Symmetry-Protected Tensor Networks

Dissertation der Fakultät für Physik
der Ludwig-Maximilians-Universität München

Vorgelegt von
Claudius Hubig
aus
Berlin-Charlottenburg
am
31. August 2017

Erstgutachter: Prof. Dr. Ulrich Schollwöck
Zweitgutachter: Prof. Dr. Norbert Schuch
Datum der mündlichen Prüfung: 30. Oktober 2017

Deutschsprachige Zusammenfassung

Die Simulation und numerische Untersuchung großer, stark korrelierter, endlichdimensionaler Quantensysteme, welche Fermionen enthalten oder einer Realzeitentwicklung unterzogen werden sollen, ist noch immer ein faktisch ungelöstes Problem, hauptsächlich bedingt durch das exponentielle Anwachsen der Dimensionen des Hilbertraumes der Quantenzustände und das Auftreten des Vorzeichenproblems in Monte-Carlo-Rechnungen. In diesem Gebiet hat die Benutzung von Tensornetzwerkmethoden, im eindimensionalen Fall primär die Dichtematrixrenormalisierungsgruppe (DMRG) und Matrixproduktzustände (MPS), in den letzten Jahren erheblich an Bedeutung gewonnen.

Diese Arbeit fast zuerst mittels eines umfassenden Berichts über den aktuellen Forschungsstand in der veröffentlichen Literatur sowie konkreter Algorithmen und Implementierungshilfen die Benutzung nichtabelscher Symmetrien, wie zum Beispiel $\mathrm{SU}(2)_{\mathrm{Spin}}$, in beliebigen Tensornetwerken zusammen. Die Implementierung solcher Symmetrien kann zu einer erheblich effizienteren Repräsentation von Zuständen im Tensornetzwerk führen. Dieser Teil der Arbeit ist auch als implementierungsorientierte Einführung in Tensornetzwerke sowie die Benutzung nichtabelscher Symmetrien in denselben gedacht.

Als Zweites wird eine Reihe technischer Verbesserungen an MPS-Methoden vorgestellt. Hier sind vor allem ein schnelleres Konvergenzschema für DMRG, ein systematischer Ansatz für die Konstruktion von Matrixproduktoperatoren und eine verbesserte Krylov-Zeitentwicklungsmethode sowie die Kombination einiger anderer allgemein bekannter Techniken in einem umfassenden Tensornetzwerktoolkit, SyTen, zu nennen. Der Erfolg dieser Verbesserungen wird an zahlreichen numerischen Beispielen demonstriert.

Zum Dritten wird dieses Toolkit bei der Untersuchung zweier Modelle, die aktuell allgemeiner Forschungsgegenstand sind, eingesetzt: Eine eindimensionale Spinkette in einem alternierenden externen magnetischen Feld wird untersucht und das durch analytische Argumente vorhergesagte Confinement der elementaren Spinon-Anregungen mittels Realzeitentwicklung und numerischer Auswertung des dynamischen Strukturfaktors festgestellt. Weiterhin wird das Hubbard-Modell in zwei Dimensionen ausführlich bei verschiedenen Systemgrößen, Gittergeometrien, Wechselwirkungsstärken U und Elektronendichten n mit bis zu 30 000 $\mathrm{SU}(2)_{\mathrm{Spin}}$-symmetrischen Zuständen – ungefähr äquivalent zu 100 000 Zuständen in anderen Implementierungen – untersucht. Hinweise auf eine mögliche Phasenkoexistenz zwischen $0{,}85 \lesssim n \lesssim 0{,}95$ wurden bei mittlerer Wechselwirkungsstärken $U = 4$ und $U = 6$ gefunden. Weiterhin wurden konsistent kurzperiodische Modulationen der Elektronendichte im Grundzustand bei $n \approx 0{,}875$ festgestellt.

Abstract

The simulation and numerical study of large, strongly correlated quantum systems containing Fermions or using real-time evolution in finite dimensions is still an essentially unsolved problem, primarily due to the exponential growth of the Hilbert state space with system size and the occurence of the so-called sign problem in Monte Carlo studies. In this area, the use of tensor-network methods, for one-dimensional systems chief among them the density matrix renormalisation group (DMRG) and matrix-product states (MPS), has grown in importance in recent years.

This thesis first recapitulates the use of non-abelian symmetries such as $\mathrm{SU}(2)_{\mathrm{Spin}}$ in arbitrary tensor networks with an extensive review of the published literature including detailed algorithms and implementation hints. Implementing such symmetries can lead to a considerably more efficient representation of states in the tensor network. This part is intended to be suitable as an implementation-oriented introduction to tensor networks in general and the implementation of non-abelian symmetries in particular.

Second, it introduces a series of technical improvements for the MPS methods. These improvements include a faster convergence scheme for MPS-DMRG, a systematic approach to the construction of matrix-product operators and an improved Krylov time evolution method as well as the combination of several well-known techniques into a single tensor network toolkit, SyTen. The effectiveness of these improvements is demonstrated in numerical examples.

Third, the toolkit is applied to the study of two models of current research interest: A one-dimensional spin chain in a staggered external magnetic field is studied and confinement of the elementary spinon excitations, as predicted by analytical arguments, found numerically using real-time evolution and evaluation of the dynamical structure factor. Additionally, the Hubbard model in two dimensions is studied extensively at various system sizes, geometries, interaction strengths U and filling factors n using up to $30'000$ $\mathrm{SU}(2)_{\mathrm{Spin}}$-symmetric states equivalent to approx. $100'000$ states in other MPS-DMRG implementations. Hints of a possible phase coexistence in the region $0.85 \lesssim n \lesssim 0.95$ are found at intermediate interaction strengths $U = 4$ and $U = 6$ as well as a consistently striped ground state in the region $n \approx 0.875$.

Contents

List of Publications

Parts of the content of this thesis have been published before or will be published soon. Specifically:

1. The contents of Secs. 2.4.4 and 2.4.5 on matrix-product operator compression routines and numerical examples have been published in Ref. [1].

2. The DMRG3S method explained in Sec. 3.4 and in particular the numerical experiments in Sec. 3.4.2 have been published in Ref. [2].

3. The improvements to the Krylov time evolution method presented in Sec. 4.1.4 have been developed in close collaboration with Andreas Swoboda and Nils Linden and are currently being prepared for publication.[3]

4. The results on spinon confinement in one-dimensional spin chains presented in Sec. 4.2.1 have been contributed to Ref. [4].

5. The project on the Hubbard model in two dimensions (Chapter 5) was conducted in close collaboration with Leo Stenzel and parts of the data analysed in Sec. 5.4 was also analysed by him in Ref. [5].

The SyTen tensor networks library developed during the work on this thesis with contributions by Andreas Swoboda, Felix Lachenmaier, Leo Stenzel, Nils-Oliver Linden and Teresa Reinhard was already used to produce data contained in the following works:

1. The numerical experiments in Sec. VII of Ref. [1] (cf. Sec. 2.4.5 here).

2. The numerical experiments on time evolution methods conducted by Andreas Swoboda and to be published in Ref. [3] (cf. Sec. 4.1.4 here).

3. The DMRG and finite MPS-time evolution data in Ref. [4] (cf. Sec. 4.2.1 here).

4. Leo Stenzel's Master thesis on the Hubbard model, Ref. [5] (cf. Chapter 5 here).

5. Felix Lachenmaier's Bachelor thesis on a parallelised version of the time-evolution block decimation method, Ref. [6] (cf. Sec. 4.1.2 here).

6. Moritz Hahn's Master thesis on the Kagomé lattice, Ref. [7].

1 Introduction

The study of strongly correlated quantum systems has seen a major surge in importance in recent years for many reasons. Among them are the still-unsolved high-temperature superconductors, improvement of experimental techniques whose theoretical explanations require going beyond mean-field approximations and the novel field of cold atomic gases which allow for the quantum simulation of nearly entirely arbitrary systems. Correspondingly, there have been many improvements to the numerical techniques used to study such systems. However, the two most reliable and likely oldest numerical techniques for the study of quantum-mechanical systems, exact diagonalisation and Monte Carlo sampling, face two major difficulties when confronted with large strongly correlated systems: the former fails to obtain system sizes larger than the correlation length due to the exponentially increasing dimension of the Hilbert state space, the latter is susceptible to the sign problem occuring in the study of Fermionic systems or even real-time evolution of otherwise harmless bosonic systems.

Due to this shortcoming of the established methods, the use of tensor network algorithms has grown in importance. Tensor networks are a way to represent lowly-entangled states efficiently. This is done by a series of truncating basis transformations which result in bases for parts of the system in which the state in question can be represented most efficiently. Effectively, these transformations discard all unneeded parts of the Hilbert spaces of subsystems, leading to highly efficient representations of states whose cost only grows polynomially in system size, at least for such lowly-entangled states.

Thankfully, the *area law* of entanglement was proven to hold for ground states of local gapped Hamiltonians in one dimension[8] and with a logarithmic correction[9] also for ground states of critical Hamiltonians in one dimension. This area law states that the entanglement of a subsystem with the remainder of the larger system grows linearly in the *surface area* of the subsystem, not its volume. Practically, this means that in one dimension the entanglement of a subsystem with the remainder of a system eventually saturates, as its surface area is constant. While no results of similar strength are known for higher-dimensional systems, area laws are either proven to hold in some cases or assumed to (nearly) hold in others.

The tensor networks exploiting this area law will be introduced in Chapter 2 with primarily the one-dimensional case, where tensor network states and operators are called *matrix-product states* (MPS) and *matrix-product operators* (MPO), in mind. The focus will be on the implementation and use of arbitrary global symmetries of the system

which can be used to great effect in increasing computational efficiency. Furthermore, the contents of an earlier paper[1] on the generic construction of efficient MPOs will be re-presented.

Chapter 3 is a review of the *density-matrix renormalisation group* (DMRG) method which was the first permanent introduction of tensor network methods to the field of condensed matter physics and is one of the most efficient approaches to calculating ground states of one-dimensional quantum systems currently available. The method will be reviewed in detail including recent extensions, among them the subspace expansion method presented in a first paper.[2]

Chapter 4 reviews and discusses available time-evolution methods for MPS and their application to calculate Green's functions, dynamical structure factors and time-dependent observables. Four improvements to the Krylov method, subject of an upcoming paper,[3] will be presented. As an example application, we calculate the dynamical structure factor in a one-dimensional spin chain embedded into an external magnetic field representing a mean-field coupling to other chains in three-dimensional space. The associated data was contributed to a recent paper.[4]

Finally, Chapter 5 discusses the application of hybrid-space[10–12] DMRG to the two-dimensional Hubbard model. Via the implementation of non-abelian symmetries, we were able to reach much larger numbers of states than previous works (by a factor of approx. 3). Unfortunately, this was not sufficient to converge the system on cylinders of width substantially larger than previous calculations. However, we were able to confirm previous findings and provide some hints towards a possible phase coexistence region.

2 Tensor Networks

We will start with a very brief account of the history of the density matrix renormalisation group and tensor network methods (Sec. 2.1), followed by an equally brief summary of the relevance of symmetries and so-called "good quantum numbers" in the field of condensed matter physics (Sec. 2.1.1). Sec. 2.2 will continue with an introductory review to symmetry-protected tensor networks that can also serve as a motivation for the SyTen tensor network toolkit which was created during this PhD thesis.

With this powerful tool in hand, it is possible to define matrix-product states (Sec. 2.3) and matrix-product operators (Sec. 2.4) as the natural tensor network representations of states and operators on decomposable finite Hilbert spaces of one-dimensional quantum-mechanical systems. In particular, we will present a generic algorithm for the construction of efficient Matrix Product Operator representations, published as part of this thesis in Ref. [1].

2.1 Introduction

After the introduction of the *density matrix renormalisation group* (DMRG) by Steve White in the early 1990s,[13] this original, infinite-system DMRG algorithm soon grew to importance as the method of choice to find the ground states of one-dimensional quantum systems. The method was quickly expanded to handle not only infinite systems but also finite systems and in particular long-range correlations by first growing the system using the infinite method and then iteratively sweeping over the finite system of the desired size until convergence has been achieved. For a more detailed review of this relatively early work, see for example Ref. [14].

In the late 1990s, it was then realised[15,16] that DMRG is in fact a variational method over the set of *matrix-product states* (MPS, to be defined in Sec. 2.3), which can arguably be seen as an example of the simplest useful tensor network. Concepts very similar to MPS have been used in the numerical linear algebra community under the name of tensor trains[17] and also appeared on multiple occasions throughout the 20th century.[18–20] Since around 2005,[21,22] MPS-based DMRG has become more and more popular and can certainly be called the state of the art for ground-state search on one-dimensional quantum systems nowadays. Its various variants and implementation details will be discussed in more detail in Chapter 3.

It is also possible to write down one-dimensional MPS defined on infinite-size systems

with a finite unit cell length, named *infinite matrix-product states* (iMPS)[23] or *variational uniform matrix-product states* (VUMPS).[24] Either of those approaches then allows the variational ground-state search on an infinite system at costs very comparable to those of finite-size DMRG.

Furthermore, with the help of the tensor network view of DMRG, there are not one but two possible avenues to tackle two-dimensional problems: First, if the two-dimensional system is finite, it is always possible to map it to a one-dimensional system with longer-range interactions.[25] Second, one can write down an inherently two-dimensional tensor network description of a quantum-mechanical state, called *projected entangled pair state*[26] (PEPS) and then attempt to act on this state either via variational optimisation[27] or imaginary time-evolution[28] to find the ground state. The latter method also allows the study of infinite systems (though with finite unit cell) in a somewhat less one-dimensionally biased way in the form of iPEPS.[29,30] However, both approaches suffer from a very large computational cost compared to one-dimensional calculations.

Finally, the combination of global symmetries and tensor networks is a topic of ongoing research. Previous works in this area include the initial introduction of non-abelian SU(2) symmetries in the original DMRG,[31,32] the translation of that formalism to MPS-based DMRG,[22] the application of abelian U(1) and $\mathbb{Z}_k$ symmetries to finite and infinite PEPS,[33–35] and finally the first descriptions of generic tensor networks with non-abelian symmetries.[36,37]

2.1.1 Symmetries

Before introducing symmetry-protected tensors, it is useful to motivate the use of symmetries in tensor networks and fix some terminology.

Assume a finite-dimensional Hilbert space $\mathcal{H} \sim \mathbb{C}^N$ representing a quantum-mechanical state space, a Hamiltonian $\hat{H} : \mathcal{H} \to \mathcal{H}$ representing our Hamiltonian of choice on that space and a computational basis $\{|\sigma_i\rangle\}_{i=1}^N$ spanning $\mathcal{H}$. Throughout this work, $\{A_i\}_{i(=1)}^{(N)}$ will denote the set of i-indexed elements A_i where the range of i (here, $[1, \ldots, N]$) is either clear from context or explicitly given. $\{|\sigma_i\rangle\}_{i=1}^N$ hence stands for the set $\{|\sigma_1\rangle, |\sigma_2\rangle, \ldots, |\sigma_N\rangle\}$.

Then, if there is an operator $\hat{O} : \mathcal{H} \to \mathcal{H}$ s.t. $[\hat{H}, \hat{O}] = 0$, $\hat{H}$ and $\hat{O}$ can simultaneously be diagonalised and have a simultaneous eigenbasis $\{|e_i\rangle\}_i$. In particular, this means that we can assign each eigenstate[i] $|e_i\rangle$ of $\hat{H}$ a single eigenvalue O_i of $\hat{O}$. Further, we know that this eigenvalue of $\hat{O}$ is preserved under application of $\hat{H}$. This by itself is not very useful, it only tells us that $\hat{H}$ in the eigenbasis $\{|e_i\rangle\}_i$ is block-diagonal – but of course $\hat{H}$ in this basis is already diagonal!

However, now assume that the states $\{|\sigma_i\rangle\}_i$ are also eigenstates of $\hat{O}$. For complicated

[i] In the case of degeneracies, it may be necessary to first select suitable linear combinations of eigenvectors of $\hat{H}$.

$\hat{O}$, this is not true in general, but for very simple operators $\hat{O}$, it is often the case. If this is indeed so, we can also label the states $\{|\sigma_i\rangle\}_i$ by those eigenvalues O_i – since $\hat{H}$ *still* preserves them, we can then re-order $\hat{H}$ to bring it into a block-diagonal form in our original computational basis. If we were then to attempt a brute-force diagonalisation of $\hat{H}$, we would not have to take into account the full $N \times N$ matrix representing $\hat{H}$ but only small blocks of that matrix.[38] Further, we can fix the desired eigenvalue of $\hat{O}$ and concentrate on that particular block.

In this case, we say that there are *good quantum numbers* $\{O_i\}_i$ by which we can *label* the states $\{|\sigma_i\rangle\}_i$. The Hamiltonian $\hat{H}$ is said to *preserve* the quantum numbers if it commutes with $\hat{O}$. It is useful to keep in mind that, from the point of view of representation theory, this implies that there is some group $\mathcal{G}$ (usually SU(N), U(1) or $\mathbb{Z}_k$ in the context of condensed-matter physics) and that we can associate each state of $\mathcal{H}$ to an irreducible representation (IREP) of that group. The label of that IREP then coincides with the quantum number assigned to this state.[31,32,37]

A Brief Summary of Representation Theory

This section does not attempt to give a full introduction to (Lie) groups and representation theory. In any case, such an introduction is also not necessary to follow the next sections. However, it is certainly useful to define some basic concepts which may escape the memory of the working physicist over time.

A *group* $\mathcal{G}$ is a set of objects with a binary operation $\circ$ satisfying certain properties. Those properties are closure of the set under the operation $\circ$, associativity of the operation and the existence of an identity element and an inverse element. For example $\{\mathbb{R}, +\}$ is a group. $\{\mathbb{R}, \times\}$ is not, since 0 has no inverse element. $\{\mathbb{Z}, +\}$ is a group, with n the inverse element associated to $-n$, 0 the identity element and obvious closure under the operation.

A specific group is then given by the relations of its group elements (i.e. which elements, when combined with others under the binary operation, result in a specific third element).

It is possible to consider *matrix groups*, that is, groups which are defined by a set of matrices and matrix multiplication as the group operation. Matrix groups have some very favourable properties. The simplest example of a matrix group is possibly the general linear group GL($n, \mathbb{R}$) which contains all $n \times n$ invertible real matrices. There are sub-groups of GL($n, \mathbb{R}$): for example, the group of orthogonal $n \times n$ real matrices O(n) as well as SO(n) for orthogonal matrices with determinant 1. Similarly, U(n) is the group of unitary matrices of size $n \times n$, with a sub-group SU(n) containing only those unitary matrices with determinant 1. Crucially, the group of rotations in three-dimensional space around a single axis can be associated to the group SO(2), while SO(3) corresponds to the set of rotations around all axes in three-dimensional space.

On the other hand, a *matrix representation* of a group is a set of matrices which satisfy

the group relations with matrix multiplication taking the place of the group operation. That is, for group elements g_1, g_2 and g_3 with $g_1 \circ g_2 = g_3$ and a matrix representation which associates matrices M_1, M_2 and M_3 to those group elements, we have $M_1 \cdot M_2 = M_3$ where $\cdot$ is the matrix multiplication operation. A representation is called faithful if it associates a unique M_i to each g_i. The trivial representation $M_i = 1$ always exists but is not necessarily faithful.

Given a matrix group, e.g. SU(2), it is possible to take the matrices[ii] of that group 1-to-1 as an initial representation of that group. It is then possible to find *larger* matrices which still behave as described by SU(2) and also to find the smallest faithful matrix representation, i.e. the smallest matrices with a 1-to-1 mapping to and from the group elements. This smallest representation is called the fundamental representation of the group. In some cases, it is possible to decompose the matrices of a representation into invariant sub-blocks (e.g. if all of them are block-diagonal and hence don't mix certain parts of the space on which they act). In this case, the representation is called *decomposable* and can be written as the direct sum of two other (non-trivial) representations. On the other hand, if it is not possible to do so, the representation is called *indecomposable*. For most practical purposes, representations which are indecomposable are also *irreducible* (hence the name IREP above) and vice-versa.

Finally, in the case of matrix groups (and equally, matrix representations of linear groups), it is possible to consider them as smooth manifolds generatable by exponentiation of a linear combination of a finite set of *generators*. For example, all unitary 2×2 matrices with determinant 1 can be generated by appropriate choice of prefactors α_k in $\exp\left(i \sum_k \alpha_k \sigma_k\right)$, where σ_k are the Pauli matrices. This set of generators is not unique and it may be helpful to choose another set of 2×2 matrix generators instead (for example the typical spin operators $S^{\pm,z}$).

These matrix generators then fulfill certain commutation relations closely linked to the group properties and identical with the commutation relations of the physical generators of the associated physical symmetry (e.g. rotation in three dimensions generated by angular momentum). Those commutations relations then define a *Lie algebra*, which we can think of as the infinitesimal version of the Lie group. Generators of other IREPs of the same group which are associated to the same algebra fulfill the same commutation relations: For example, the generators of the 3×3 matrix IREP of SU(2) have the exact same commutation relations as the 2×2 generators.

Now in order to define a certain representation of a group, it is only necessary to store the generators of that representation (which are usually a small number of matrices). For example, to describe a particular representation of $\mathrm{SU}(2)_{\mathrm{Spin}}$, it is sufficient to store three matrices: The S^z, S^+ and S^- operators. All representations of other elements of the $\mathfrak{su}(2)$ algebra can be written as linear combinations of those, while the representations of

[ii] In the case of SU(2), those would be the matrices $\begin{pmatrix} \alpha & -\beta^\star \\ \beta & \alpha^\star \end{pmatrix}$ with $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$.

elements of the group can be achieved via exponentiation.

2.2 Symmetry-Protected Tensors

The reasoning for the introduction of symmetry-protected tensors in DMRG or related applications closely follows the argument given above for the case of exact diagonalisation: Given an operator $\hat{O}$ which commutes with the Hamiltonian, it should be possible to find a relatively simple computational basis for the Hilbert space which only contains eigenstates of $\hat{O}$. Furthermore, if we wish to find the ground-state of the Hamiltonian, we can be certain that we will not have to mix states with different quantum numbers later on, i.e. that the blocking of the computational basis is preserved. The following sections will provide a generic introduction to tensors as used in condensed-matter physics, the implementation of exactly-protected symmetries on such tensors and some basic tensor operations required later on.

2.2.1 Definition and Graphical Representation

For now, define a tensor T as a map from a set of *input vector spaces* $\{\mathbb{K}_a^{N_a}\}_a$ with dimensions $\{N_a\}_a$ to linear combinations of elements of a set of *output vector spaces* $\{\mathbb{K}_b^{N_b}\}_b$ with dimensions $\{N_b\}_b$. The coefficients of this map are given by the elements of the tensor. $\mathbb{K}$ is usually $\mathbb{C}$, but a restriction to $\mathbb{R}$ may be possible and is computationally cheaper. The differentiation between input and output vector spaces and indices is relevant. For example, consider three input vector spaces A, B, C and two output vector spaces D and E. To write down the tensor, we have to fix a basis in each space, let those be $\{|a\rangle_A\}_{a=1}^{N_A}$, $\{|b\rangle_B\}_{b=1}^{N_B}$ etc. The tensor with coefficients T_{abc}^{de} then describes the map:

$$T : A \otimes B \otimes C \to D \otimes E \tag{2.2.1}$$

$$T : |a\rangle_A \otimes |b\rangle_B \otimes |c\rangle_C \mapsto \sum_{de} T_{abc}^{de} |d\rangle_D \otimes |e\rangle_E \quad . \tag{2.2.2}$$

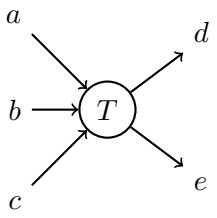


Figure 2.1: The tensor T from Eq. (2.2.2) represented graphically. It maps states $|a\rangle_A \otimes |b\rangle_B \otimes |c\rangle_C$ from the input spaces A, B, C to linear combinations of states $|d\rangle_D \otimes |e\rangle_E$ in the output state spaces D, E.

We can graphically represent this tensor (cf. Fig. 2.1) by a blob with five legs corresponding to each of the five vector spaces (and associated indices) with arrows indicating the direction (input and output). T is said to have *rank* five, with for example the a leg (or a index) having *dimension* $\dim(A)$.[iii]

It is useful to assume that neither of the two sets of input and output state spaces is empty, i.e. that there is at least one input state space and at least one output state space. If this is not the case, one should add a single 1-dimensional input (or output) leg. This is equivalent to viewing a column vector $v \in \mathbb{C}^N$ as a matrix $v' \in \mathbb{C}^{N\times 1}$.

2.2.2 Physical Symmetries in Tensors

Abelian Symmetries

Let us continue with the previously-defined tensor T. Now assume that there is some symmetry with associated group $\mathcal{G}$ (e.g. U(1) or SU(2)) such that we can identify each of the states $|a\rangle_A$ etc. in A, B, C, D and E with an IREP a, b, c, d or e of that group $\mathcal{G}$. This means that firstly the states $|a\rangle_A$ etc. do not break the symmetry and that we can secondly assign a single quantum number to each of those states. Even if a state does not break the symmetry, the second requirement does not have to hold: For example, it is not possible to assign a single quantum number to a superposition of two states with different quantum numbers. In this case, one first has to decompose the not-uniquely-transforming state into a direct sum (i.e. superposition) of states which do transform uniquely. Note that of course different states $|a\rangle_A$ and $|a'\rangle_A$ may share the same quantum number $a = a'$, dealing with such a degeneracy is no problem. An example is the abelian $\mathrm{U}(1)_{S^z}$ rotational symmetry, which allows us to label states $|a\rangle_A$ etc. by their S^z quantum numbers.

In the case of an abelian group $\mathcal{G}$, the tensor T is then said to preserve the symmetry if the statement

$$d \otimes e \notin a \otimes b \otimes c \Rightarrow T^{de}_{abc} \equiv 0 \tag{2.2.3}$$

holds, where $\otimes$ means the *product* of two representations and $\notin$ denotes that the set of IREPs arising from the decomposition of the left operand and that arising from the right operand have no elements in common.

For U(1) symmetries and quantum numbers, i.e. IREP labels, q_a, q_b, q_c, q_d and q_e one can simplify this to the statement[22]

$$q_d + q_e \neq q_a + q_b + q_c \Rightarrow T^{de}_{abc} \equiv 0 \quad . \tag{2.2.4}$$

In the case of $\mathrm{U}(1)_{S^z}$, this means that we can map input states with total quantum number $S^z_i = S^z_a + S^z_b + S^z_c$ only to output states with total quantum number $S^z_o = S^z_d + S^z_e$

[iii] This is in contrast to the numerical linear algebra community, where T has *order* five and the a leg or a index has *size* $\dim(A)$. In this context, T has *rank* r if it can be expressed as the sum over r outer products of five vectors.

equal to S_i^z.

For $\mathbb{Z}_k$ symmetries, we also have to take the periodicity with respect to k into account:

$$(q_d + q_e) \bmod k \neq (q_a + q_b + q_c) \bmod k \Rightarrow T^{de}_{abc} \equiv 0 \quad . \tag{2.2.5}$$

Here, similarly, we can map for example a parity-odd input state only to a parity-odd output state.

Effectively, these requirements can be used to reduce the in-memory size of the tensor: After sorting the input and output states by their quantum numbers, we can write the tensor as a sum of dense blocks which are allowed to be non-zero. Tensor operations (defined later) then act sequentially on these dense blocks instead of the whole tensor.

Non-Abelian Symmetries

For symmetries with an associated non-abelian group, e.g. the total spin S with group SU(2), an additional concern arises regarding the correct treatment of the inner multiplicities of each IREP. For example, the IREPs of SU(2) with non-zero quantum numbers S' not only contain one overall state S' but actually $2S'+1$ sub-states with different S^z quantum numbers. In addition to the requirement posed by Eq. (2.2.3), these "sub-states" also have to be treated correctly in order to preserve the overall non-abelian symmetry.[31,34,37]

Let us first look at the former requirement. Assume that we have input states with quantum numbers of total spin $S_a = \frac{1}{2}$, $S_b = \frac{1}{2}$ and $S_c = 1$. The product of these IREPs can be decomposed as[39]

$$\begin{aligned} &\frac{1}{2} \otimes \frac{1}{2} \otimes 1 && (2.2.6)\\ &= (1 \oplus 0) \otimes 1 && (2.2.7)\\ &= (1 \otimes 1) \oplus (0 \otimes 1) && (2.2.8)\\ &= (2 \oplus 1 \oplus 0) \oplus 1 \quad . && (2.2.9) \end{aligned}$$

It is now obvious that if the product of the output IREPs contains a $S = \frac{1}{2}$ IREP, there is no way to map to this state given the above input and preserving the SU(2) symmetry.

The second requirement concerns the inner multiplicity[37] of each state. Each state with non-zero total spin S is actually composed of $2S + 1$ states with different S^z quantum numbers, which, by the SU(2) rotational symmetry, are required to behave identically. Other non-abelian symmetries (e.g. spatial rotation SO(3)) behave in much the same way. To preserve this symmetry, our tensor T now has to map each sub-state of e.g. the input $S = 1$ state to its corresponding partner in the output $S = 1$ state (this is essentially Schur's lemma). Equivalently, one could require that not only the "visible" quantum numbers associated to the group labels are preserved but also their hidden subdivisions, e.g. the $S^z = \pm 1, 0$ labels hidden below the $S = 1$ label.

The second requirement can be used to reduce the size of each dense block of the tensor by decomposing it into a reduced, dense block and a set of very sparse tensors, one for each symmetry in the system. The full block is then given as the tensor product of the dense block and the symmetry-defined blocks.

2.2.3 Construction and Implementation of Symmetry-Protected Tensors

The implementation of symmetries allows two decompositions. It becomes possible to decompose the full tensor T as a sum of dense blocks by the use of abelian symmetries and their good quantum numbers. That is, we can write:

$$T = \bigoplus_{i \in \text{blocks}} b_i \quad . \tag{2.2.10}$$

where b_i is a dense block which has a single good quantum number per symmetry on each tensor leg (hence transforms uniquely) and the $\bigoplus$ denotes a tensor-sum: Two blocks with different quantum numbers are placed "next to each other", since they relate to different states, while blocks with the same quantum number on a given leg relate to the same states on that leg. In particular, blocks with identical quantum numbers on all legs should be understood as being added together directly. Furthermore, for IREPs of non-abelian symmetries with inner multiplicity not equal to one we can decompose every block b_i as the tensor product of a reduced, dense block m_i and a set of symmetry-defined, usually very sparse, tensors $\{c_i^\gamma\}_\gamma$, where γ runs over the list of symmetries defined in the system (e.g. one SU(2) for particle-hole symmetry and one SU(2) for total spin).

In total, one can then write

$$T = \bigoplus_{i \in \text{blocks}} \left(m_i \bigotimes_\gamma c_i^\gamma \right) \tag{2.2.11}$$

To construct such a symmetry-protected tensor, first consider a rank-2 tensor with one input and one output space. In this case, we know that the input and output quantum numbers on each block must coincide, the resulting tensor is block-diagonal. Further, since the IREPs must coincide, each inner state of a non-abelian input IREP must be mapped to its exact equivalent in the output IREP, i.e. the c_i^γ must be proportional to identity matrices of the size of the corresponding IREP. Note that there is a freedom in the choice of prefactors: multiplying a certain m_i by a scalar α preserves the overall tensor value if one of its partners c_i^γ is multiplied by $1/\alpha$. Additionally, for consistency it is useful to also include the c_i^γ tensors even for IREPs with inner multiplicity equal to one, e.g. $S = 0$. In those cases, the c_i^γ tensors are simply singleton values of appropriate rank.

For example, assume that we have a rank-2 tensor with a single input and a single output index as well as a single SU(2) symmetry, namely the total spin, with quantum numbers $S = 0, \frac{1}{2}$ and $S = 1$ occuring on the input and output index. These quantum

numbers can then be used to label the blocks as $b_{0,0}$, $b_{\frac{1}{2},\frac{1}{2}}$ and $b_{1,1}$. A block $b_{0,\frac{1}{2}}$ cannot appear, as it would break the SU(2) symmetry. The tensor is hence

$$T = b_{0,0} \oplus b_{\frac{1}{2},\frac{1}{2}} \oplus b_{1,1} = \begin{pmatrix} b_{0,0} & 0 & 0 \\ 0 & b_{\frac{1}{2},\frac{1}{2}} & 0 \\ 0 & 0 & b_{1,1} \end{pmatrix} \quad , \tag{2.2.12}$$

where each $b_{i,i}$ is a dense matrix $\mathbb{K}^{N_i \times N_i'}$ of the appropriate size and content which implements the tensor map. However, now also exploiting the known inner structure of each IREP, namely that it treats each S^z value identically due to the rotational symmetry, we can rewrite each $b_{i,i}$ as a tensor product

$$b_{i,i} = m_{i,i} \bigotimes \mathbf{1}^{d_i \times d_i} \tag{2.2.13}$$

with the identity matrix $\mathbf{1}^{d_i \times d_i}$ and $d_i = 2S + 1$ in this case being the inner multiplicity of the associated IREP. Each of those d_i states of $\mathbf{1}^{d_i \times d_i}$ corresponds to one S^z value. Each column of $m_{i,i}$ corresponds to one of $\mathrm{cols}\,(m_{i,i})$ degenerate states with the same S quantum number and which – precisely due to the rotational $\mathrm{SU}(2)_{\mathrm{Spin}}$ symmetry – actually occurs d_i times.

