$

is the averaged density profile. Here, n_{i_x,i_y} is the particle density at the site with coordinates (i_x, i_y), and $|Y|$ is the number of elements in a set Y. Again, I present density deviation $\delta n_2 = \delta n(Y_2)$ obtained from a central strip with the width of two sites in the y direction, corresponding to $Y_2 = \{3,4\}$, and $\delta n_6 = \delta n(Y_6)$ obtained from full width of six sites in the y direction, corresponding to $Y_6 = \{1,\ldots,6\}$.

The system was evolved for a time interval $t = 187\,\tau$. The final asymmetry depends on the disorder strength Δ and also on the particular realization of the disorder potential. The calculation was repeated for $M = 100$ different potential realizations for each parameter set, and the results were statistically averaged. Each measured observable X corresponds to an averaged observable

$$\bar{X} = \frac{1}{M} \sum_{i=1}^{M} X_i, \tag{10.13}$$

where X_i are individual samples. The standard error of its mean is

$$\sigma_{\bar{X}} = \frac{\sigma_X}{\sqrt{M}}, \tag{10.14}$$

where

$$\sigma_X = \sqrt{\frac{1}{M-1} \sum_{i=1}^{M} \left(X_i - \bar{X}_i\right)^2} \tag{10.15}$$

is the sample standard deviation. σ_X is nonzero generally, but $\sigma_{\bar{X}}$ becomes smaller with the increasing number of samples.

10.6 Calculation

For the simulation of MBL I used the TTNS algorithm described in the previous chapters with the following parameters:

- The performance of TEBDOL does not depend strongly on the size of the physical dimension. As there are $N = 6$ particles, I set the physical dimension to $P = N + 1 = 7$. All states in the Hilbert space of the model are therefore accessible during the time evolution.

- The bond dimension was scaled dynamically with parameters $\varepsilon = 2_m^{33} \approx 9.54 \times 10^{-7}$ and $D = 512$, where $\varepsilon_m \approx 1.11 \times 10^{-16}$ is the double-precision machine epsilon in the IEEE 754 standard. The smallest singular values, whose sum is relatively smaller than ε are truncated after each tensor decomposition. If there remain more than D states, TEBDOL further truncates the smallest values, so there are D values kept at most.
- The time step in the second-order Suzuki–Trotter decomposition was $\delta t = 1/16\,\tau$.
- Full simulated time interval was $T = 256\,\tau$.

The above parameters ensured that a calculation with a single potential realization finished in reasonable time and also that the results had reasonable accuracy. The accuracy of the results was measured by the TTNS norm. At $t = 187\,\tau$, the lowest norm was $\|\psi\| \approx 0.65$.

10.7 Results

The first calculation was performed for the average potential energy $E_A = 0.28\,U$. This is the same energy as in the experiment [40]. The experimental value of E_A includes both potential and thermal energy, and additional energy was present due to the heating of the atomic cloud. In contrast, there is no thermal contribution or heating effect present in my calculation.

Both imbalance I and density deviation δn were calculated. The results were obtained from a two-site-wide strip and from a six-site-wide strip. Each data point is an average over 100 random potential realizations. The corresponding standard errors of the mean for the average quantities are present in the following figures as well.

The first set of figures shows the results for $E_A = 0.28\,U$. The imbalances $\bar{I}_2$ and $\bar{I}_6$ are shown in Figure 10.3 and Figure 10.4, and density deviations $\bar{\delta n}_2$ and $\bar{\delta n}_6$ are shown in Figure 10.5 and Figure 10.6, respectively.

The results for $E_A = 0.28\,U$ do not show any transition from a thermalized to a localized phase. Both imbalance and density deviation grow with increasing Δ. Further investigation showed that the trapping potential was too strong for a 6×6 lattice. The dynamics was mostly confined to its central 4×4 region. I therefore decided to decrease the strength of the initial average energy to $E_A = 0.14\,U$, that is to decrease the strength of the harmonic potential. The following figures show the results for this case.

The calculated average imbalances $\bar{I}_2$ and $\bar{I}_6$ for $E_A = 0.14\,U$ are shown in Figure 10.7 and Figure 10.8. Both figures exhibit a clear distinction between a thermalized and a localized phase. The corresponding average density deviations $\bar{\delta n}_2$ and $\bar{\delta n}_6$ are shown in Figure 10.9 and Figure 10.10.

The results for $E_A = 0.14\,U$ support the hypothesis that there is a critical strength of the disorder potential that changes the dynamics of the system from thermalization to localization. Both imbalance figures show that for weak disorder the systems thermalize. The critical disorder strength was obtained from a simple double-linear fit [40]

$$\bar{I}(\Delta) = A + B \times \max\left[(\Delta - \Delta_{c,\bar{I}}), 0\right], \tag{10.16}$$

Figure 10.3: Dependence of the average imbalance on the disorder strength in a central strip with the width of two sites for $E_A = 0.28\,U$.

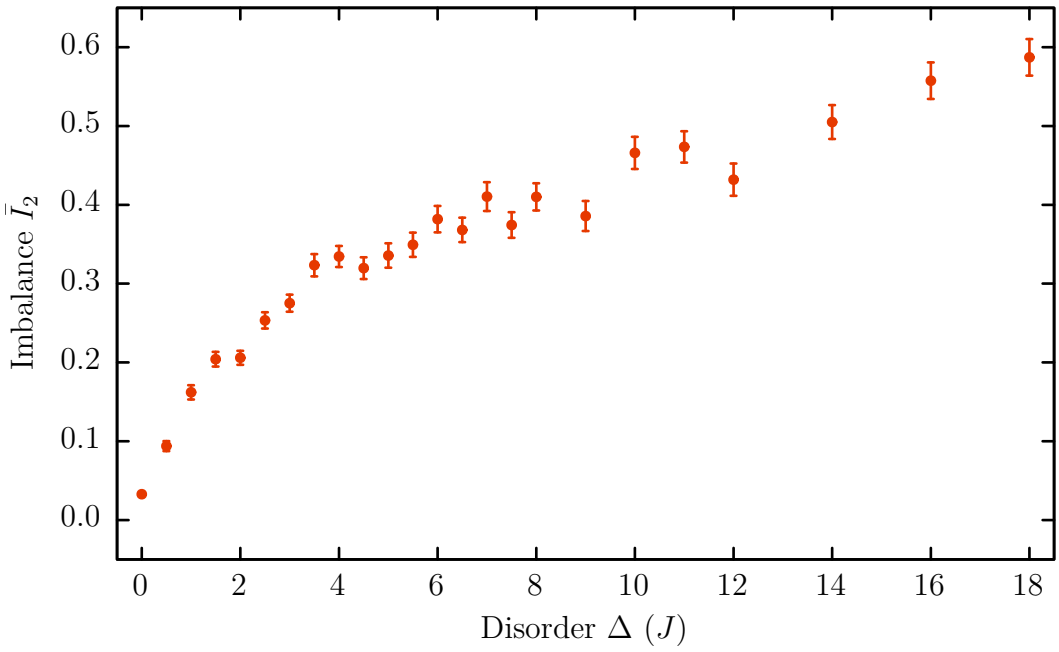


Figure 10.4: Dependence of the average imbalance on the disorder strength in a full strip with the width of six sites for $E_A = 0.28\,U$.

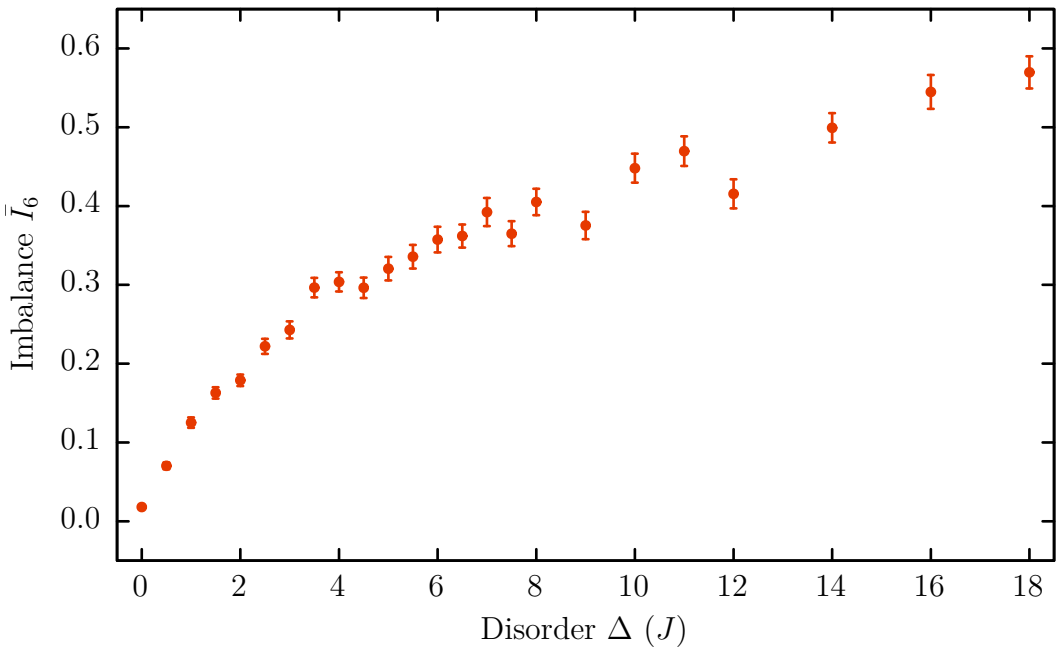


Figure 10.5: Dependence of the average density deviation on the disorder strength in a central strip with the width of two sites for $E_A = 0.28\,U$.

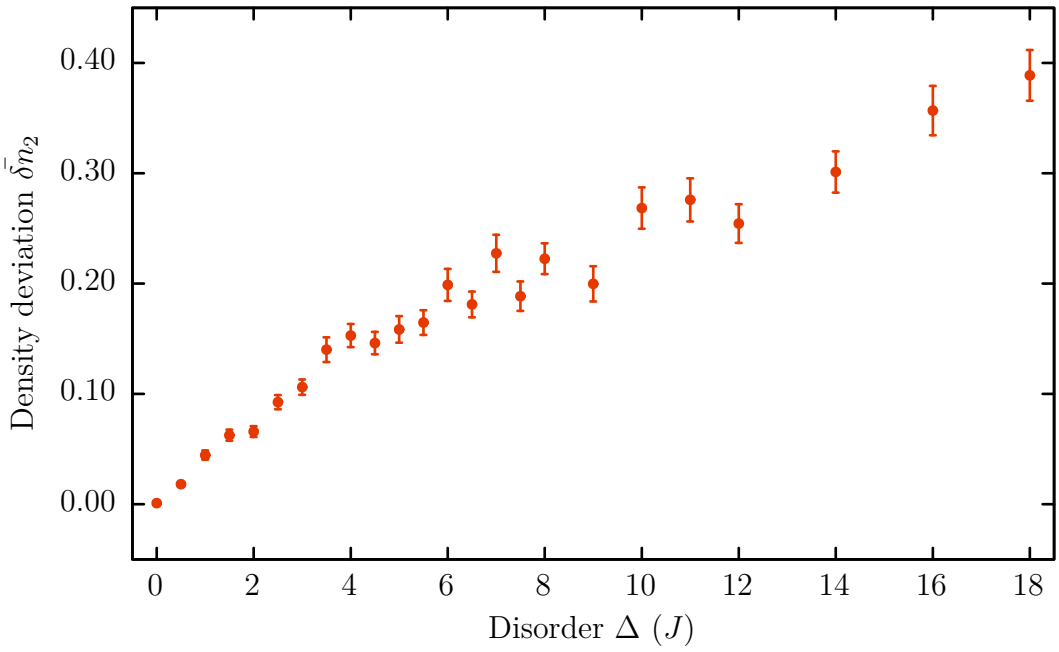


Figure 10.6: Dependence of the average density deviation on the disorder strength in a full strip with the width of six sites for $E_A = 0.28\,U$.

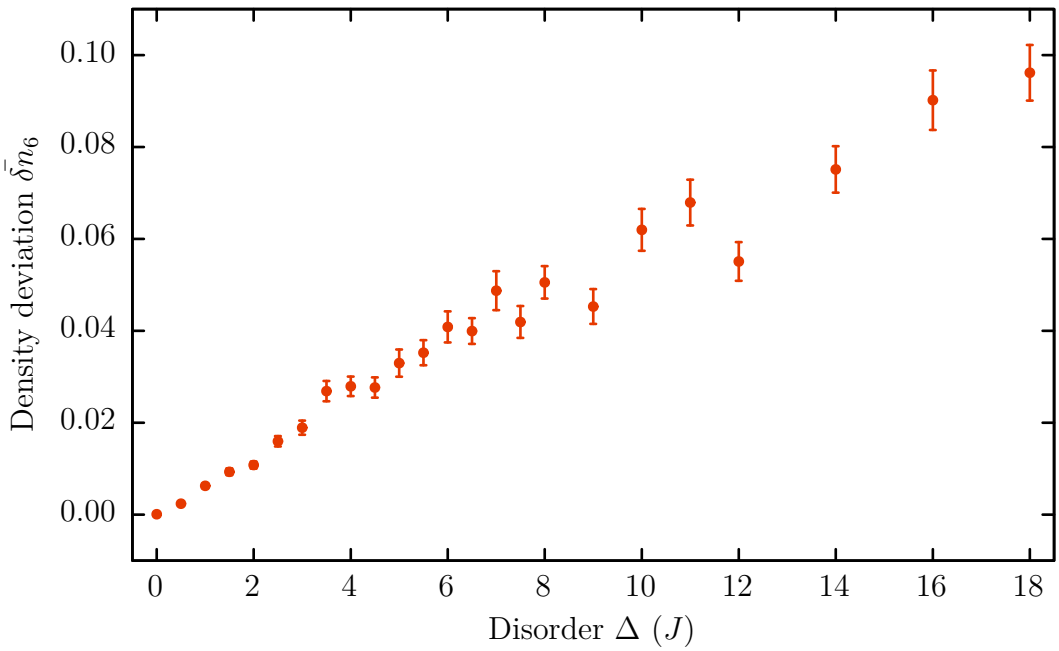


Figure 10.7: Average imbalance calculated from a strip of two sites for $E_A = 0.14\,U$. The critical disorder strength obtained from a double-linear fit is $\Delta = 4.3(3)\,J$ with $A = 0.030(7)$ and $B = 0.040(2)$.

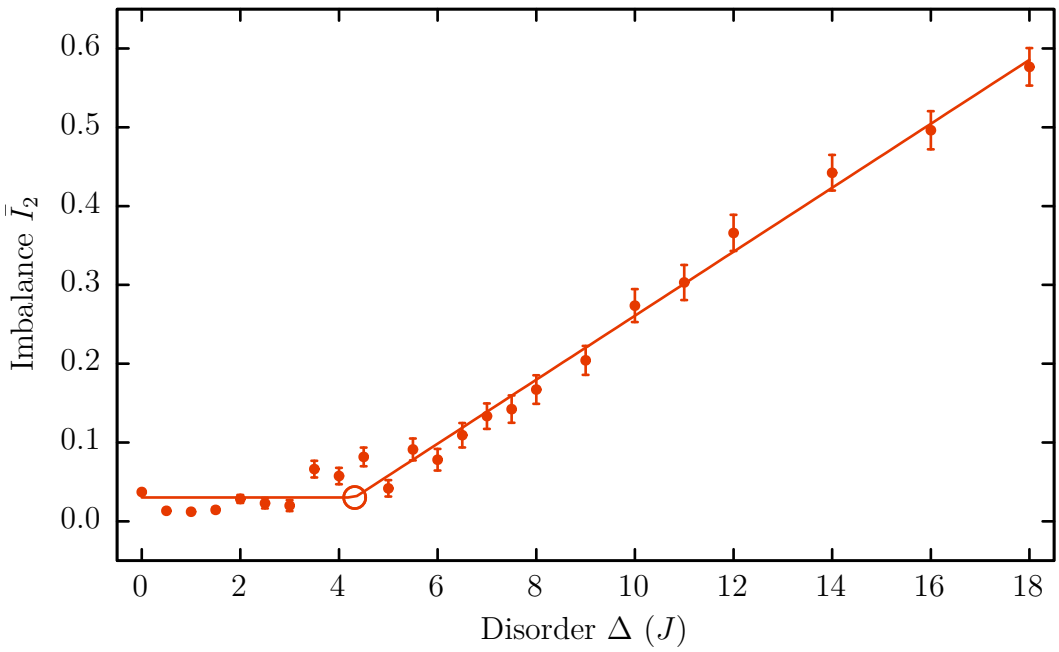


Figure 10.8: Average imbalance calculated from a strip of six sites for $E_A = 0.14\,U$. The critical disorder strength obtained from a double-linear fit is $\Delta = 3.9(3)\,J$ with $A = 0.025(7)$ and $B = 0.036(2)$.

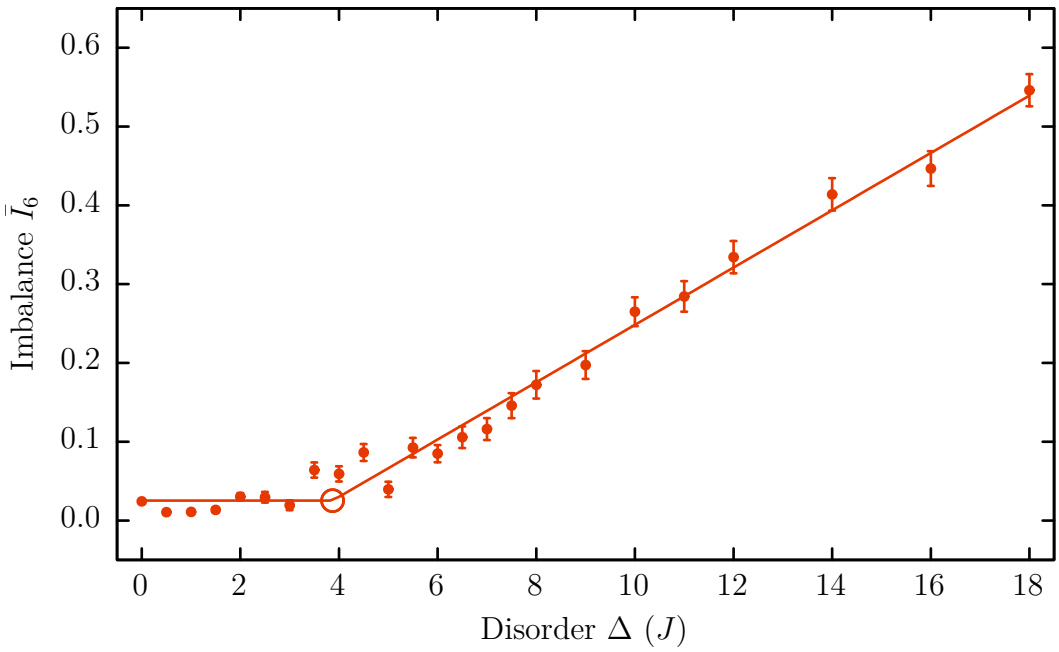


Figure 10.9: Average density deviation calculated from a strip of two sites for $E_A = 0.14\,U$. The critical disorder strength obtained from a double-linear fit is $\Delta = 2.9(2)\,J$ with $C = 0.005(3)$ and $D = 0.0230(4)$.

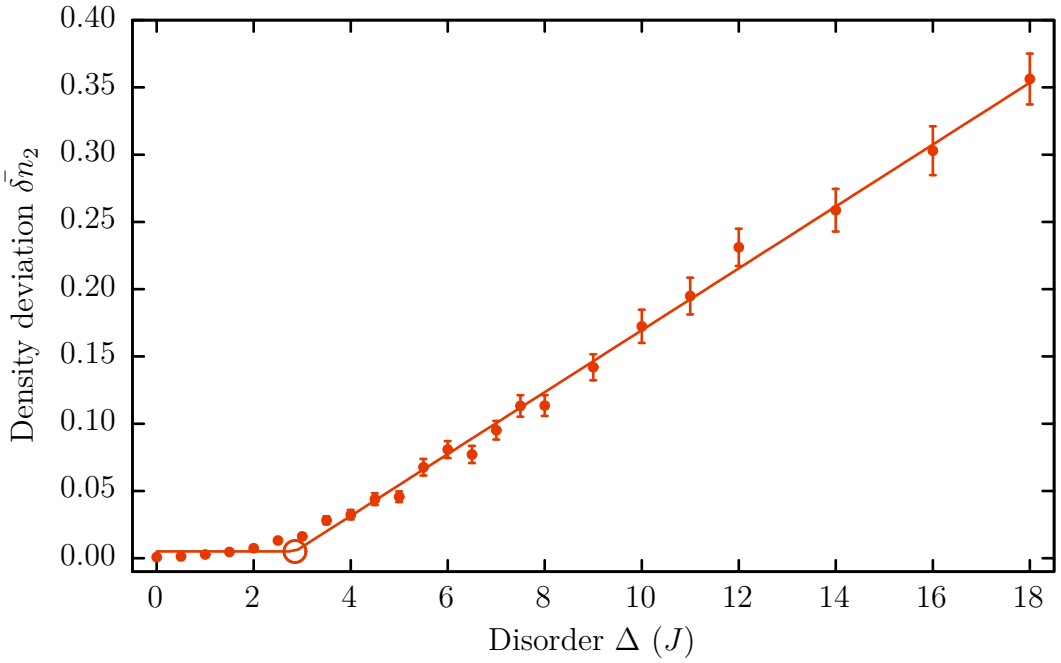


Figure 10.10: Average density deviation calculated from a strip of six sites for $E_A = 0.14\,U$. The critical disorder strength obtained from a double-linear fit is $\Delta = 4.4(2)\,J$ with $C = 0.0021(9)$ and $D = 0.0058(2)$.