For the $S = 0$ case, nothing changes, but for the other two, the side lengths of the dense matrices are reduced to a half and a third of their original size respectively at the cost of a 2×2 and 3×3 identity matrix! This already demonstrates why the explicit use of such non-abelian symmetries is so interesting in tensor network methods.

After the construction of the rank-2 tensor, it is natural to continue with the rank-3 tensor. In this case, the c^γ coincide exactly with the Clebsch-Gordan coefficients (CGC), as they describe valid combinations of two input states into one output state (or vice versa). There are then two possibilities of implementation: Either one can rely on the fact that closed expressions for these coefficients are known exactly for SU(2). This works for both the original DMRG and MPS-DMRG and has been the first method by which non-abelian symmetries have been implemented in DMRG in 2002 by Ian McCulloch.[31] While MPS-DMRG can be expressed purely in terms of rank-3 tensors, higher-dimensional tensor networks become somewhat more difficult to implement.[36] Furthermore, groups for which appropriate closed expressions for these coefficients are not known (e.g. SU(3) or Sp($2n$)) cannot be used.

Alternatively, one can calculate the coefficients explicitly from the decomposition of product representations and then handle those coefficients during all tensor operations. This second approach was first suggested by Andreas Weichselbaum in 2012.[37] SyTen also follows this approach. Unless the reduced tensors are very small, the overhead from the additional, sparse symmetry tensors c^γ is negligible in practice, while they at the same time allow for much greater flexibility during tensor products, the implementation of

other symmetries and higher-rank tensors without an explicit decomposition into rank-3 tensors, as was suggested to be necessary by Sukhwinder Singh et al.[36]

Calculation of Rank-3 Clebsch-Gordan Coefficients

This subsection describes how to build the product representation of two IREPs and how to decompose that product representation into a series of irreducible representations. As a by-product of this decomposition, the Clebsch-Gordan coefficients (CGC) for the tensor decomposition above are calculated. The following is fairly technical and likely only required if one wishes to implement this decomposition oneself. The presented algorithm closely follows the diagram given in Appendix B of Ref. [37].

The calculation requires the raising (R), lowering (L) and z-operators (Z), i.e. the generators of the algebra, of the involved IREPs. Given two input IREPs a and b of an arbitrary group $\mathcal{G}$, we can directly calculate the raising R^ρ, lowering L^ρ and z-operators Z^ρ of the product representation ρ (which in general is not an *irreducible* representation). Accounting for the case of multiple operators per IREP with an additional label i, we have with the Kronecker product $\otimes$:

$$R_i^\rho = R_i^a \otimes \mathbf{1}_i^b + \mathbf{1}_i^a \otimes R_i^b \tag{2.2.14}$$

$$L_i^\rho = L_i^a \otimes \mathbf{1}_i^b + \mathbf{1}_i^a \otimes L_i^b \tag{2.2.15}$$

$$Z_i^\rho = Z_i^a \otimes \mathbf{1}_i^b + \mathbf{1}_i^a \otimes Z_i^b \tag{2.2.16}$$

As an example, let us consider $\mathcal{G} = \mathrm{SU}(2)$, $a = \frac{1}{2}$, $b = 1$. In this case, we have:

$$R_1^\rho = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & \sqrt{2} & 0 \\ 0 & 0 & \sqrt{2} \\ 0 & 0 & 0 \end{pmatrix} \tag{2.2.17}$$

$$= \begin{pmatrix} 0 & \sqrt{2} & 0 & 1 & 0 & 0 \\ 0 & 0 & \sqrt{2} & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \sqrt{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sqrt{2} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{2.2.18}$$

$$L_1^\rho = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 & 0 \\ \sqrt{2} & 0 & 0 \\ 0 & \sqrt{2} & 0 \end{pmatrix} = (R_1^\rho)^T \tag{2.2.19}$$

$$Z_1^\rho = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & -\frac{1}{2} \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} \tag{2.2.20}$$

$$= \begin{pmatrix} \frac{3}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{3}{2} \end{pmatrix} \tag{2.2.21}$$

The decomposition of ρ into IREPs then occurs as described in Alg. 2.1. Applied to the example of $\mathcal{G} = \mathrm{SU}(2)$, $a = 1/2$ and $b = 1$, the decomposition proceeds as follows:

1. The first maximum weight state is $m_1 = (1, 0, 0, 0, 0, 0)^T$
2. Applying the only lowering operator, L_1^ρ, on m_1 gives the states (with orthonormalisation):

$$m_1 = (1, 0, 0, 0, 0, 0)^T \tag{2.2.22}$$
$$m_2 = (0, \sqrt{2/3}, 0, \sqrt{1/3}, 0, 0)^T \tag{2.2.23}$$
$$m_3 = (0, 0, \sqrt{1/3}, 0, \sqrt{2/3}, 0)^T \tag{2.2.24}$$
$$m_4 = (0, 0, 0, 0, 0, 1)^T \tag{2.2.25}$$

   m_4 is annihilated by L_1^ρ.
3. These states have eigenvalues $z_1 = 3/2$, $z_2 = 1/2$, $z_3 = -1/2$, $z_4 = -3/2$ with respect to Z_1^ρ, m_1 hence defines the IREP label (this should be true in general).
4. The IREP label is $c = 3/2$.
5. The operators of the new IREP are given by $O_1^{3/2} = C^T O_1^\rho C$, specifically (and as expected):

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \sqrt{2/3} & 0 & 0 \\ 0 & 0 & \sqrt{1/3} & 0 \\ 0 & \sqrt{1/3} & 0 & 0 \\ 0 & 0 & \sqrt{2/3} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.2.26}$$

$$R_1^{3/2} = \begin{pmatrix} 0 & \sqrt{3} & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & \sqrt{3} \\ 0 & 0 & 0 & 0 \end{pmatrix} = L_1^{3/2^T} \tag{2.2.27}$$

Algorithm 2.1 The decomposition of an arbitrary representation ρ into IREPs. The original factor IREPs a and b are only necessary if the Clebsch-Gordan coefficients $\mathbf{C}^{cc_i}_{aa_i;bb_i}$ are to be calculated. ORTHOGONALISE$(x, \mathcal{Y})$ orthogonalises x against all states in $\mathcal{Y}$ and normalises x. EIGVAL$(\{O_i\}_i^d, x)$ returns a tuple of size d, containing the eigenvalue of x with respect to each operator O_i.

procedure IREP-DECOMPOSITION(REP $\rho = \{R_i^\rho, L_i^\rho, Z_i^\rho\}_i^{\text{DEG}(\mathcal{G})}$, IREP a, IREP b)
  $\mathcal{A} \leftarrow \{\}$ ▷ Set of all found states
  $\mathcal{T} \leftarrow \{\}$ ▷ Set of states found from current max weight state
  $\mathcal{R} \leftarrow \{\}$ ▷ Set of found IREPs with associated CGC spaces
  repeat
    $q \leftarrow$ FIND-MAX-WEIGHT-STATE$(\dim(\rho), \mathcal{A})$
    repeat ▷ Generate chain of orthogonal states in $\mathcal{T}$
      $\mathcal{T} \leftarrow \mathcal{T} \cup \{q\}$
      $i \leftarrow 1$
      repeat ▷ Generate new orthogonal state by any lowering operator
        $q' \leftarrow L_i^\rho q$
        $q' \leftarrow$ ORTHOGONALISE$(q', \mathcal{T})$
        $i \leftarrow i + 1$
      until $q' \neq 0 \vee i >$ DEG$(\mathcal{G})$
      $q \leftarrow q'$
    until $q = 0$
    SORT$(\mathcal{T},$ EIGVAL$(\{Z_i^\rho\}_i^{\text{DEG}(\mathcal{G})}))$ ▷ sort by Z_i^ρ eigenvalues in descending order
    $c \leftarrow$ EIGVAL$(\{Z_i^\rho\}_i^{\text{DEG}(\mathcal{G})}, \mathcal{T}[0])$ ▷ IREP c defined by z-eigenvalues
    $M \leftarrow$ HSTACK$(\mathcal{T})$ ▷ $M \in \mathbb{K}^{\dim(\rho)\times|\mathcal{T}|}$, projector onto span $(\mathcal{T})$
    for $O \in \{R, L, Z\} \wedge i \in [1,$ DEG$(\mathcal{G})]$ **do**
      $O_i^c \leftarrow M^T O_i^\rho M$ ▷ Definition of raising, lowering and z-operators
    end for
    for $c_i \in [1, \dim(c)]$ **do** ▷ Definition of Clebsch-Gordan coefficients $\mathbf{C}^{cc_i}_{aa_i;bb_i}$
      for $a_i \in [1, \dim(a)]$ **do**
        for $b_i \in [1, \dim(b)]$ **do**
          $\mathbf{C}^{cc_i}_{aa_i;bb_i} \leftarrow \left\langle \mathcal{T}[c_i] \middle| e_{a_i}^{\dim(a)} \otimes e_{b_i}^{\dim(b)} \right\rangle$
        end for
      end for
    end for
    $\mathcal{A} \leftarrow \mathcal{A} \cup \mathcal{T}$
    $\mathcal{T} \leftarrow \{\}$
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{[c, \{R_i^c\}_i^{\text{DEG}(\mathcal{G})}, \{L_i^c\}_i^{\text{DEG}(\mathcal{G})}, \{Z_i^c\}_i^{\text{DEG}(\mathcal{G})}, \mathbf{C}^{cc_i}_{aa_i;bb_i}]\}$
  until $\dim(\rho) = |\mathcal{A}|$ ▷ Space exhausted
  return $\mathcal{R}$
end procedure
procedure FIND-MAX-WEIGHT-STATE(dimension d, set of states $\mathcal{S}$)
  for $i \in [1, d]$ **do**
    $m \leftarrow e_i^d$ ▷ i-th standard vector of size d
    $m \leftarrow$ ORTHOGONALISE$(m, \mathcal{S})$
    if $m \neq 0$ **return** m **end if**
  end for
end procedure

$$Z_1^{3/2} = \begin{pmatrix} 3/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 \\ 0 & 0 & -1/2 & 0 \\ 0 & 0 & 0 & -3/2 \end{pmatrix} \tag{2.2.28}$$

6. For example the Clebsch-Gordan coefficient $C^{3/2,1/2}_{1/2,1/2;1,0}$ is defined by $\langle m_2 | e_1^2 \otimes e_2^3 \rangle = \langle (0, \sqrt{2/3}, 0, \sqrt{1/3}, 0, 0)^T | (0,1,0,0,0,0,)^T \rangle = \sqrt{2/3}$. m_2 is used since we are looking for the second $(1/2)$ state in the $c = 3/2$ IREP. e_1^2 and e_2^3 are used as we are interested in the combination of the first $(1/2)$ state of the two-dimensional $a = 1/2$ IREP and the second (0) state of the three-dimensional $b = 1$ IREP.

7. We have found only four out of six states, so we would have to repeat this procedure with a new maximum weight state.

The next maximum weight state is found as follows:

1. Starting at $i = 1$, we have $m = e_1^6$. Orthogonalised against the four previous states, this is zero, so we continue with the next i.

2. At $i = 2$, we have initially $m = e_2^6 = (0,1,0,0,0,0)^T$. After orthogonalisation against the four other states, this is $m = (0, \sqrt{1/3}, 0, -\sqrt{2/3}, 0, 0)$. This has non-zero norm and hence can be used as the new maximum-weight state.

When repeating the first procedure with this new maximum weight state, we will find two additional states belonging to the $d = 1/2$ IREP. The dimension of ρ is then exhausted and we can conclude that for the IREPs of SU(2)

$$1/2 \otimes 1 = 3/2 \oplus 1/2 \quad . \tag{2.2.29}$$

Once the rank-3 CGC spaces are known, one can build arbitrary tensors from combinations of rank-3 tensors.

2.2.4 Tensor Operations

There are three nontrivial tensor operations of interest to be discussed: contractions of two tensors, addition of two tensors and the expansion of one tensor by another. Multiplication by a scalar is somewhat trivial, as one only has to multiply each reduced dense block m_i of the tensor by that scalar.

Tensor Contractions

Tensor contractions are a generalisation of matrix-matrix products. Given two tensors $A^{o_1,o_2,\cdots}_{i_1,i_2,\cdots}$ and $B^{p_1,p_2,\cdots}_{j_1,j_2,\cdots}$ and a set C of pairs $\{i_m, p'_m\}$ or $\{o_n, j'_n\}$ which relate to the same

Figure 2.2: Graphical representation of a matrix-matrix product $R^p_m = \sum_n B^p_n A^n_m$ (left) and a more complicated tensor product $R^d_{abf} = \sum_{ce} C^{de}_{abc} D^c_{ef}$ (right).

vector space (once as output and once as input), we can define a new tensor

$$R^{q_1,q_2,\cdots}_{k_1,k_2,\ldots} = \sum_C A^{o_1,o_2,\cdots}_{i_1,i_2,\ldots} B^{p_1,p_2,\cdots}_{j_1,j_2,\ldots} \tag{2.2.30}$$

where the set $\{k_m\}_m$ contains all input indices i and j of A and B not contained in C and similarly the set $\{q_m\}_m$ contains all output indices not contained in C.

The matrix-matrix product is a special case of this tensor contraction. Given two matrices $A : \mathbb{K}^m \to \mathbb{K}^n$ and $B : \mathbb{K}^n \to \mathbb{K}^p$, we can calculate a new matrix $R : \mathbb{K}^m \to \mathbb{K}^p$ as

$$R^p_m = \sum_n B^p_n A^n_m = B \cdot A \quad . \tag{2.2.31}$$

Such contractions can be represented graphically by connecting the contracted legs of the tensors (cf. Fig. 2.2). By convention, the expression $A \cdot B$ will denote the contraction over all common indices of A and B.

The implementation of a tensor contraction proceeds in three steps: First, we pair blocks a of A and b of B such that the quantum numbers on the contracted legs match. Secondly, for each of these pairs, the new block r is given as

$$r = \left(\sum_C m_a m_b \right) \bigotimes_\gamma \left(\sum_C c^\gamma_a c^\gamma_b \right) \quad . \tag{2.2.32}$$

The $\sum_C$ operation is now just a sparse or dense tensor product, the objects m and c have no further subdivisions, labels or somesuch. The best way to implement the dense sum is to reshape m_a and m_b into matrices such that contracted and uncontracted indices are put together as rows and columns (and vice versa for m_b), use a standard BLAS call to execute the product of the two and then reshape the result into the desired form with the desired index ordering.[iv] The c^γ are very sparse, so that a coordinate format is

[iv] Note that the reordering can be avoided if the input tensors indices and output tensor indices are ordered correctly. While asymptotically irrelevant (reordering takes time $O(mn + np + mp)$ while the matrix multiplication requires approximately $O(mnp)$ operations), very often the dense tensors are

likely the best way to store them. The contraction can then be implemented by hand very easily and at negligible computational cost – in real calculations, usually 5-10% of runtime are spent on handling the c^γ while 80-90% of runtime are spent inside BLAS or LAPACK functions dealing with the dense subblocks m_i.

Thirdly, there will likely be many blocks r_i with parallel CGC tensors. That is, if we have a tensor R which contains two blocks, p and q, with identical IREP labels and the CGC tensors of the two for all symmetries are exactly parallel $c_p^\gamma = \alpha^\gamma c_q^\gamma$ with $\alpha^\gamma \in \mathbb{R}$, we should add up the corresponding reduced dense tensors:

$$R = \left(m_p \bigotimes_\gamma c_p^\gamma \right) + \left(m_q \bigotimes_\gamma c_q^\gamma \right) \tag{2.2.33}$$

$$= \left(m_p \bigotimes_\gamma c_p^\gamma \right) + \left(m_q \bigotimes_\gamma \alpha^\gamma c_p^\gamma \right) \tag{2.2.34}$$

$$= \left(m_p + \left(\prod_\gamma \alpha^\gamma \right) m_q \right) \bigotimes_\gamma c_p^\gamma \quad . \tag{2.2.35}$$

Adding up blocks with parallel CGC spaces will be called *reducing* the tensor in the following. Note that for rank-2 tensors, *all* CGC spaces will be parallel. For rank-3 tensors, the spaces associated to SU($n \leq 2$) will be parallel, while larger groups (e.g. SU(3)) will result also in non-parallel CGC spaces. The latter is a result of the outer multiplicity of those groups. Finally, for rank-4 tensors, all non-abelian groups will produce non-parallel CGC space tensors. This effect can already be observed in the introductory example of this section – in the IREP decomposition of the product of SU(2) IREPs $\frac{1}{2} \otimes \frac{1}{2} \otimes 1$, the IREP 1 occurs twice (cf. Eq. (2.2.9)). There are hence two different ways to get from three input IREPs (from three input vector spaces) to one output IREP (on one output vector space). While for SU(2), we need at least three input vector spaces to observe this effect, for groups with outer multiplicity, such as SU(3), two input vector spaces/input IREPs suffice to generate the same output IREP multiple times.

Tensor Addition

The addition of two tensors is possible if they are defined on the same vector spaces, the tensors are then said to be compatible. Since it is not necessary to store all-zero blocks, such a compatibility is given if, for every leg and for every symmetry sector (i.e. distinct set of IREPs on that leg), the reduced, dense tensors either have the same size or only one of the addends has tensor blocks for this symmetry sector. For the addition, the reduction algorithm also used during the tensor contraction can be employed.

relatively small and hence receptive to such optimisations.

Tensor Expansion

The tensor expansion increases one or more of the associated vector spaces additively by another set of vector spaces. It is best explained via some examples:

For two dense vectors, the expansion would be the concatenation of the two vectors, creating a vector of different length. For two dense matrices A and B, we can either a) expand the columns by placing the matrices next to each other (the number of rows then have to match), b) expand the rows by placing the matrices atop of each other (the number of columns then have to match) or c) expand both rows and columns, placing the matrices as diagonal blocks of a larger matrix. For higher-rank tensors, the operation occurs analogously. This expansion only acts on the dense blocks of the tensor, the symmetry-protected CGC elements are left untouched.

For the implementation, we have to distinguish the case where only one tensor has blocks in a given symmetry sector of an expanded leg (in which case we can keep those blocks the same) or where both tensors have blocks in that sector. In this case, we have to expand the reduced, dense blocks of each of the blocks and place the contents either in the first half or the second half of the enlarged blocks. The following tensor reduction will then take care of all required additions.

2.2.5 Fusing and Inverting Tensor Legs

When working with tensor networks, it is sometimes necessary to combine two indices of a tensor into a single, larger index or to switch the direction of a tensor leg. Such operations can only be done by inserting appropriate resolutions of identity tensor operators into the overall tensor network. Changing the legs of a single tensor is not in general a useful operation and of course changes the *meaning* of the tensor, to e.g. act on a combined or dual space.

As an example, consider a rank-4 tensor T^{ab}_{cd} where we wish to combine the two indices c and d into a larger index γ, for example in order to do a singular value decomposition on this common index. If T was not a symmetry-protected tensor but simply a dense collection of scalars, we would only have to re-shape those scalars. However, such an operation would destroy the symmetry of the tensor when done naively.[v] Instead, we generate a pair of rank-3 tensors $S^{c'd'}_{\gamma}$ and F^{γ}_{cd} which resolve to a set of identity tensors when contracted over their common index γ (cf. Fig. 2.3):

$$\sum_{\gamma} S^{c'd'}_{\gamma} F^{\gamma}_{cd} = \mathbf{1}^{c'}_{c} \mathbf{1}^{d'}_{d} \tag{2.2.36}$$

[v] In the special case of SVD or similar decomposition over just a single leg, it is possible to implement it without an explicit previous reshaping, see Sec. 2.3.2

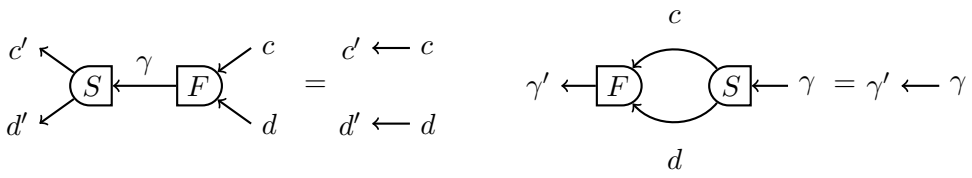


Figure 2.3: Left: A fusing tensor F and a splitting tensor S contracted over their single common index γ result in a rank-4 tensor which acts as the identity between two legs each. Right: F and S contracted over their two common indices c and d result in a rank-2 identity tensor.

and similarly to a single identity tensor when contracted over c and d instead:

$$\sum_{cd} S_{\gamma}^{cd} F_{cd}^{\gamma'} = \mathbf{1}_{\gamma}^{\gamma'} \quad . \tag{2.2.37}$$

Returning to our example, we would first calculate

$$T'^{ab}_{\ \ \gamma} = \sum_{cd} T_{cd}^{ab} S_{\gamma}^{cd} \quad , \tag{2.2.38}$$

act on T' with the operation we had in mind and afterwards contract with F to get

$$T''^{ab}_{\ \ cd} = \sum_{\gamma} T'^{ab}_{\ \ \gamma} F_{cd}^{\gamma} = \sum_{\gamma} \sum_{c'd'} T_{c'd'}^{ab} S_{\gamma}^{c'd'} F_{cd}^{\gamma} = \sum_{c'd'} T_{c'd'}^{ab} \mathbf{1}_{c}^{c'} \mathbf{1}_{d}^{d'} = T_{cd}^{ab} \quad . \tag{2.2.39}$$

As can be seen, if our operation was the identity, we have $T'' = T$. This of course does not hold if we changed T' in some other way in-between the initial fusing and later splitting.

In the reduced dense tensor sector, the reduced blocks of F and S are simply the expected rank-3 identity tensors, while the CGC spaces are exactly the spaces arising from a IREP decomposition of the IREPs on the legs c and d. Conveniently, those are the same regardless of whether both c and d are incoming or both c and d are outgoing indices, hence the tensors F and S have identical structure except for inverted tensor legs.[36]

For such a fusing or splitting of indices to work, it is necessary that either both indices are incoming or both indices are outgoing indices. In general, this is not true, we may wish to act on a matrix-like structure (e.g. for SVD) while combining an outgoing and an incoming index. This problem can be avoided by another insertion of the identity: Given the tensor above, we can construct a fuse of the indices a and $a^{\dagger}$, where $a^{\dagger}$ has the same structure as a except that all IREPs are adjointed: For SU(2), this changes nothing, U(1) and $\mathbb{Z}_k$ IREPs x are changed into $-x$ (or $-x$ mod k). We then project the generated fusing leg γ onto the dummy vacuum sector, hence generating effectively a rank-2 tensor with two incoming indices, $I_{aa^{\dagger}}$. A similar operation can give us a tensor

with two outgoing indices $O^{aa^\dagger}$. Once we fix the normalisation of each CGC space by the square root of its dimension in either I or O (cf. Ref. [36], Eq. (B80)), we can ensure that[vi]

$$\sum_{a^\dagger} I_{aa^\dagger} O^{a'a^\dagger} = \mathbf{1}_a^{a'} \quad . \tag{2.2.40}$$

With this further identity in hand, it is now possible to contract e.g. $I_{aa^\dagger}$ with T_{cd}^{ab} to arrive at $T'^{b}_{\ a^\dagger cd}$, combine the indices $a^\dagger$ and c in the same way as before with S, operate on the resulting rank-3 tensor and first revert the combination with F and then the direction inversion with O.

2.3 Matrix-Product States

Matrix-product states (MPS) are a potentially very efficient way to represent quantum-mechanical states on a global Hilbert space $\mathcal{H}$ which can be decomposed as the tensor product of a set of L local Hilbert spaces $\{\mathcal{H}_i\}_{i=1}^L$, s.t. $\mathcal{H} = \bigotimes_i^L \mathcal{H}_i$. The basic idea is to build, via successive basis transformations, a basis in which the desired state can be expressed most efficiently and then store both the set of transformations as well as the representation of the state in this basis.

Let $\{|\sigma_i\rangle\}_\sigma$ be a set of d_i basis states of the spaces $\mathcal{H}_i$. We can then write any pure quantum state $|\psi\rangle \in \mathcal{H}$ as

$$|\psi\rangle = \sum_{\sigma_1} \sum_{\sigma_2} \cdots \sum_{\sigma_L} c_{\sigma_1 \sigma_2 \cdots \sigma_L} |\sigma_1\rangle \otimes |\sigma_2\rangle \otimes \cdots \otimes |\sigma_L\rangle \quad . \tag{2.3.1}$$

The rank-L tensor c will in general have $\prod_i^L d_i$ (or if all $d_i = d$, d^L) entries. Storing this tensor explicitly is manageable for small systems (e.g. spins with $d = 2$, $L = 10$, $|c| = 2^{10} = 1024$), but due to the exponential growth quickly becomes impossible for larger systems.

However, it is clear that at least for *some* states, there are also efficient (meaning polynomial in system size) parametrisations. For example, we can describe a fully ferromagnetic product state on a spin chain of arbitrary length perfectly well with "all spins up", without having to explicitly specify d^L coefficients. The magic of MPS is now to achieve an efficient parametrisation even for strongly correlated quantum states. The cost of that parametrisation primarily depends on the entanglement in the system,[40] but for example for ground-states of gapped Hamiltonians, we know that entanglement obeys an area law[41] and the cost of the MPS parametrisation of such states in one dimension will be effectively linear in system size.

[vi] Correct index ordering is crucial here to obtain the identity tensor – accidentally contracting a and $a^\dagger$ will result in zero tensors for U(1) IREPs, while exchanging a and $a^\dagger$ during the creation of either one will result in spurious minus signs for SU(2) IREPs – they are called *non*-abelian for a reason!

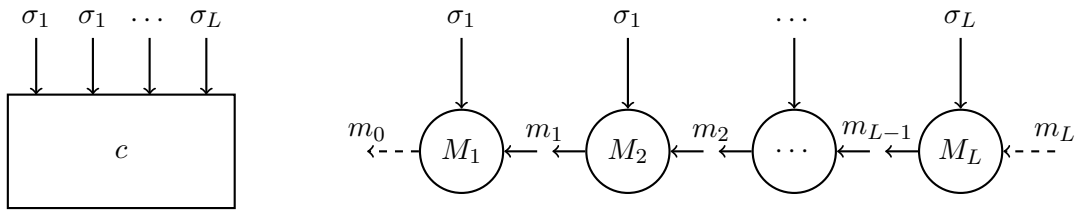


Figure 2.4: Left: The tensor c from equation (2.3.1) with the physical indices associated to the local spaces $\mathcal{H}_i$ coming in from above. Right: A MPS with four tensors explicitly shown. Physical indices from the local spaces $\mathcal{H}_i$ come in from above corresponding to states $|\sigma_i\rangle$. The direction of bond indices is from right-to-left. The tensor c can be restored by contracting over all bond indices $\{m_i\}_{i=1}^{L-1}$. The special indices m_0 and m_L are shown as dashed lines – the former corresponds to the symmetry sector of the state itself, the latter to the vacuum sector; both are essentially one-dimensional dummy indices.

2.3.1 Definition

Specifically, the MPS formalism attempts to decompose the tensor c from Eq. (2.3.1) into a series of L rank-3 tensors $\{M^{m_{i-1}}_{i;\sigma_i m_i}\}_{i=1}^{L}$ such that the contraction of those tensors over all *bond indices* $\{m_i\}_i$ results in the tensor c:

$$c_{\sigma_1\sigma_2\cdots\sigma_L} = \sum_{m_1}\sum_{m_2}\cdots\sum_{m_{L-1}} M^{m_0}_{i;\sigma_1 m_1} M^{m_1}_{i;\sigma_2 m_2}\cdots M^{m_2}_{i;\sigma_3 m_3}\cdots M^{m_{L-1}}_{i;\sigma_L m_L} \tag{2.3.2}$$

In a non-symmetry-protected view, the tensor c and the $\{M\}_i$ are just dense tensors. Then for consistency, the indices m_0 and m_L must be one-dimensional dummy indices to ensure that c and the tensor product have the same shape.[vii]

If we wish to use symmetry-protected tensors, there are two potential issues: Firstly, we have to fix the directions of the tensor legs in a consistent manner. In the SyTen toolkit, the tensor $M^{m_{i-1}}_{i;\sigma_i m_i}$ has incoming indices σ_i (which is also called *physical index*) and m_i (*right bond index*) and an outgoing index m_{i-1} (*left bond index*). This is graphically represented in Fig. 2.4.

Secondly, to actually implement symmetries, we must assign each index the good quantum numbers associated to the symmetry. For the physical indices, those are simply the quantum numbers also associated to the local, physical states. The bond indices can then be labelled consistently assuming that we fix the rightmost index m_L. It is convenient to associate this index to the vacuum, i.e. all labels zero. This naturally results in the leftmost index m_0 carrying the quantum numbers of the state itself – if there

[vii] There exists an alternative description whereby we trace out the remaining first and last index. This description implements periodic boundary conditions of the underlying system in the MPS, but implemented naively has some computational disadvantages. Since periodic boundary conditions of the system Hamiltonian can also be implemented with "open boundary condition" MPS, I will not discuss the other description further. However, see also Ref. [42, 43] for potentially more efficient approaches beyond the scope of this thesis.

are five particles in the state, the one-dimensional m_0 index will have quantum number $N = 5$.

Construction of an MPS

Given a tensor c, it is in theory possible to iteratively decompose it into a MPS by a series of tensor leg combinations and singular value decompositions. As this is impractical in all but the simplest cases, it won't be discussed here. The following alternatives exist:

- For normalised product states, each $M^{m_{i-1}}_{i;\sigma_i m_i}$ is a $1 \times 1 \times d_i$ dimensional tensor. The entries are either one or zero, depending on whether the particular σ_i occurs in the product state or not.

- As a special case of the above, a Néel spin state can be found by generating the MPS tensors iteratively. Given a right-hand basis from the previous iteration (or the vacuum) and the local physical basis, an isometry tensor which explores the entire available left-hand basis is generated. Then, all states but a single one from the absolute-value-wise smallest spin quantum number sector are discarded. For example, the local basis $\{|\uparrow\rangle, |\downarrow\rangle\}$ combined with a right-hand basis of a single state with $S^z = \frac{1}{2}$ gives two states in total, one with $S^z = 0$ and one with $S^z = 1$. The former is chosen, resulting in a new right-hand basis on the next iteration with $S^z = 0$. The state built in this way will either have $S^z = 0$ for even chain lengths or $S^z = \pm S^z_l$ where S^z_l is the minimal local spin for odd chain lengths.

- A "random" state can be generated by first building a state corresponding to the vacuum (either no particles or a Néel state for spins) and then acting with random local operators in random places until a state transforming as desired has been found. By using different weights for different operators according to the expected change in distance to the target state, this can be made very fast and result in decent initial states for e.g. DMRG ground-state search. A measure for the expected change in distance would be based on the current quantum number of the state ($N_c = 5$), the target quantum number we wish to achieve ($N_t = 10$) and the change in quantum numbers given by the transformation of the operator ($N_\Delta = \pm 1$ for $c^{(\dagger)}$), e.g. as $\operatorname{sign}(N_c - N_t)\, N_\Delta$.