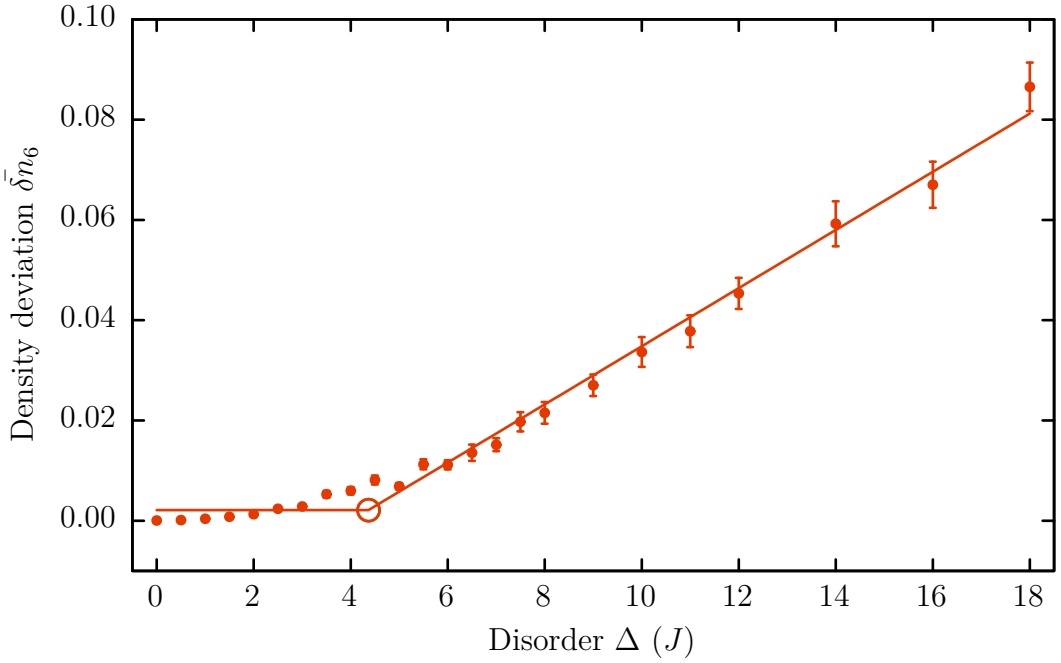
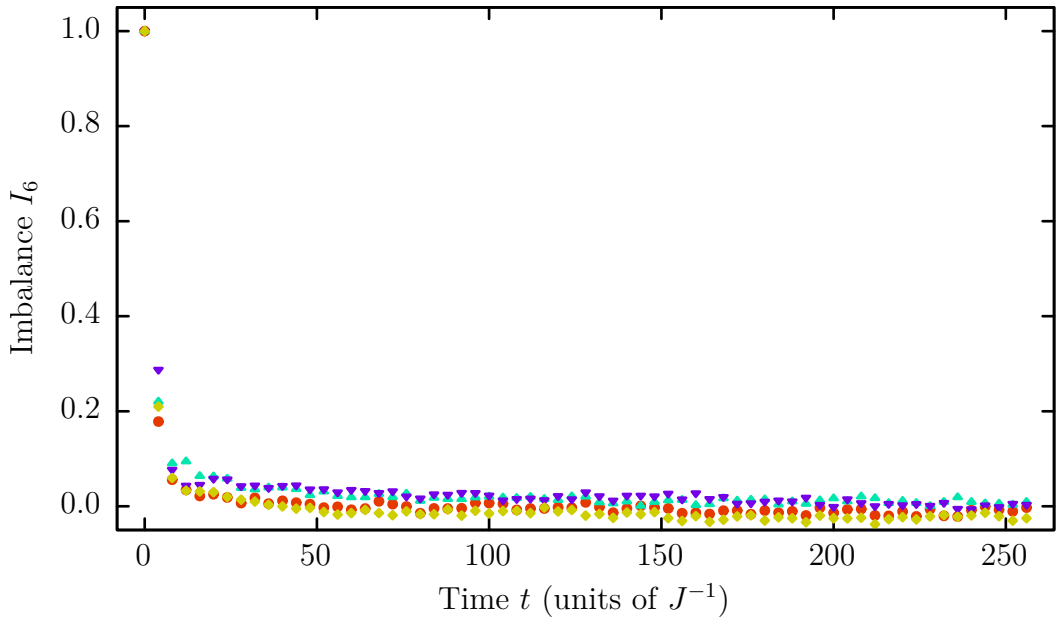


Figure 10.11: Evolution of the imbalance I_6 for $\Delta = 1\,J$ in a model with $E_A = 0.14\,U$. The figure shows the imbalance for four random potential realizations.



where $\Delta_{c,\bar{I}}$ is the critical disorder strength. In contrast to the article [40], where the fit was performed in the interval $\Delta/J \in [0, 8]$, I performed the fit taking into account all calculated values in the interval $\Delta/J \in [0, 18]$. The reason is that there is no apparent deviation from linearity for large Δ in the calculated results. The critical disorders are $\Delta_{c,\bar{I}_2} = 4.3(3)\,J$ for $\bar{I}_2$ and $\Delta_{c,\bar{I}_6} = 3.9(3)\,J$ for $\bar{I}_6$. The critical disorder obtained in the experiment [40] was $\Delta_{c,I} = 5.5(4)\,J$.

The results for the density deviation are less clear. Figures 10.9 and 10.10 show that the density deviation slowly increases with Δ until is becomes linear for large Δ. To follow the experiment, I again fitted the data with a double-linear fit

$$\bar{\delta n}(\Delta) = C + D \times \max\left[(\Delta - \Delta_{c,\bar{\delta n}}), 0\right]. \tag{10.17}$$

The extracted critical disorders are $\Delta_{c,\bar{\delta n}_2} = 2.9(2)\,J$ for $\bar{\delta n}_2$ and $\Delta_{c,\bar{\delta n}_6} = 4.4(2)\,J$ for $\bar{\delta n}_6$. The critical disorder obtained in the experiment was $\Delta_{c,\delta n} = 5.3(2)\,J$.

All above results corresponds to averaged quantities. The dynamics can differ substantially for a particular potential realization. To better understand the results, the following figures show the evolution of I_6 for four potentials taken out of the 100 realizations in each case. Figure 10.11 show the imbalance for $\Delta = 1\,J$, Figure 10.12 for $\Delta = 4\,J$, and Figure 10.13 for $\Delta = 12\,J$. The differences in the imbalances are small for small Δ, but they become prominent for large Δ. Figure 10.14 shows the average imbalance $\bar{I}_6$ for four values of Δ. The average imbalance is very regular compared to individual samples.

Figure 10.12: Evolution of the imbalance I_6 for $\Delta = 4\,J$ in a model with $E_A = 0.14\,U$. The figure shows the imbalance for four random potential realizations.

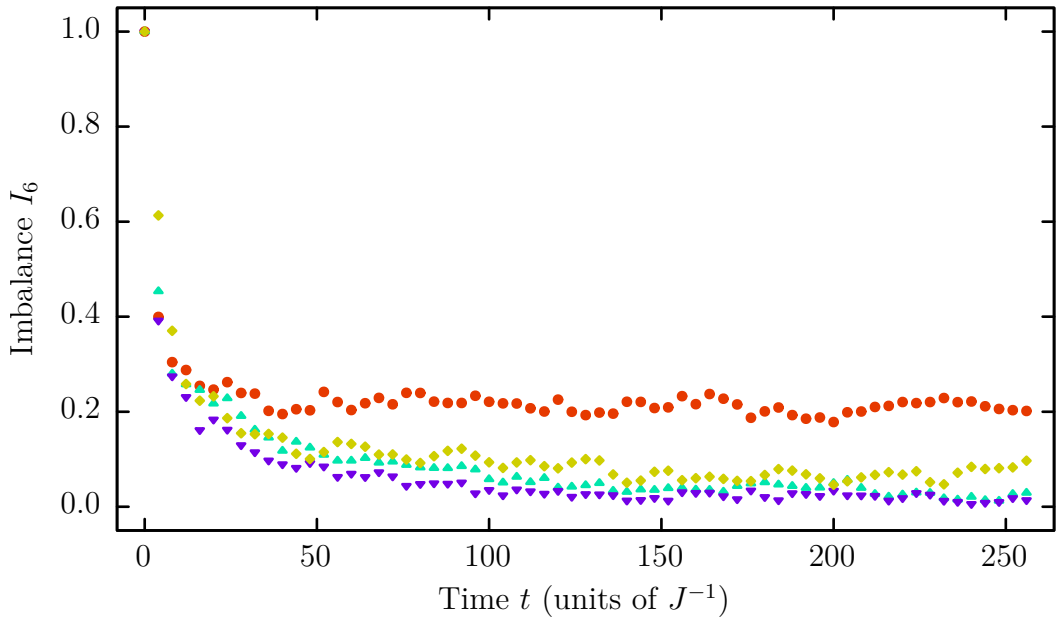


Figure 10.13: Evolution of the imbalance I_6 for $\Delta = 12\,J$ in a model with $E_A = 0.14\,U$. The figure shows the imbalance for four random potential realizations.