Generality of a symmetry-protected MPS

It is fairly straight-forward to see that a MPS which transforms globally irreducibly (i.e. has a well-defined particle number, S^z quantum number or somesuch) can be constructed from symmetry-preserving tensors if the local basis also preserves the symmetry. Assume the opposite, that some tensor of the system does not preserve the symmetry of the system and hence has (e.g.) a state on its left outgoing leg which does not transform

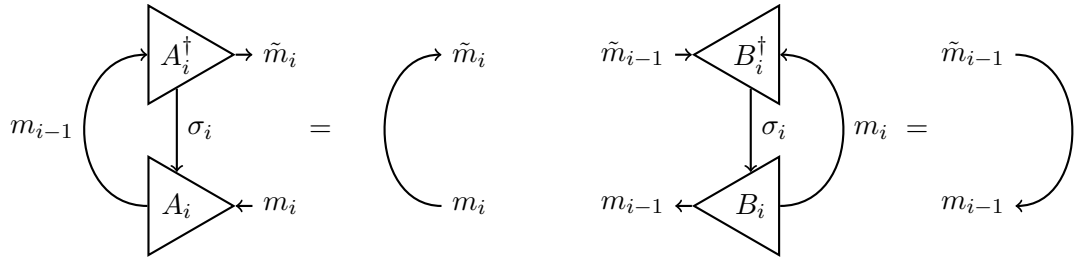


Figure 2.5: Left: A left-normalised MPS tensor A_i contracted with its conjugate $A_i^\dagger$ over the physical and left indices σ_i and m_{i-1} results in a left-hand identity tensor mapping m_i to $\tilde{m}_i$, represented by a single line. Right: Similarly, a right-normalised MPS tensor B_i contracted with its conjugate $B_i^\dagger$ over the physical and right indices σ_i and m_i results in a right-hand identity tensor mapping $\tilde{m}_{i-1}$ to m_{i-1}.

irreducibly. However, given such a reducible state mixing quantum numbers, subsequent tensors cannot "disentangle" those states and hence, when arriving at the left edge of the MPS chain, the overall state still transforms reducibly, not irreducibly as originally assumed. Put another way, we cannot combine a local eigenstate of the symmetry and some state which transforms reducibly in a tensor product to build a global eigenstate of the symmetry. If the states on the first (or last) site transform irreducibly, then also the combined states on sites 2 through L (or 1 through $L-1$) have to transform irreducibly. By applying this reasoning again to the second site, we can iteratively argue that every tensor of the state has to preserve the global symmetries.

Note that this argument of course does not hold for states which do *not* transform irreducibly. Hence if one wants to study symmetry breaking, it is absolutely necessary to *allow* the breaking of the symmetry in question in the first place by not implementing the symmetry-to-be-broken as part of the symmetries preserved in the tensor network.

2.3.2 Left- and Right-Normalisation

The left- and right-normalisation of MPS tensors is a computationally very useful tool. Specifically, a tensor $A^{m_{i-1}}_{i;\sigma_i m_i}$ is said to be left-normalised if and only if

$$\sum_{m_{i-1}\sigma_i} A^{m_{i-1}}_{i;\sigma_i m_i} \left(A^{m_{i-1}}_{i;\sigma_i \tilde{m}_i}\right)^\dagger = \mathbf{1}^{\tilde{m}_i}_{m_i} \quad . \tag{2.3.3}$$

The dual-space conjugation † is defined for tensors as firstly complex-conjugating every element and secondly inverting the direction of every tensor leg, i.e. $(X^m_n)^\dagger = X^{\star n}_{\ m}$ with $\star$ denoting complex conjugation. It is also customary to draw a conjugated MPS tensor with a downward physical leg (in part to ease contraction with its unconjugated counterpart).

Similarly, the tensor $B^{m_{i-1}}_{i;\sigma_i m_i}$ is right-normalised if and only if

$$\sum_{m_i\sigma_i} B^{m_{i-1}}_{i;\sigma_i m_i} \left(B^{\tilde{m}_{i-1}}_{i;\sigma_i m_i}\right)^\dagger = \mathbf{1}^{m_{i-1}}_{\tilde{m}_{i-1}} \quad . \tag{2.3.4}$$

These contractions are graphically presented in Fig. 2.5. By convention, in the context of MPS, the letter A is used for left-normalised tensors only, the letter B for right-normalised tensors and the letter M for unnormalised tensors. Triangles are used instead of circles to denote normalised tensors (also cf. Fig. 2.8 later).

Process of Normalisation: QR or SVD

A key property of MPS is a certain gauge freedom: Given two MPS tensors M_i and M_{i+1}, we only actually care about their contraction over their common MPS bond index. In turn, if we can find a decomposition such that

$$M^{m_{i-1}}_{i;\sigma_i m_i} = \sum_{m'_i} A^{m_{i-1}}_{i;\sigma_i m'_i} T^{m'_i}_{m_i} \quad , \tag{2.3.5}$$

we may multiply the transfer tensor T also into M_{i+1} instead:

$$M_{i+1} \to \sum_{m_i} T^{m'_i}_{m_i} M^{m_i}_{i+1;\sigma_{i+1}m_{i+1}} = M^{m'_i}_{i+1;\sigma_{i+1}m_{i+1}} \quad . \tag{2.3.6}$$

A singular value decomposition or a QR decomposition may be used to compute a normalised tensor A_i (or B_i) and a transfer tensor T from an input tensor M_i. The QR decomposition has the advantage that it is computationally cheaper, while the SVD decomposition allows the compression of the state (cf. Sec. 2.3.4). The decompositions, when applied to matrices and without regard for tensor leg directions, are defined here as follows:

$$M_{ab} \to Q_{ac}\,;\, R_{cb} \text{ s.t. } \sum_a Q_{ac} Q^\star_{a\tilde{c}} = \mathbf{1}_{c\tilde{c}} \tag{2.3.7}$$

$$\text{and R upper triangular} \tag{2.3.8}$$

$$\text{and } M_{ab} = \sum_c Q_{ac} R_{cb} \tag{2.3.9}$$

$$M_{ab} \to U_{ac}\,;\, S_{cd}\,;\, V_{db} \text{ s.t. } \sum_a U_{ac} U^\star_{a\tilde{c}} = \mathbf{1}_{c\tilde{c}} \tag{2.3.10}$$

$$\text{and } \sum_b V_{db} V^\star_{\tilde{d}b} = \mathbf{1}_{d\tilde{d}} \tag{2.3.11}$$

$$\text{and } S \text{ diagonal, real and with } S_{i,i} \geq S_{i+1,i+1} \tag{2.3.12}$$

$$\text{and } M_{ab} = \sum_{cd} U_{ac} S_{cd} V_{db} \tag{2.3.13}$$

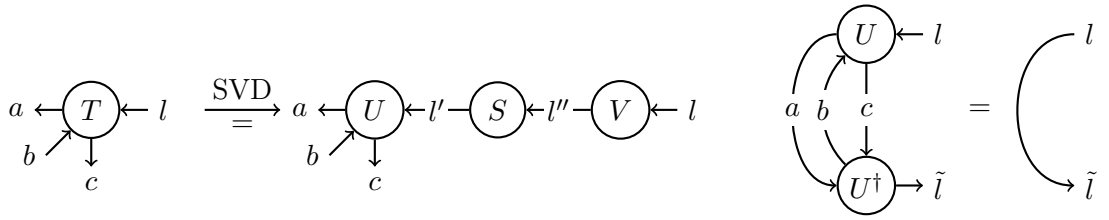


Figure 2.6: Left: Decomposition of a rank-4 tensor T over a single leg l into tensors U, S and V. The contraction over the legs l' and l'' of U, S and V would result in the original tensor T. Of course, instead of U, one can also take V to correspond to the original tensor. Right: The contraction of U and $U^\dagger$ results in an identity tensor on the space of l'.

The QR or SVD decomposition over one leg (m_i in Eq. (2.3.5)) of an arbitrary-rank tensor proceeds as described in Alg. 2.2. It makes extensive use of the fact that the only valid rank-2 symmetry tensors are proportional to the identity and that contractions of valid symmetry tensors also result in valid symmetry tensors, which allows us to transfer the scaling factor easily from the then-orthonormal symmetry tensors into the reduced tensor, where it is handled by the matrix decomposition. A possible result of such a decomposition is depicted in Fig. 2.6.

2.3.3 Level-1 Operations on MPS

Three operations are very easy to implement on MPS and will be reviewed here briefly:

Scalar Multiplication

Given a MPS described by tensors $\{M_i\}_i$, multiplication by a scalar α can be implemented by multiplying the dense reduced components of the blocks of the first tensor M_1 by α.

Scalar Product

Given two states $|\psi\rangle$ and $|\phi\rangle$ with associated MPS representations $\{M_i\}_i$ and $\{N_i\}_i$, the scalar product $\langle\phi|\psi\rangle$ can be calculated by the complete contraction of all tensors. This is best represented graphically (cf. Fig. 2.7). It must be stressed that this contraction should work from left to right, i.e. first contracting $M_1 \cdot N_1^\dagger = C_1$, then the result with M_2 and $N_2^\dagger$ etc.

MPS-MPS Addition

If two states transform in the same way on their leftmost dummy index, their sum will again transform uniquely on this leg. The MPS tensors $\{R_i\}_i$ representing the sum of two MPS $\{M_i\}_i$ and $\{N_i\}_i$ can be written via tensor expansion on the MPS bond indices as

$$R_{1;\sigma_1} = \begin{pmatrix} M_{1;\sigma_1} & N_{1;\sigma_1} \end{pmatrix} \tag{2.3.14}$$

Algorithm 2.2 The decomposition of a tensor either via SVD or QR decomposition over a single leg. In the SVD, either U or V can be used to take the place of the original tensor and result in an orthogonal basis upon contraction; in the QR, only Q can be used. The transfer tensor into the next site is then given either by $S \cdot V$ or R. In principle, any matrix decomposition can be used in Line 14 depending on the required properties of the resulting tensors.
The loop from Line 4 to Line 12 rescales each dense block such that the associated CGC block results in an identity matrix when contracted over all but the l-th leg. This is reversed in Line 17 so the values in c_i^γ are never actually changed.
If the target tensor is the U tensor from the SVD or the Q tensor from the QR decomposition, the collection on Line 13 re-shapes each block such that all indices but the l-th index build the rows and the l-th index is the sole column index. Those blocks should then be stacked on top of each other. It is necessary to remember which rows of the combined matrix m belongs to which block r_i for the following re-distribution. If the target tensor is the $V^\dagger$ tensor from the SVD, rows and columns have to be interchanged in the previous two sentences.

1: **procedure** TENSOR-DECOMPOSITION(Tensor T, Leg l)
2: **for each** Sector s on leg l **do**
3: Collect all blocks $b_i \in T$ which transform as s on l into the set $\mathcal{S}$.
4: **for** block $\{r_i, c_i^\gamma\} = b_i \in \mathcal{S}$ **do**
5: $\alpha_i \leftarrow 1$
6: **for** each symmetry γ in the system **do**
7: $t^l_{l'} \leftarrow \sum_{\text{all but } l,l'} c_i^\gamma \, (c_i^\gamma)^\dagger$
8: $t^l_{l'} = \alpha' \mathbf{1}^l_{l'}$
9: $\alpha_i \leftarrow \alpha_i \cdot \alpha'$
10: **end for**
11: $r_i \leftarrow \alpha_i \cdot r_i$
12: **end for**
13: Collect all $r_i \in \mathcal{S}$ into matrix m
14: Decompose $m \rightarrow u \cdot s \cdot v$ or $m \rightarrow q \cdot r$
15: Distribute u, v or q into r_i'
16: **for** block $b_i \in \mathcal{S}$ **do**
17: Add $\{r_i'/\alpha_i, c_i^\gamma\}$ to tensors U, V or Q
18: **end for**
19: Add $\{s, \mathbf{1}^\gamma\}$ and $\{v, \mathbf{1}^\gamma\}$ to tensor S and V or $\{r, \mathbf{1}^\gamma\}$ to R
20: **end for**
21: **return** tensors U, S, V or Q and R
22: **end procedure**

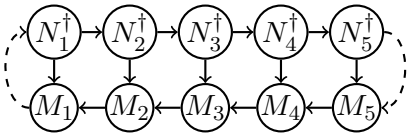


Figure 2.7: The scalar product of two MPS represented by tensors $\{M_i\}_i$ and $\{N_i\}_i$. The tensors on top are conjugated, with directions of legs reversed and entries complex-conjugated. The right-hand side dashed line connects the two vacuum sectors of the right-hand side dummy indices. The left-hand side dashed line connects the two sector indices containing the transformation information of each of the two states. Since the scalar product results in a scalar value, no legs of the tensor network are left open after the contraction operation completes.

$$R_{1<i<L;\sigma_i} = \begin{pmatrix} M_{i;\sigma_i} & 0 \\ 0 & N_{i;\sigma_i} \end{pmatrix} \tag{2.3.15}$$

$$R_{L;\sigma_L} = \begin{pmatrix} M_{L;\sigma_L} \\ N_{L;\sigma_L} \end{pmatrix} \tag{2.3.16}$$

where $X_{i;\sigma_i}$ is a matrix in the MPS bond dimensions. Note that in principle, it is also possible to add two states which do not transform the same way, i.e. where it is *not* possible to place the leftmost tensors $M_{1;\sigma_1}$ and $N_{1;\sigma_1}$ in the same row since their m_0 indices transform differently. This would then result in a non-uniquely transforming state with a non-trivial leftmost basis. Something very similar occurs when handling a state transforming as $S \neq 0$ under $\mathrm{SU}(2)_{\mathrm{Spin}}$. In this case, the symmetry tensors are not one-dimensional but actually $2S+1$-dimensional, even on the leftmost leg. Taking care of such states correctly (in particular during scalar products) is crucial to obtaining correct data, but results naturally from an otherwise correct tensor implementation.

2.3.4 Schmidt Coefficients and MPS Truncation

If all MPS tensors to the left of a given bond k are left-normalised as described above (with the transfer tensors successively multiplied into the next right-hand neighbour), then the left-hand MPS bond basis is orthonormal (as it results from a series of orthonormal transformations of an orthonormal basis). If we attempt the same for the right-hand MPS bond basis, we will generally require an additional singular value tensor on this bond. This tensor results from the last right-hand side normalisation, but we cannot multiply it into the left-hand side without destroying the other normalisation. Graphically, this situation is represented in Fig. 2.8. We can then write the MPS as

$$|\psi\rangle = \sum_{\boldsymbol{\sigma}} \sum_{\boldsymbol{m}} A^{m_0}_{1;\sigma_1 m_1} \cdots A^{m_{k-1}}_{k;\sigma_k m_k} S^{m_k}_{m'_k} B^{m'_k}_{k+1;\sigma_{k+1} m_{k+1}} \cdots B^{m_L}_{L;\sigma_L m_{L-1}} |\boldsymbol{\sigma}\rangle \tag{2.3.17}$$

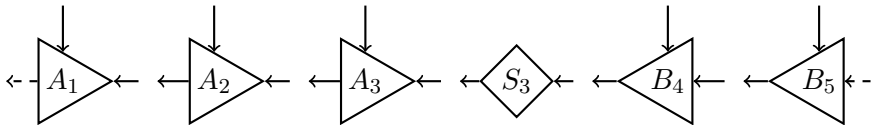


Figure 2.8: A MPS of five sites; the leftmost three sites are left-normalised while the two right sites are right-normalised. A singular value tensor S_3 lives on the third bond and contains the Schmidt coefficients.

which, with the orthonormality of the left- and right-hand side and the diagonalicity of $S^{m'_k}_{m_k} = S_i$, is equivalent to

$$|\psi\rangle = \sum_{i=1}^{t} S_i |\alpha_i\rangle \otimes |\beta_i\rangle \tag{2.3.18}$$

where states $|\alpha_i\rangle$ and $|\beta_i\rangle$ are orthonormal states on the left- and right-hand side of the system, defined via the series of basis transformations contained in $A_{j\leq k}$ and $B_{j>k}$. This decomposition is exactly the Schmidt decomposition of the state between subsystems of size k and $L-k$ respectively.[44] It is always possible to choose the $S_i \in \mathbb{R}$, furthermore, they can be sorted in descending order, $S_i \geq S_{i+1}$.

The von Neumann entanglement entropy between the subsystems to the left and right of the bond k is given as[21]

$$\mathcal{S}_{j\leq k, j>k} = \frac{-\sum_{i=1}^{t} S_i^2 \log_2\left(S_i^2\right)}{\sum_{i=1}^{t} S_i^2} \quad . \tag{2.3.19}$$

The denominator here is simply the normalisation condition, for a normalised state, it will be 1. The number t of Schmidt coefficients S_i is given by the dimension of the k-th MPS bond – if the system has little entanglement (t small), the MPS bond dimension is small, if it is strongly entangled (t large), that bond dimension also has to be large. The set of Schmidt coefficients can be considered the entanglement spectrum of the system: For most ground-states, they will decay exponentially to zero, while for strongly entangled or critical states, the decay will be algebraic.

This decomposition also provides a way to *truncate* a MPS, i.e. to approximate a state $|\psi\rangle$ (w.l.o.g. assumed to be normalised) by a different state $|\psi'\rangle$ with a smaller MPS bond dimension. By setting the smallest coefficients in the tensor S to zero, we can generate this new state $|\psi'\rangle$ while closely monitoring the induced error. For this, first note that the norm of a state in the format of Eq. (2.3.17) or Eq. (2.3.18) is given by $\sum_{m_k m'_k} S^{m'_k}_{m_k} S^{\star m_k}_{\ \ m'_k} = \sum_{k=1}^{t} S_k^2$. Second, if two states $|\psi\rangle$ and $|\psi'\rangle$ share the left- and right-hand Schmidt basis $|\alpha_k\rangle$ and $|\beta_k\rangle$, with different coefficients S_k and S'_k, then their overlap is given by $\langle\psi'|\psi\rangle = \sum_{k=1}^{t} S_k S'_k$.

Hence, if we select $S'_k = S_k$ if $k \leq r$ and $S'_k = 0$ if $r < k \leq t$, we will generate a state $|\psi'\rangle$ with norm $\sqrt{R} = \sqrt{\sum_{k=1}^{r} S_k^2}$. Further, let $D = \sum_{k=r+1}^{t} S_k^2$ be the squared sum of

discarded singular values, $D \in [0,1]$ and note that $R + D = 1$. We then consider the error induced by such a truncation from t to r singular values between *normalised* states $|\psi_N\rangle$ and $|\psi'_N\rangle = R^{-1/2}|\psi'\rangle$ and find:

$$\left|\left||\psi'_N\rangle - |\psi_N\rangle\right|\right|^2 = \left|\left|R^{-\frac{1}{2}}|\psi'\rangle - |\psi\rangle\right|\right|^2 \tag{2.3.20}$$

$$= \frac{1}{R}\langle\psi'|\psi'\rangle - 2R^{-\frac{1}{2}}\langle\psi'|\psi\rangle + \langle\psi|\psi\rangle \tag{2.3.21}$$

$$= 1 - 2\sqrt{R} + 1 \tag{2.3.22}$$

$$= 2 - 2\sqrt{1-D} = 2 - 2\sqrt{1 - \sum_{k=r+1}^{t} S_k^2} \tag{2.3.23}$$

$$\left|\left||\psi'_N\rangle - |\psi_N\rangle\right|\right| = \sqrt{2 - 2\sqrt{1 - \sum_{k=r+1}^{t} S_k^2}} \tag{2.3.24}$$

Making use of the triangle inequality, it is obvious that if we truncate $|\psi'_N\rangle$ on another bond creating a new state $|\psi''_N\rangle$, the error between $|\psi_N\rangle$ and $|\psi''_N\rangle$ is bounded by the sum of the individual errors:

$$\left|\left||\psi''_N\rangle - |\psi_N\rangle\right|\right| \leq \left|\left||\psi''_N\rangle - |\psi'_N\rangle\right|\right| + \left|\left||\psi'_N\rangle - |\psi_N\rangle\right|\right| \tag{2.3.25}$$

Then, given a normalised state $|\psi_N\rangle$, we can bound the error incurred by a series of SVD truncations on each bond resulting in a new normalised state $|\psi'_N\rangle$ as

$$\left|\left||\psi'_N\rangle - |\psi_N\rangle\right|\right| \leq \sum_{i=1}^{L} \sqrt{2 - 2\sqrt{1 - \sum_{k=r_i+1}^{t_i} S_{i;k}^2}} \quad . \tag{2.3.26}$$

In comparison to this simple calculation, there exists a different bound proven[45] by Frank Verstraete, namely that

$$\left|\left||\psi'_N\rangle - |\psi_N\rangle\right|\right| \leq \sqrt{2\sum_{i=1}^{L}\sum_{k=r_i+1}^{t_i} S_{i;k}^2} \tag{2.3.27}$$

which is the commonly-used bound[21,45] and stricter if the SVD has been used to truncate more than once, e.g. on different bonds. In the case of only a single truncation via a single SVD, Eq. (2.3.24) gives the exact induced error and is hence in particular suitable for DMRG (cf. Chapter 3).

2.3.5 Variational Orthogonalisation

Given a MPS $|\psi\rangle$, it may be desirable to orthogonalise it against a set of other MPS $\{|o^A\rangle, |o^B\rangle, \ldots, |o^M\rangle\}$, resulting in a new MPS $|r\rangle$. One application of this is the Krylov time evolution algorithm, explained in Sec. 4.1.4. In principle, this orthogonalisation is possible using MPS-MPS scalar products and addition with a Gram-Schmidt-like procedure. However, the increase in bond dimension resulting from a MPS-MPS addition may require a subsequent truncation operation which inevitably leads to a loss of precision and hence orthogonality. In contrast, the variational algorithm presented here works at the desired bond dimension and precision throughout, hence preserving orthogonality between the vectors to within the machine epsilon. This algorithm is not new, but a certain lack of detailed explanations in the published literature motivates this re-presentation here.

Assume that the input MPS $|\psi\rangle$ on a system of size N is given as a series of tensors $\{\Psi_i\}_{i=1}^N$, the M target-orthogonal vectors are given as $\{\{O_i^A\}_{i=1}^N, \ldots, \{O_i^M\}_{i=1}^N\}$ and our current best estimate for the resulting state is $|r\rangle$, stored in MPS form as $\{R_i\}_{i=1}^N$. We wish to minimise

$$\Big|\Big||\psi\rangle - |r\rangle\Big|\Big| \tag{2.3.28}$$

under the constraint that $\langle r|o^J\rangle = 0 \; \forall \; J \in [1, M]$. Using a series of Lagrange multipliers[46] λ^J, this is equivalent to minimising

$$\langle r|\psi\rangle + \langle r|r\rangle + \sum_{J=A}^{M} \lambda^J \langle r|o^J\rangle \tag{2.3.29}$$

with respect to $\langle r|$ and λ^J, where we have already removed terms not dependent on $\langle r|$ nor λ^J.

To obtain an individual optimal site tensor R_i, we must partially differentiate this function with respect to $R_i^\dagger$ and each λ^J. The resulting equations are

$$-\psi_E + R_i + \sum_{J=A}^{M} \lambda^J o_E^J = 0 \tag{2.3.30}$$

$$o_E^J \cdot R_i^\dagger = 0 \Leftrightarrow o_E^{J\dagger} \cdot R_i = 0 \quad \forall J \in [A, M] \tag{2.3.31}$$

where X_E denotes the *environment tensor* generated by taking the overlap between $|r\rangle$ and $|x\rangle$ and removing the tensor $R_i^\dagger$. Note that we assume the state $|r\rangle$ to be in mixed-canonical form with respect to site i. We can rewrite these equations as follows: Let r be the vectorisation of the target tensor R_i, similarly ψ the vectorisation of the environment tensor ψ_E, let o be a matrix which contains the orthogonal environment tensors o_E^J as its columns and let λ be a vector which contains the Lagrange multipliers.

With this, the above becomes:

$$r = \psi - o\lambda \tag{2.3.32}$$

$$0 = o^\dagger r \tag{2.3.33}$$

By left-multiplying the first equation with $o^\dagger$, we can equate its left-hand side to zero to obtain:

$$0 = o^\dagger \psi - o^\dagger o\lambda \tag{2.3.34}$$

$$\Rightarrow \lambda = \left(o^\dagger o\right)^{-1} o^\dagger \psi \quad . \tag{2.3.35}$$

Here, the object $o^\dagger\psi$ is simply the vector of overlaps between target-orthogonal vectors $\langle o^J|$ with the input vector $|\psi\rangle$ and the matrix $o^\dagger o$ is the matrix of overlaps between the target-orthogonal vectors. Due to its relatively small size, it can be inverted exactly (using e.g. a SVD). This then gives the vector of Lagrange multipliers λ which, when inserted into Eq. 2.3.32 gives us the optimal value for the tensor R_i. Proceeding in this fashion through the MPS, we can optimise each individual site tensor sequentially. The full procedure is summarised in Alg. 2.3.

Two caveats have to be observed here: First, the above approach intentionally limits each individual orthogonalisation to the space spanned by a single tensor, i.e. does not allow for a potentially necessary increase in bond dimensions. In practice, this means that the result may well differ from the "exact" result produced using MPS-MPS addition and the Gram-Schmidt procedure, however, the difference is usually negligible and the exact orthogonality with the vectors $|o^J\rangle$ much preferable. If for some reason the incurred errors are unacceptably large, e.g. because the bond dimensions of the input states are already very small and do not allow for a good variational approach, one may use the same procedure but optimise two neighbouring tensors at a time, in this way introducing a variable bond dimension which may be increased as necessary. The SVD necessary to split the merged MPS tensors again will, however, usually introduce some sort of truncation *after* the orthogonalisation. For this reason, one should first use the two-site orthogonalisation and then a few additional sweeps with the single-site orthogonalisation. The latter does not require truncations and hence orthogonalises exactly.

Second, when starting at the boundary of a MPS, the variational space may be very small (e.g. only containing d parameters on the first site). Orthogonalising a state against multiple other states while restricted to such a small space may prove impossible. Our solution to this problem is to only orthogonalise against certain states on an initial sweep if the norm of the resulting tensor R_i does not become too small. Once the centre of the MPS chain is reached, there are usually sufficiently many parameters to allow full orthogonalisation. During a second sweep, this criterion is then ignored, resulting in a null state only where necessary due to the imposed restrictions. If there are many

Algorithm 2.3 Variational orthogonalisation of a MPS $\{\Psi_i\}_{i=1}^N$ against a series of MPS $\{|o^j\rangle\}_j$ using the single-site variant. $|o^j\rangle$ do not have to be orthogonal among each other. A two-site variant would work equivalently and simply merge two neighbouring MPS tensors into one prior to all operations and split them afterwards using a SVD.

procedure VARIATIONALLY-ORTHOGONALISE(MPS $\{\Psi_i\}_{i=1}^N$, M MPS $\{\{O_i^j\}_{i=1}^N\}_{j=1}^M$)
  Right-normalise the MPS $\{\Psi_i\}_i$
  $R_i \leftarrow \Psi_i \quad \forall \quad i \in [1, N]$ ▷ Initialise returned MPS $\{R_i\}$
  $\psi_0^L \leftarrow 1_{m_0^\Psi}^{m_0^R} \qquad \psi_{L+1}^R \leftarrow 1_{m_{L+1}^R}^{m_{L+1}^\Psi}$ ▷ Dummy leftmost/rightmost contraction
  for $i \in [L, 2]$ **do** $\psi_i^R \leftarrow R_i^\dagger \cdot \Psi_i \cdot \psi_{i+1}^R$ **end for** ▷ iterative overlaps from right
  for $j \in [1; M]$ **do** ▷ Add orthogonal states one-by-one
    $o_0^{j;L} \leftarrow 1_{m_0^{O^j}}^{m_0^R} \qquad o_{L+1}^{j;R} \leftarrow 1_{m_{L+1}^R}^{m_{L+1}^{O^j}}$ ▷ Dummy leftmost/rightmost contraction
    for $i \in [L, 2]$ **do** $o_i^{j;R} \leftarrow R_i^\dagger \cdot O_i^j \cdot o_{i+1}^{j;R}$ **end for** ▷ iterative overlaps from right
    while not converged **do**
      for $i \in [1, L-1]$ **do** ▷ Left-to-right sweep
        $\psi_E \leftarrow \psi_{i-1}^L \cdot \Psi_i \cdot \psi_{i+1}^R$
        for $k \in [1, j]$ **do** $o_E^k \leftarrow o_{i-1}^{k;L} \cdot O_i^k \cdot o_{i+1}^{k;R}$ **end for**
        $R_i \leftarrow$ ORTHOGONALISE(ψ_E, $\{o_E^k\}_{k=1}^j$, is first while-loop iteration?)
        Left-normalise $R_i \to R_i^A, T; \quad R_i \leftarrow R_i^A; \quad R_{i+1} \leftarrow T \cdot R_{i+1}$
        $\psi_i^L \leftarrow \psi_{i-1}^L \cdot R_i^\dagger \cdot \Psi_i$
        for $k \in [1, j]$ **do** $o_i^{k,L} \leftarrow_{i-1}^{k;L} \cdot R_i^\dagger \cdot O_i^k$ **end for**
      end for
      for $i \in [L, 2]$ **do** ▷ Right-to-left sweep
        $\psi_E \leftarrow \psi_{i-1}^L \cdot \Psi_i \cdot \psi_{i+1}^R$
        for $k \in [1, j]$ **do** $o_E^k \leftarrow o_{i-1}^{k;L} \cdot O_i^k \cdot o_{i+1}^{k;R}$ **end for**
        $R_i \leftarrow$ ORTHOGONALISE(ψ_E, $\{o_E^k\}_{k=1}^j$, false)
        Right-normalise $R_i \to T, R_i^B; \quad R_i \leftarrow R_i^B; \quad R_{i-1} \leftarrow R_{i-1} \cdot T$
        $\psi_i^R \leftarrow \psi_{i+1}^R \cdot R_i^\dagger \cdot \Psi_i$
        for $k \in [1, j]$ **do** $o_i^{k,R} \leftarrow_{i+1}^{k;R} \cdot R_i^\dagger \cdot O_i^k$ **end for**
      end for
    end while
  end for
  return $\{R_i\}_{i=1}^N$
end procedure
procedure ORTHOGONALISE(ψ_E, $\{o_E^k\}_{k=1}^j$, bool careful)
  $\left(o^\dagger o\right)_{kl} \leftarrow {o_E^k}^\dagger \cdot o_E^l \quad \forall \quad k, l \in [1, j]$
  $\left(o^\dagger \psi\right)_k \leftarrow {o_E^k}^\dagger \cdot \psi_E \quad \forall \quad k \in [1, j]$
  $\lambda \leftarrow \left(o^\dagger o\right)^{-1} \cdot \left(o^\dagger \psi\right)$ ▷ Use SVD here, $\left(o^\dagger o\right)$ may be degenerate
  for $k \in [1, j]$ **do**
    if $\neg$ careful $\vee$ $||\psi_E - \lambda_k \cdot o_E^k||/||\psi_E|| > 10^{-3}$ **then**
      $\psi_E \leftarrow \psi_E - \lambda_k \cdot o_E^k$ ▷ Do not kill ψ_E in first iteration
    end if
  end for
  return ψ_E
end procedure

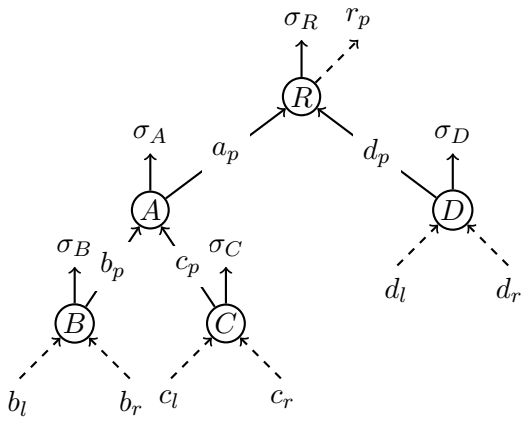


Figure 2.9: A binary tree tensor network state of five sites. The child legs of tensors B, C and D are dummy legs associated to the vacuum and behave the same way as the rightmost MPS leg m_L. The parent leg of the root node, r_p, contains the transformation information of the state and is equivalent to the leftmost MPS leg m_0. The interconnecting legs b_p, c_p, a_p and d_p take the role of MPS bond indices. Each tensor is associated to a physical site index σ_i.

target-orthogonal states, it is usually helpful to add them one-by-one to the set of states against which the sweeping procedure orthogonalises to avoid too many constraints hitting the unoptimised state at once. In any case, it is usually beneficial to sweep multiple times over the system due to the variational nature of the approach.

2.3.6 Generalisation to Tree Tensor Networks

Tree tensor networks in this context are connected, loop-free networks of tensors. Each tensor may have one or more child tensors and each but one tensor has exactly one parent tensor. For implementation purposes, in particular in the context of SyTen, it is advantageous to concentrate on binary tree tensor network states (BTTS): Each tensor is a rank-4 tensor with one physical index, 2 child indices and one root index. The latter three take the role of the previous MPS bond indices. If the tensor has no children or if it is the root node, the associated indices are dummy indices much like the rightmost and leftmost MPS indices. Fig. 2.9 gives an example of a BTTS with five tensors.