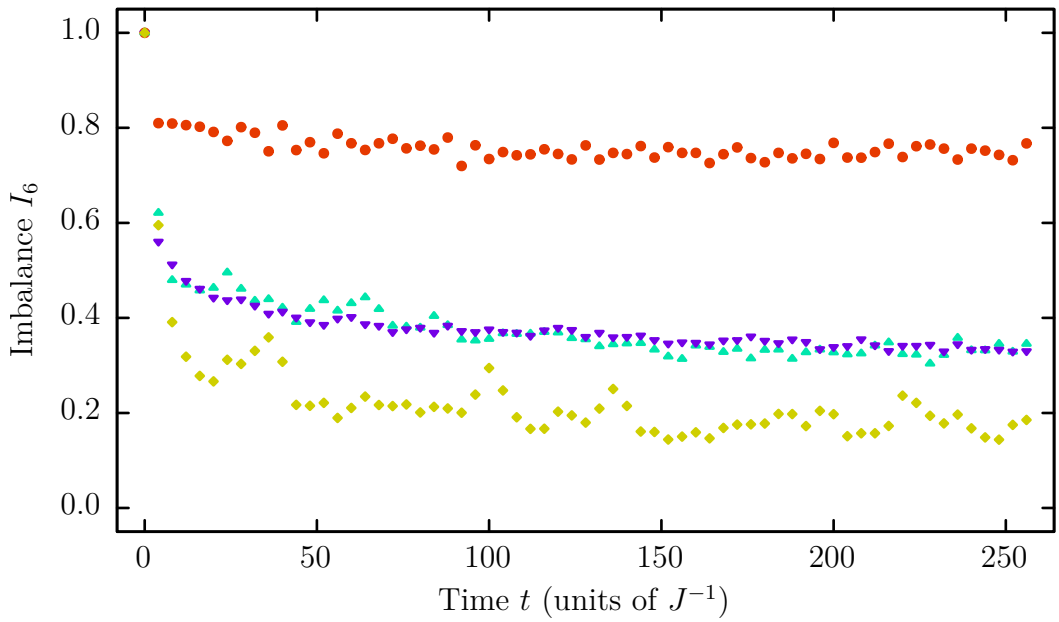
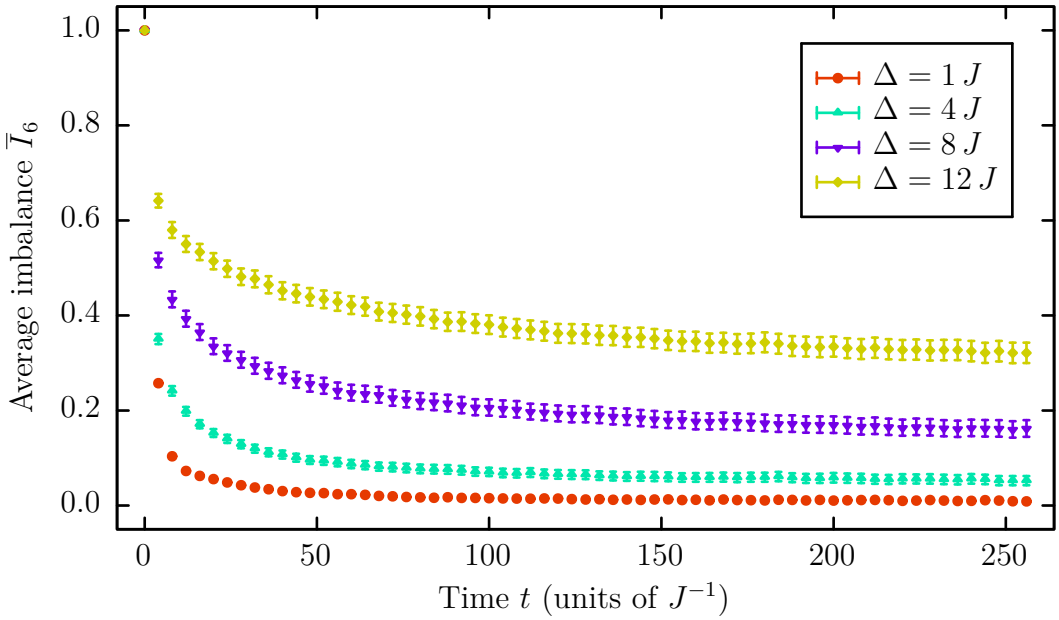


Figure 10.14: Evolution of the average imbalance $\bar{I}_6$ for $\Delta = 1\,J$, $\Delta = 4\,J$, $\Delta = 8\,J$, and $\Delta = 12\,J$ in a model with $E_A = 0.14\,U$. Each average imbalance was calculated from 100 random potential realizations. Error bars represent the standard error of the mean.



10.8 Conclusions

Simulating MBL in a two-dimensional lattice confirmed that there are two regimes of system dynamics for $E_A = 0.14\,U$. The particle density becomes symmetric for small Δ, but stays asymmetric for large Δ. The critical disorder separating these two regimes is not as clear as in the experimental data. The values of $\bar{I}_6$ and $\bar{\delta n}_6$ obtained when taking into account full strip of six sites are fairly close to each other and also to the values obtained in the experiment. It can be also seen from the figures above that the data become slightly irregular near the point $\Delta = 4\,J$. It can be a sign of a complicated dynamics in this region. Further investigation in this region could reveal the reason behind the observed behaviour.

There are several reasons for blurred critical disorder. The most important two are the small system size and the small number of particles. I expect that the transition becomes clearer with increasing number of particle. Another reason is the statistical nature of the calculations. The differences in dynamics for different potential realization are substantial. Again, I expect that the influence of a particular potential realization becomes smaller in larger lattices. The future calculations should therefore focus on increasing both the lattice size and the number of particles.

11. Multidimensional systems

There are many ways how to further extend the presented two-dimensional algorithm. This chapter outlines several ideas about multidimensional systems in general.

A natural extension is to use tensor networks to model three-dimensional systems. One approach is to generalize PEPS topology to three dimensions, that is to assign a tensor to each lattice site, and to connect each tensor to its closest neighbours. Another approach, more aligned with the present work, is to represent a system state as a three-dimensional TTNS (Figure 11.1). The network is composed of layers in the z direction. Each layer is a two-dimensional tree tensor network connected to neighbouring layers through a corner site. The calculation complexity increases with the number of tensor indices. The two-dimensional algorithm rearranges a network in a such way that each tensor carries at most three indices at any given moment. In this generalization to three dimensions, each tensor carries at most four indices. The algorithm for time evolution can work in a similar fashion as in two dimensions. The connections in the network can be rearranged so that each pair of neighbouring sites is eventually connected. TTNO can be constructed analogously to TTNS.

Another extension is to model systems with nontrivial topologies. A fundamental one-dimensional example is a ring of lattice sites. It differs from ordinary one-dimensional lattices by a tunnelling term between the first and the last site. This is equivalent to replacing the open boundary conditions with periodic boundary conditions. The ring topology presents a problem for the current method, because the corresponding tensor network contains a cycle. Canonical states therefore cannot be defined for ring tensor networks. Two-dimensional models with nontrivial topology include a cylinder or a torus. The Bose–Hubbard model on a cylinder has been investigated [90], as well as the quantum Ising model on a torus [30]. Lattice models on a torus are also important for studying topological phases of matter. Additionally, large systems with periodic boundary conditions are appropriate models of infinite lattices in the thermodynamic limit.

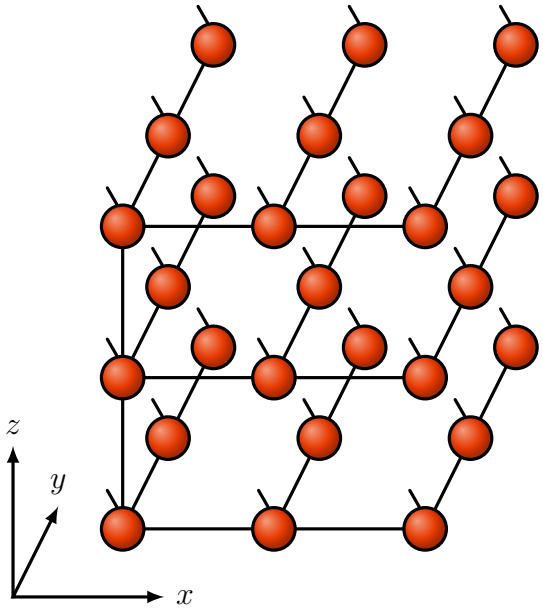
In this work, the TTNS topology was explicitly specified. The topology is also prescribed for PEPS, where each bulk tensor is connected to its four nearest neighbours. A question then arises whether a particular set of connections is an optimal network topology. In one dimension, it is natural to connect neighbouring sites only. It is unclear what should be an appropriate topology in higher dimensions. A basic idea is to connect only strongly correlated sites directly. It would also be beneficial to optimize the network structure automatically, as the optimal topology could be different at each time step.

There has been some progress in this area [31, 98, 99]. An idea coming from application of DMRG to quantum chemistry is to calculate the mutual information between each pair of sites, and connect sites with high mutual information directly. In particular, a suitable one-dimensional ordering can be found by minimizing the entanglement distance given by [98]

$$I_{\mathrm{dist}} = \sum_{\mathbf{i},\mathbf{j}} I_{\mathbf{i},\mathbf{j}} \|\mathbf{i} - \mathbf{j}\|^{\eta}, \tag{11.1}$$

where $I_{\mathbf{i},\mathbf{j}}$ is the mutual information between sites $\mathbf{i}$ and $\mathbf{j}$, and η is a constant,

Figure 11.1: A three-dimensional TTNS. Each ball represents a tensor corresponding to a single lattice site. Open lines represent physical indices, and lines connecting the balls represent tensor contractions.



for example $\eta = 1$ or $\eta = 2$. I_{dist} is used to sort sites so that sites with large mutual information are close to each other. A similar approach could be used for multidimensional systems. The first step is to perform a calculation with small bond dimension D. The aim is to obtain approximate mutual information for each pair of sites. Then a suitable network topology with a small I_{dist} can be constructed. The subsequent calculation with a large D uses a TTNS with this topology.

Extensions of current methods to three-dimensional systems and to systems with nontrivial topologies present several challenges, both conceptually and computationally. Hopefully, the advances in tensor networks theory will lead to studies of large multidimensional systems in future. New insights would be beneficial to simulations of one-dimensional and two-dimensional models as well.

Conclusions

This thesis has summarized my research in modelling of ultracold atoms in multi-dimensional optical lattices. Physics of bosonic atoms moving and interacting in a lattice can be described by the Bose–Hubbard model. There exist several analytic and computational methods to investigate this model. I have focused exclusively on numerical methods based on tensor networks. The model can be investigated for the full range of its parameters with this approach. Additionally, I have focused only on its time evolution and especially on the resulting nonequilibrium dynamics.

The work has involved extending methods for modelling one-dimensional quantum systems to two dimensions, developing numerical software, and calculating the time evolution of interesting physical systems. I have considered several algorithms for solving the two-dimensional problem. My approach was to create a working implementation that can be applied to realistic models that follow current experiments. I have found that some methods, for example PEPS, have sound theoretical foundations, but is it difficult to achieve good performance with them. My final approach was to use the TTNS representation. The decisive argument in favour of TTNS was that the notion of canonical states can be extended from one-dimensional MPS to multidimensional TTNS. The existence of canonical states considerably simplifies the complexity of the algorithm.