Such loop-free networks behave largely the same way as standard MPS: Nodes can be *normalised* in one of three directions (towards the left child, the right child and its parent) rather than two, but apart from this difference, every concept carries over as is. In particular, it is possible to efficiently calculate the Schmidt decomposition and hence optimal truncation at a given bond by appropriately normalising all other nodes to point towards this bond. Furthermore, the contraction of one state with another is equally efficiently possible by working upwards over each leg and discarding child leg contractions once the parent has been calculated.

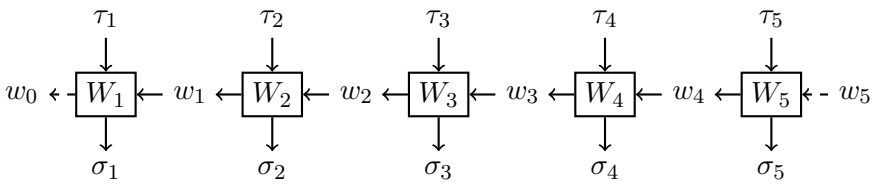


Figure 2.10: Graphical representation of a matrix-product operator on five sites. The MPO acts on a MPS from above by connecting the physical legs of the MPS with the lower physical legs of the MPO.

2.4 Matrix-Product Operators

This section starts with a definition of *matrix-product operators* (MPOs) in Sec. 2.4.1 and how to construct MPOs representing single-site operators (Sec. 2.4.2). It continues with an overview of MPO-MPO and MPO-MPS arithmetic (Sec. 2.4.3) as well as MPO compression (Sec. 2.4.4) and concludes with some examples of MPOs constructed by combinations of these ingredients. While Secs. 2.4.1 and 2.4.3 are mostly a review of well-known facts, Sec. 2.4.4 contains an investigation of the failure of SVD compression when applied to MPOs carried out during this PhD thesis as well as a newly-designed MPO compression algorithm.

2.4.1 Definition

Similar to matrix-product states, matrix-product operators are a potentially extremely efficient way to represent large matrices. They are in particular useful to define the Hamiltonian matrices describing the behaviour of one-dimensional quantum-mechanical systems.

Given a Hilbert space $\mathcal{H} = \otimes_i^L \mathcal{H}_i$ with $\{|\sigma_i\rangle\}_{\sigma_i}$ being a set of d_i basis states of the space $\mathcal{H}_i$, we can write any operator $\hat{H} : \mathcal{H} \to \mathcal{H}$ as

$$\hat{H} = \sum_{\sigma_1\tau_1} \sum_{\sigma_2\tau_2} \cdots \sum_{\sigma_L\tau_L} c^{\sigma_1\sigma_2\cdots\sigma_L}_{\tau_1\tau_2\cdots\tau_L} |\tau_1\rangle \otimes |\tau_2\rangle \otimes \cdots \otimes |\tau_L\rangle \langle\sigma_1| \otimes \langle\sigma_2| \otimes \cdots \otimes \langle\sigma_L| \ . \quad (2.4.1)$$

The rank-$2L$ tensor c of size $\prod_i d_i^2$ can then be decomposed in much the same way as a state $|\psi\rangle$ into a series of L rank-4 tensors $\{W^{\sigma_i w_i}_{i;\tau_i w_{i-1}}\}_i$, namely (cf. Fig. 2.10)

$$c^{\sigma_1\sigma_2\cdots\sigma_L}_{\tau_1\tau_2\cdots\tau_L} = \sum_{w_1} \sum_{w_2} \cdots \sum_{w_L} W^{\sigma_1 w_0}_{1;\tau_1 w_1} W^{\sigma_2 w_1}_{2;\tau_2 w_2} \cdots W^{\sigma_L w_{L-1}}_{L;\tau_L w_L} \quad . \quad (2.4.2)$$

In the same way that the leftmost MPS bond index m_0 labels the transformation properties of the state, the leftmost MPO bond index w_0 defines the transformation properties of the operator. A particle creation operator will for example transform as $N = 1$ on w_0. This IREP combines with the IREP of the state if we apply the operator to a state to

result in (one or more[viii]) IREPs of the target state.

Of course, much as for generic states $|\psi\rangle$, for a generic operator $\hat{H}$, the MPO bond dimensions w_i will grow exponentially with system size. However, for most physical Hamiltonians with finite-range interactions in one dimension, this size is constant and very small (e.g. $|w_i| = 5$). For another, very large class of physical Hamiltonians, modelling systems as diverse as the two-dimensional Fermi-Hubbard model on a cylinder or the full four-operator interaction from quantum chemistry, the MPO bond dimension grows at most polynomially (linear in the former and quadratic in the latter case) in system size. Usually, if there is reason to assume that the ground-state of a Hamiltonian can efficiently be represented as a matrix-product state, then the Hamiltonian itself can efficiently be represented as a matrix-product operator.

The construction of a MPO representation of a given matrix can, roughly speaking, proceed using three different approaches: First, if the matrix is given as a dense $\mathbb{K}^{n\times n}$ matrix, a series of SVD factorisations can be used to achieve a MPO representation of the desired length. In the context of condensed matter physics, this approach is irrelevant, as $n \sim d^L$ and hence by far too large even for systems of moderate size $L \approx 100$. Second, if an analytic expression of the matrix is available, e.g. of the form $\sum_i \hat{s}_i \cdot \hat{s}_{i+1}$, it is possible to construct a finite-state machine (FSM) that corresponds closely to the MPO form. For details on this approach, see Ref. [21] pp. 142. It works generally very well as long as the system is sufficiently simple to capture all rules required to construct the MPO in a small FSM. However, once the system becomes more complicated, the complexity of the FSM grows very quickly.[11] In turn, a considerable amount of effort has to be put into every new model under study which makes the method not very generic. Third, it is possible to construct MPO representations using only the analytical representation without resorting to finite-state machines.

This last approach has been published as part of this thesis in Ref. [1] and will be presented here in three steps: We start with the definition of single-site operators. By implementing MPO arithmetic, it becomes possible to construct e.g. $\hat{H} = \sum_i \hat{c}_i^\dagger \cdot \hat{c}_{i+1} + \text{h.c.}$ via a series of multiplications and additions in a very simple fashion (especially when using a programming language implementing object-oriented semantics such as `C++`). Finally, to achieve the most efficient MPO representation, a compression method is used, of which we present three together with a suggestion for an in our experience optimal compression procedure based on combining two of these compression methods.

2.4.2 MPOs for Single-Site Operators

Single-site operators (SSO) are those operators which act nontrivially only on a single site, examples include the particle annihilator $\hat{c}_i$ on site i, the spin operator along the

[viii] For example, applying a $S = 1$ operator on a $S = 1$ state will result in three states transforming as $S = 0$, $S = 1$ and $S = 2$. In principle, it is also possible to represent such nonuniquely transforming states with MPS, but one usually projects into just a single sector instead.

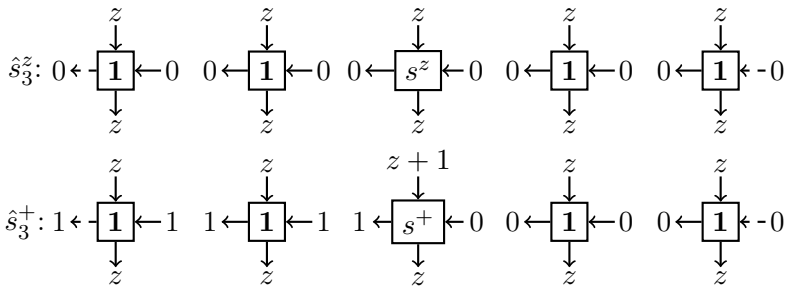


Figure 2.11: Graphical representation of MPOs for $\hat{s}_3^z$ and $\hat{s}_3^+$ on a five-site system. Numbers $1, 0$ and letters z denote the incoming and outgoing S^z quantum numbers on each tensor leg. By convention, the leftmost MPO bond index w_0 transforms the same as the represented operator, while the rightmost MPO bond index w_L always transforms as the vacuum of the system. The MPO acts on the MPS below it, mapping states with $S^z = z$ to those with $S^z = z + 1$ in the second example on the third site.

z-axis $\hat{s}_i^z$ on site i or the local particle number operator $\hat{n}_i$ on site i. The site i is called the *active* site, since the operator acts there. The MPO representations of SSO are very simple, as they have MPO bond dimension 1 throughout the system. They are hence fully described by a set of L matrices of size $d_i \times d_i$.

Bosonic Single-Site Operators

The easiest case is that of bosonic operators transforming trivially with respect to the implemented symmetries, e.g. $\hat{s}_i^z$ or $\hat{n}_i$. If no symmetries are implemented, all bosonic operators behave this way. To construct the MPO representation of $\hat{s}_i^z$, we only need the local representation, given as $\mathrm{diag}(-S, -S+1, \ldots, S-1, S)$, and an identity operator acting on each site but i. In the case of a homogenous system, these are all the same and we can construct the MPO for $\hat{s}_i^z$ as

$$W_{<i} = \mathbf{1}_i \quad W_i = s_i^z \quad W_{>i} = \mathbf{1}_i \quad . \tag{2.4.3}$$

This construction can occur on-the-fly so that we only have to store one identity tensor of size $(1, 1, d, d)$ and one active site tensor of the same size.

Operators which do change a quantum number, such as $\hat{s}_i^+$ or $\hat{c}_i^\dagger$, are more complicated. Since each tensor has to locally preserve symmetries and hence quantum numbers, the additional quantum number must be carried from the active site i to the left (or right) edge of the system, depending on convention. In turn, the chain of identity operators to the left of the active site must allow for this quantum number on their MPO bond indices, while those on the right of the active site only carry the vacuum quantum numbers (cf. Fig. 2.11). Since those quantum numbers are part of the tensor structure, it is necessary to store different identity operator tensor representations for the left and right half of the system. Because all involved tensors are very small, it is possible to always store left-

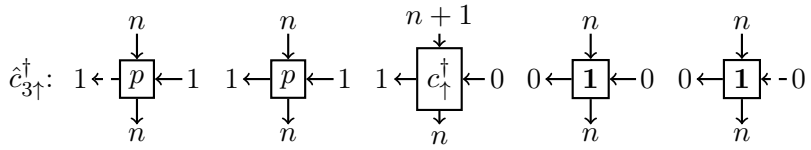


Figure 2.12: Graphical representation of the fermionic creation operator $c^\dagger_{3\uparrow}$ on a five-site system. Labels correspond to the fermionic particle quantum numbers. To the left of site 3, the identity tensors $\mathbf{1}$ have been replaced by parity tensors p implementing the non-local CAR.

and right identities and avoid the additional logic to differentiate between operators with identical left- and right identity chains and those with differing chains.

Fermionic Single-Site Operators

The *local* anti-commutation relations of fermionic operators can be implemented by a suitable definition of the active site tensor. However, non-local anti-commutation relations also have to be implemented.[47] To do so, we first employ the natural ordering of MPO sites from e.g. left to right through the system to define an ordering of the fermionic operators. It is then sufficient to place parity operators instead of identity operators either to the left or right of the active site (cf. Fig. 2.12). Constructed in such a fashion, the MPO representing a fermionic operator can be treated in exactly the same way as any other MPO.

Non-Homogenous Systems

It is sometimes useful to model a non-homogenous system. For example, this could be a spin chain with alternating $S = 1$ and $S = 1/2$ spins[48,49] or a system with alternating bosonic and fermionic sites modeling electrons and phonons.[50] Alternatively, the underlying system could also be homogenous, but due to the explicit implementation of quantum numbers (e.g. momentum k along a ring), it becomes necessary to assign different sites different quantum numbers: an electron on the i-th site has momentum $2\pi i/L$, such that a state with an electron on the first site has (say) total quantum numbers $N = 1$, $S^z = 1/2$, $k = 2\pi/L$ while an electron on the third site has total quantum numbers $N = 1$, $S^z = 1/2$, $k = 6\pi/L$.[10] To implement this properly, we have to assign the local one-electron basis states on site one the momentum quantum number $2\pi/L$ while the local one-electron basis states on site three have momentum quantum number $6\pi/L$.

Such a system can be implemented by associating a specific type to each site with the requirements that all sites of the same type carry the same local Hilbert space. The $S = \{1, 1/2\}$ spin chain would have two types of sites, as would the mixed bosonic/fermionic system while the momentum-space ring would have L different types of sites.

Then, to be able to construct a MPO representation of a single-site operator, we need

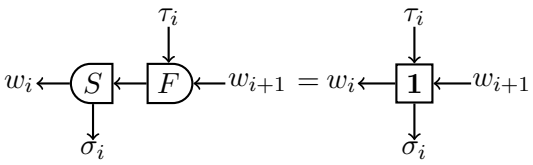


Figure 2.13: An identity MPO component with arbitrary non-abelian symmetries generated from a fusing and a splitting tensor.

to define an active site tensor for each type of site and, for each type of site on which the operator could act, a proper left and right identity for all sites to the left and right of that site for a total of $T + 2T^2$ rank-4 tensors for each single-site operator, where T is the number of different types of sites. However, this is not a problem: The tensors are all of dimension $(1, 1, d_i, d_i)$ with d_i typically small. Additionally, the number of types T is usually much smaller than the total system size – even in the case of the momentum-space system, one often combines multiple rings into a hybrid-space cylinder of a given length.[11] The total number of types is then given by the circumference of the cylinder, which is often bounded by 10 or at most 20 due to computational costs associated with wider cylinders.[25,51]

Non-Abelian Single-Site Operators

The representations of single-site operators which transform according to a non-abelian symmetry are often the only place where one has to explicitly handle the Clebsch-Gordan coefficient spaces. Two possible avenues exist:

First, by generating a fusing and a splitting tensor, it becomes possible to generate a rank-4 identity MPO component with two physical indices and two MPO bond indices (cf. Fig. 2.13). This procedure is independent of the symmetries involved and works for both abelian as well as non-abelian symmetries. Care must be taken to correctly order the legs as to connect the physical and auxiliary indices with each other as opposed to connecting a physical to an auxiliary leg. If the MPO in question is diagonal in the local basis (e.g. occupation number $\hat{n}$ or parity $\hat{p}$), one can then multiply each block of the generated identity by the appropriate prefactor (for $\hat{n}$, 0, 1 and 2) to build the local component of the corresponding MPO tensor. This approach is *also* applicable if the left and right MPO bond basis are not identical. In this case, the Clebsch-Gordan tensors will have the correct structure for the MPO component in question (e.g. for a spin operator going from $S = 0$ on the RHS MPO basis to $S = 1$ on the LHS MPO basis), but additional tweaking of coefficients will still be necessary.

Second, the MPO component tensors can of course be defined explicitly. This is often a faster and more flexible method to build the correct MPOs than the generation of the fusing and splitting tensors. However, it does require a bit of work up-front to figure out the correct prefactors. The $\hat{c}$ spinor operator which annihilates particles in the

Fermi-Hubbard model is tabulated in Tab. 2.2a while the $\hat{s}$ local spin-flip operator is given in Tab. 2.2b.

For spin chains, the $\hat{s}$ spin operator is tabulated in Tab. 2.2c for the $S = 1/2$ case and in Tab. 2.2d for the case of $S = 1$. In particular for the spin chains, the generation via fusing and splitting tensors as outlined above combined with the appropriate normalisation factor of $-\sqrt{S^2 + S}$ is preferable to a hard-coded implementation, as one may want to consider arbitrary large values for S.

2.4.3 MPO Arithmetic

Six arithmetic operations involving MPOs and MPS can reasonably be defined. Most of these are either very simple or well-covered elsewhere, so they will only briefly be reviewed. However, there does not seem to be a good definition of the MPO-MPO dot product in the public literature, so it will be covered slightly more extensively here.

These operations can be used to iteratively build up a MPO given an analytical description of the operator. For example, to build a MPO representation of $\hat{H} = \sum_{i=1}^{L} \hat{s}_i^\dagger \cdot \hat{s}_{(i+1)\%L}$, we accumulate L results of dot products of SSO-MPOs and then sum those products together. The resulting MPO will represent $\hat{H}$.

Scalar Multiplication

The multiplication of a MPO $\{W_i\}_i$ by a scalar α proceeds exactly as in the MPS case: The dense blocks of the first tensor W_1 are multiplied by α. As a slight improvement to evenly spread that value throughout the tensor, it can be considered to multiply W_1 by $\alpha/\sqrt{|\alpha|}$ and W_L by $\sqrt{|\alpha|}$. In the case of a SSO-MPO which acts trivially everywhere but on the k-th site, it is best to multiply the tensor on that site by α.

MPO-MPO Scalar Product

The scalar product of two MPOs $\hat{A} = \{A_i\}_i$ and $\hat{B} = \{B_i\}_i$ can be written as

$$\mathrm{tr}\left(\hat{A}\hat{B}^\dagger\right) = \sum_{\boldsymbol{\sigma\tau ww'}} A^{\sigma_1 w_0}_{1;\tau_1 w_1} B^{\star\tau_1 w'_1}_{1;\sigma_1 w'_0} \cdots A^{\sigma_L w_{L-1}}_{L;\tau_L w_L} B^{\star\tau_L w'_L}_{L;\sigma_L w'_{L-1}} \quad . \tag{2.4.4}$$

This trace can in particular be used to calculate the Frobenius norm of an operator, but is much more rarely used than the scalar product of two MPS.

MPO-MPO Multiplication

Given two MPOs $\hat{A} = \{A_i\}_i$ and $\hat{B} = \{B_i\}_i$, it is straightforward to calculate the MPO representation of the operator $\hat{R} = \hat{A}\hat{B}$ (with $\hat{B}$ applied first). Each pair of site tensors A_i and B_i is contracted over the upper (for B_i) or lower (for A_i) index and the two left and right indices are merged into one fat index using a fusing tensor on the left hand and

Table 2.1: The active MPO component tensors of different non-abelian spinors are written out, a single block of the tensor per line. Columns of the tables are the quantum numbers of the right, left, up and down MPO tensor leg and the Clebsch-Gordan coefficient tensor blocks. Subscripts of the quantum numbers are the sizes of the associated Clebsch-Gordan subspaces and hence the corresponding size of the CGC tensor. Indices of the CGC tensor blocks c^i are zero-indexed and in the same order as the columns, i.e. r, l, u, d. The reduced, dense blocks are always singleton tensors with entry 1 and hence not given here. For the definition of the dot product $\cdot$ see Sec. 2.4.3; in particular, note that for the dot product $\hat{A}^\dagger \cdot \hat{A}$, only the MPO form of $\hat{A}$ is required, *not* the definition of the corresponding creator $\hat{A}^\dagger$.

(a) The $\hat{c}_i = (-\hat{c}_{i\downarrow}, \hat{c}_{i\uparrow})^T$ annihilation operator in a $\mathrm{U}(1)_{\mathrm{Charge}} \times \mathrm{SU}(2)_{\mathrm{Spin}}$-symmetric Fermi-Hubbard system. The hopping term between sites i and j is $\hat{c}_i^\dagger \cdot \hat{c}_j|_{S=0} = \sum_{\sigma=\uparrow,\downarrow} c_{i\sigma}^\dagger c_{i\sigma}$. Note that $\hat{c}_i^\dagger \cdot \hat{c}_i|_{S=0} = \hat{n}_i$, while $\sqrt{3/4}\hat{c}_i^\dagger \cdot \hat{c}_i|_{S=1} = \hat{s}_i$. The minus sign in the first CGC tensor corresponds to the minus sign arising from the implementation of the local anticommutation relation.

Q_R	Q_L	Q_U	Q_D	CGC
$(0,0)_{1\times1}$	$(-1,1/2)_{1\times2}$	$(0,0)_{1\times1}$	$(1,1/2)_{1\times2}$	$c_{0,0,0,1} = -1\ ;\ c_{0,1,0,0} = +1$
$(0,0)_{1\times1}$	$(-1,1/2)_{1\times2}$	$(1,1/2)_{1\times2}$	$(2,0)_{1\times1}$	$c_{0,0,0,0} = +1\ ;\ c_{0,1,1,0} = +1$

(b) The $\hat{s}_i$ local spin flip operator in the same $\mathrm{U}(1)_{\mathrm{Charge}} \times \mathrm{SU}(2)_{\mathrm{Spin}}$-symmetric Fermi-Hubbard system as above. This operator should be the same as $\sqrt{3/4}\hat{c}_i^\dagger \cdot \hat{c}_i|_{S=1}$.

Q_R	Q_L	Q_U	Q_D	CGC
$(0,0)_{1\times1}$	$(0,1)_{1\times3}$	$(1,1/2)_{1\times2}$	$(1,1/2)_{1\times2}$	$c_{0,0,0,1} = -\sqrt{1/2}\ ;\ c_{0,1,0,0} = 1/2$ $c_{0,1,1,1} = -1/2\ ;\ c_{0,2,1,0} = \sqrt{1/2}$

(c) The $\hat{s}_i$ local spin operator of a $S = 1/2$ Heisenberg spin chain. The SU(2)-invariant spin-spin interaction is $\hat{s}_i \cdot \hat{s}_j|_{S=0} = \sum_{a=x,y,z} \hat{s}_i^a \hat{s}_j^a$. Note that this operator is essentially the same as the spin-flip operator of the Fermi-Hubbard model.

Q_R	Q_L	Q_U	Q_D	CGC
$(0)_1$	$(1)_3$	$(1/2)_2$	$(1/2)_2$	$c_{0,0,0,1} = -\sqrt{1/2}\ ;\ c_{0,1,0,0} = 1/2$ $c_{0,1,1,1} = -1/2\ ;\ c_{0,2,1,0} = \sqrt{1/2}$

(d) The $\hat{s}_i$ local spin operator of a $S = 1$ Heisenberg (or Haldane) spin chain. The SU(2)-invariant spin-spin interaction is $\hat{s}_i \cdot \hat{s}_j|_{S=0} = \sum_{a=x,y,z} \hat{s}_i^a \hat{s}_j^a$.

Q_R	Q_L	Q_U	Q_D	CGC
$(0)_1$	$(1)_3$	$(1)_3$	$(1)_3$	$c_{0,0,0,1} = -1\ ;\ c_{0,0,1,2} = -1$ $c_{0,1,0,0} = 1\ ;\ c_{0,1,2,2} = -1$ $c_{0,2,1,0} = 1\ ;\ c_{0,2,2,1} = 1$

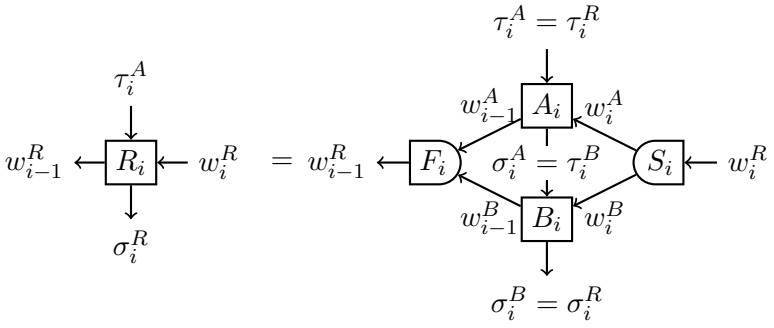


Figure 2.14: Product of two MPOs for the tensors on a single site i. The product is built the same way on all sites $i \in [1, L]$ of the system. Matching physical indices are contracted and the two left and right MPO bond indices merged into one on each side.

a splitting tensor on the right hand. This is best represented graphically, cf. Fig. 2.14. It is noteworthy that the MPO bond dimension will increase to the product of the two input MPO bond dimensions, i.e. $w_i^R = w_i^A w_i^B$.

MPO-MPO Addition

The sum of two MPOs $\hat{A} = \{A_i\}_i$ and $\hat{B} = \{B_i\}_i$, $\hat{R} = \hat{A} + \hat{B}$ is defined exactly the same as for MPS with result tensors (written as matrices in the MPO bond dimensions):

$$R^{\sigma_1}_{1;\tau_1} = \begin{pmatrix} A^{\sigma_1}_{1;\tau_1} & B^{\sigma_1}_{1;\tau_1} \end{pmatrix} \tag{2.4.5}$$

$$R^{\sigma_i}_{1<i<L;\tau_i} = \begin{pmatrix} A^{\sigma_i}_{i;\tau_i} & 0 \\ 0 & B^{\sigma_i}_{i;\tau_i} \end{pmatrix} \tag{2.4.6}$$

$$R^{\sigma_L}_{L;\tau_L} = \begin{pmatrix} A^{\sigma_L}_{L;\tau_L} \\ B^{\sigma_L}_{L;\tau_L} \end{pmatrix} \quad . \tag{2.4.7}$$

The resulting MPO bond dimension is the sum of the two input bond dimensions, $w_i^R = w_i^A + w_i^B$.

MPO-MPS Application

Naively, the product of a MPO $\hat{O} = \{O_i\}_i$ with a MPS $|\psi\rangle$ can be evaluated very similarly to the product of two MPOs, except that the lower "MPO" is actually a MPS and hence only has one physical index. The resulting state has a very large bond dimension $m_i^R = w_i^O m_i^\psi$, which is usually immediately truncated to some $\tilde{m}_i^R \ll m_i^R$. It is hence advisable to pick a different application strategy, e.g. the zip-up algorithm (cf. Ref. [52], Sec. 2.1.3) or a variational approach (cf. Ref. [21], Sec. 4.5.2). Especially the former algorithm has proven to be very successful and essentially as accurate as the naive method in practice while the intermediate tensors produced by it are only of size $\tilde{m}_i^R \times d_i \times w_i^O m_i^\psi$

compared to the large tensors of size $w_i^O m_i^\psi \times d_i \times w_i^O m_i^\psi$ produced by the naive application method.

For MPOs with a relatively large bond dimension, it may be better to settle on a variational optimisation of a MPS to represent the result of the operator application. This proceeds in exactly the same way as the variational orthogonalisation described in Sec. 2.3.5, except that the "target" environment tensor is given by a contraction of the input MPS, input MPO and current estimate MPS rather than just the input MPS and current estimate MPS. To introduce flexibility in the MPS bonds, it is advisable to introduce a subspace expansion step as described in Sec. 3.4.1, in effect growing the bond dimension to the minimal necessary $\tilde{m}_i^R$ instead of first generating a very large bond dimension and then truncating it. Notably, with this variational operator application, it is possible to simultaneously orthogonalise against a set of MPS as well with exactly the same procedure as described in Sec. 2.3.5.

MPO-MPO Dot Product

The dot product of two MPOs is again an operator. It is particularly useful to implement, in a SU(2)-invariant way, the usual term

$$\hat{c}_i^\dagger \cdot \hat{c}_j = \begin{pmatrix} \hat{c}_{i;\uparrow}^\dagger & \hat{c}_{i;\downarrow}^\dagger \end{pmatrix} \cdot \begin{pmatrix} \hat{c}_{j;\uparrow} \\ \hat{c}_{j;\downarrow} \end{pmatrix} = \sum_\sigma \hat{c}_{i;\sigma}^\dagger \hat{c}_{j;\sigma} \tag{2.4.8}$$

in Fermionic systems where $\hat{c}$ is a spinor and $\hat{c}^\dagger$ the dual-space conjugate of $\hat{c}$ or the term $\hat{s}_i^\dagger \cdot \hat{s}_j$ in spin systems. It must be stressed that the dual-space conjugates of the spinors $\hat{c}$ or $\hat{s}$ used here are *not* the usual spinors one would write down for e.g. the creation operator $\hat{c}^\dagger$! As such, implementing two spinors $\hat{c}$ and $\hat{c}^\dagger$ and taking their MPO-MPO product $\hat{c}^\dagger \hat{c}$ will *not* work.[31,37] Instead, these expressions are best thought of as fulfilling

$$\langle p| \left(\left(\hat{A}^\dagger \cdot \hat{B} \right) |q\rangle \right) = \left\langle p\hat{A} \middle| \hat{B}q \right\rangle \quad \forall |p\rangle, |q\rangle \quad , \tag{2.4.9}$$

which gives a convenient interpretation in terms of a tensor network that is already available in a straight-forward manner.

Extracting the operator $\hat{A}^\dagger \cdot \hat{B}$ from the tensor network in drawn Fig. 2.15 representing Eq. (2.4.9), and writing it as a MPO is not trivial and requires the insertion of three identities as follows:

Concentrating on the inner block of four tensors, we can insert two identities to bring this into MPO format (cf. Fig. 2.16). Then also notice that the upper leg of each $A_i^\dagger$ connects to the MPS $\langle p|$, i.e. is the *lower* leg of A_i, which means that we have to connect the *upper* leg of the original tensor A with the *upper* leg of B_i during the evaluation of the product. The total connection network inside the MPO to obtain R_i representing $\hat{R} = \hat{A}^\dagger \cdot \hat{B}$ is then given in Fig. 2.17. At the left and right boundary, we have to decide on

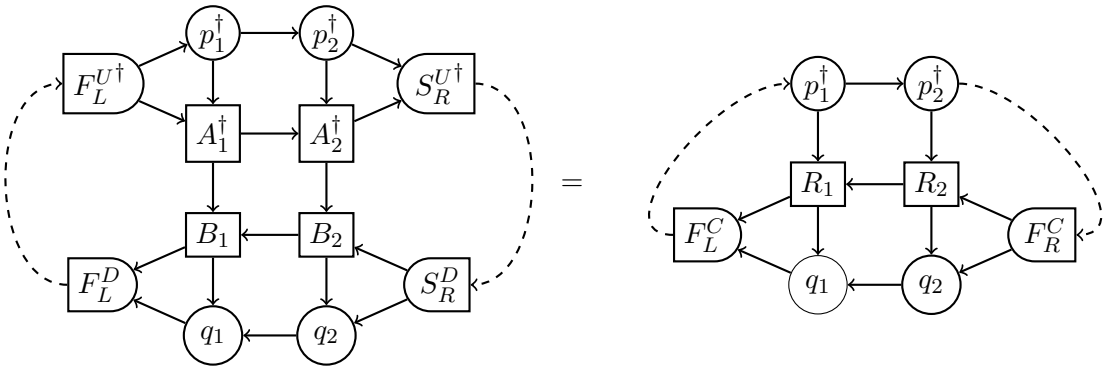


Figure 2.15: The graphical representation of Eq. (2.4.9), which gives rise to the tensor-network expression for MPO-MPO dot products. Also compare with Fig. 2.7, which is the same expression but without the additional operators in the center. The fusing and splitting operators on the inner legs of each MPO-MPS product have already been resolved to identity operators. Labels have been left off, as they are clear from the context (but note that the original *upper* legs of A_i now connect as the *lower* legs of $A_i^\dagger$ to B_i).

a quantum number sector, this is equivalent to taking $\hat{c}^\dagger \cdot \hat{c}$ either in the $S = 0$ (hopping or particle number) or $S = 1$ (spin-flip) sector. Given this sector, we can construct the leftmost tensor combining the two leftmost MPO indices as the split into the w_0^a leg and this sector. Similarly, on the rightmost site, we fuse the vacuum and the rightmost leg w_L^a together and contract the resulting fused leg with w_L^b.

2.4.4 Compression of a Matrix-Product Operator

MPO arithmetic is very useful, in particular, as we will see, to construct MPO representations of more complex operators. However, both addition and the two multiplications (multiplication and dot product) will inevitably grow the bond dimension of a MPO. While this cannot be avoided in general, there is still often room for *compression* of a MPO towards a smaller bond dimension while representing the same operator. The easiest example is the MPO representation of $\hat{A} + \hat{A}$: Obviously, this will have twice the bond dimension of $\hat{A}$, while the same effect could be achieved by multiplying the first tensor of $\hat{A}$ by a factor of two.

For MPS, SVD compression is a perfectly suitable tool, however, when applying the same methodology to a MPO, we face four issues: First, doing a SVD on a matrix with a very sparse structure will in general result in a dense matrix. MPO tensors often have such an extremely sparse structure; preserving that structure may allow for computational gain and a better introspection, as each element of the tensor can easily be associated to some term in the original operator.

Second, MPOs are not normalised, i.e. their Frobenious norm is not one. Indeed, it is easy to see that the Frobenius norm of an identity operator on a L-site system of local

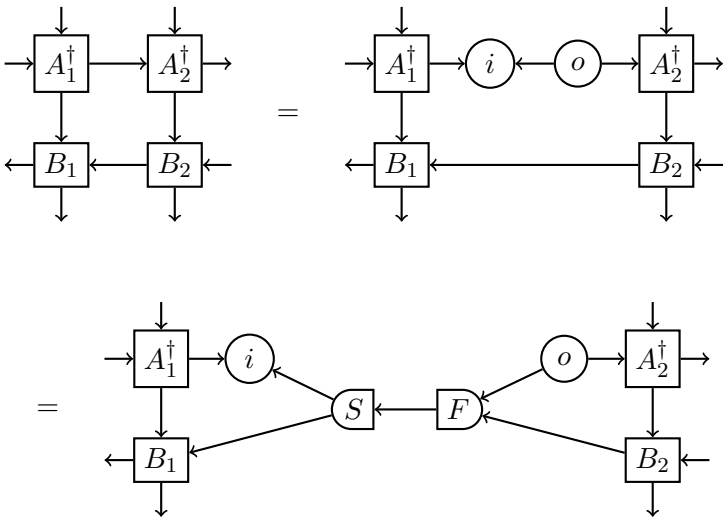


Figure 2.16: The centre part of Fig. 2.15, which would be repeated as necessary for a larger MPO. Insertion of two identities in the form of the input-input/output-output tensors and the splitting/fusing tensors allows us to combine the two inner indices into one to bring this into the standard MPO format.