The primary output of my work is TEBDOL. It is a program for simulating time evolution in one-dimensional and two-dimensional lattice models. Its main method is a generalization of the TEBD algorithm to TTNS. It has been tested thoroughly and provides good performance on realistic problems. Another useful output is the tensor library that underlies TEBDOL.

I used TEBDOL to study three phenomena in two-dimensional optical lattices — phase revivals, boson expansion, and many-body localization. All three have already been observed in experiments. Limited applicability of theoretical methods to two-dimensional models makes it hard to analyze the experimental findings in detail. The first study explored the role of two-dimensional tunnelling on the attenuation of phase revivals. In the simulation of boson expansion, I improved published theoretical results by investigating a larger system. I am not aware of any numerical studies of many-body localization in a two-dimensional lattice, so the last part of this work represents an original result. The study has confirmed that the transition from thermalization to many-body localization takes places in the model. I have also found a critical disorder strength comparable to the value measured in the experiment.

Studying time evolution with tensor networks presents several challenges. Accessible system sizes are limited by available computing resources. In particular, the system size and the accuracy of results are bounded by the amount of available memory. The maximum simulated time interval is bounded by the CPU power. Presented results correspond to the largest models I could simulate with reasonable resources. The calculations can be extended to even larger systems and longer time intervals by using better hardware in future.

There are several ways how to extend the present work. A natural direction is to develop a similar method for three-dimensional systems. However, the experience with the two-dimensional algorithm suggests that modelling three-

dimensional system is going to be a very computationally intensive task. It would be nonetheless interesting to determine how large three-dimensional systems are accessible to simulations with tensor networks on current computers. A similar area is to study models with other geometries, and to implement support for periodic boundary conditions. They are also important for investigations of infinite systems in the thermodynamic limit.

A useful research idea is to parallelize the program. I have already parallelized the one-dimensional algorithm. There is a trade-off that results become slightly inaccurate after performing large truncations of the tensor network. Truncations in two-dimensional systems are generally more prominent than in one-dimensional systems, so it is problematic to obtain robust parallelization in this case. In a sense, a correct truncation of any virtual index in a tensor network is a global operation, so it is not well-suited for parallelization. A thorough theoretical analysis is required to develop an appropriate parallelization strategy.

Another direction is to study fermions in optical lattices. Their physics is closely related to the physics of electrons in crystal lattices. The fermionic anti-commutation relations make it necessary to use a slightly different representation. Fermionic models have been investigated with MPS, therefore it should be possible to generalize existing ideas to TTNS and to two-dimensional lattices. A closely related area are particles with anyonic statistics in two dimensions.

Ultimately, the simulation of quantum systems on a classical computer is a formidable task. The dimensions of Hilbert spaces for systems with moderate numbers of particles are huge. If the amount of entanglement is low, current methods provide a way to extract useful information about a system. Generally, the amount of entanglement grows quickly, especially after quantum quenches. The time interval accessible in simulations is therefore rather short.

It has been known for some time that one way to overcome these obstacles is to implement and solve quantum models on quantum computers. In fact, an optical lattice is an analogue quantum computer that can simulate, for instance, the evolution of the Bose–Hubbard model. The model can be implemented on a digital quantum computer that can be programmed to perform any quantum calculations in future. There has been a substantial progress in this area in recent years.

I hope the present work contributes to understanding time evolution in multi-dimensional quantum systems. It shows that even the simplest models exhibit complicated dynamics. Solving a model numerically and comparing the obtained results to experimental observations not only confirms that the model describes reality well enough, but also allows one to predict its behaviour under altered circumstances.

Appendix A. Matrix product states compression

An important part of several tensor-network algorithms is an approximation of an MPS with an MPS with smaller bond dimensions. The operation is called MPS compression, but similar operation is also performed in a search for the ground state of a lattice model. There exist two approaches, compression with SVD and variational compression. Both have advantages and disadvantages. SVD compression is not optimal, its result depends on the order of decompositions, and it is slow. Variational compression is fast for large compressions with $D' \ll D$, where D and D' are the maximum bond dimensions of the original and of the compressed MPS, respectively. It is also better at finding the optimal result, but can stuck in a local minimum.

This appendix explains the details of SVD compression using a simple example. It is helpful in getting a better understanding of MPS. It also discusses why the parallel algorithm loses accuracy with large truncations. The notation closely follows article [16].

Compression with singular value decomposition

A simple method to compress an MPS is based on singular value decomposition. The procedure starts from a left-canonical form

$$c_{\sigma_1,\ldots,\sigma_L} = A_1^{\sigma_1} A_2^{\sigma_2} \ldots A_{L-1}^{\sigma_{L-1}} A_L^{\sigma_L}, \tag{11.2}$$

where $A_i^{\sigma_i}$ are left-normalized tensors. Left-normalized tensors $A_i^{\sigma_i}$ and right-normalized tensors $B_i^{\sigma_i}$ obey the conditions

$$\begin{aligned} \sum_{\sigma_i} A_i^{\sigma_i \dagger} A_i^{\sigma_i} &= I_i^A, \\ \sum_{\sigma_i} B_i^{\sigma_i} B_i^{\sigma_i \dagger} &= I_i^B, \end{aligned} \tag{11.3}$$

where I_i^A and I_i^B are unit matrices. The tensors in (11.2) are consecutively decomposed and truncated from the right-hand side. First, the tensor $A_L^{\sigma_L}$ is decomposed using SVD into $A_L^{\sigma_L} = M_{\alpha,\sigma_L\beta} = U_{\alpha,\gamma} S_{\gamma,\gamma'} V^\dagger_{\gamma',\sigma_L\beta}$. The decomposed matrices are truncated to contain only D' singular values. This means S is replaced by a $D' \times D'$ matrix S' that has only the singular values with the largest absolute value on the diagonal. Similarly, matrices U and $V^\dagger$ are replaced by matrices U' and $V'^\dagger$ that contain only the D' corresponding columns and rows, respectively. The matrix $V'\dagger$ can be identified with a right-normalized tensor $B_L^{\sigma_L}$. The truncated decomposition is then given by

$$c_{\sigma_1,\ldots,\sigma_L} = A_1^{\sigma_1} A_2^{\sigma_2} \ldots A_{L-2}^{\sigma_{L-2}} A_{L-1}^{\sigma_{L-1}} U' S' V'^\dagger = A_1^{\sigma_1} A_2^{\sigma_2} \ldots A_{L-2}^{\sigma_{L-2}} M_{L-1}^{\sigma_{L-1}} B_L^{\sigma_L}, \tag{11.4}$$

where $M_{L-1}^{\sigma_{L-1}} = A_{L-1}^{\sigma_{L-1}} U' S'$. The compression then continues with the decomposition of $M_{L-1}^{\sigma_{L-1}}$.

Tensors in a compressed MPS depend on the compression direction in the chain. If the compression starts from a left-canonical MPS and truncations proceed from

the right-hand side to the left-hand side, one obtains a different MPS than if the compression starts from the right-canonical MPS and truncations proceed in the opposite direction. Each truncation creates essentially a new MPS. Additionally, each truncation breaks the normalization conditions, and the MPS ceases to be in a canonical form afterwards. The final MPS depends on the order in which the bonds are truncated.

The computational cost of calculating the SVD of an $m \times n$ matrix with $m \geq n$ is $\mathcal{O}(mn^2)$. The matrix corresponding to the tensor $M_i^{\sigma_i}$ above has dimensions $D \times dD'$, where d is the physical dimension. The SVD cost is therefore either $\mathcal{O}(dD^2D')$ or $\mathcal{O}(d^2DD'^2)$. The matrix multiplication following the decomposition costs $\mathcal{O}(dD^2D')$. In practice, D' is usually proportional to D, so the complexity of the SVD compression scales as $\mathcal{O}(D^3)$. This is also the cost of bringing an MPS into a canonical state which is a required initial state in many algorithms.

Example

To see why a compressed MPS depends on the order of truncations, it is instructional to consider a chain of three qubits as an example. The goal is to compress a state

$$|\psi\rangle = c_1|100\rangle + c_2|010\rangle + c_3|001\rangle. \tag{11.5}$$

Its Schmidt decomposition [16] at the first bond is given by

$$\sum_{i=1}^{r} \alpha_i |u_i\rangle \otimes |v_i\rangle, \tag{11.6}$$

where r is its Schmidt rank, α_i are its Schmidt coefficients, $|u_i\rangle$ are orthonormal vectors from the Hilbert space of the first qubit, and $|v_i\rangle$ are orthonormal vectors from the Hilbert space of the second and third qubits combined. The rank is $r = 2$ for the state (11.5), and the decomposition coefficients and vectors are given by

$$\begin{aligned} &\alpha_1 = |c_1|, && u_1 = |1\rangle, \quad v_1 = \frac{c_1}{|c_1|}|00\rangle, \\ &\alpha_2 = \sqrt{|c_2|^2 + |c_3|^2}, && u_2 = |0\rangle, \quad v_2 = \frac{1}{\sqrt{|c_2|^2 + |c_3|^2}} \left(c_2|10\rangle + c_3|01\rangle\right). \end{aligned} \tag{11.7}$$

Similarly, the Schmidt decomposition at the second bond reads

$$\sum_{i=1}^{s} \beta_i |w_i\rangle \otimes |z_i\rangle, \tag{11.8}$$

where s is its Schmidt rank, β_i are its Schmidt coefficients, $|w_i\rangle$ are orthonormal vectors from the Hilbert space of the first and second qubits combined, and $|z_i\rangle$ are orthonormal vectors from the Hilbert space of the third qubit. Again, for the state (11.5) the rank is $s = 2$, and the decomposition coefficients and vectors are given by

$$\begin{aligned} &\beta_1 = \sqrt{|c_1|^2 + |c_2|^2}, && w_1 = \frac{1}{\sqrt{|c_1|^2 + |c_2|^2}} \left(c_1|10\rangle + c_2|01\rangle\right), && z_1 = |0\rangle, \\ &\beta_2 = |c_3|, && w_2 = \frac{c_3}{|c_3|}|00\rangle, && z_2 = |1\rangle. \end{aligned} \tag{11.9}$$

If all coefficients c_i are nonzero, all Schmidt coefficients α_i and β_i are nonzero as well. The bond dimension of an MPS decomposition into $M_1^{\sigma_1} M_2^{\sigma_2} M_3^{\sigma_3}$ is therefore $D = 2$. In this example, the goal is to compress $|\psi\rangle$ into $|\psi'\rangle$ with a bond dimension $D' = 1$.