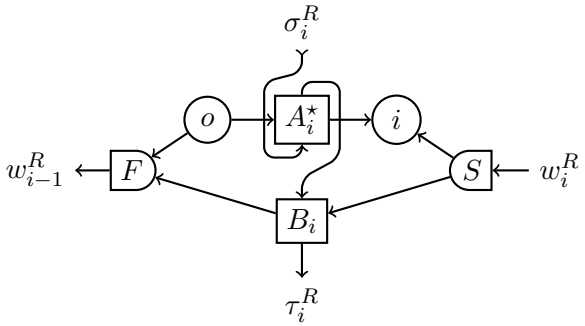


Figure 2.17: The final connection diagram for the inner tensors R_i representing the MPO-MPO dot product. We use $A_i^\star$ to express that, while the directions of each leg have changed, they are still in their original places (i.e. the upper physical leg of A_i is painted upwards, not downwards as is the case in Fig. 2.16). Named indices are those of the resulting tensor R_i.

physical dimension d is $\sqrt{d^L}$, hence exponential in system size. This large norm will be concentrated in the single unnormalised tensor when applying a naive SVD to a MPO – we would hence have $L-1$ tensors with elements of order 1 and orthonormal columns or rows and one tensor with exponentially large elements. That is poison to preserving any degree of accuracy during floating-point calculations.

Third, and as a result of the second point, different MPOs may have wildly different norms. The SVD will truncate based on that norm. Hence compressing the sum $\hat{\mathbf{1}} + \hat{P}$ where $\hat{P}$ projects e.g. on the fully ferromagnetic state (and has norm 1) will result in just $\hat{\mathbf{1}}$ once the system is sufficiently large. This is equivalent to compressing $|\psi\rangle + \varepsilon |\phi\rangle$ with ε very small and $|\phi\rangle, |\psi\rangle$ of norm 1 via SVD. Of course, the small contribution will be discarded and we are left with $|\psi\rangle$.

Fourth, the terms in a MPO often have a very specific meaning to the point that many of the exact zeros mentioned before correspond to unphysical or unwanted interactions (e.g. a three-operator term). Applying a SVD will in general introduce spurious small terms of the order of the machine epsilon there which then could potentially break e.g. particle number conservation against the expectations of the user. When using symmetry-protected tensors to implement all desired symmetries, this is slightly less relevant, but could still give rise to e.g. four-operator interactions in a strictly two-operator Hamiltonian.

In the following, we will present three compression methods to alleviate these problems: the *rescaled SVD* is a trivial extension of standard SVD which avoids the exponential build-up in a single tensor. The *deparallelisation* (DPL) method is able to compress simple MPOs very well and often results in expressions which one would also write down analytically. The *delinearisation* (DLN) is an extension of the DPL and compresses also more complex operators nearly perfectly. Both DPL and DLN avoid the loss of sparsity (and hence the introduction of spurious terms), preserve contributions of very different norms (e.g. $\hat{\mathbf{1}} + \hat{P}$) and spread the norm of the operator evenly throughout the system.

Rescaled SVD

The standard SVD, employed to compress a MPO, would work as follows: First, either via a series of QR decompositions or SVDs without truncation of the singular value spectrum, each tensor from right to left is brought into right-normalised form. Then, the first tensor is decomposed via SVD, resulting in a left-normalised first tensor, a singular value tensor and a transfer tensor. The singular value tensor is truncated to remove singular values associated to numerical noise. This is best detected algorithmically, for most MPOs, there is a clear cut-off in the singular value spectrum. The singular value tensor and the transfer tensor are then multiplied into the second site before the decomposition proceeds

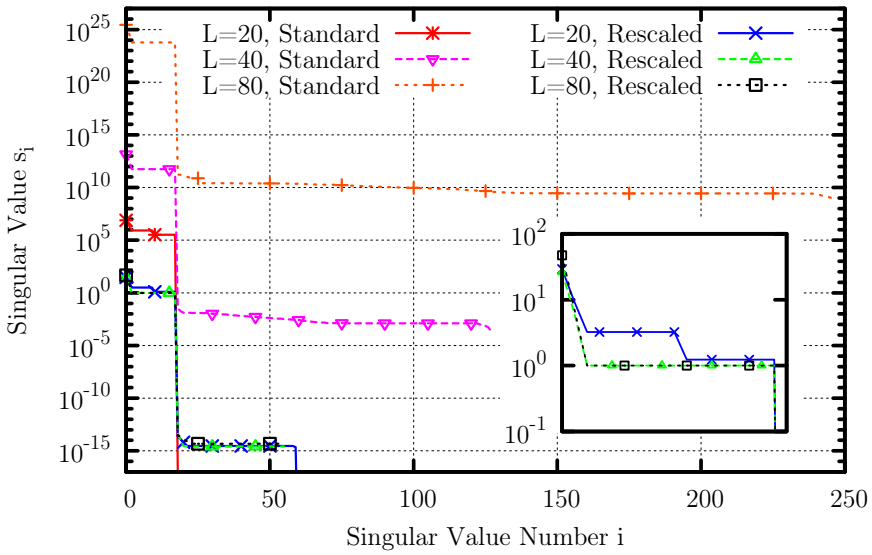


Figure 2.18: The singular value spectrum of a Fermi-Hubbard Hamiltonian on a one-dimensional chain of the specified length on the central bond. Without rescaling, the singular values grow exponentially with system size. With rescaling, the magnitude of the singular values is essentially constant, the remaining differences can be explained by numerical details. (First published in Ref. [1].)

there. Specifically, we have by reshaping, SVD and combination into the new tensors:

$$W^{\sigma_i w_{i-1}}_{i;\tau_i w_i} \to M^{\gamma}_{w_i} \to \left[U^{\gamma}_{w'_i}\right]\left[S^{w'_i}_{w''_i} V^{w''_i}_{w_i}\right] \to W'^{\sigma_i w_{i-1}}_{i;\tau_i w'_i} T^{w'_i}_{w_i} \quad . \tag{2.4.10}$$

The rescaling now introduces a simple factor α here:

$$W^{\sigma_i w_{i-1}}_{i;\tau_i w_i} \to M^{\gamma}_{w_i} \to \left[\alpha U^{\gamma}_{w'_i}\right]\left[\alpha^{-1} S^{w'_i}_{w''_i} V^{w''_i}_{w_i}\right] \to W'^{\sigma_i w_{i-1}}_{i;\tau_i w'_i} T^{w'_i}_{w_i} \quad . \tag{2.4.11}$$

where α is chosen as the average of the singular values in S after compression (i.e. after the near-zero terms have been discarded). The effect of this rescaling can be observed in Fig. 2.18. This figure gives the singular-value spectrum prior to truncation on the central bond of a 1-dimensional Fermi-Hubbard Hamiltonian on systems of varying size. Without rescaling, we can clearly see that the singular values grow exponentially with system size, to the point where the numerical noise grows as large as 10^{14}. With rescaling, however, the singular values are largely independent of system size; hence resulting in the desired behaviour.

Deparallelisation

The deparallelisation (DPL) of a tensor with respect to a given leg is based on the same overall framework as described in Alg. 2.2 except that the decomposition of the dense

matrix via SVD (Line 14) is replaced by a deparallelisation of the dense matrix. We will hence only present the decomposition via deparallelisation of a dense matrix here. The DPL results in a single transfer matrix and tensor, much like the QR decomposition, instead of the two matrices/tensors generated by the SVD. We want to have a decomposition $M \to M'$; T fulfilling the properties:

- all columns of M' are also columns of M
- no two columns of M' are parallel to each other
- $M_{ab} = \sum_c M'_{ac} T_{cb}$.

Such a decomposition can be achieved using Alg. 2.4. It has been used extensivley in the Matrix Product Toolkit[53] but to our knowledge not been published elsewhere, in particular in the connection with matrix-product operators.

As an example, consider the matrix

$$M = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 1 & 0 & 3 \end{pmatrix} \quad . \tag{2.4.12}$$

When supplied to the DEPARALLELISE algorithm, we would obtain matrices

$$M' = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 1 & 0 \end{pmatrix} \quad ; \quad T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad . \tag{2.4.13}$$

In practice, the matrices encountered during MPO compression usually have many more parallel columns, which gives this method a certain degree of effectivity and makes it fully sufficient in the case of relatively simple MPOs. However, as can be observed in the example above, it is not guaranteed to obtain the optimal result. Clearly there is more room for compression by not only deparallelising columns but also delinearising them – that is, to remove linearly dependent columns (such as the third column of M).

Algorithm 2.4 The deparallelisation algorithm applied to a dense matrix. To apply it to a tensor, substitute it in Alg. 2.2 Line 14. An equivalent method can be defined to deparallelise the rows as DEPARALLELISE-ROWS.

1: **procedure** DEPARALLELISE-COLS(Matrix M)
2: $\mathcal{F} \leftarrow \{\}$ ▷ List of tuples to construct T
3: $\mathcal{K} \leftarrow \{\}$ ▷ List of kept columns
4: **for each** Column c of M **do**
5: **if** $\text{norm}(c) \equiv 0$ **then** ▷ Zero columns proportional to first with factor 0
6: $\mathcal{F} \leftarrow \mathcal{F} + [1, 0]$
7: **continue**
8: **end if**
9: Bool *found* $\leftarrow$ false
10: **for** $i \in [1, |\mathcal{K}|]$ **do** ▷ Search for kept parallel column
11: **if** $c \equiv \alpha \cdot \mathcal{K}[i]$ **then** ▷ c parallel to i-th kept column
12: $\mathcal{F} \leftarrow \mathcal{F} + [i, \alpha]$ ▷ Create c using i-th column with factor α
13: *found* $\leftarrow$ true
14: **break**
15: **end if**
16: **end for**
17: **if** $\neg$*found* **then**
18: $\mathcal{K} \leftarrow \mathcal{K} + c$ ▷ Add c to list of kept columns
19: $\mathcal{F} \leftarrow \mathcal{F} + [|\mathcal{K}|, 1]$ ▷ Create c using $|\mathcal{K}|$-th column with factor 1
20: **end if**
21: **end for**
22: **if** $|\mathcal{K}| = 0$ **then** ▷ No non-zero columns in M
23: **return** $M' = 0^{\text{rows}(M)\times 1}$; $T = 0^{1\times\text{cols}(M)}$ ▷ Zero row and column vectors
24: **end if**
25: $M' \leftarrow \text{HSTACK}(\mathcal{K}) \in \mathbb{K}^{\text{rows}(M)\times|\mathcal{K}|}$
26: $T \leftarrow 0^{|\mathcal{K}|\times\text{cols}(M)}$ ▷ Initialise as zero matrix
27: **for** $i \in [0, |\mathcal{F}|]$ **do** ▷ Construct T from stored factors in $\mathcal{F}$
28: $r \leftarrow \mathcal{F}[i]_1$
29: $T_{ri} \leftarrow \mathcal{F}[i]_2$
30: **end for**
31: **return** M' ; T
32: **end procedure**

Algorithm 2.5 The outer parts of the DLN method. INNER-DELINEARISE is defined in Alg. 2.6.

procedure DELINEARISE(Matrix M, Matrix Δ, Bool *relaxed*)
  if *relaxed* **then**
    Set all $\Delta_{ab} \equiv 0$ to ε_D
  end if
  $(M^C, T^C, \Delta^C) \leftarrow$ INNER-DELINEARISE(M, Δ)
  $(M^{CR}, T^R, \Delta^{CR}) \leftarrow$ INNER-DELINEARISE$(M^{C\dagger}, \Delta^{C\dagger})$
  if $\mathrm{cols}(T^{R\dagger}) < \mathrm{cols}(M^C)$ **then**
    return $T^{R\dagger}$; $M^{CR\dagger} \cdot T^C$
  end if
  if $\mathrm{cols}(M^C) < \mathrm{cols}(M)$ **then**
    return M^C ; T^C
  end if
  return $M, \mathbf{1}^{\mathrm{cols}(M)\times\mathrm{cols}(M)}$
end procedure

Delinearisation

The basic idea of the delinearisation method (DLN) is to find a minimal set of columns in the matrix which are sufficient to reconstruct the original matrix with the side-condition that no cancellation to exact-zero may occur. This latter requirement is necessary to ensure that we do not introduce spurious small terms. To fulfill this requirement, we need to construct a threshold tensor Δ of the same format as our original tensor W_i with elements defined as:

$$\Delta^{\sigma_i w_{i-1}}_{\tau_i w_i} = \varepsilon_D \sum_{\sigma_i \tau_i} \left|\left| W^{\sigma_i w_{i-1}}_{i;\tau_i w_i} \right|\right| \quad , \tag{2.4.14}$$

where ε_D is a small threshold, e.g. 10^{-8}. During Alg. 2.2, the dense blocks of this tensor Δ are reshaped in exactly the same way as the dense blocks of tensor W_i, resulting in two matrices M_{ab} (from W_i) and Δ_{ab} (from Δ). Δ_{ab} gives an error estimate for the M_{ab}. In particular, $\Delta_{ab} \equiv 0$ if M_{ab} arises from a zero operator in the original W_i tensor. This way, we can distinguish zeros in M_{ab} coming from part of an otherwise non-zero operator (e.g. off-diagonal elements of $\hat{n}$) from those coming from exactly-zero operators. The latter would give rise to the non-physical interactions we wish to avoid. However, sometimes it may be beneficial to allow such cancellations to achieve the optimal result. This is called the *relaxed* delinearisation.

The DELINEARISATION algorithm is described in Alg. 2.5. It is relatively complex, as attempts are made to break cyclic linear dependencies without introducing cancellation to zero nor entirely new columns (as a QR decomposition would do). It is split into an outer part, which attempts delinearisation of both rows and columns, an inner part, which deparallelises rows and delinearises the matrix columns and the innermost part which only attempts to find linearly dependent columns. The initial deparallelisation is done to

Algorithm 2.6 The inner part of the matrix delinearisation. See Alg. 2.4 for DEPARALLELISE-ROWS. Δ^p in line 2 are selected as smallest elements from the corresponding column of Δ from non-zero rows which were parallel to the kept row. The permutation of columns in line 3 should result in an as much upper-triangular form as possible.

1: **procedure** INNER-DELINEARISE(Matrix M, Matrix Δ)
2: $(R, M^p, \Delta^p) \leftarrow$ DEPARALLELISE-ROWS(M, Δ)
3: $(M^{pP}, \Delta^{pP}, P) \leftarrow$ Sort columns of M^p
4: $\mathcal{K} \leftarrow \{\}$ ▷ Set of kept columns
5: $\mathcal{C} \leftarrow \{\}$ ▷ Set of coefficients
6: **for each** column $\mu \in M^{pP}$ and associated $\delta \in \Delta^{pP}$ **do**
7: **if** $x \leftarrow$ DELINEARISE-SOLVE$(\mu, \delta,$ HSTACK$(\mathcal{K}))$ **then**
8: $\mathcal{C} \leftarrow \mathcal{C} + x$ ▷ Store coefficients if successful
9: **else**
10: $\mathcal{K} \leftarrow \mathcal{K} + \mu$ ▷ Store column if linearly independent
11: **end if**
12: **end for**
13: $M^{pC} \leftarrow$ STACK$(\mathcal{K})$
14: Construct Δ^{pC} from columns associated to M^{pC} in Δ^{pP}
15: Construct transfer matrix T^C from $\mathcal{C}$ times the permutation matrix P
16: $M^C \leftarrow R \cdot M^{pC}$
17: $\Delta^C \leftarrow R \cdot \Delta^{pC}$
18: **if** $\text{cols}(M^C) = \text{cols}(M)$ **then**
19: **return** M ; Δ ; $\mathbf{1}^{\text{cols}(M) \times \text{cols}(M)}$
20: **end if**
21: **return** M^C ; Δ^C ; T^C
22: **end procedure**
23: **procedure** DELINEARISE-SOLVE(Column μ, Column δ, Matrix A)
24: Remove columns from A which are non-zero where δ is exactly zero
25: **loop**
26: Attempt to solve $Ax = \mu$ for x using QR with row and column scaling
27: **for each** x_i in x **do**
28: **if** $|x_i| < \varepsilon \vee |x_i| > 1/\varepsilon$ **then** ▷ Avoid very small/large x_i
29: $x_i \leftarrow 0$
30: remove i-th column from A
31: **continue** to next outer **loop** iteration
32: **end if**
33: **end for**
34: If any $x_i \approx \pm 1$, set $x_i = \pm 1$
35: **if** each $(Ax - \mu)_i$ of residual is smaller than $\delta_i \times \text{cols}(A)$ **then**
36: **return** coefficients x as success
37: **end if**
38: **return** no coefficients and failure
39: **end loop**
40: **end procedure**

remove as many columns as possible and make the subsequent QR decompositions both faster and more accurate. The sorting of columns will achieve an upper-triangular form if possible and prefer sparser over denser columns. Since the first columns are more likely to be kept, this also increases the sparsity of the resulting tensor. The combination of row- and column delinearisation often succeeds in breaking cyclic linear dependencies and hence achieve optimal compression.

Suggested MPO Compression Procedure

The rescaled SVD method is fairly stable if one only wishes to compress a Hamiltonian or similar operator composed of sums of terms with roughly equal norms. After an initial left-to-right sweep without compression to achieve an orthogonal basis, one can compress the MPO iteratively sweeping right-to-left with compression, either detected based on the singular value cutoff or to some fixed maximal bond dimension or minimal singular value. The resulting MPO will have optimal bond dimension, but small contributions will have been discarded, spurious operators introduced and sparsity will be lost.

Algorithm 2.7 MPO compression algorithm: A full sweep with DPL removes most superfluous terms and speeds up the following, more expensive sweeps with DLN. Only after no further improvements can be done without allowing cancellation to zero do we accept it. Even then, if that allows no improvement either, no spurious terms are introduced.

procedure COMPRESS-MPO(MPO $\{W_i\}$)
  COMPRESS-MPO-SWEEP($\{W_i\}$, *deparallelisation*)
  repeat
    COMPRESS-MPO-SWEEP($\{W_i\}$, *delinearisation*)
  until no change in bond dimensions
  repeat
    COMPRESS-MPO-SWEEP($\{W_i\}$, *relaxed delinearisation*)
  until no change in bond dimensions
  return $\{W_i\}$
end procedure
procedure COMPRESS-MPO-SWEEP(MPO $\{W_i\}$, Method *method*)
  for $i \in [1, L-1]$ **do**
    Decompose with *method*: $W^{\sigma_i w_{i-1}}_{i;\tau_i w_i} \to W'^{\sigma_i w'_{i-1}}_{i;\tau_i w_i} T^{w'_i}_{w_i}$
    $W_i \leftarrow W'_i$
    $W_{i+1} \leftarrow T \cdot W_{i+1}$
  end for
  for $i \in [L, 2]$ **do**
    Decompose with *method*: $W^{\sigma_i w_{i-1}}_{i;\tau_i w_i} \to T^{w_{i-1}}_{w'_{i-1}} W'^{\sigma_i w'_{i-1}}_{i;\tau_i w_i}$
    $W_i \leftarrow W'_i$
    $W_{i-1} \leftarrow W_{i-1} \cdot T$
  end for
end procedure

Alternatively, Alg. 2.7 provides a stable compression method based on DPL and DLN which is relatively fast (the asymptotic scaling of all methods is identical in any case), preserves both sparsity and small contributions and only introduces spurious contributions if absolutely necessary.

2.4.5 Examples of Generated MPOs

The construction method was applied to three exemplary Hamiltonians: nearest-neighbour interactions on a spin chain, the Fermi-Hubbard model on a cylinder in hybrid real-/momentum space and, as a proof of principle, a four-body long-range quantum chemistry toy model Hamiltonian. In all three cases, the construction is very straightforward after the definition of the necessary local tensors. However, the three compression methods (rescaled SVD, DLN and DPL) on their own handle the increasing complexity differently well.

Spin Chain with Nearest-Neighbour Interactions

Consider the Hamiltonian

$$\hat{H} = \sum_{i=1}^{L} h\hat{S}_i^z + \sum_{a=x,y,z} \sum_{i=1}^{L-1} J_a S_i^a S_{i+1}^a \tag{2.4.15}$$

without any explicitly preserved symmetry. It is relatively easy to set up a finite-state machine by hand which corresponds to the MPO representing this Hamiltonian.[21] Alternatively, we can very easily construct it using the generic construction algorithm, even with just DPL-based compression. Plotting the dimension of each bond over the system length (cf. Fig. 2.19), we recover the optimal bond dimension of five in the bulk and even a slight improvement at the edges, where only $|w_1| = |w_{L-1}| = 4$ is strictly necessary. At the same time, introspection of the tensor yields essentially the analytic solution, namely (with $h = J_a = 1$):

$$W_{\text{bulk}} = \begin{pmatrix} 1 & s^z & s^z & s^y & s^x \\ 0 & 1 & 0 & 0 & 0 \\ 0 & s^z & 0 & 0 & 0 \\ 0 & s^y & 0 & 0 & 0 \\ 0 & s^z & 0 & 0 & 0 \end{pmatrix} . \tag{2.4.16}$$

It is also possible to construct powers $\hat{H}^n$ of this Hamiltonian using the standard MPO-MPO multiplication and subsequent compression. We used coefficients $h = 1$, $J_x = \frac{1}{2}$, $J_y = \frac{1}{3}$, $J_z = \frac{1}{5}$. The maximal bond dimensions of the resulting MPOs quickly grow (cf. Tab. 2.3). We compare the results from SVD and DLN with earlier data,[54] where a variational method was used to find an optimal low-dimensional MPO

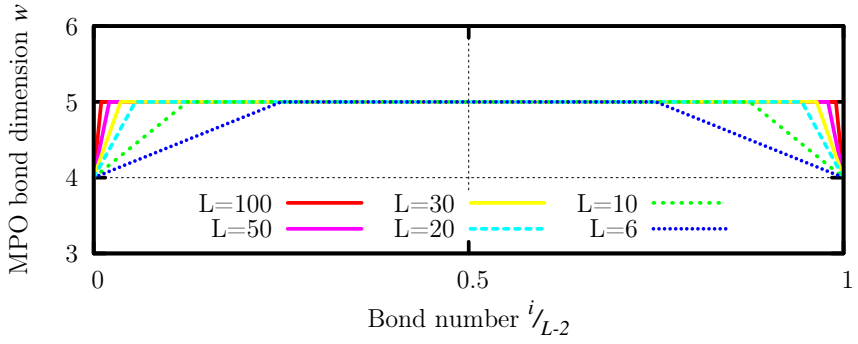


Figure 2.19: The bond dimensions of the MPO representation of $\hat{H}$ from Eq. (2.4.15) plotted over the relative system size for different sizes. The saturation to the optimal result at $|w_i| = 5$ is very obvious. (First published in Ref. [1].)

Table 2.3: Relative sparsity and bond dimension in the center of the $L = 100$ chain for MPO representations of $\hat{H}^n$.

Order $\hat{H}^n$	1	2	3	4	5	6	7
SVD: $w_{L/2}$	5	9	16	32	51	64	92
DLN: $w_{L/2}$	5	9	16	32	51	81	126
DLN: Sparsity	81%	84%	82%	89%	88%	88%	85%
Variational Method:[54] $w_{\max}$	5	9	16	32	51	79	110

representation of the XXZ-Hamiltonian (the DPL on its own quickly resulted in very suboptimal representations).

For small powers, all three methods give identical bond dimensions. Starting at $n = 6$, the SVD consistently gives the smallest bond dimension, while the DLN method results in somewhat larger dimensions than the variational approach. It may be that the SVD discards some very small contributions which are preserved by the other two methods or that the other two methods fail to find the optimal representation. To numerical accuracy, the MPOs generated by SVD and DLN behave identically, however.

It should be noted that when taking very large powers of $\hat{H}$, the singular value spectrum becomes increasingly smooth instead of exhibiting the drastic cut-off seen in Fig. 2.18. It hence becomes also increasingly appropriate to use a singular value threshold or even a maximal bond dimension to find good approximations for $\hat{H}^n$. In this case, the SVD is the method of choice to at least estimate the error compared to the exact representation, which is not readily available from the variational approach. A comparison

$$1 - \frac{\mathrm{tr}\left(\hat{O}_{\mathrm{exact}}\hat{O}_{\mathrm{fit}}\right)}{\sqrt{\left|\left|\hat{O}_{\mathrm{exact}}\right|\right|\left|\left|\hat{O}_{\mathrm{fit}}\right|\right|}} \tag{2.4.17}$$

after the variational optimisation has completed is usually of little value, since the involved large numbers easily lend themselves to numerical cancellation and loss of accuracy.

Fermi-Hubbard Model on a Cylinder in Hybrid Space

The study of two-dimensional models is an area of ongoing research. While tensor network methods do not suffer from the sign problem encountered in Monte Carlo, calculations on large system sizes are still very difficult. The usual approach in the context of DMRG and MPS is to consider a cylinder of some length L and circumference W. The surface of the cylinder is then mapped onto a one-dimensional chain.[25] The periodic boundary condition along the y-axis allows the implementation of a Fourier transformation into momentum space along that axis.[11] After this transformation, it becomes possible to explicitly implement momentum conservation (associated to a group $\mathbb{Z}_W$). This allows for a further reduction of the sizes of the dense blocks in the tensor and much larger bond dimensions than the fully real-space approach. For details of this transformation and the mapping, cf. Sec. 5.3. The result is a Hamiltonian

$$\hat{H}' = -\sum_{x=1}^{L}\sum_{\alpha=1}^{W} 2\cos\left(2\pi\frac{\alpha}{W}\right)\hat{c}^\dagger_{x,\alpha}\cdot\hat{c}_{x,\alpha} \tag{2.4.18}$$

$$+ -\sum_{x=1}^{L-1}\sum_{\alpha=1}^{W}\left(\hat{c}^\dagger_{x,\alpha}\cdot\hat{c}_{x+1,\alpha} + \hat{c}^\dagger_{x,\alpha}\cdot\hat{c}_{x+1,\alpha}\right) \tag{2.4.19}$$

$$+ \frac{1}{2}\sum_{x=1}^{L}\sum_{\alpha=1}^{W}\left(\left[\sum_{\beta\gamma=1}^{W}\frac{1}{W}\hat{c}^\dagger_{x,\alpha}\cdot\hat{c}_{x,\beta}\times\hat{c}^\dagger_{x,\gamma}\cdot\hat{c}_{x,\alpha-\beta+\gamma}\right] - \hat{c}^\dagger_{x,\alpha}\cdot\hat{c}_{x,\alpha}\right) \quad . \tag{2.4.20}$$

Constructing this Hamiltonian using a finite-state machine is essentially impossible.[ix] On the other hand, using the formalism presented above, the construction is still easy to implement. Comparing the different compression methods for different system widths (cf. Fig. 2.20), we find that SVD and DLN result in identical bond dimensions and sizes for the dense blocks (3 independent of system width). DPL gives minimally larger bond dimensions. The largest dense block resulting from DPL is however much larger than that found with DLN or SVD. Since computational effort cubically depends on these sizes, this is an important difference. Investigating the relative sparsity of the dense blocks produced by DLN, we find that 8% of stored values are exactly zero and 46% are ± 1. In comparison, the SVD produced entirely dense blocks. Hence even at size 3×3, the DLN results in a considerably simpler MPO structure.

Independent of the compression method, we find the expected approximately linear growth of the maximal bond dimension with the cylinder width: This linearity is due to the momentum conservation in the interaction term, which, once two sites on one half of

[ix] This didn't stop Motruk[11] et. al. from doing it to achieve first results.

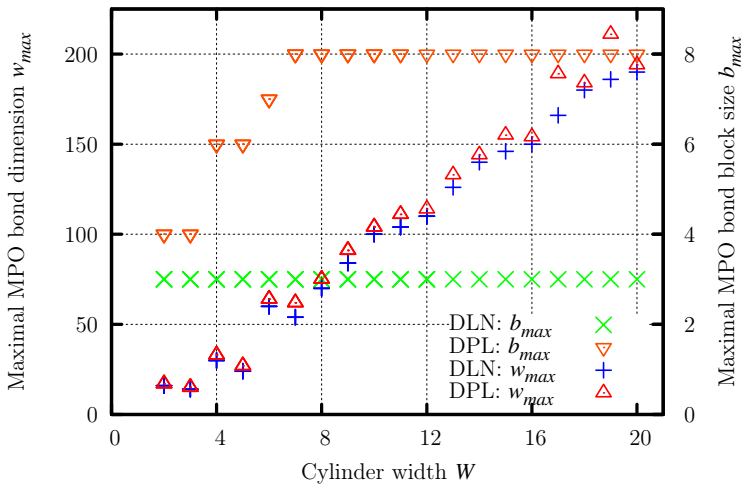


Figure 2.20: Maximal MPO bond dimension $w_{\max}$ (left axis) and maximal block size $b_{\max}$ (right axis) for the Fermi-Hubbard Hamiltonian on a cylinder. The centre bond of each ring has the largest dimension and also the largest dense block, while the inter-ring connections are smallest independent of the compression method. (First published in Ref. [1].)

the cylinder and a third on the second half are fixed, results in only one valid location on the second half for the fourth site.

Four-Body Quantum Chemistry Hamiltonian

As a last example, we consider a toy model for the quantum chemistry Hamiltonian

$$\hat{H} = V_{ijkl} \sum_{\sigma\tau=\uparrow\downarrow} \sum_{ijkl}^{L} \hat{c}^\dagger_{i\sigma} \hat{c}^\dagger_{k\tau} \hat{c}_{l\tau} \hat{c}_{j\sigma} \quad . \tag{2.4.21}$$

V_{ijkl} are chosen randomly with absolute value less than two and the necessary symmetry $V_{ijkl} = V_{jilk}$. Construction of this Hamiltonian with the generic construction method is very expensive (L^4 MPO-MPO additions and many compression sweeps) and not suitable for larger systems. For small $L \leq 35$, it is however possible to generate the Hamiltonian in a sufficiently short time. Using some preprocessing such as an initial SVD decomposition of V_{ijkl}, it should be possible to also construct the Hamiltonian for larger systems. Nevertheless, we can apply the compression methods and check that they arrive at good MPO representations.

The maximal bond dimension always occurs at bond $L/2$ for systems of even length and bonds $L/2-1$ and $L/2$ for systems of odd length. Analytically, we find that it should be possible to sum up the terms of the Hamiltonian to give a scaling $w_{L/2} = 2L^2 + 3L + 2$. The constant term 2 can be explained by either all i, j, k, l being to the left or right of

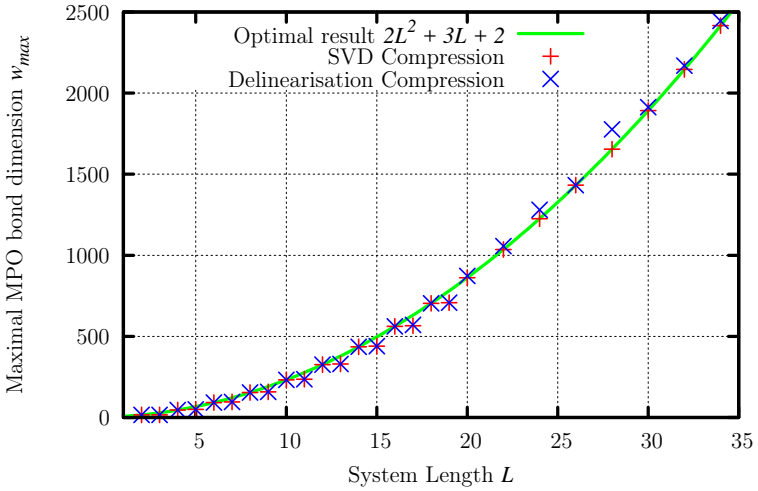


Figure 2.21: MPO bond dimensions of the representations of the toy model four-body interaction and the optimal scaling (for even dimensions) for different system sizes L. SVD always results in the optimal bond dimension, while DLN scales decidedly quadratically, but fails to find the optimal representation for large L. (First published in Ref. [1].)

the central bond. The prefactor 3 of the linear term corresponds to L terms with one out of i, j, k, l being on the left side and three on the right side (or vice versa) plus the L contributions each with $i = j$ or $k = l$. Finally, there are L^2 ways to distribute two out of i, j, k, l on the left or right half of the system. This is in agreement with recent results[55] which also find a leading term $2L^2$.