Let $c_1 = 4/\sqrt{77}$, $c_2 = 5/\sqrt{77}$, and $c_3 = 6/\sqrt{77}$. If the compression starts from the right-canonical state $B_1^{\sigma_1} B_2^{\sigma_2} B_3^{\sigma_3}$, a decomposition $A_1^{\sigma_1} S_1 V_1^\dagger B_2^{\sigma_2} B_3^{\sigma_3}$ is obtained in the first step. The singular values of S_1 are the Schmidt coefficients $\alpha_1 = |c_1| = 4/\sqrt{77}$ and $\alpha_2 = \sqrt{|c_2|^2 + |c_3|^2} = \sqrt{61/77}$. The truncation retains only the vectors corresponding to the largest Schmidt coefficient α_2, and results in a state

$$|\psi_L^T\rangle = c_2|010\rangle + c_3|001\rangle = \frac{5}{\sqrt{77}}|010\rangle + \frac{6}{\sqrt{77}}|001\rangle. \tag{11.10}$$

The decomposition in the second step is given by $A_1^{\sigma_1} A_2^{\sigma_2} S_2 V_2^\dagger B_3^{\sigma_3}$, where the singular values in S_2 correspond to the Schmidt coefficients of a decomposition at the second bond. However, the decomposition is performed on the truncated vector (11.10) now, and it reads

$$\begin{aligned} \beta_1 = |c_2| = \frac{5}{\sqrt{77}}, \quad w_1 = \frac{c_2}{|c_2|}|01\rangle, \quad z_1 = |0\rangle, \\ \beta_2 = |c_3| = \frac{6}{\sqrt{77}}, \quad w_2 = \frac{c_3}{|c_3|}|00\rangle, \quad z_2 = |1\rangle. \end{aligned} \tag{11.11}$$

Again, the truncation retains only vectors corresponding to the largest Schmidt coefficient β_2. The final compressed state is therefore

$$|\psi_L'\rangle = c_3|001\rangle = \frac{6}{\sqrt{77}}|001\rangle. \tag{11.12}$$

If the compression starts from the left-canonical state $A_1^{\sigma_1} A_2^{\sigma_2} A_3^{\sigma_3}$, a decomposition $A_1^{\sigma_1} A_2^{\sigma_2} U_3 S_3 B_3^{\sigma_3}$ is obtained, where the matrix S_3 contains the singular values (11.9). They are $\beta_1 = \sqrt{|c_1|^2 + |c_2|^2} = \sqrt{41/77}$ and $\beta_2 = |c_3| = 6/\sqrt{77}$. Only the largest singular value is retained, which gives a truncated state

$$|\psi_R^T\rangle = c_1|100\rangle + c_2|010\rangle = \frac{4}{\sqrt{77}}|100\rangle + \frac{5}{\sqrt{77}}|010\rangle. \tag{11.13}$$

The second decomposition into $A_1^{\sigma_1} U_2 S_2 B_2^{\sigma_2}$ and the following truncation produces a state

$$|\psi_R'\rangle = c_2|010\rangle = \frac{5}{\sqrt{77}}|010\rangle. \tag{11.14}$$

Vector (11.14) differs from vector (11.12), therefore the compression result depends on the order of truncations.

An alternative idea is to truncate the tensors without performing any SVD. This method keeps only D' pairs of Schmidt vectors at each bond. The resulting state is a projection of the original vector onto the subspaces spanned by all preserved Schmidt vectors. In the above example, it would keep the vector $|\psi_L^T\rangle = c_2|010\rangle + c_3|001\rangle$ at the first bond and the vector $|\psi_R^T\rangle = c_1|100\rangle + c_2|010\rangle$ at the second bond. Truncation of all tensors without performing the SVD leads to a state $\psi_N' = c_2|010\rangle = \frac{5}{\sqrt{77}}$ which is again not the optimal outcome. This method generally gives worse results than the SVD compression because it always

picks the common vector $c_2|010\rangle$ without any regards to c_1 or c_3. It is also what happens in the parallel one-dimensional algorithms with independent truncations.

SVD compression can be understood as sequential projections of a vector. Each neglected part depends on the bond the decomposition is performed on. The final state therefore depends on the order in which the bonds are selected.

Variational compression

Let ψ be an MPS with bond dimension D. The goal is to find an MPS ψ' with a fixed bond dimension D' that minimizes $\|\psi - \psi'\|^2$. Variational compression starts from an initial guess ψ'_0 and optimizes all its tensors. ψ'_0 can be a randomly generated MPS or an MPS obtained from the SVD compression. It is difficult to optimize all tensors at the same time, because it is a nonlinear optimization problem. The algorithm therefore sweeps through the chain and optimizes one tensor at the time while keeping all other tensors fixed. This approach is also called the alternating least squares method (ALS). It does not guarantee finding the global minimum but works well in practice. It is also similar to the DMRG algorithm for finding ground states and to its improvement called the alternating minimal energy method (AME) [100, 101].

Appendix B. Implementation notes

TEBDOL [41] is implemented in Common Lisp. It is a language rarely used in high-performance computing nowadays, despite its many benefits. Common Lisp is a very high-level language that makes it possible to define and work with sophisticated data structures easily. It also provides high numerical performance [102, 103]. TEBDOL uses open-source implementation Steel Bank Common Lisp (SBCL) [104]. The goal of the work has been to develop high-level and fast code. Implementation in Common Lisp achieved both [105].

The basic layer of TEBDOL is a library for tensor manipulation. There is a distinction between dense tensors called *arrays* and symmetric tensors called simply *tensors*. Symmetric tensors use arrays for storing their nonzero blocks.

Arrays are implemented as ordinary Common Lisp arrays, and are stored as contiguous blocks of complex double-precision floating-point numbers. Basic operations on arrays are performed using Common Lisp functions. Additionally, TEBDOL uses BLAS [106] and LAPACK [107] to perform linear algebra operations on arrays. Arrays are first reshaped into matrices, and external routines are called to perform the requested operations. The obtained matrices are then reshaped back into arrays. TEBDOL attempts to minimize the amount of reshaping. If array indices are already in the correct order, it directly calls the external routine. Array operations performed by external routines are array contraction, array decomposition, and diagonalization of a Hermitean matrix. The last operation is used to calculate time-evolution operators.

Array permutation and array contraction are operations that take advantage of the Common Lisp macro system. They support arrays of arbitrary rank. A naive implementation of permutations and contractions for general arrays was slow. TEBDOL therefore uses a different method. It generates a specific function for any combinations of array ranks and indices. The functions are created and compiled during runtime. This approach ensures good runtime performance of both operations.

TEBDOL uses matrix SVD to decompose arrays. There are several LAPACK routines to calculate a SVD. Routine ZGESVD uses the standard approach while routine ZGESDD uses a divide and conquer algorithm, which is much faster for large matrices. I have found that ZGESDD is numerically unstable in all tested versions of LAPACK. For certain matrices, ZGESDD fails to converge, or crashes due to a floating-point error. TEBDOL therefore falls back to ZGESVD if it detects a problem in ZGESDD for a particular matrix. I have not observed any numerical instabilities in ZGESVD. This way TEBDOL achieves a good performance of SVD without any instabilities.

Tensors are built on top of arrays. Each tensor has a list of indices and a list of blocks. The blocks are represented by arrays. Tensor operations include index fusion, tensor conjugation, tensor permutation, tensor contraction, and tensor decomposition. TEBDOL also supports a calculation of tensor exponential for a special class of tensors that are analogous to Hermitean matrices. The operations are fairly complicated, as TEBDOL must determine which blocks contribute to the resulting tensor. Operations on blocks are performed using array routines described above.

TTNS and TTNO are constructed from tensors. Quantum operators discussed

in this work are implemented as TTNO, which are a generalization of MPO [16, 41]. Because each operator has a distinct structure, they are implemented independently.

Parallel version for one-dimensional models uses MPI for communication between compute nodes. A tensor is a structure that consists of multiple data types on several levels. It is necessary to exchange tensors between the nodes during a calculation. TEBDOL uses its own serialization and deserialization routines. They can convert a tensor to a block of bytes and such a block back to a tensor. They are used for exchanging tensors between nodes. TEBDOL supports both OpenMPI and MPICH libraries. It has been run with up to 256 MPI processes on 64 nodes.

The two-dimensional version does not use MPI parallelization. However, TEBDOL uses BLAS and LAPACK for linear algebra operations. Most of implementations of these libraries are parallel on multi-core systems. TEBDOL can therefore utilize all CPU cores on a single node. Exact parallel scaling depends on the problem size.

The program performance depends on all calculation parameters, including the model parameters, the lattice size, the total number of particles, the time step, and the simulated time interval. Generally, the complexity of a calculation depends mostly on the number of particles, because particle interactions lead to an increase of the entanglement entropy in the system. The calculation parameters discussed in this thesis were chosen so that a single run could finish in a day or two. This typically led to a bond dimension of about 500–1000 for two-dimensional models. The calculations required about 10–30 GiB of available memory. The memory requirements and calculation time grows substantially with increasing maximum bond dimension D.

There are several other open-source packages for quantum calculations using tensor networks [108–126]. Advantages of TEBDOL are a parallel version of the one-dimensional algorithm and the focus on two-dimensional problems.

Appendix C. List of publications

[1] M. Urbanek and P. Soldán, "Matter-wave revival of binary mixtures in optical lattices", Phys. Rev. A **90**, 033610 (2014).

[2] M. Urbanek, "Quantum physics simulations in Common Lisp", in Proceedings of the 8th European Lisp Symposium, edited by J. Padget (2015), pp. 71–78.

[3] M. Urbanek and P. Soldán, "Parallel implementation of the time-evolving block decimation algorithm for the Bose–Hubbard model", Comput. Phys. Commun. **199**, 170–177 (2016).

Bibliography

[1] D. Jaksch and P. Zoller, "The cold atom Hubbard toolbox", Ann. Phys. **315**, 52–79 (2005).

[2] O. Morsch and M. Oberthaler, "Dynamics of Bose–Einstein condensates in optical lattices", Rev. Mod. Phys. **78**, 179–215 (2006).

[3] I. Bloch, J. Dalibard and W. Zwerger, "Many-body physics with ultracold gases", Rev. Mod. Phys. **80**, 885–964 (2008).

[4] M. Lewenstein, A. Sanpera and V. Ahufinger, *Ultracold atoms in optical lattices: Simulating quantum many-body systems* (Oxford University Press, London, 2012), ISBN: 978-01-995-7312-7.

[5] T. Stöferle, H. Moritz, C. Schori, M. Köhl and T. Esslinger, "Transition from a strongly interacting 1D superfluid to a Mott insulator", Phys. Rev. Lett. **92**, 130403 (2004).