Fig. 2.21 gives the maximal bond dimension for different system sizes as returned by SVD and DLN. The SVD always results in the optimal bond dimension. The DLN fails to find that optimal representation for larger system sizes, but still scales decidedly quadratically. We can hence conclude that even on such a difficult operator, the constructive approach is a viable candidate at least for small systems, e.g. to allow quick proto-typing or investigation of simple cases. Alternative approaches[55,56] suffer from a general need to adapt them to each model and hence come with a certain over-head attached before research can start.

2.5 Concluding Remarks

In this section, the usefulness of symmetries in tensor network applications has been motivated and a framework for symmetry-protected tensors has been defined. While based on earlier work by Ian McCulloch[31] and Andreas Weichselbaum,[37] the implementation in the form of SyTen has been done entirely independently with great focus on generality, modularity, ease of code-reuse and adaptability to other problems both by the original author as well as current and future collaborators.

Based on this framework, matrix-product states and matrix-product operators as

commonly known in the literature have been defined. The MPO dot product is clearly defined for the first time in the published literature in this thesis, while the other MPS and MPO operations have been reviewed briefly. Extensive work was done together with Ian McCulloch to build a better MPO compression algorithm, culminating in the very general DELINEARISATION procedure[1] capable of compressing arbitrary MPOs without loss of precision to their (nearly) optimal form.

With these building blocks in hand, it is now possible to introduce the density-matrix renormalisation group as the variational optimisation of a MPS in the next chapter.

3 Density Matrix Renormalisation Group

This chapter will review the Density Matrix Renormalisation Group (DMRG) as a variational optimisation algorithm to find the ground state for a given Hamiltonian. It will concentrate on the application of DMRG to matrix-product states (MPS) using matrix-product operators (MPO), i.e. MPS-DMRG; for definitions of and details on MPS and MPO see Secs. 2.3 and 2.4 in Chapter 2. The strictly single-site DMRG method (DMRG3S), developed during this thesis, will be presented here again and will be explained in detail.

However, while the generalisation of MPS-DMRG to tree-tensor networks is again straightforward, it will not be discussed explicitly here. Equally out-of-scope is the application of DMRG to the solution of infinite one-dimensional systems in the form of iDMRG[23] or VUMPS.[24]

3.1 Ideas behind MPS-DMRG

Given a hermitian operator $\hat{H}$ represented as a MPO $\{W_i\}_i$, the aim of MPS-DMRG is to find the state $|\psi\rangle$ represented as a MPS $\{M_i\}_i$ which minimises the energy

$$\min_{|\psi\rangle} \left(E = \frac{\langle\psi| \hat{H} |\psi\rangle}{\langle\psi|\psi\rangle} \right) \quad . \tag{3.1.1}$$

This problem is equivalent to minimising $\langle\psi| \hat{H} |\psi\rangle$ under the constraint that $\langle\psi|\psi\rangle = 1$ and hence can be solved using a Lagrangian multiplier[46] λ, resulting in

$$\min_{|\psi\rangle} \left(\langle\psi| \hat{H} |\psi\rangle - \lambda \langle\psi|\psi\rangle \right) \quad . \tag{3.1.2}$$

The next key ingredient is then to optimise the MPS tensors $\{M_i\}_i$ iteratively one after the other. When optimising M_i, differentiation with respect to $M_i^\dagger$ and equating to zero for the extremal point results in the generalised eigenvalue equation

$$H_{i;\mathit{eff}} M_i - \lambda N_i M_i = 0 \tag{3.1.3}$$

$$H_{i;\mathit{eff}} M_i = \lambda N_i M_i \tag{3.1.4}$$

where $H_{i;\mathit{eff}}$ is a rank-6 tensor given by the contraction of all $\{W_k\}_k$ and the $\{M_k\}_{k\neq i}$ and $\{M_k^\dagger\}_{k\neq i}$ on sites k other than i. Similarly, N_i is the rank-4 tensor given by the

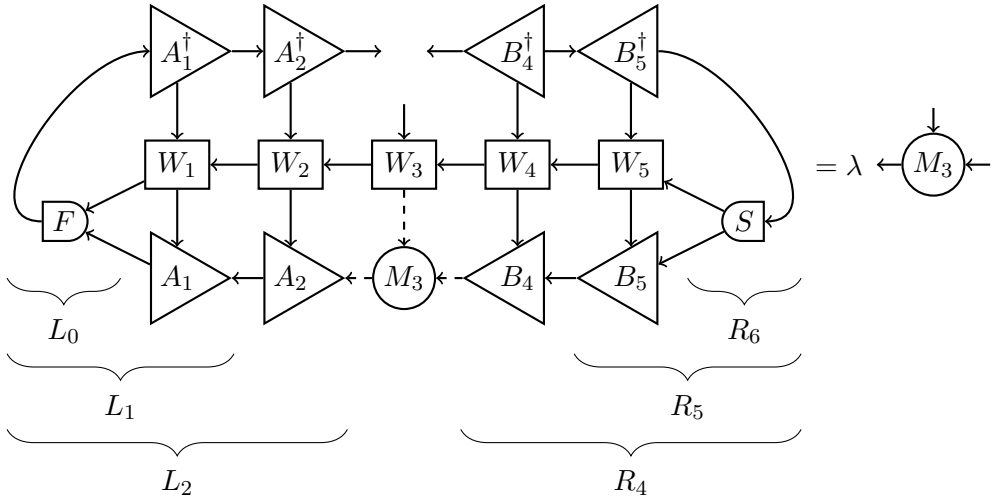


Figure 3.1: Graphical representation of the basic DMRG eigenvalue equation (3.1.5). Dashed tensor legs of the tensor M_3 on the left-hand side correspond to the $H_{i;\mathit{eff}}M_i$ product to be evaluated. Brackets denote the intermediate contractions to be calculated (cf. Alg. 3.1).

contraction of just the $\{M_k\}_{k\neq i}$ and $\{M_k^\dagger\}_{k\neq i}$. A crucial simplification now occurs if all $\{M_k\}_{k<i}$ are left-normalised and all $\{M_k\}_{k>i}$ are right-normalised. In this case, the tensor N_i is simply a two-fold identity tensor and we have the standard eigenvalue problem:

$$H_{i;\mathit{eff}}M_i = \lambda M_i \tag{3.1.5}$$

This equation is best represented graphically, cf. Fig. 3.1. Solving it for the lowest eigenvalue λ and associated eigenvector M_i will result in the locally optimal MPS tensor.

3.2 1DMRG: Single-Site DMRG

The single-site variant of the DMRG method, 1DMRG, is the easiest variant to implement. It is not advisable to use it in practice as it suffers from severe convergence problems. In particular, it is very likely to get stuck in a local energy minimum or unfavourable symmetry sector. However, all other variants of DMRG are relatively simple extensions of this easiest method. Hence, the 1DMRG method will be used to define the overall framework; essentially a stepping stone towards the more useful variants.

The 1DMRG method is described in Alg. 3.1. It first calculates the partial contractions of the MPO sandwiched between the MPS tensors from the right edge of the system to the left edge of the system and stores those contractions as R_i (cf. Fig. 3.1). It then sweeps back and forth over the system. During each sweep, each site is visited and updated using the left- and right contractions L_{i-1} and R_{i+1} and the local MPO tensor W_i to represent $H_{i;\mathit{eff}}$. This local update is done using a sparse eigenvalue solver, aiming for the

Algorithm 3.1 1DMRG, the single-site DMRG algorithm without any extensions.

procedure 1DMRG(MPS $\{M_i\}_i$, MPO $\{W_i\}_i$)
  $\left[\{B_i\}_i, L_0, \{R_i\}_{i=2}^{L+1}\right] \leftarrow$ 1DMRG::SETUP($\{M_i\}_i$, $\{W_i\}_i$)
  while not converged **do** ▷ e.g. until change in energy, i.e. λ, is small
    $\left[\{A_i\}_i, \{L_i\}_{i=0}^{L-1}, R_{L+1}\right] \leftarrow$ 1DMRG::SWEEP-RIGHT($\{B_i\}_i, L_0, \{R_i\}_{i=2}^{L+1}$)
    $\left[\{B_i\}_i, L_0, \{R_i\}_{i=2}^{L+1}\right] \leftarrow$ 1DMRG::SWEEP-LEFT($\{A_i\}_i, \{W_i\}_i$, $\{L_i\}_{i=0}^{L-1}, R_{L+1}$)
  end while
  return $\{B_i\}_i$
end procedure
procedure 1DMRG::SETUP(MPS $\{M_i\}_i$, MPO $\{W_i\}_i$)
  Right-normalise the MPS, $\{M_i\}_i \to \{B_i\}_i$, discard the norm
  $R_{L+1} \leftarrow$ SPLIT($\tilde{m}_L \to w_L \times m_L$) ▷ S in Fig. 3.1
  for $i \in [L, 2]$ **do** ▷ Iteratively calculate contractions of MPO with MPS
    $R^{w_{i-1}m_{i-1}}_{i;\tilde{m}_{i-1}} \leftarrow \sum_{\tilde{m}'_i w_i m_i} R^{w_i m_i}_{i+1;\tilde{m}_i} B^{m_{i-1}}_{i;\sigma_i m_i} \left(B^{m_{i-1}}_{i;\tau_i m_i}\right)^\dagger W^{\sigma_i w_{i-1}}_{i;\tau_i w_i}$
  end for
  $L^{\tilde{m}_0}_{0;w_0 m_0} \leftarrow$ FUSE($m_0 \times w_0 \to \tilde{m}_0$) ▷ F in Fig. 3.1
  return $\left[\{B_i\}_i, L_0, \{R_i\}_{i=2}^{L+1}\right]$
end procedure
procedure 1DMRG::SWEEP-RIGHT(MPS $\{B_i\}_i$, MPO $\{W_i\}_i$, L_0, $\{R_i\}_{i=2}^{L+1}$)
  for $i \in [1, L-1]$ **do**
    Solve Eq. (3.1.5): $(W_i \cdot (L_{i-1} \cdot M_i)) \cdot R_{i-1} = \lambda M_i$ for $\tilde{M}_i$
    Left-normalise $\tilde{M}_i \to A_i T$
    Discard R_{i+1}
    $M_i \leftarrow A_i$
    $M_{i+1} \leftarrow T \cdot B_{i+1}$
    $L_i \leftarrow ((L_{i-1} \cdot A_i) \cdot W_i) \cdot A_i^\dagger$
    return $\left[\{A_i\}_i, \{L_i\}_{i=0}^{L-1}, R_{L+1}\right]$
  end for
end procedure
procedure 1DMRG::SWEEP-LEFT(MPS $\{A_i\}_i$, MPO $\{W_i\}_i$, $\{L_i\}_{i=0}^{L-1}, R_{L+1}$)
  for $i \in [L, 2]$ **do**
    Solve Eq. (3.1.5): $(W_i \cdot (L_{i-1} \cdot M_i)) \cdot R_{i-1} = \lambda M_i$ for $\tilde{M}_i$
    Right-normalise $\tilde{M}_i \to T B_i$
    Discard L_{i-1}
    $M_i \leftarrow B_i$
    $M_{i-1} \leftarrow A_{i+1} \cdot T$
    $R_i \leftarrow ((R_{i+1} \cdot B_i) \cdot W_i) \cdot B_i^\dagger$
  end for
  return $\left[\{B_i\}_i, L_0, \{R_i\}_{i=2}^{L+1}\right]$
end procedure

lowest possible eigenvalue. The eigenvalue solver best suited for DMRG is the Lanczos method, as it only requires the application of $H_{i;\mathit{eff}}$ onto the local M_i. This operation can be done efficiently at cost $O\left(2m^3dw + m^2d^2w^2\right)$ in the order displayed in the algorithm. Alternatively, if the effective d is 1, the alternative ordering $((L_{i-1} \cdot W_i) \cdot M_i) \cdot R_{i+1}$ allows for the precomputation of the first product prior to all Lanczos iterations, which can speed up calculations. When using quantum numbers, the local Hilbert space often decomposes into one-dimensional blocks, leading to this effective $d = 1$.

A common alternative is the Davidson method.[57] However, using its preconditioner is not trivial and hence often left out.[i] In practice, both Lanczos and Davidson achieve comparable results, with either choice having the edge in some problems.

Constructing the left and right contractions incurs a cost $O(2m^3dw + m^2d^2w^2)$ for each contraction. The SVD costs $O(m^3d)$, as does the multiplication of the transfer tensor T into the next MPS tensor. The cost of the overall DMRG method scales linearly in the system length L and cubically in the MPS bond dimension m. It scales primarily linearly in the local physical dimension d and the MPO bond dimension w. However, there are subleading terms of quadratic order in d and w. In the 1DMRG algorithm, the eigensolver step usually takes between 75% and 95% of the total runtime.

A matter of ongoing research is hence the improvement of the DMRG eigensolver to require fewer applications of $H_{i;\mathit{eff}}$ onto the MPS tensor (typically, one uses 5 to 20 Lanczos iterations, depending on the problem at hand, with a trade-off between fewer sweeps and fewer eigensolver iterations).

In this spirit, it should be noted that there were two relatively recent approaches to combine DMRG-obtained MPS results with traditional projector or Lanczos methods to obtain improved ground state estimates.[58,59] These methods start from a ground state obtained via DMRG and then generate a set of orthonormal states. The former approach by Emanuele Tirrito et. al. does this based on the normally-discarded singular values, while Rui-Zhen Huang et. al. proposed a Lanczos method implemented on top of the MPS structure. Both approaches then diagonalise the Hamiltonian projected into this generated space to obtain a better ground state approximation. While the first method claims an effective increase in bond dimension by a factor of approx. 1.2, the second method only converges to the ground state approximation after many hundred Lanczos iterations. As such, simply increasing the bond dimension during the initial DMRG calculation is usually a faster and cheaper way to achieve better energy estimates.

3.2.1 Parallelisation of DMRG

By its nature, DMRG is a largely serial process, updating a single site at a time, with the updated result potentially affecting all other tensors and hence influencing future updates.

[i] For example, the ITensor toolkit uses the Davidson method without preconditioning in its DMRG implementation.

As such, there is no straightforward or trivial way to parallelise DMRG apart from the parallelisations possible within tensor-tensor operations in general (such as parallelising over different symmetry blocks or parallelising dense matrix operations). At the moment, the most prominent way to parallelise DMRG is the real-space parallelisation[60] introduced by Miles Stoudenmire and Steve White in 2013 (see also Ref. [61] for a detailed explanation). This method relies on the assumption that updating MPS tensors far from each other can happen nearly independently by a number of different workers which sweep through *parts* of the whole system. Unfortunately, the number of sequential operations required to transmit information from the left edge of the system to the right edge of the system remains unchanged. Therefore, the method is useful when applied to large systems late during the calculation, when most updates result in minor local changes, but not, for example, global redistributions of particles. Additionally, the amount of RAM required grows linearly in the number of workers, while for example tensor-block parallelisation only barely increases the amount of necessary memory. Another "layer" of parallelisation[55,62] may be provided by decomposing the Hamiltonian MPO into a sum of mutually-orthogonal MPOs, where the added bond dimensions of the summands is approximately equal to the bond dimension of the compressed sum. In this way, one may distribute each part of the Hamiltonian to its own worker, which only calculates the application of this particular part onto the state on each bond. The workers then have to send and receive only the relatively small state tensors from a central master, while most of the expensive contractions can occur in parallel.

However, due to the multiplicative nature of such parallelisations, it is quite feasible to use a few tens of cores for a single DMRG calculation: for example, one could use 4 real-space DMRG workers, split the Hamiltonian into at least two orthogonal components, each using 8 cores for tensor-block parallelisation and up to two cores each for dense matrix operations to employ a total of 128 cores. Combine this with multiple calculations for different parameter values and using a cluster to its full extent is often possible.

In practice, tensor-block parallelisation nearly always works better than real-space parallelisation at least for a small number of cores and the parallelisation of dense matrix operations achieves very little gain as soon as sufficiently many symmetries are used. MPO-based parallelisation has not been implemented yet and hence also not been tested within the SyTen toolkit.

3.3 Previous Approaches to Avoid Convergence Failures

The 1DMRG algorithm as written above suffers from severe convergence issues on all but the most trivial problems. The most striking example is certainly its requirement to keep the left and right MPS bond basis constant: This leads to a failure to include other relevant quantum number sectors (when using symmetries) and to redistribute particles over a larger scale (in any case).

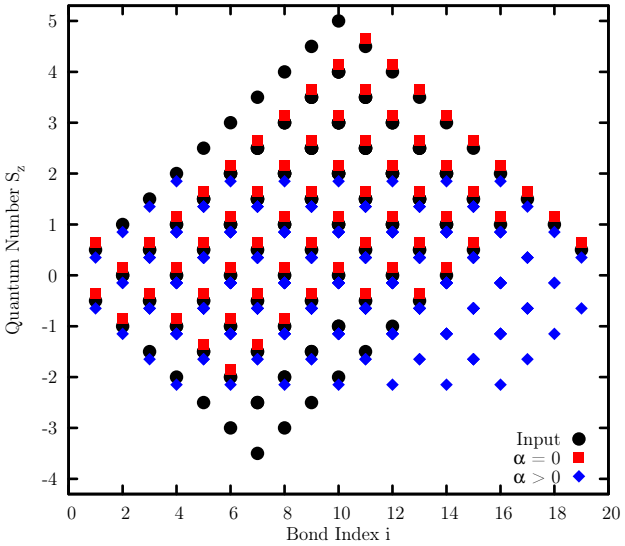


Figure 3.2: S^z quantum number distribution on a $L = 20$ Heisenberg spin chain. An asymmetric initial state with its three rightmost spins all pointing up is prepared (black circles). The basic 1DMRG algorithm then fails to properly add new quantum number sectors and does not restore the overall symmetry ($\alpha = 0$, red squares), resulting in a high energy of $E_{\alpha=0} = -6.35479$. If some perturbation is added (cf. Sec. 3.4), the overall symmetry can be restored (blue diamonds) and the true ground-state energy is achieved at $E_{\alpha\neq 0} = -8.6824724$ (Figure and data first published as part of Ref. [2]).

The former failure is particular easy to illustrate (cf. also Ref. [2]): We consider a Heisenberg spin chain of $L = 20$ spins and start from an initial random state with the three rightmost states pointing up. The resulting S^z quantum number distribution can be plotted on each bond (Fig. 3.2, black circles). Then running the 1DMRG algorithm on this system leads to the distribution plotted with red squares: Crucially, while some sectors have been removed, no new sectors have been added (red always occurs together with black, never on its own) and the overall mirror symmetry of the system is not restored.

Multiple avenues exist to improve the 1DMRG algorithm to avoid convergence issues such as this one: Three of them will be discussed briefly in the following before the next section introduces the strictly-single site DMRG method with subspace expansion.

3.3.1 Two-Site DMRG

Today's two-site DMRG method,[16] 2DMRG, based on MPS is motivated by the original DMRG method[13,14] with its system and environment blocks. The primary difference from the 1DMRG method introduced above is the optimisation of two sites at a time. That is, two neighbouring MPS component tensors are contracted over their common index and the resulting rank-4 tensor is optimised in largely the same way as in 1DMRG before being split via SVD into two rank-3 tensors again. The difference has the following

consequences: First, the cost of DMRG now scales in the leading term quadratically and in subleading terms (in the MPS bond dimension) cubically in the local physical dimension.[63] Especially for bosonic systems, this is very unfavourable. Second, the MPS bond between the two optimised tensors is now also a variational parameter: this means that the size of the MPS bond indices can be adapted throughout the calculation and that it is possible to distribute particles through the system, as the left/right basis on this bond can change.

In practice, the method works reliably, but is firstly very slow to pick up long-range correlations[64] (as only nearest-neighbour sites can e.g. exchange particles) and secondly extremely slow as soon as the local physical dimension becomes large. However, it is still a reliable tool to get first results. Furthermore, the truncation error obtained from the SVD to split the rank-4 tensor into two after each optimisation serves as a useful parameter over which to extrapolate observables such as the energy or occupation numbers.[14,65] For a particular impressive example of such extrapolation, see Ref. [12], where both occupation numbers and energy where extrapolated to great success from a truncation error $t \approx 10^{-5}$ to zero.

3.3.2 Density-Matrix Perturbation

The density matrix perturbation method proposed by Steve White in 2005[64] was the first extension to single-site DMRG to give reliable results. This was the first form of an additional *enrichment* step inserted after the optimisation step. Its main idea is to construct the reduced density matrix (RDM) on one bond, add a perturbation term to it and use the diagonalisation of that perturbed density matrix to select relevant states. However, if one were to restrict the RDM to the states available on that bond, no perturbation would be possible. Hence, it is necessary to first artificially increase the dimension of the bond in question. This is done by replacing one of the neighbouring site tensors with an isometry combining the nearest-neighbour bond states and the local physical states into a new bond space of dimension $d_i \times m$. The RDM constructed on this bond hence has dimension $d_i^2 m_i^2$. During a left-to-right sweep, the perturbation term is calculated from the left-contraction of the Hamiltonian contracted with its complex conjugate over one of the pairs of MPS-indices and the MPO indices (cf. Fig. 3.3), multiplied by a small prefactor α. Similarly, during a right-to-left sweep, the perturbation is calculated using the right-contractions of the Hamiltonian. This perturbation adds some global information and leads to a faster inclusion of long-range correlations.[64]

Calculation of this perturbation term scales as $O(d^3 m^3 w)$. While the scaling in the MPS and MPO bond dimensions is on-par with the other operations of DMRG, the scaling with the local physical dimension is comparatively bad. For small physical dimensions (e.g. $d = 2$ for a spin $S = 1/2$ system) this is very acceptable, especially since it only has to be done once per site and allows for the faster 1DMRG method. However, it may

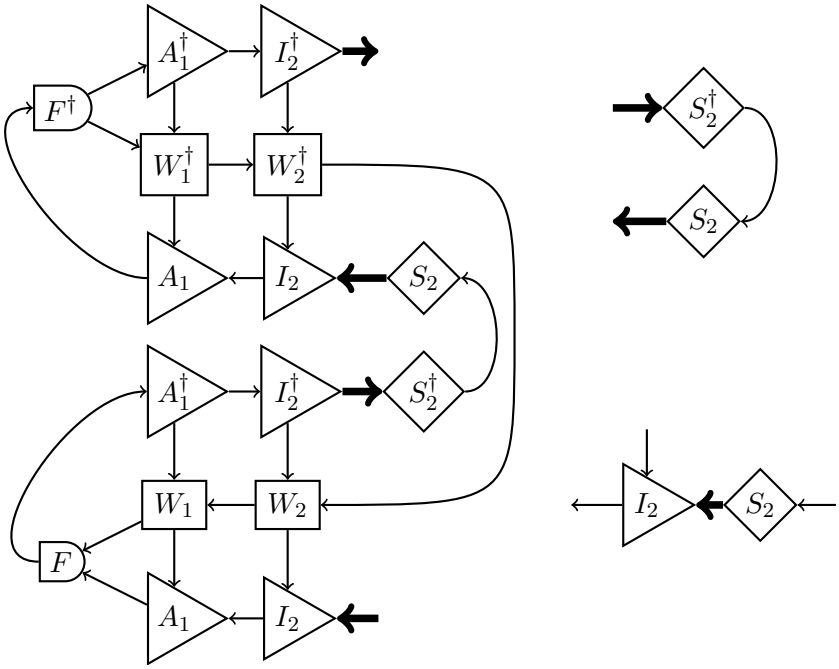


Figure 3.3: Left: The perturbation term added to the reduced density matrix in White's perturbation scheme during a left-to-right sweep.[64] The tensor A_2 has been renamed to I_2 to denote that it is a non-truncating isometry combining the spaces m_1 and d_2 into the (larger, thick) bond $m_2 = d_2 \times m_1$.
Right Top: The reduced density matrix on the second bond in the enlarged space is given as the singular value tensor on the second bond contracted with itself over its right index.
Right Bottom: The singular value tensor S_2 contracted with the isometry I_2 gives the original MPS site tensor A_2.

still be advantageous to avoid this scaling. Furthermore, it should be pointed out that a standard implementation of the method in the form of the Centermatrix Wavefunction approach (which works directly on the S_2 tensor of Fig. 3.3 during the local update) does not allow for the optimal contraction sequence normally offered by the 1DMRG algorithm and hence also scales quadratically in d.

During the work on this thesis, extensive numerical experiments where done to firstly evaluate whether other perturbation terms yield better convergence and secondly which value of the small prefactor α is optimal at which point during the DMRG convergence. For the first point, two other perturbation terms were tested: First, a symmetrised version which also adds the corresponding right-contractions to the density matrix and second an environment term[22] which contracts left- and right-contractions into one overall term. However, neither of the two had a clear advantage over the original scheme but performed roughly comparably in terms of convergence per computational effort, with the environment scheme converging slightly faster per sweep at slightly higher computational effort per sweep.

The second point of interest was an optimal scheme to select the mixing factor α. In the original publication it was set at 10^{-4} during the first few sweeps and then at 10^{-6}. However, it was found that selecting a much larger value initially of order 1 greatly sped up convergence. By considering the decrease in energy during the eigensolver step and the increase in energy during the subsequent perturbation it then becomes possible to adaptively increase or decrease the value of the mixing factor. This scheme has first been published in Ref. [2] and is explained in detail in Sec. 3.4.1.

3.3.3 Multi-Grid DMRG

Multi-Grid DMRG (MG-DMRG) was first proposed by Michele Dolfi et. al. in 2012.[66] The idea is to map a very large system of e.g. $L = 512$ sites to a series of smaller systems by combining two neighbouring sites into one and truncating the local state space. For example, in a bosonic system this could work by combining two spaces holding up to N bosons each into a single combined state holding up to $2N$ bosons. Once the system only has a few physical sites (e.g. $L_0 = 4$), DMRG is run to obtain a ground-state approximation. This approximation is then mapped onto the larger system of $L_1 = 8$ sites and taken as an initial state for the next DMRG calculation. The procedure is repeated until the original system length is reached.

The 2012 paper presents an example of a large bosonic system in a slowly modulated optical lattice and shows how standard DMRG fails to re-distribute particles properly while MG-DMRG essentially always achieves a very even distribution. This problem was reproduced during the work on this thesis. However, it was found that using single-site DMRG with some sort of perturbation and many sweeps (up to e.g. 100) resulted in an equally even distribution of particles at comparable computational effort.

Furthermore, while the method is well-applicable to problems where states in the combined space of two sites are not 1-to-1 associated with a given quantum number, it becomes very costly to use it while conserving many quantum numbers, as the combined state space is still decomposed into many small quantum number blocks. One hence either has to discard some of those blocks, making entire regions of phase space inaccessible to the coarser grid, or deal with a very large local dimension. On the other hand, if there are only few quantum numbers preserved, the combined space is only decomposed into a few blocks which can be truncated internally and it is possible to keep one representative of each quantum number sector.

3.4 Strictly Single-Site DMRG (DMRG3S)

The subspace expansion variant of DMRG was developed during the work on this thesis and first published in Ref. [2]. The variant allows for the optimisation of strictly a single-site tensor, as opposed to the 2DMRG variant or the Centermatrix Wavefunction

approach to implement the density matrix perturbation method. As such, it aims to scale as $O(m^3dw)$ with only sub-leading terms in m scaling worse than $O(dw)$. This scaling is in contrast to the scaling of the 2DMRG method ($O(m^3d^2w)$) or the density matrix perturbation method ($O(m^3d^3w)$).

The resulting strictly single-site DMRG algorithm, (1)DMRG3S, proceeds in exactly the same way as the standard 1DMRG algorithm Alg. 3.1 with the addition of a subspace expansion step as *enrichment* after each solution (i.e. after lines 20 and 31 of Alg. 3.1 respectively). This enrichment step is explained in detail in the next section before the results of some numerical experiments are presented in Sec. 3.4.2. Summarising conclusions and acknowledgements are given in Sec. 3.4.4.

3.4.1 Subspace Expansion

The subspace expansion method was originally introduced in the numerical linear algebra community[67,68] as part of the AMEn algorithm. It relies on the fact that if only the product of two matrices is of relevance, it becomes possible to extend both matrices in such a way that the product of the two matrices stays invariant. As an example, consider two matrices $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times p}$ where we are only interested in their product $A \cdot B \in \mathbb{R}^{m\times p}$. We can then expand A by another matrix P column-wise into A' and as long as we also expand B row-wise with zeros into B', the product will stay invariant:

$$A \cdot B \to \begin{bmatrix} A & P \end{bmatrix} \cdot \begin{bmatrix} B \\ 0 \end{bmatrix} = A \cdot B + P \cdot 0 = A \cdot B \tag{3.4.1}$$

This method works equally well for MPS, except that the matrices A, B and P in the example above carry an additinal index σ_i or σ_{i+1} (during a left-to-right sweep) respectively. That is, for the expansion over the bond m_i from the left with expansion term $P^{m_{i-1}}_{i;\sigma_i m_{P_i}}$, we write:

$$M_{i;\sigma_i} \to \tilde{M}_{i;\sigma_i} = \begin{bmatrix} M_{i;\sigma_i} & P_{i;\sigma_i} \end{bmatrix} \tag{3.4.2}$$

$$M_{i+1;\sigma_{i+1}} \to \tilde{M}_{i+1;\sigma_{i1}} = \begin{bmatrix} M_{i+1;\sigma_{i+1}} \\ 0 \end{bmatrix} \tag{3.4.3}$$

such that $\tilde{M}_i$ and $\tilde{M}_{i+1}$ share a common bond of dimension $m_i + m_{P_i}$ while their other dimensions remain unchanged. It is easy to see that this operation leaves the overall MPS unchanged, since the elements of $P_{i;\sigma_i}$ will only be multiplied by zero.

The advantage is now that the optimisation step on the next site (in this case one step to the right) can use the larger, new space of size $m_i + m_{P_i}$ to its left. If no elements of the enlarged space are suitable to lower the energy, they can be discarded by simply keeping their associated factors zero. This results in a flexibility in the left- and right basis states of each MPS tensor while at the same time keeping well within the required

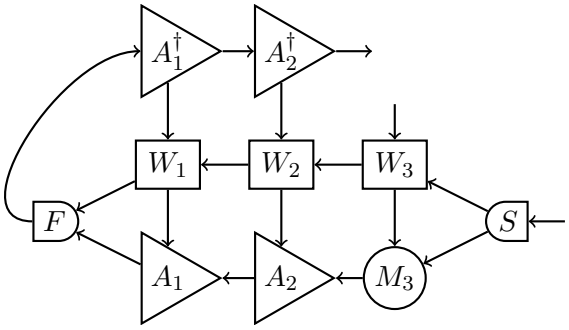


Figure 3.4: The expansion term used in DMRG3S during a left-to-right sweep after optimisation of the third site tensor M_3.

computational cost at just $O\left(\left(m_i^2 + m_{P_i}^2\right) d\right)$ not counting the cost to calculate P_i.

After the expansion, the SVD proceeds as normal, selecting the most relevant m_i states from the larger set of $m_i + m_{P_i}$ states. Note that while the states from the expansion term will still have prefactor zero after the SVD as well, some of the original states will have been discarded. This will usually lead to a slight increase in energy (unless the expansion states are actually lower in energy).

Expansion Term

The original work by Sergey Dolgov and Daniel Kressner in the numerical linear algebra community focused on using the local components of the exact residual, written as a MPS, as the expansion term in the subspace expansion. This leads to some very favourable guarantees regarding global convergence to the minimum, provided the required number of states is used.

Unfortunately, calculating this residual exactly is very costly; furthermore, the guarantee to globally converge does not necessarily hold as soon as truncation of the MPS bond dimensions is used. We hence proposed an alternative expansion term of the form

$$P_i = \alpha L_{i-1} M_i W_i \tag{3.4.4}$$

where we contract over all possible indices and combine the RHS MPO and MPS indices into one larger index. This expansion term is pictured in Fig. 3.4. The scalar prefactor α serves to control the magnitude of the perturbation. It can be motivated by observing that if one were to apply the MPO $\{W_i\}_i$ to the current MPS $\{M_i\}_i$, then the site tensor on the i-th site is given as $W_i M_i$. The left-contraction L_{i-1} can then be seen as projecting the left-hand side basis of this new site tensor into the basis of the original site tensor. As such, the states on the RHS of the new site tensor are precisely those generated by the Hamiltonian throughout the entire left or right half of the system and hence again provide some global information similar to the reduced density matrix perturbation.