[6] B. Paredes, A. Widera, V. Murg, O. Mandel, S. Fölling, I. Cirac, G. V. Shlyapnikov, T. W. Hänsch and I. Bloch, "Tonks–Girardeau gas of ultracold atoms in an optical lattice", Nature **429**, 277–281 (2004).

[7] H. A. Gersch and G. C. Knollman, "Quantum cell model for bosons", Phys. Rev. **129**, 959–967 (1963).

[8] M. P. A. Fisher, P. B. Weichman, G. Grinstein and D. S. Fisher, "Boson localization and the superfluid-insulator transition", Phys. Rev. B **40**, 546–570 (1989).

[9] D. Jaksch, C. Bruder, J. I. Cirac, C. W. Gardiner and P. Zoller, "Cold bosonic atoms in optical lattices", Phys. Rev. Lett. **81**, 3108–3111 (1998).

[10] T. D. Kühner and H. Monien, "Phases of the one-dimensional Bose–Hubbard model", Phys. Rev. B **58**, R14741–R14744 (1998).

[11] M. Greiner, O. Mandel, T. Esslinger, T. W. Hänsch and I. Bloch, "Quantum phase transition from a superfluid to a Mott insulator in a gas of ultracold atoms", Nature **415**, 39–44 (2002).

[12] S. R. White, "Density matrix formulation for quantum renormalization groups", Phys. Rev. Lett. **69**, 2863–2866 (1992).

[13] S. R. White, "Density-matrix algorithms for quantum renormalization groups", Phys. Rev. B **48**, 10345–10356 (1993).

[14] M. A. Martín-Delgado, G. Sierra and R. M. Noack, "The density matrix renormalization group applied to single-particle quantum mechanics", J. Phys. A Math. Gen. **32**, 6079 (1999).

[15] U. Schollwöck, "The density-matrix renormalization group", Rev. Mod. Phys. **77**, 259–315 (2005).

[16] U. Schollwöck, "The density-matrix renormalization group in the age of matrix product states", Ann. Phys. **326**, 96–192 (2011).

[17] G. Vidal, "Efficient classical simulation of slightly entangled quantum computations", Phys. Rev. Lett. **91**, 147902 (2003).

[18] G. Vidal, "Efficient simulation of one-dimensional quantum many-body systems", Phys. Rev. Lett. **93**, 040502 (2004).

[19] A. J. Daley, C. Kollath, U. Schollwöck and G. Vidal, "Time-dependent density-matrix renormalization-group using adaptive effective Hilbert spaces", J. Stat. Mech. Theor. Exp. **2004**, P04005 (2004).

[20] S. R. White and A. E. Feiguin, "Real-time evolution using the density matrix renormalization group", Phys. Rev. Lett. **93**, 076401 (2004).

[21] G. Vidal, "Classical simulation of infinite-size quantum lattice systems in one spatial dimension", Phys. Rev. Lett. **98**, 070201 (2007).

[22] N. Schuch, I. Cirac and F. Verstraete, "Computational difficulty of finding matrix product ground states", Phys. Rev. Lett. **100**, 250501 (2008).

[23] Z. Landau, U. Vazirani and T. Vidick, "A polynomial time algorithm for the ground state of one-dimensional gapped local Hamiltonians", Nat. Phys. **11**, 566–569 (2015).

[24] J. Eisert, M. Cramer and M. B. Plenio, "Colloquium: Area laws for the entanglement entropy", Rev. Mod. Phys. **82**, 277–306 (2010).

[25] F. Verstraete and J. I. Cirac, "Renormalization algorithms for quantum-many body systems in two and higher dimensions", arXiv.org (2004), arXiv:`cond-mat/0407066v1 [cond-mat.str-el]`.

[26] R. Orús, "A practical introduction to tensor networks: Matrix product states and projected entangled pair states", Ann. Phys. **349**, 117–158 (2014).

[27] M. B. Hastings, "Solving gapped Hamiltonians locally", Phys. Rev. B **73**, 085115 (2006).

[28] M. M. Wolf, F. Verstraete, M. B. Hastings and J. I. Cirac, "Area laws in quantum systems: Mutual information and correlations", Phys. Rev. Lett. **100**, 070502 (2008).

[29] Y.-Y. Shi, L.-M. Duan and G. Vidal, "Classical simulation of quantum many-body systems with a tree tensor network", Phys. Rev. A **74**, 022320 (2006).

[30] L. Tagliacozzo, G. Evenbly and G. Vidal, "Simulation of two-dimensional quantum systems using a tree tensor network that exploits the entropic area law", Phys. Rev. B **80**, 235127 (2009).

[31] V. Murg, F. Verstraete, Ö. Legeza and R. M. Noack, "Simulating strongly correlated quantum systems with tree tensor networks", Phys. Rev. B **82**, 205105 (2010).

[32] G. Vidal, "Entanglement renormalization", Phys. Rev. Lett. **99**, 220405 (2007).

[33] G. Vidal, "Class of quantum many-body states that can be efficiently simulated", Phys. Rev. Lett. **101**, 110501 (2008).

[34] M. Greiner, O. Mandel, T. W. Hänsch and I. Bloch, "Collapse and revival of the matter wave field of a Bose–Einstein condensate", Nature **419**, 51–54 (2002).

[35] F. Gerbier, S. Trotzky, S. Fölling, U. Schnorrberger, J. D. Thompson, A. Widera, I. Bloch, L. Pollet, M. Troyer, B. Capogrosso-Sansone, N. V. Prokof'ev and B. V. Svistunov, "Expansion of a quantum gas released from an optical lattice", Phys. Rev. Lett. **101**, 155303 (2008).

[36] J. P. Ronzheimer, M. Schreiber, S. Braun, S. S. Hodgman, S. Langer, I. P. McCulloch, F. Heidrich-Meisner, I. Bloch and U. Schneider, "Expansion dynamics of interacting bosons in homogeneous lattices in one and two dimensions", Phys. Rev. Lett. **110**, 205301 (2013).

[37] S. Trotzky, Y.-A. Chen, A. Flesch, I. P. McCulloch, U. Schollwöck, J. Eisert and I. Bloch, "Probing the relaxation towards equilibrium in an isolated strongly correlated one-dimensional Bose gas", Nat. Phys. **8**, 325–330 (2012).

[38] J. Eisert, M. Friesdorf and C. Gogolin, "Quantum many-body systems out of equilibrium", Nat. Phys. **11**, 124–130 (2015).

[39] A. M. Kaufman, M. E. Tai, A. Lukin, M. Rispoli, R. Schittko, P. M. Preiss and M. Greiner, "Quantum thermalization through entanglement in an isolated many-body system", Science **353**, 794–800 (2016).

[40] J.-y. Choi, S. Hild, J. Zeiher, P. Schauß, A. Rubio-Abadal, T. Yefsah, V. Khemani, D. A. Huse, I. Bloch and C. Gross, "Exploring the many-body localization transition in two dimensions", Science **352**, 1547–1552 (2016).

[41] M. Urbanek and P. Soldán, "Parallel implementation of the time-evolving block decimation algorithm for the Bose–Hubbard model", Comput. Phys. Commun. **199**, 170–177 (2016).

[42] M. Urbanek and P. Soldán, "TEBDOL – parallel simulation of ultracold atoms in optical lattices", in *Supercomputing in science and engineering*, edited by K. Pešatová, B. Poláková and J. Cawley, 1st ed. (VŠB – Technical University of Ostrava, Ostrava, 2017), pp. 213–215, ISBN: 978-80-248-4037-6.

[43] R. Grimm, M. Weidemüller and Y. B. Ovchinnikov, "Optical dipole traps for neutral atoms", Adv. At. Mol. Opt. Phy. **42**, 95–170, ISSN: 1049-250X (2000).

[44] M. Urbanek, "Binary mixtures in one-dimensional optical lattices", Diploma thesis (Czech Technical University, Prague, 2008).

[45] C. Weitenberg, M. Endres, J. F. Sherson, M. Cheneau, P. Schauß, T. Fukuhara, I. Bloch and S. Kuhr, "Single-spin addressing in an atomic Mott insulator", Nature **471**, 319–324 (2011).

[46] D. M. Weld, P. Medley, H. Miyake, D. Hucul, D. E. Pritchard and W. Ketterle, "Spin gradient thermometry for ultracold atoms in optical lattices", Phys. Rev. Lett. **103**, 245301 (2009).

[47] H. Miyake, G. A. Siviloglou, G. Puentes, D. E. Pritchard, W. Ketterle and D. M. Weld, "Bragg scattering as a probe of atomic wave functions and quantum phase transitions in optical lattices", Phys. Rev. Lett. **107**, 175302 (2011).

[48] G. H. Golub and C. F. Van Loan, *Matrix computations*, 4th ed. (Johns Hopkins University Press, Baltimore, 2012), ISBN: 978-1-4214-0794-4.

[49] S. Singh, R. N. C. Pfeifer and G. Vidal, "Tensor network decompositions in the presence of a global symmetry", Phys. Rev. A **82**, 050301 (2010).

[50] S. Singh, R. N. C. Pfeifer and G. Vidal, "Tensor network states and algorithms in the presence of a global $U(1)$ symmetry", Phys. Rev. B **83**, 115125 (2011).

[51] D. Muth, "Particle number conservation in quantum many-body simulations with matrix product operators", J. Stat. Mech. Theor. Exp. **2011**, P11020 (2011).

[52] I. P. McCulloch, "From density-matrix renormalization group to matrix product states", J. Stat. Mech. Theor. Exp. **2007**, P10014 (2007).

[53] F. Verstraete, V. Murg and J. I. Cirac, "Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems", Adv. Phys. **57**, 143–224 (2008).

[54] S. Östlund and S. Rommer, "Thermodynamic limit of density matrix renormalization", Phys. Rev. Lett. **75**, 3537–3540 (1995).

[55] J. Dukelsky, M. A. Martín-Delgado, T. Nishino and G. Sierra, "Equivalence of the variational matrix product method and the density matrix renormalization group applied to spin chains", Europhys. Lett. **43**, 457 (1998).

[56] I. V. Oseledets, "Tensor-train decomposition", SIAM J. Sci. Comput. **33**, 2295–2317 (2011).

[57] S. R. White and D. J. Scalapino, "Density matrix renormalization group study of the striped phase in the 2D t–J model", Phys. Rev. Lett. **80**, 1272–1275 (1998).

[58] S. R. White and D. J. Scalapino, "Stripes on a 6-leg Hubbard ladder", Phys. Rev. Lett. **91**, 136403 (2003).

[59] F. Verstraete and J. I. Cirac, "Matrix product states represent ground states faithfully", Phys. Rev. B **73**, 094423 (2006).

[60] 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**, 207204 (2004).

[61] B. Pirvu, V. Murg, J. I. Cirac and F. Verstraete, "Matrix product operator representations", New J. Phys. **12**, 025012 (2010).

[62] F. Fröwis, V. Nebendahl and W. Dür, "Tensor operators: Constructions and applications for long-range interaction systems", Phys. Rev. A **81**, 062337 (2010).

[63] H. F. Trotter, "On the product of semi-groups of operators", P. Am. Math. Soc. **10**, 545–551 (1959).

[64] M. Suzuki, "Generalized Trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems", Commun. Math. Phys. **51**, 183–190 (1976).

[65] M. Suzuki, "Relationship between d-dimensional quantal spin systems and (d+1)-dimensional Ising systems: Equivalence, critical exponents and systematic approximants of the partition function and spin correlations", Prog. Theor. Phys. **56**, 1454 (1976).

[66] M. Suzuki, "General theory of fractal path integrals with applications to many-body theories and statistical physics", J. Math. Phys. **32**, 400–407 (1991).

[67] A. T. Sornborger and E. D. Stewart, "Higher-order methods for simulations on quantum computers", Phys. Rev. A **60**, 1956–1965 (1999).

[68] M. L. Wall and L. D. Carr, "Out-of-equilibrium dynamics with matrix product states", New J. Phys. **14**, 125015 (2012).

[69] N. Schuch, M. M. Wolf, F. Verstraete and J. I. Cirac, "Computational complexity of projected entangled pair states", Phys. Rev. Lett. **98**, 140506 (2007).

[70] H. C. Jiang, Z. Y. Weng and T. Xiang, "Accurate determination of tensor network state of quantum lattice models in two dimensions", Phys. Rev. Lett. **101**, 090603 (2008).

[71] M. Lubasch, J. I. Cirac and M.-C. Bañuls, "Unifying projected entangled pair state contractions", New J. Phys. **16**, 033014 (2014).

[72] I. Pižorn, L. Wang and F. Verstraete, "Time evolution of projected entangled pair states in the single-layer picture", Phys. Rev. A **83**, 052321 (2011).

[73] M. Lubasch, J. I. Cirac and M.-C. Bañuls, "Algorithms for finite projected entangled pair states", Phys. Rev. B **90**, 064425 (2014).

[74] M. Suzuki, "Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations", Phys. Lett. A **146**, 319–323 (1990).

[75] M. Suzuki, "Convergence of general decompositions of exponential operators", Commun. Math. Phys. **163**, 491–508 (1994).

[76] M. A. Cazalilla, R. Citro, T. Giamarchi, E. Orignac and M. Rigol, "One dimensional bosons: From condensed matter systems to ultracold gases", Rev. Mod. Phys. **83**, 1405–1466 (2011).

[77] M. Urbanek and P. Soldán, "Matter-wave revival of binary mixtures in optical lattices", Phys. Rev. A **90**, 033610 (2014).

[78] A. Kleine, C. Kollath, I. P. McCulloch, T. Giamarchi and U. Schollwöck, "Spin-charge separation in two-component Bose gases", Phys. Rev. A **77**, 013607 (2008).

[79] S. Will, T. Best, S. Braun, U. Schneider and I. Bloch, "Coherent interaction of a single fermion with a small bosonic field", Phys. Rev. Lett. **106**, 115305 (2011).

[80] J. Schachenmayer, A. J. Daley and P. Zoller, "Atomic matter-wave revivals with definite atom number in an optical lattice", Phys. Rev. A **83**, 043614 (2011).

[81] M. B. Hastings, "Light-cone matrix product", J. Math. Phys. **50**, 095207 (2009).

[82] M. L. Wall, *Quantum many-body physics of ultracold molecules in optical lattices: Models and simulation methods* (Springer, 2015), ISBN: 978-3-319-14252-4.

[83] U. R. Fischer and R. Schützhold, "Tunneling-induced damping of phase coherence revivals in deep optical lattices", Phys. Rev. A **78**, 061603 (2008).

[84] P. R. Johnson, E. Tiesinga, J. V. Porto and C. J. Williams, "Effective three-body interactions of neutral bosons in optical lattices", New J. Phys. **11**, 093022 (2009).

[85] S. Will, T. Best, U. Schneider, L. Hackermüller, D.-S. Lühmann and I. Bloch, "Time-resolved observation of coherent multi-body interactions in quantum phase revivals", Nature **465**, 197–201 (2010).

[86] F. A. Wolf, I. Hen and M. Rigol, "Collapse and revival oscillations as a probe for the tunneling amplitude in an ultracold Bose gas", Phys. Rev. A **82**, 043601 (2010).

[87] M. Buchhold, U. Bissbort, S. Will and W. Hofstetter, "Creating exotic condensates via quantum-phase-revival dynamics in engineered lattice potentials", Phys. Rev. A **84**, 023631 (2011).

[88] U. R. Fischer and B. Xiong, "Many-site coherence revivals in the extended Bose–Hubbard model and the Gutzwiller approximation", Phys. Rev. A **84**, 063635 (2011).

[89] Á. Perales and G. Vidal, "Entanglement growth and simulation efficiency in one-dimensional quantum lattice systems", Phys. Rev. A **78**, 042337 (2008).

[90] J. Hauschild, F. Pollmann and F. Heidrich-Meisner, "Sudden expansion and domain-wall melting of strongly interacting bosons in two-dimensional optical lattices and on multileg ladders", Phys. Rev. A **92**, 053629 (2015).

[91] M. Rigol and A. Muramatsu, "Fermionization in an expanding 1d gas of hard-core bosons", Phys. Rev. Lett. **94**, 240403 (2005).

[92] A. Minguzzi and D. M. Gangardt, "Exact coherent states of a harmonically confined Tonks–Girardeau gas", Phys. Rev. Lett. **94**, 240404 (2005).

[93] T. Kinoshita, T. Wenger and D. S. Weiss, "A quantum Newton's cradle", Nature **440**, 900–903 (2006).

[94] P. W. Anderson, "Absence of diffusion in certain random lattices", Phys. Rev. **109**, 1492–1505 (1958).

[95] E. Altman and R. Vosk, "Universal dynamics and renormalization in many-body-localized systems", Annu. Rev. Condens. Matter Phys. **6**, 383–409 (2015).

[96] R. Nandkishore and D. A. Huse, "Many-body localization and thermalization in quantum statistical mechanics", Annu. Rev. Condens. Matter Phys. **6**, 15–38 (2015).

[97] J.-y. Choi, S. Hild, J. Zeiher, P. Schauß, A. Rubio-Abadal, T. Yefsah, V. Khemani, D. A. Huse, I. Bloch and C. Gross, "Supplementary materials for Exploring the many-body localization transition in two dimensions", Science **352**, 1547–1552 (2016).

[98] G. Barcza, Ö. Legeza, K. H. Marti and M. Reiher, "Quantum-information analysis of electronic states of different molecular structures", Phys. Rev. A **83**, 012508 (2011).

[99] C. Krumnow, L. Veis, Ö. Legeza and J. Eisert, "Fermionic orbital optimization in tensor network states", Phys. Rev. Lett. **117**, 210402 (2016).

[100] S. V. Dolgov and D. V. Savostyanov, "Alternating minimal energy methods for linear systems in higher dimensions", SIAM J. Sci. Comput. **36**, A2248–A2271 (2014).

[101] C. Hubig, I. P. McCulloch, U. Schollwöck and F. A. Wolf, "Strictly single-site DMRG algorithm with subspace expansion", Phys. Rev. B **91**, 155115 (2015).

[102] D. Verna, "Beating C in scientific computing applications", in Third European Lisp Workshop at ECOOP (2006).

[103] D. Verna, "How to make Lisp go faster than C", IAENG International Journal of Computer Science **32**, 499–504, ISSN: 1819-656X (2006).

[104] *Steel Bank Common Lisp*, `http://sbcl.org/`.

[105] M. Urbanek, "Quantum physics simulations in Common Lisp", in Proceedings of the 8th European Lisp Symposium, edited by J. Padget (2015), pp. 71–78.

[106] *Basic Linear Algebra Subprograms (BLAS)*, `http://www.netlib.org/blas/`.

[107] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney and D. Sorensen, *LAPACK users' guide*, 3rd ed. (Society for Industrial and Applied Mathematics, Philadelphia, 1999), ISBN: 0-89871-447-8.

[108] G. Alvarez, "The density matrix renormalization group for strongly correlated electron systems: A generic implementation", Comput. Phys. Commun. **180**, 1572–1578 (2009).

[109] G. Alvarez, "Implementation of the SU(2) Hamiltonian symmetry for the DMRG algorithm", Comput. Phys. Commun. **183**, 2226–2232 (2012).

[110] S. Wouters, W. Poelmans, P. W. Ayers and D. Van Neck, "CheMPS2: A free open-source spin-adapted implementation of the density matrix renormalization group for ab initio quantum chemistry", Comput. Phys. Commun. **185**, 1501–1514 (2014).

[111] M. Dolfi, B. Bauer, S. Keller, A. Kosenkov, T. Ewart, A. Kantian, T. Giamarchi and M. Troyer, "Matrix product state applications for the ALPS project", Comput. Phys. Commun. **185**, 3430–3440 (2014).

[112] *Algorithms and Libraries for Physics Simulations (ALPS)*, `http://alps.comp-phys.org/`.

[113] *BLOCK*, https://sanshar.github.io/Block/.

[114] *CheMPS2*, https://github.com/SebWouters/CheMPS2.

[115] *DMRG Applet*, http://www.theorie.physik.uni-goettingen.de/~thomas.koehler/.

[116] *DMRG++*, https://g1257.github.io/dmrgPlusPlus/.

[117] *evoMPS*, https://amilsted.github.io/evoMPS/.

[118] *Intelligent Tensor (ITensor)*, http://itensor.org/.

[119] *Matrix Product States*, https://github.com/juanjosegarciaripoll/mps.

[120] *Matrix Product Toolkit*, https://people.smp.uq.edu.au/IanMcCulloch/mptoolkit/.

[121] *Open Source Matrix Product States (OpenMPS)*, https://sourceforge.net/projects/openmps/.

[122] *Open Source Time-Evolving Block Decimation (OpenTEBD)*, https://sourceforge.net/projects/opentebd/.

[123] *POWDER with Power*, http://qti.sns.it/dmrg/phome.html.

[124] *Simple DMRG*, https://github.com/simple-dmrg/simple-dmrg.

[125] *Snake DMRG*, https://github.com/entron/snake-dmrg.

[126] *Universal Tensor Network Library (Uni10)*, https://yingjerkao.github.io/uni10/.