An important optimisation here which was not published in our first paper is to restrict the splitting half-isometry combining the MPS and MPO RHS bond indices to only access a subset of the full space which is normally of size $m_i \times w_i$. This can be done very easily by only constructing a near-isometry of the selected size; in our experience, restricting that size to approx. 10-15 per quantum number block serves to speed up calculations with very much equal convergence behaviour. However, in some cases, in particular complicated spin systems, this may lead to a over-representation of high-quantum-number sectors (as all sectors can grow by an equal number of states during each expansion). In this case, it may be more appropriate to restrict each block to a small percentage (5% to 10%) of its full size.

Fig. 3.2 illustrates how the subspace expansion is able to *add* additional states on the bonds (once the prefactor α is non-zero). The resulting quantum number distribution (blue diamonds in Fig. 3.2) is symmetrical and furthermore corresponds to the true ground-state distribution and energy.

Optimal Choice of the Mixing Factor α

When choosing a value for the scalar coefficient α, the *mixing factor*, multiple concerns have to be taken into account: First, setting it to zero (literally) nullifies the expansion. Second, if the expansion term is not a better ground state estimate than the current value of M_i, the expansion will result in a slightly higher energy and hence offset the energy gain achieved by the previous eigensolver step. Third, if the expansion term does have a lower energy than the current value of M_i, it would be ideal to have a very large mixing factor. Fourth, if the current state is an eigenstate of the Hamiltonian, the perturbation term will be parallel to the current MPS tensor, implying that the state is unchanged by the subspace expansion (even with subsequent truncation). Note that the last point holds for the perturbation term as described above in subspace expansion, but not generally for any enrichment term. In particular, when using White's density matrix perturbation, it does not hold.

The second case is in particular common if the true ground state of the Hamiltonian cannot be properly captured with the current maximal bond dimension. In this case, the DMRG eigensolver attempts to find the absolute lowest energy state under the constrain of the limited bond dimension, while the expansion term as written above tends to favour an *eigenstate* of the current bond dimension. The resulting competition leads to an increase in energy if the mixing factor is non-zero and is the most commonly-observed behaviour. In those cases, the mixing factor should slowly be reduced to zero to allow for convergence at the current bond dimension.

The third case, that the energy of the perturbed state is lower than the energy of the eigensolver, is in particular common during the initial sweeps of the calculation.

Finally, the fourth case occurs if the selected bond dimension is sufficient to represent

the ground state of the Hamiltonian. In this case, a useful test is to increase the mixing factor until it becomes (relatively) large: if the expansion then still does not result in a perturbation of the state, one can be reasonable certain to have found the ground state of the Hamiltonian: If the state was an eigenstate, but not the ground state, it is extremely unlikely that the optimisation would not have found a single site tensor with slightly lower energy.[ii] Conversely, if the lowest-energy state found by the optimisation was not an eigenstate of the Hamiltonian, the expansion term would have resulted in a very large perturbation of the state.

In the original paper introducing the density matrix perturbation method, the suggestion was made to start with a value of (e.g.) $\alpha = 10^{-4}$ and decrease it during the calculation to as low as 10^{-6}.

However, it was found that a much larger value initially, e.g. $\alpha = 100$ is not unreasonable during the initial sweeps. If the bond dimension is not sufficient, the value should then slowly be reduced during the calculation. This can be achieved by adapting α after each local iteration based on the energy reduction resulting from the eigensolver optimisation compared to the energy increase resulting from the SVD following the expansion.

An examplary algorithm to update α is given in Alg. 3.2. In this algorithm, cases C1 and C2 are those typically encountered: both optimisation and truncation decrease and increase the energy respectively and an attempt is made to keep the ratio between the two changes between 5% and 30%. Case B occurs if the perturbation resulted in a lower energy; in this case, we increase the mixing factor considerably. If the optimisation does not result in a decrease of energy while the truncation does increase the energy, case A1 triggers to reduce the mixing factor. Alternatively (case A2), if neither operation changes the energy beyond some threshold $\varepsilon \approx 10^{-9}$, the mixing factor will be increased slightly. Over time, the mixing factor will grow to either escape a local minimum or illustrate that even with a large mixing factor, the expansion does not result in measurable truncation – this occurs if we are converged to the true ground state.

3.4.2 Numerical Experiments

To test the strength of the new approach, we apply it to four different systems: first, a $S = 1$ Heisenberg spin chain, second dilute bosons on an optical lattice, third the Fermi-Hubbard model and fourth free fermions. Calculations using the Centermatrix Wavefunction implementation of the density matrix perturbation method are also done as well as pure two-site DMRG calculations. While one would usually slowly increase the bond dimension during a DMRG calculation, each algorithm prefers a different "staging

[ii] This assumes that the ground state is contained in the Krylov space spanned over the initial state by $\hat{H}$ or conversely that the ground state is reachable using imaginary time evolution with $\hat{H}$. If the Hamiltonian does not connect the ground state to the initial state, some perturbative term (usually some form of hopping) has to be introduced initially and then slowly taken to zero to allow exploration of the full Hilbert space. While rare in model Hamiltonians, the situation does occur relatively frequently when attempting to handle more complicated systems.

Algorithm 3.2 Method to dynamically adapt the mixing factor α in the DMRG3S subspace expansion method. Numerical values, in particular the limits at the end, are entirely heuristic and require some implementation-dependant fine-tuning.

procedure DMRG3S::ADAPTALPHA(α, E_{initial}, $E_{\text{optimised}}$, $E_{\text{truncated}}$)
  $\Delta_{\text{opt}} \leftarrow E_{\text{initial}} - E_{\text{optimised}}$ ▷ Energy decrease during optimisation
  $\Delta_{\text{trunc}} \leftarrow E_{\text{truncated}} - E_{\text{optimised}}$ ▷ Energy increase due to truncation
  if $\text{abs}(\Delta_{\text{opt}}) < \varepsilon \vee \text{abs}(\Delta_{\text{trunc}}) < \varepsilon$ **then**
    if $\text{abs}(\Delta_{\text{trunc}}) > \varepsilon$ **then**
      $f \leftarrow 0.9$ ▷ Case A1
    else
      $f \leftarrow 1.001$ ▷ Case A2
    end if
  else
    $r \leftarrow \text{abs}(\Delta_{\text{trunc}}) / \text{abs}(\Delta_{\text{opt}})$
    if $\Delta_{\text{trunc}} < 0.$ **then**
      $f \leftarrow 2 * (r + 1.)$ ▷ Case B
    else if $r < 0.05$ **then**
      $f \leftarrow 1.2 - r$ ▷ Case C1
    else if $r > 0.3$ **then**
      $f \leftarrow 1/(r + 0.75)$ ▷ Case C2
    end if
  end if
  $f \leftarrow \max(0.1, \min(2, f))$ ▷ Limit multiplicative factor between $[0.1, 2]$
  $\alpha \leftarrow \alpha \cdot f$
  $\alpha \leftarrow \max(10^{-11}, \min(100, \alpha))$ ▷ Limit α between $[10^{-11}, 100]$
  return α
end procedure

description", i.e. a different number of sweeps at each bond dimension as well as a different increase in bond dimensions. To make results comparable, we have decided to fix the bond dimension from the first sweep onwards and then repeat the experiment using three different bond dimension to observe behaviour at different values of m – while not optimal for any of the algorithms presented, the collected data should allow for a reasonable comparison.

Hence, for each system, an appropriate maximal bond dimension $m_{\max}$ is selected based on its physical properties. Each algorithm is then run at this bond dimension as well as at two lower bond dimensions $m_{\max}/2$ and $m_{\max}/4$. Calculations are run until convergence based on the relative change in energy between two subsequent sweeps. The runtime to convergence is taken as the CPU time used until the converged energy value was output for the first time.

The error in energy relative to some near-exact value E_0 is plotted over the number of local updates as well as CPU time consumed during the calculation. Unfortunately, this error in energy is not directly comparable to the variance $\langle \hat{H}^2 \rangle - \langle \hat{H} \rangle^2$, nor to the 2DMRG truncation error and very different physical states may lie very close in energy.

Convergence to the best possible value in the energy is hence of utmost importance.

Calculations were performed on a single core of a Xeon E5-2650 with the DMRG3S, CWF-DMRG and 2DMRG implementations in the Matrix Product Toolkit.[53] The DMRG3S and 2DMRG methods have later also been implemented in the (newer) SyTen toolkit with comparable characteristics.

The results are summarised in Figs. 3.5 and Tabs. 3.1 and will be discussed in detail in the following paragraphs. These figures and tables were all previously published in Ref. [2].

The $S = 1$ Heisenberg spin chain on $L = 100$ sites with periodic boundary conditions is described by the Hamiltonian

$$\hat{H} = \sum_{i=1}^{100} \hat{s}_i \cdot \hat{s}_{(i+1)\%100} \,. \tag{3.4.5}$$

This system is a standard benchmarking system with a well-known near-exact ground-state energy[64] value $E_0 = -140.148\,404$. The periodic boundary conditions result in a somewhat non-local optimisation problem, as the entanglement between the first and last site has to be introduced by the DMRG method during the calculation. Furthermore, while the system itself is non-critical, the PBC do increase the number of states required to represent the ground state accurately. As such, we choose $m_{\max} = 800$ with calculations also run at $m = 200$ and $m = 400$. Only the U(1) symmetry of the problem was exploited to facilitate comparison with other implementations.

During the calculation, DMRG3S initially exhibits a smaller convergence rate per sweep than CWF-DMRG. After the fourth to fifth sweep, however, convergence rates per sweep are comparable (at low accuracies) or greatly improved (at high accuracies), while the computational effort of each sweep is much lower with DMRG3S than with CWF-DMRG. The relative speed-up of DMRG3S over CWF-DMRG is then approx. $2.6, 1.3$ and 2.7 for $m = 200, 400$ and 800. The 2DMRG method, on the other hand, failed to pick up the periodic boundary conditions properly, resulting in an approximately ten times larger relative error in energy. The runtime to convergence of 2DMRG is hence not directly comparable.

Dilute Bosons on an optical lattice are used as an example of a non-homogeneous system due to the modulated potential. The lattice consists of 10 unit cells of the periodic external potential, each discretised as 16 sites. Each of those sites can carry at most $n_{\max} = 5$ bosons. The Hamiltonian of the system is

$$\hat{H} = + \sum_{i=1}^{160} \hat{n}_i \left\{ \cos^2 \left(2\pi \frac{i - 0.5}{16} \right) + (\hat{n}_i - 1) \right\}$$

$$-\sum_{i=1}^{159}\left\{\hat{c}_i^\dagger\hat{c}_{i+1}+\text{h.c.}\right\}\ . \tag{3.4.6}$$

$n=80$ bosons are placed in the system initially, making use of the $\mathrm{U}(1)_{\text{Charge}}$ symmetry of the Hamiltonian.

The system is relatively easy for DMRG assuming a good initial distribution, however, creating this distribution requires large-scale reshuffling of particles (which, in part, motivated the MG-DMRG variant[66]). We hence allow $m=50,100,200$ states which is sufficient to describe the ground state. The ground-state energy is taken as $E_0=-103.646\ 757$, which is achieved by all three algorithms at $m=200$ due to their reshuffling and basis optimisation abilities. If one were to apply the single-site update without perturbation or expansion, it would fail very badly here.

At small bond dimensions, the improved convergence behaviour per sweep leads to a speed-up of DMRG3S over CWF-DMRG of approximately 2. At larger bond dimensions, CWF-DMRG also converges relatively well, but its numerical operations become more costly, leading again to a speed-up of 2 between DMRG3S and CWF-DMRG. In comparison, the larger local dimension leads to slower calculations for 2DMRG, resulting in a total speed-up of 3.3 between DMRG3S and 2DMRG at $m=200$.

The Fermi-Hubbard model in one dimension at intermediate interaction strength already proves a difficult problem and relatively large bond dimensions are required to represent the ground state accurately. Furthermore, in combination with the free fermions in the next section, we can study how criticality and increased entanglement affect the three methods. We consider a system of $L=100$ sites with open boundary conditions at quarter-filling. The Hamiltonian is given by

$$\hat{H}=-\sum_{i=1}^{99}\sum_{\sigma=\uparrow,\downarrow}\left[\hat{c}_{i,\sigma}^\dagger\hat{c}_{i+1,\sigma}+\text{h.c.}\right]+\sum_{i=1}^{100}\hat{n}_{i,\uparrow}\hat{n}_{i,\downarrow}\ . \tag{3.4.7}$$

$\mathrm{U}(1)_{\text{Charge}}$ and $\mathrm{U}(1)_{S_z}$ symmetries are both employed with a total of fifty fermions and $S_z^{\text{total}}=0$. The selected bond dimensions are $m=300,600$ and 1200. At $m=1200$, all methods converge to the value $E_0=-84.255\ 525\ 4$.

Comparing the convergence of DMRG3S and CWF-DMRG, we can see that DMRG3S converges faster both in the number of sweeps and in the computational time used, while CWF-DMRG exhibits a very long tail of slow convergence. All three methods converge to approximately the same energy values at the different bond dimensions. DMRG3S is consistently the fastest method, while 2DMRG is slightly faster than CWF-DMRG at small bond dimensions. At the largest bond dimension, DMRG3S is approximately 2.6 times faster than CWF-DMRG and approximately 3.9 times faster than 2DMRG.

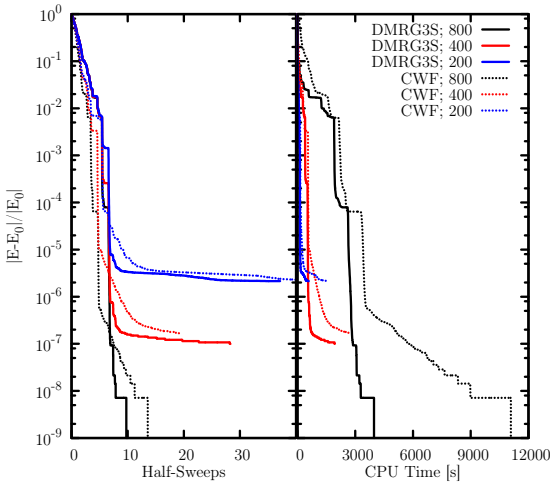
Free fermions in real-space are maximally delocalised and hence one of the most difficult problems for MPS-DMRG to solve in one dimension. However, apart from the increased entanglement, the other parameters of the system (local physical dimension, MPO bond dimension etc.) are exactly the same as in the Fermi-Hubbard model of the previous section. The free fermionic Hamiltonian is given by

$$\hat{H} = -\sum_{i=1}^{99} \sum_{\sigma=\uparrow,\downarrow} \hat{c}_{i,\sigma}^{\dagger} \hat{c}_{i+1,\sigma} + \text{h.c.} \tag{3.4.8}$$

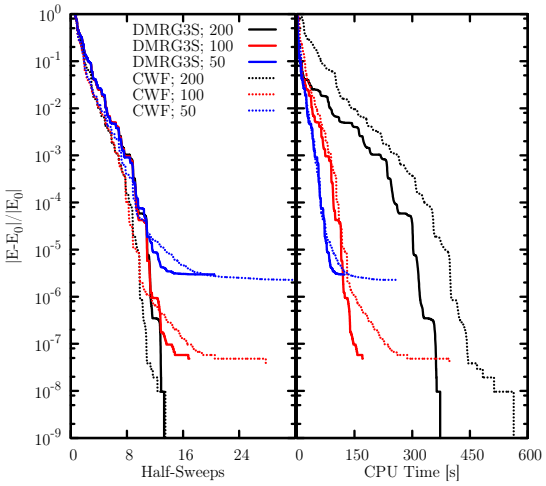
and we place $N = 100$ fermions in the system (half-filling). The values of m are the same as before, $m = 300, 600, 1200$. E_0 is given as $-126.602\,376$.

The accuracy of all methods is essentially identical, meaning that the bond dimension and not so much convergence problems are the limiting factor in this setting. However, the time to convergence varies greatly, especially at large bond dimensions: with $m = 1200$ states, DMRG3S is more than twice as fast as CWF-DMRG and converges more than six times faster than 2DMRG. At smaller bond dimensions, the differences are not as stark, as the (equally expensive) sub-leading terms in m become more relevant in all three methods.

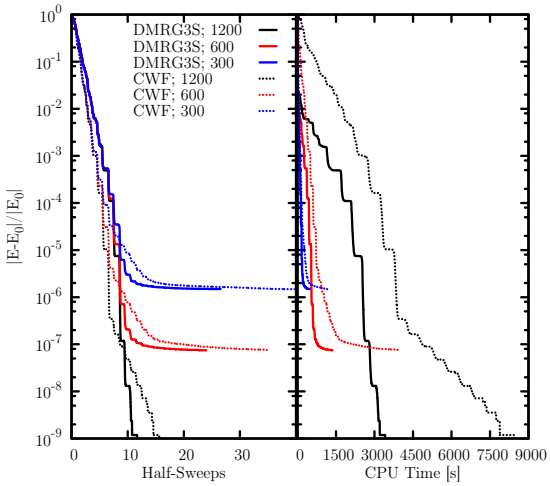
Comparing with the easier Fermi-Hubbard model with finite interactions, convergence becomes slower and the errors in energy become larger as we go to the non-interacting system, as expected. Likely due to the more expensive calculations, the speed-up between DMRG3S on the one hand and CWF-DMRG and 2DMRG on the other hand also becomes more pronounced.



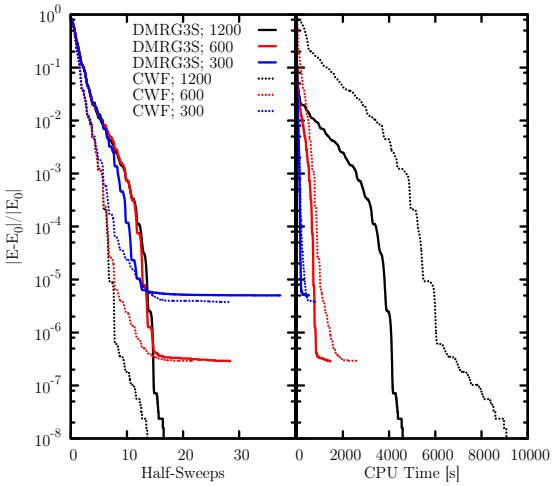
(a) Heisenberg spin chain Eq. (3.4.5)



(b) Bosons on an optical lattice Eq. (3.4.6)



(c) Fermi-Hubbard chain Eq. (3.4.7)



(d) Free fermions Eq. (3.4.8)

Figure 3.5: Convergence behaviour of DMRG3S and CWF-DMRG applied to a series of physical systems at three different bond dimensions. The energy is given with respect to some near-exact reference value E_0. The faster convergence of DMRG3S (solid lines) compared to CWF-DMRG (dotted lines) over computational time (right panes) is obvious, especially at the largest bond dimension (in black). Compared on a sweep-by-sweep basis (left panes), CWF-DMRG sometimes fares better, in particular early during the calculation. However, each individual sweep is always faster with DMRG3S. Curves for 2DMRG are not shown to preserve readability (First published in Ref. [2]).

Table 3.1: Runtime to convergence of DMRG3S, CWF-DMRG and 2DMRG applied to a series of physical systems at three different bond dimensions. The energy errors with respect to some reference value E_0 achieved by all three methods are comparable except for the Heisenberg spin problem, where 2DMRG struggles to establish the long-range interaction representing periodic boundary conditions. In all cases, DMRG3S takes the least computational time, with the relative speed-up to 2DMRG and CWF-DMRG increasing at higher accuracies and larger bond dimensions (First published in Ref. [2]).

(a) Heisenberg spin chain Eq. (3.4.5)

	$m = 200$	$m = 400$	$m = 800$
DMRG3S Energy Error	2.1×10^{-6}	1.0×10^{-7}	7.1×10^{-9}
CWF Energy Error	2.8×10^{-6}	1.7×10^{-7}	7.1×10^{-9}
2DMRG Energy Error	1.1×10^{-5}	8.6×10^{-7}	1.0×10^{-7}
DMRG3S Runtime	583 s	1935 s	3990 s
CWF Runtime	1519 s	2695 s	11 133 s
2DMRG Runtime	762 s	3181 s	21 963 s

(b) Bosons on an optical lattice Eq. (3.4.6)

	$m = 50$	$m = 100$	$m = 200$
DMRG3S Energy Error	2.9×10^{-6}	4.8×10^{-8}	$< 10^{-9}$
CWF Energy Error	2.3×10^{-6}	3.9×10^{-8}	$< 10^{-9}$
2DMRG Energy Error	1.9×10^{-6}	2.8×10^{-8}	$< 10^{-9}$
DMRG3S Runtime	124 s	171 s	469 s
CWF Runtime	260 s	397 s	951 s
2DMRG Runtime	210 s	462 s	1550 s

(c) Fermi-Hubbard chain Eq. (3.4.7)

	$m = 300$	$m = 600$	$m = 1200$
DMRG3S Energy Error	1.5×10^{-6}	7.5×10^{-8}	$< 10^{-9}$
CWF Energy Error	1.5×10^{-6}	7.6×10^{-8}	$< 10^{-9}$
2DMRG Energy Error	1.3×10^{-6}	6.4×10^{-8}	$< 10^{-9}$
DMRG3S Runtime	474 s	1367 s	3955 s
CWF Runtime	1215 s	3917 s	10 122 s
2DMRG Runtime	727 s	2950 s	15 596 s

(d) Free fermions Eq. (3.4.8)

	$m = 300$	$m = 600$	$m = 1200$
DMRG3S Energy Error	5.0×10^{-6}	2.8×10^{-7}	$< 10^{-9}$
CWF Energy Error	3.8×10^{-6}	2.8×10^{-7}	$< 10^{-9}$
2DMRG Energy Error	3.7×10^{-6}	2.6×10^{-7}	$< 10^{-9}$
DMRG3S Runtime	533 s	1452 s	4643 s
CWF Runtime	863 s	2590 s	9586 s
2DMRG Runtime	794 s	4584 s	29 698 s

3.4.3 Applied to DMRG with Local Basis Optimisation

Recent work[69] introduced the *local basis optimisation* (LBO-MPS) method for MPS applications. The LBO-MPS method works by first transforming the *full* local basis into an optimised local basis on each site via a matrix V_i. This matrix then attaches to the physical leg of the MPS rank-3 tensor M_i as usual. The advantage is that while the full basis may be very large (e.g. containing between 0 and $N = 500$ bosons on each site), the effective basis that is actually needed may be much smaller. If the boson number is not conserved, V_i may transform the bare boson modes into some sort of effective modes (of course depending on the model at hand) containing different superpositions of bare modes. If the boson number is conserved, such superpositions are not allowed (V_i has to be diagonal in the boson number), but we may still discard some or most of the states. Florian Dorfner[70] implemented this MPS-LBO method first for DMRG applications, using the subspace expansion method presented here to dynamically select different sectors. However, in that work they only considered non-conserved boson numbers and it turns out that the expansion as presented there is not suitable to achieve convergence if the boson number is conserved.

In the implementation for SyTen, a different expansion scheme was found which appears to lead to convergence in all cases. The expansion starts after the optimisation of the local M_i tensor and prior to the (say) left-normalisation. It first builds the full basis tensor $M_i = M_i \cdot V_i$ on the local site and the corresponding expansion tensor as described above. It then uses the subspace expansion on the full site tensor as usual and afterwards decomposes the full tensor on site i again into the new matrix V_i (normalised downwards) and the rank-3 MPS tensor. The tensor M_i is left-normalised with the usual truncation. On the next site, V_{i+1} will in general still be orthogonal to potential states now-contained in M_{i+1}. Hence, after a local optimisation of M_{i+1}, we subspace-expand M_{i+1} *upwards* using $L_i \cdot M_{i+1} \cdot V_{i+1} \cdot W_{i+1} \cdot R_{i+2}$ with corresponding zero-padding in V_{i+1}. The optimisation of V_{i+1} can then take those states into account if appropriate and a subsequent optimisation of M_{i+1} may also use those states (as they are not orthogonal anymore to V_{i+1}).

The scaling of the LBO-MPS-DMRG with this subspace expansion is slightly worse than without: All other operations scale as $O(m^3 d_o w) + O(m^2 d d_o w^2)$ where d is the dimension of the full basis and d_o the optimised basis dimension. However, the SVD following the expansion scales as $O(m^2 d^2)$ if $m^2 > d$ and $O(m^4 d)$ if $m^2 < d$. This means that if $d > d_o w^2$, the scaling in d is now (possibly substantially) worse. If boson numbers are conserved, the expansion term will often not exploit the full d-dimensional basis, resulting in better behaviour.

It is useful to point out that while a truncation on the LBO bond between V_i and M_i incurs the exact same error in the overall state as a truncation on the MPS bond between M_i and M_{i+1}, it may still be preferable to incur a larger error in the LBO bond

than the MPS bond. To this end it is useful to specify two different bond dimensions and truncation thresholds, one applied to MPS-MPS tensor SVDs and one applied to MPS-LBO matrix SVDs.

In practice, the method now also converges if the optimised boson number is conserved with similar speed-ups as presented in Ref. [70]. Of course, the exact speed-up always strongly depends on the system at hand, but an about 50% faster calculation is entirely possible. Furthermore, since each optimisation occurs in a smaller Hilbert space, the inner Lanczos optimisations are also more stable and faster, leading to nicer convergence behaviour.

3.4.4 Conclusions and Acknowledgements

The DMRG3S method results in a theoretical speed-up of $\approx (d+1)/2$ at each Lanczos iteration step compared to the CWF-DMRG method, assuming that d^2w/m is small. Numerical experiments show a consistent speed-up over CWF-DMRG, both due to faster numerical operations and faster convergence per sweep via the improved enrichment step after each local optimisation. This speed-up is of particular importance during the computationally most expensive last phase of calculations, where the desired accuracy and the utilised bond dimension m are particularly high. Additionally, we provided further evidence that 1DMRG methods with enrichment provide a consistent speed-up over 2DMRG at approximately equal accuracies.

We would like to thank Sergey Dolgov, Dmitry Savostyanov and Ilya Kuprov for very helpful discussions kickstarting the initial work on this method.

4 Spinon Confinement: A Time Evolution Study

The SyTen toolkit was used to calculate the dynamical structure factors and the excitation gaps of a one-dimensional $S = 1/2$ spin chain inside a staggered magnetic field, which can be used to model the behaviour of certain weakly-linked three-dimensional materials.[4]

These calculations will be motivated and discussed in detail in Sec. 4.2. First, however, the opportunity will be used as an excuse to provide a brief overview over available time-evolution methods for matrix-product states, as the time evolution is a crucial ingredient for the calculation of the dynamical structure factor. The same methods can also be used in the context of DMFT[71] to calculate the necessary Green's functions on the embedded impurity systems or to evaluate arbitrary time-dependent quantities in a generic setting, e.g. after a local or global quench or again in an embedding setting, e.g. for time-dependent DMET.

4.1 Overview of Available Time Evolution Methods

In the following, we will consider methods which, given a state $|\psi\rangle$ and an operator $\hat{H}$, are able to provide a time-evolved state

$$|\psi(\tau)\rangle = \exp\left(-\tau\hat{H}\right)|\psi(0)\rangle \quad . \tag{4.1.1}$$

τ here may be purely real (for imaginary time evolution) or purely imaginary (for real-time evolution). Complex τ are also feasible, but are less applicable to practical situations. $|\tau|$ will usually be very small, e.g. between 10^{-3} and 10^{-1}.

Some length will be spend on a qualitative comparison of each method, but no large-scale numerical comparison will be carried out within this thesis, as the individual methods each have various strengths and weaknesses tested by different physical systems. A fair comparison of all methods would hence require the inclusion of an impractically large representative sample of these physical systems and in any case would not necessarily be useful if the system in question was not contained within that sample.

We will generally assume that the states $|\psi(0)\rangle$ and $|\psi(\tau)\rangle$ are expressed as matrix-product states. Three of the four methods discussed in detail (Taylor expansion, TDVP and Krylov approximation) only require the matrix-product operator representation

of the operator $\hat{H}$. The fourth method, TEBD, however does require some (analytic) preprocessing and hence an analytical expression for $\hat{H}$.

4.1.1 Taylor Expansion

The certainly most straightfoward way to evaluate Eq. (4.1.1) is to expand the exponential as a Taylor series:

$$\exp\left(-\tau\hat{H}\right) = \hat{1} - \tau\hat{H} + \frac{\tau^2}{2}\hat{H}^2 - \frac{\tau^3}{6}\hat{H}^3 + \frac{\tau^4}{24}\hat{H}^4 - \dots \quad . \tag{4.1.2}$$

If τ is very small, this expansion will in principle be very accurate. If $\hat{H}$ is provided as a MPO, it is possible to calculate $\hat{H}^n$ as $\hat{H} \cdot \hat{H}^{n-1}$ at cost approximately $O(w^n)$, excluding a certainly possible compression of the resulting MPO. For small n (e.g. $n \leq 5$) and relatively simple operators ($w \leq 10$), this is feasible. Once the time-evolution operator $\hat{U}(\tau) \approx \exp\left(-\tau\hat{H}\right)$ is available, it can be applied to arbitrary states as often as necessary at relatively small cost (compared to the initial construction of $\hat{U}(\tau)$).

Unfortunately, of course, the method is severely limited by the range of convergence of the Taylor series, which is in principle uncontrolled and the exponentially-increasing cost to go to higher terms of the series. Furthermore, the time-evolution operator is not unitary by construction, resulting in a change of the norm of the time-evolved state. For simple systems and a few, small, time-steps, it is however an extremely simple method to provide relatively good data, which is useful for the validation and testing of the more advanced methods.

4.1.2 Time-Evolution with Block Decimation (TEBD)

The TEBD algorithm (usually "time-evolution with block decimation") was originally developed[72] by Guifré Vidal directly for the application to MPS in the early 2000s and may well be one of the earliest uses of the matrix-product formulation as opposed to the environment-system DMRG formulation. Equivalent ideas had been pursued from the traditional DMRG viewpoint at the same time, culminating in a very fast adaption of the TEBD to traditional DMRG.[73,74] By now, the naming of this and related methods is entirely confused, all of "tDMRG", "tMPS", "Trotter decompositions" and "TEBD" are used[21,75–77] (or explicitly not-used) to refer to the same underlying idea when applied directly to MPS or to environment-system DMRG.

The key insight behind the method is that while $\exp\left(-\tau\hat{H}\right)$ is difficult to compute, $\exp\left(-\tau\hat{H}_s\right)$, where $\hat{H}_s$ only acts non-trivially on a very small part of the system, can be calculated exactly. The go-to example is the one-dimensional spin chain with nearest-neighbour interactions. There, we have

$$\hat{H} = \sum_{i=1}^{L} \hat{s}_i \cdot \hat{s}_{i+1} \quad . \tag{4.1.3}$$

Calculating $\exp\left(-\tau\hat{H}\right)$ is impossible, but calculating $\exp\left(-\tau\hat{s}_i \cdot \hat{s}_{i+1}\right)$ only requires the exponentiation of a $(2S+1)^2 \times (2S+1)^2$ matrix, where S is the length of the local spins. For example, in the case of $S = 1/2$, this matrix is a four-by-four matrix which can be diagonalised (and hence exponentiated) exactly.

To make this method work, it is necessary to split $\hat{H}$ into parts $\hat{H}_A$, $\hat{H}_B$ etc. which only contain mutually-commuting small operators. In the example above, we could collect all $\hat{s}_i \cdot \hat{s}_{i+1}$ with i even into $\hat{H}_A$ and all those with i odd into $\hat{H}_B$. $\exp\left(-\tau H_A\right)$ etc. can then be calculated exactly by individually exponentiating each of the contained parts.

This is essentially the first step towards a first-order Suzuki-Trotter[78] decomposition, namely writing

$$\exp\left(-\tau\hat{H}\right) = \exp\left(-\tau\hat{H}_A\right)\exp\left(-\tau\hat{H}_B\right) + O(\tau^2) \quad . \tag{4.1.4}$$

This approach has the advantage that, if $\exp\left(-\tau\hat{H}_{\{A,B\}}\right)$ is calculated exactly, even the approximated evolution operator is unitary and hence preserves the norm of the state.

Similarly, a second-order decomposition

$$\exp\left(-\tau\hat{H}\right) = \exp\left(-\tau/2\hat{H}_A\right)\exp\left(-\tau\hat{H}_B\right)\exp\left(-\tau/2\hat{H}_A\right) + O(\tau^3) \tag{4.1.5}$$

and a fourth-order decomposition

$$\begin{aligned}
\exp\left(-\tau\hat{H}\right) = {} & \exp\left(-\frac{s}{2}\tau\hat{H}_A\right)\exp\left(-s\tau\hat{H}_B\right)\exp\left(-s\tau\hat{H}_A\right) \\
& \times \exp\left(-s\tau\hat{H}_B\right)\exp\left(-\frac{1-3s}{2}\tau\hat{H}_A\right) \\
& \times \exp\left(-(1-4s)\tau\hat{H}_B\right) \\
& \times \exp\left(-\frac{1-3s}{2}\tau\hat{H}_A\right)\exp\left(-s\tau\hat{H}_B\right) \\
& \times \exp\left(-s\tau\hat{H}_A\right)\exp\left(-s\tau\hat{H}_B\right)\exp\left(-\frac{s}{2}\tau\hat{H}_A\right) \\
& + O(\tau^5) \qquad (4.1.6) \\
s = {} & \frac{1}{4 - \sqrt[3]{4}} \qquad (4.1.7)
\end{aligned}$$

were found by Masuo Suzuki (cf. Ref. [79] for a good overview). Both the second- and fourth-order decomposition have the additional advantage that they are symmetric in the sense that $\exp\left(-\tau\hat{H}\right)\exp\left(\tau\hat{H}\right) = \hat{1}$. Once the time-evolution operator has been found following the above prescription, it can be applied straight-forwardly to the state to be

time-evolved.

Alternatively, the individual gates formed by the exponentials of $\hat{H}_A$ and $\hat{H}_B$ may be applied one-by-one on the relevant bonds – this saves (minor) computational effort, as we explicitly exploit the fact that the MPO bonds are one-dimensional on every other bond. Additionally, this allows for a real-space parallelisation[80] by applying each gate individually. A bachelor student supervised during this PhD[6] implemented the method and found an acceptable speed-up compared to the unparallelised version. Unfortunately, the additional error introduced due to a non-optimal truncation (each parallel truncation necessarily assumes the rest of the state to be unchanged) led to a much larger error resulting from the parallelised approach. There is reason to assume, however, that with minor tweaks, this may become a viable parallelised method. Such tweaks could include periodical serial re-orthonormalisation to propagate the information from the time evolution thoughout the state.

The time evolution with TEBD is extremely fast and has only two well-controlled error sources: the Trotter error arising from the decomposition and the truncation error incurred during the evolution. Both can easily be taken as small as necessary, though of course at increased computational effort.

The main drawback of this method is then the analytical effort required to decompose the desired operator $\hat{H}$ into mutually-commuting parts as well as the restriction to nearest-neighbour interactions: if longer-range interactions exist, the corresponding sites have to be moved next to each other prior to each individual time-evolution step[81] which, however, increases the computational effort.

4.1.3 Time-Dependent Variational Principle (TDVP)

The time-dependent variational principle essentially attempts to find the time-evolved state by deriving time-evolution equations for each individual site tensor of the MPS and solving them subsequently. In the context of MPS, it was first proposed by Jutho Haegeman et. al.[82] That work also shows a very nice correspondence between TDVP-based time evolution of each individual tensor and ground-state search DMRG: the latter is simply the case of taking infinite steps along the imaginary axis during each individual tensor update. Then, similarly to how $\lim_{\tau\to\infty}\left(e^{-\tau\hat{H}}|\psi\rangle\right)$ will converge to the lowest-energy state of $\hat{H}$ with which $|\psi\rangle$ has non-zero overlap, each infinitely-imaginary-time-evolved tensor will approach the lowest-energy ground-state tensor as found by DMRG.

The TDVP has a number of very appealing features: First, it only requires a MPO representation of the Hamiltonian $\hat{H}$ in the time-evolution operator $\hat{U}(t) = e^{-\tau\hat{H}}$, meaning that no analytical decomposition (as for TEBD) is necessary. Second, it exists in both a single-site and a two-site variant. The single-site variant scales in exactly the same way as standard 1-DMRG, i.e. linearly in the local physical dimension. Third, the TDVP inherently preserves both the norm and the energy of the state, i.e. is exactly unitary.

Fourth, similarly to the TEBD method and different from the Krylov method, it generates the time-evolved state directly instead of first building a set of potentially very large Krylov vectors which then may or may not cancel out to form the evolved state.

On the other hand, there are certain problems with it: compared to the exact TDVP equation, the finite time-step of size Δt incurs an error $O(\Delta t^3)$. Furthermore, we have observed relatively large errors due to the TDVP projection itself (compared to an exact method) as soon as the Hamiltonian is not restricted to nearest-neighbour interactions any more. This can be understood intuitively when comparing the single-site TDVP (where even a nearest-neighbour Hamiltonian $\hat{H}$ incurs an error due to the TDVP projection) to the two-site TDVP, where such a Hamiltonian incurs no projection error.[82] It is then reasonable to assume that longer-range interactions require "three-site TDVP" to be captured exactly. While this projection error can be measured and monitored in principle, there does not seem to be a straight-forward way to avoid it. Worse, this error becomes larger if many small timesteps are taken as compared to taking one larger time step – in a certain sense, the TDVP equations provide the exact evolution followed by a projection back onto the MPS manifold. Chaining many such projections then incurs a larger error than only projecting back once. This implies a competition between the integration error, which becomes smaller as the time step becomes smaller, and the projection error, which becomes smaller as the number of time steps decreases.

However, in problems with very short-range interactions (nearest- or next-nearest-neighbour only), the two-site TDVP provides nearly error-free and fast results. In very complicated problems where the truncation errors dominate any other errors, even the single-site TDVP is also competitive error-wise and much faster than all other methods.

4.1.4 Krylov Approximation

The method presented here under the name "Krylov approximation" is a variant of the Lanczos algorithm. As such, it first builds an orthonormal space span $\left(|\psi\rangle, \hat{H}|\psi\rangle, \hat{H}^2|\psi\rangle, \dots\right)$, projects $\hat{H}$ into that space, diagonalises the resulting small matrix T and evaluates Eq. (4.1.1) by approximating the action of the exponential onto $|\psi\rangle$ using that diagonalisation.

This approach is well-known in the general numerical linear algebra community[83] for its advantage of only needing to evaluate the action of the operator onto $|\psi\rangle$ instead of, say, evaluating operator-operator products. It is in principle straightforward to use it while representing the states $\left\{|\psi\rangle, \hat{H}|\psi\rangle, \hat{H}^2|\psi\rangle, \dots\right\}$ as MPS and the operator $\hat{H}$ as a MPO.[75,77] However, due to the special matrix-product structure, multiple caveats have to be taken into account:

Loss of orthogonality is much more pronounced when using naive MPS-MPS arithmetic to (re-)orthogonalise states. MPS arithmetic usually results in an increased bond dimension

Algorithm 4.1 A re-ordered version of the classical Lanczos algorithm to generate a Krylov subspace approximation of a Hermitian operator $\hat{H}$. The matrix T represents $\hat{H}$ projected into the low-dimensional space and can be exponentiated exactly. Well-known bounds on the errors incurred by the projection are available.[83] The most costly step is the operator application with simultaneous orthonormalisation in Line 7, evaluation of expectation values usually only takes 10%-20% of total runtime. A good measure of convergence is the change in the approximation of $e^{-\tau\hat{H}}|\psi\rangle$ due to the last added vector $|v_n\rangle$.

1: **procedure** KRYLOV-GENERATE(MPO $\{W_i\}_i = \hat{H}$, normalised MPS $\{M_i\}_i = |\psi\rangle$)
2: $|v_1\rangle \leftarrow |\psi\rangle$ ▷ First Krylov vector
3: $T_{11} \leftarrow \langle v_1|\hat{H}|v_1\rangle$ ▷ Projected matrix representing $\hat{H}$
4: $i \leftarrow 1$ ▷ Counter of Krylov vectors
5: **while** not converged **do**
6: $i \leftarrow i + 1$
7: $|v_i\rangle \leftarrow \hat{H}|v_{i-1}\rangle \perp \{|v_j\rangle\}_{j=1}^{i-1}$ ▷ Apply $\hat{H}$ and orthonormalise against all $|v_j\rangle$
8: $T_{i,i-1} \leftarrow T_{i-1,i} \leftarrow \langle v_i|\hat{H}|v_{i-1}\rangle$ ▷ b_i in a usual Lanczos procedure
9: $T_{i,i} \leftarrow \langle v_i|\hat{H}|v_i\rangle$ ▷ a_i in a usual Lanczos procedure
10: **end while**
11: **return** T, i, $\{|v_j\rangle\}_{j=1}^{i}$
12: **end procedure**

which then has to be truncated to the desired precision again. However, this truncation does not take the previously-achieved orthogonality into account and may well destroy it. The solution to this is a variational orthogonalisation procedure (cf. Sec. 2.3.5) which optimises a state to be orthogonal against a set of other states without increasing its bond dimension.

The cost of operations differs between "dense" numerical linear algebra and tensor networks. In particular, while for dense matrices X and vectors a, b, the expectation value $b \cdot Xa$ can only be evaluated by first applying X to a and then taking the overlap with b, this is not true for tensor networks. In fact, evaluating $\langle a|\hat{X}|b\rangle$ is much cheaper than evaluating $\hat{X}|b\rangle$ alone. This motivates a slight re-shuffling of the Krylov iterations which results in only needing $N-1$ MPO-MPS applications in order to produce a N-dimensional Krylov space. The re-ordered algorithm is summarised in Alg. 4.1.

The application of a MPO to a MPS may be very costly, in particular for large MPOs. Furthermore, using either the zip-up method[52] or the naive approach requires subsequent orthogonalisation against (at least the previous two) other Krylov vectors. However, when variationally applying a MPO to a MPS as summarised in Sec. 2.4.3 and previously published e.g. in Ref. [75], it is possible to orthogonalise against the other Krylov vectors *at the same time* using the procedure outlined in Sec. 2.3.5. This combination of two steps leads to a much faster and much more scalable algorithm –

where before the computational cost for every additional Krylov vector grows very quickly (in line with the initial exponential growth of the bond dimension by a factor of w during every operator application step), now the computational effort per vector still grows, but at a much slower rate.

Re-using the Krylov space may be feasible and potentially very advantageous. If Krylov vectors have been built which proved sufficient to evaluated $e^{-\tau\hat{H}}|\psi\rangle$, the same space may also suffice to evaluate $e^{-2\tau\hat{H}}|\psi\rangle$ and potentially $3\tau, 4\tau$ etc. If it doesn't, adding another Krylov vector to increase the maximal reach in time may still be preferable over generating an entirely new set of vectors. A. Swoboda has developed an estimating heuristic which compares the time that was required to build the last Krylov vector to the potential maximal time achievable with an additional Krylov vector to make this decision. The primary concern here is not a loss of orthogonality which potentially occurs with very large subspaces, but rather the increasing cost to generate each Krylov vector: without compression, bond dimensions of the MPS will grow by a factor of the MPO bond dimension at each step, leading to potentially exponentially growing costs. Hence, generating the first three vectors a second time may well be faster than building the fourth Krylov vector, even if building a third vector was faster than regenerating the first two vectors. Where exactly this trade-off occurs depends strongly on the system as well as the initial state and time-step, necessitating a runtime heuristic to measure and adapt the algorithm as needed.

The variational operator application with immediate truncation has the added benefit that it allows for much larger Krylov vector spaces (with e.g. 10 or 20 vectors rather than just 10), which allows to reach superlinearly longer time scales due to the quadratic convergence behaviour of the Krylov method when the number of Krylov vectors is relatively small.[84]

Combining the improvements in the previous four paragraphs leads to a scalable and relatively fast Krylov-based time evolution method with an extremely well-controlled error: the MPS truncation error is of course exactly the same as elsewhere, while there is no inherent time-step error but only the (well-controllable and reducable) error from an insufficiently large Krylov space. Furthermore, the method generalises very easily to other tensor network states, such as MPS with a local basis optimisation or tree tensor networks, as the only network-specific operation is the calculation of expectation values as well as the operator application with simultaneous orthogonalisation against the previous Krylov vectors.

4.1.5 Other Methods and Summary

Various other methods exist to evaluate time-dependent quantities in a MPS setting. Among those are Chebyshev polynomials,[85,86] a decomposition of MPOs similar to TEBD but also applicable to long-range interactions[76] as well as Padé and Runge-Kutta approximations of the exponential.[77]

However, for most practical purposes, the choice of method largely depends on the desired precision and range of interactions. At low precision, it is no problem to reduce the time-step error inherent in TDVP and TEBD below the truncation error; in particular the TDVP then computes time-evolved states extremely quickly. At high precision, however, this becomes more and more difficult: while the MPS precision increases exponentially with the bond dimension, the time-step error only decreases polynomially with the time-step size. To match a small truncation error, exponentially small time-steps are necessary. In this area, the Krylov method can potentially overtake both other methods. Additionally, both TEBD and TDVP incur additional errors or increased computational complexity when used with long-range Hamiltonians – here, too, the Krylov method can control errors with costs only increasing polynomially in the MPO bond dimension, not the range of interactions itself.

4.2 Spinon Confinement in a Quasi One-Dimensional Heisenberg Magnet

The material representing the quasi one-dimensional Heisenberg magnet is $SrCo_2V_2O_8$ and was studied by Anup Bera and collaborators.[4] Interestingly, it crystallises in a series of twisted chains where the inter-chain interactions support long-range antiferromagnetic order via an effective mean-field h and, crucially, the Co^{2+} ions can be described as anisotropically coupled $S = 1/2$ Heisenberg spins in that staggered magnetic field h. While $S = 1/2$ spin chains are generally critical, the original assumption was that the staggered field h was sufficiently large to open a large gap and allow for an efficient treatment with MPS. It turned out, however, that the experimentally necessary value of h is ≈ 0.0063, in turn greatly complicating further MPS calculations and necessitating the involvement of another iMPS-based method.[87]

The qualitative picture however carries over very well from large values of $h \approx 0.1$ to this small-h regime, as can be seen from the good agreement between the MPS-based calculations carried out using SyTen, the iMPS calculations done by Laurens Vanderstraeten and finally the experimental results obtained by Bella Lake.

The effective Hamiltonian used to model the system under the assumption of the mean-field inter-chain interactions is given as

$$\hat{H} = \sum_{i=1}^{L} J\hat{s}_i^z\hat{s}_{i+1}^z + {}^{J\epsilon}/_2\left(\hat{s}_i^+\hat{s}_{i+1}^- + \hat{s}_i^-\hat{s}_{i+1}^+\right) - h(-1)^i\hat{s}_i^z \quad , \tag{4.2.1}$$

where we identify sites $L+1 \equiv 1$ for periodic boundary conditions. The anisotropy is given by ϵ while the magnitude of the staggered field is given by h. The system would be $\mathrm{SU}(2)_{\mathrm{Spin}}$-symmetric if $\epsilon = 1$ and $h = 0$, as it stands for the experimentally relevant $h \approx 0.0063$ and $\epsilon \approx 0.56$, only the $\mathrm{U}(1)_{S^z}$ symmetry is preserved.

The elementary excitations of this system are $S = {}^1/_2$ spinons,[88] arising from spin-charge separation of the Co^{2+} ions in the original compound. The observed spinon confinement then relates to a split of the spinon continuum observed in this compound above a certain temperature into a series of bound $S = 1$ magnon modes below the ordering temperature. Since our method currently works strictly at $T = 0$, we do not observe the spinon continuum. However, when setting the external field $h = 0$, the typical continuum of the standard Heisenberg spin chain, including gapless excitations, can in principle be observed.[i]

Here, the quantities of interest are the dynamical structure factor, the excitation gap and potentially the ordered magnetic moment. The dynamical structure factor is directly related to the experimental data obtained via neutron scattering while the excitation gaps relate to the energies of the bound states and hence δ-peaks in the dynamical structure factor or neutron scattering data.

In the following, we will first discuss the calculation of the dynamical structure factor using MPS-Krylov time evolution. This is followed by a brief discussion of the excitation gap calculations and a more thorough comparison of experimental and theoretical results.

4.2.1 Dynamical Structure Factors with MPS

Given the time-evolution methods presented above, it is straightforward to calculate real-time real-space two-point correlators. Specifically, we will evaluate quantities

$$C^{aa}(t,j) \equiv \langle 0|\hat{s}_j^a(t)\hat{s}_0^a(0)|0\rangle - \langle 0|\hat{s}_j^a(t)|0\rangle\langle 0|\hat{s}_0^a(0)|0\rangle \tag{4.2.2}$$

where $a = x, y, z$, j describes the position of the second excitation (the chain is centered on site 0) and $|0\rangle$ is the ground state of the Hamiltonian. The subtraction of the ground-state expectation values is necessary due to the staggered magnetic field leading to $\langle 0|\hat{s}_i^z|0\rangle \neq 0$. The quantity $C^{aa}(t,j)$ can be evaluated by time-evolving the state $\hat{s}_0^a|0\rangle$ to time t and taking the overlap with a series of states $\hat{s}_j^a|0\rangle$ while multiplying with the phase factor $e^{i\hat{H}t}$ arising from the second Heisenberg-picture exponential. Exemplary illustrations of these correlators $C^{aa}(t,j)$ are plotted via their absolute value in Fig. 4.1. There, the initially

[i] Entanglement growth during real-time evolution however makes it extremely difficult to obtain valid dynamical structure factors as calculated during this project.

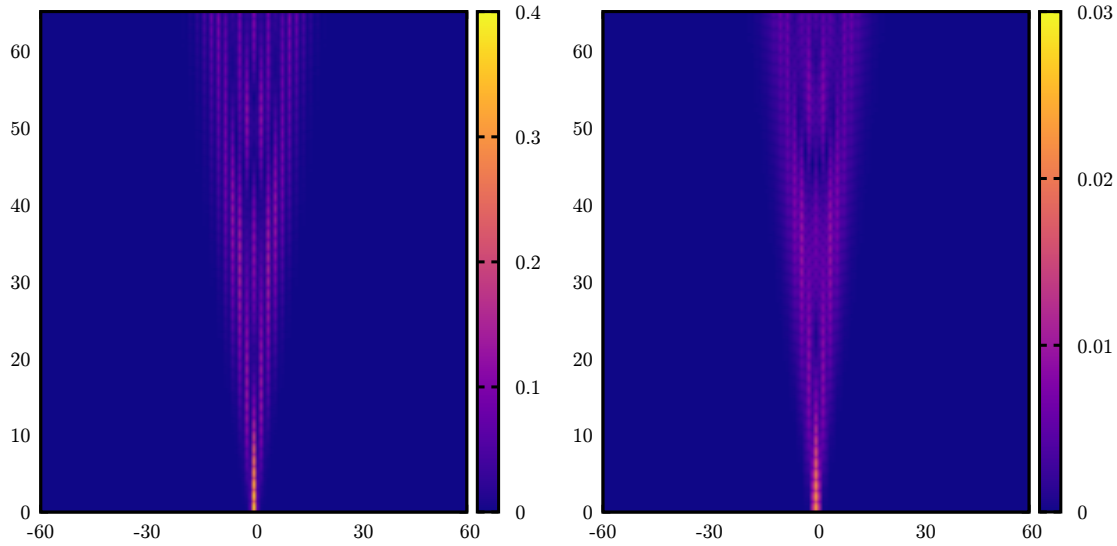


Figure 4.1: Illustrative example of real-space real-time correlators $C^{xx}(t,j)$ (left) and $C^{zz}(t,j)$ (right) as obtained from the time evolution; plotted are the absolute values of the correlators $|C^{aa}(t,j)|$. The relative position j of the second operator is on the horizontal axis while the time t is given on the vertical axis. Parameters for this plot are $\epsilon = 0.2$, $h = 0.05$.

strongly localised excitation spreads through the system over time. Since entanglement grows in the system, only a maximal time $t_{\max} \approx 65$ can be reached using the MPS-Krylov method. During this time frame, the excitation does not reach the edge of the system of size $L = 128$, implying that the system size was chosen sufficiently large.

Once the real-space real-time correlators are obtained, it is straightforward to perform a Fourier transformation into momentum space. Since the excitation is localised to the central part of the system for all times t, this operation is very easy and does not require any further numerical tricks. We obtain the momentum-space real-time correlators $C^{aa}(t,k)$ as

$$C^{aa}(t,k) = \frac{1}{L} \sum_{j=-L/2}^{L/2-1} e^{ijk} C^{aa}(t,j) \quad . \tag{4.2.3}$$

An exemplary illustration of this quantity for $a = x, z$ is given in Fig. 4.2, again plotting the absolute values of the correlators. As can be seen, the mirror symmetry around $k = \pi$ emerges very nicely without further input or symmetrisation.

However, given Fig. 4.2 and its underlying data, it becomes clear that the signal does *not* decay in time t or at least not sufficiently quickly to observe this decay using MPS time evolution methods. This makes the Fourier transformation into frequency space problematic. Ideally, we would like to define[73,89–92]

$$C^{aa}(\omega,k) = \int_{-\infty}^{\infty} \mathrm{d}t e^{i\omega t} C^{aa}(t,k) = 2\int_{0}^{\infty} \mathrm{d}t \mathrm{Re}\left\{e^{i\omega t} C^{aa}(t,k)\right\} \quad , \tag{4.2.4}$$

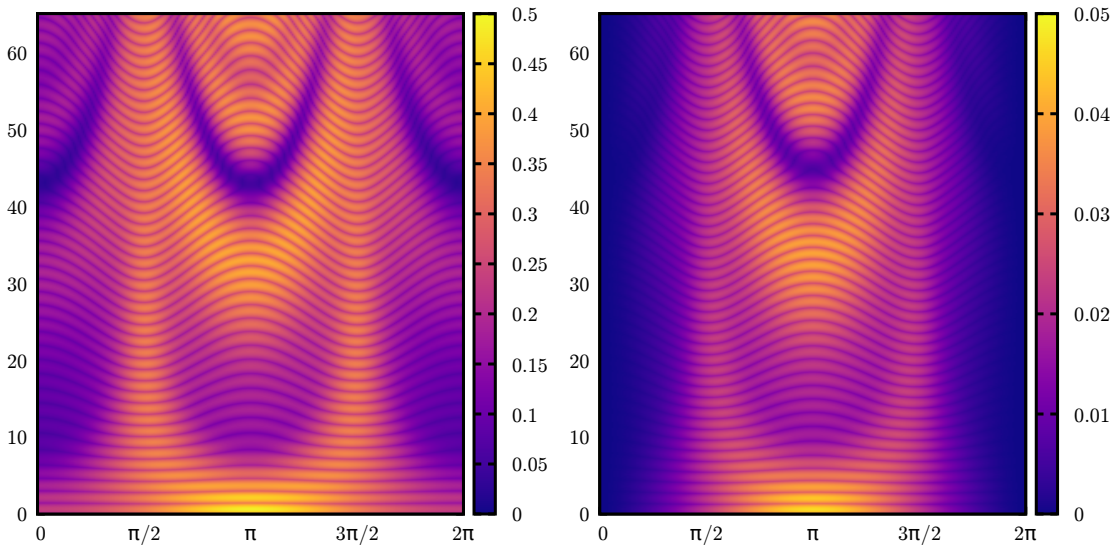


Figure 4.2: Illustrative example of momentum-space real-time correlators $C^{xx}(t,k)$ (left) and $C^{zz}(t,k)$ (right) as obtained from the time evolution; plotted are the absolute values of the correlators $|C^{aa}(t,k)|$. The momentum k is on the horizontal axis while the time t is given on the vertical axis. Parameters for this plot are $\epsilon = 0.2$, $h = 0.05$.

where the second equality sign relies in temporal and spatial translation invariance, spatial mirror symmetry over the site $j = 0$ as well as the hermiticity (or anti-hermiticity) of the operators $\hat{s}^a$. Due to the limited time frame available, this transformation leads to strong spectral leakage.[93] Our solution is the multiplication of an additional damping factor $e^{-\eta t}$ such that

$$C_\eta^{aa}(\omega,k) = 2\int_0^\infty \mathrm{d}t \mathrm{Re}\left\{e^{i\omega t}e^{-\eta t}C^{aa}(t,k)\right\} \quad . \tag{4.2.5}$$

For one of the investigated choices $\epsilon = 0.2$, this damping with $\eta \approx 1$ was sufficient to arrive at a clear signal due to a very strong input signal $C^{aa}(t,k)$. However, for the experimentally relevant value of $\epsilon = 0.56$, entanglement grew faster, leading to shorter attainable times and hence stronger spectral leakage. Increasing η to alleviate this problem lead to essentially no signal being left. To avoid this, we have decided to use a numerical prediction method similar to the previously-implemented[91] linear prediction method. Since our signal does not decay exponentially, we cannot use the linear prediction itself, but attempting to model

$$C_{\mathrm{fit}}^{aa}(t,k) \approx \sum_{m=1}^{M} a_m e^{-it\nu_m} \quad ; \quad a_m \in \mathbb{C} \quad \nu_m \in \mathbb{C} \tag{4.2.6}$$

was successful for $M \approx 10$ using the NLopt library[94] with its `LD_SLSQP` algorithm[95] to calculate the a_m and ν_m based on the achieved time frame. The expression can then be used to extend the value to much longer maximal times $t'_{\mathrm{max}} \approx 1000$. Using the extended time frame, setting $\eta \approx 10^{-2}$ provides a clear signal without spectral leakage and clearly

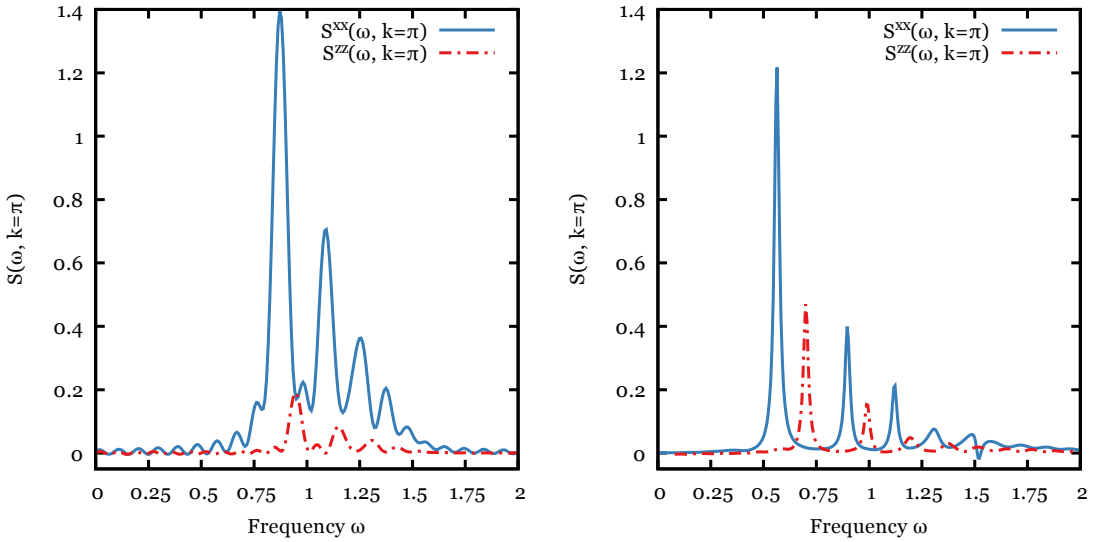


Figure 4.3: Dynamical structure factors as obtained from the MPS-Krylov calculation. Parameters are $h = 0.05$, $\epsilon = 0.2$ (left) and $\epsilon = 0.5$ (right). Numerical extension in time was used for the calculation with $\epsilon = 0.5$, subsequently leading to less spectral leakage than the (originally less problematic) case of $\epsilon = 0.2$.

identifiable peaks.

Of course, this numerical extension in time via a sum of exponentials is essentially a "poor-man's Fourier transform" and as such should be treated with some care. Agreement between the results thus obtained, other theoretical methods and the experimental data at least somewhat justifies the approach.

Finally, the dynamical structure factor is then simply given as

$$S^{aa}(\omega, k) = \frac{1}{4\pi} C^{aa}_{(\mathrm{fit}),\eta}(\omega, k) \quad . \tag{4.2.7}$$

Its value at $k = \pi$ is plotted in Fig. 4.3 for $h = 0.05$, $\epsilon = 0.2$ and $\epsilon = 0.5$ and is the primary quantity to be compared with the experimental data. In the former case, no extension in time was necessary when accepting minor spectral leakage, while the latter case did require the numerical extension in time to remove spectral leakage sufficiently. This leads to the counter-intuitively cleaner signal in the more difficult case.

4.2.2 Excitation Gaps

The dynamical structure factor gives the energy dependency of the relative weights of excitations. In a system with confinement and bound excited states, there are (hopefully) in principle relatively few relevant excitations directly above the ground state. This can be verified directly by the finite number of peaks of the dynamical structure factor calculated in the previous section and in particular the absense of a excitation (or spinon) continuum. It then becomes feasible to calculate the excitation gap both within the $S^z = 0$ sector

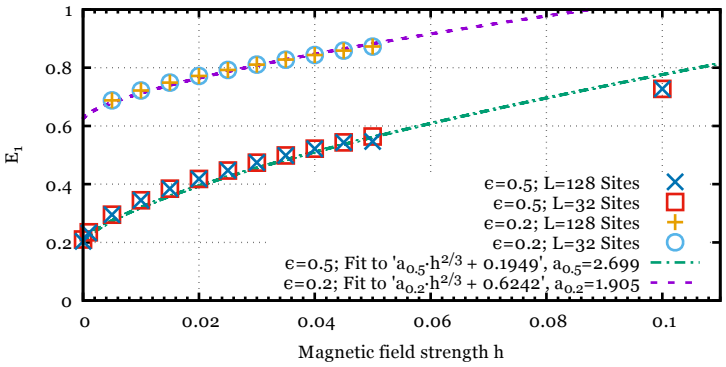


Figure 4.4: Excitation gaps into the transversal $S^z = 1$ sector for $\epsilon = 0.2$ and $\epsilon = 0.5$ for various values of h. Comparison with the experimentally-obtained data allows the determination of the effective staggered field strength h. Zero-field values 0.1949 and 0.6242 provided by Fabian H.L. Essler.

(longitudinal excitations) and to the $S^z = 1$ sector (transversal excitations) with DMRG. The former calculation first requires the ground state in that sector and then attempts to find another low-energy state under the condition that it must be orthogonal to the ground state. The latter calculation is simply a DMRG "ground state" calculation in a different, namely $S^z = 1$, quantum number sector. Both calculations are extremely stable if only the first excited state is required, but calculating higher excited states using this method quickly becomes unfeasible. In the present system, this problem is excarberated by the L-fold degeneracy of the first excitation energy, requiring $L + 1$ orthogonal states to get to the second excitation energy.

Nevertheless, the first excitation gap can be calculated reliably and very quickly. The value also quickly converges to the thermodynamic limit when increasing the system size – often, there is only negligible difference between the gaps for a $L = 32$ and $L = 128$ sites system. This then allows the determination of the effective staggered mean-field h present in the experiment by calculating the excitation gap for a series of values for h and comparing to the excitation energies measured experimentally.

For small values of h, field-theoretical results[88,96,97] predict a scaling of the excitation gap $E_1 \propto h^{\frac{2}{3}}$. As evident from Fig. 4.4, this scaling can be reproduced nearly perfectly for values of h up to ≈ 0.05, using the analytically-known values for the gap at $h = 0$ to determine the constant term.

4.2.3 Comparison with Analytical Approaches and Experimental Data

Recall the Hamiltonian of interest,

$$\hat{H} = \sum_{i=1}^{L} J\hat{s}_i^z\hat{s}_{i+1}^z + {}^{J\epsilon}/_2\left(\hat{s}_i^+\hat{s}_{i+1}^- + \hat{s}_i^-\hat{s}_{i+1}^+\right) - h(-1)^i\hat{s}_i^z \quad . \tag{4.2.8}$$

Exact analytical treatment of this Hamiltonian is possible in two limits: First, the strong anisotropy-weak field case $h, \epsilon \ll J$ can be treated with perturbation theory[4,98,99] to obtain the ground-state and the elementary excitations as domain-wall states between regions of the system in the ground state and continuous spin-flipped regions. The different parity of the number of flipped spins then manifests in the transverse (for odd parity, i.e. $S^z = 1$) and longitudinal (for even parity, i.e. $S^z = 0$) excitation sectors. Additionally, it is easy to see that with the external magnetic field, the additional energy of the excitation increases with the number of flipped spins. This is in contrast to the case $h = 0$, where the energy difference *only* depends on the number of domain walls and *not* their relative position. We can hence understand how the additional magnetic field leads to a confinement of the $S = {}^1/_2$ spinon excitations, binding them into a single $S = 1$ magnon excitation.

Alternatively, one may apply the powerful bosonisation approach[100] to the problem. This approach is explained in great detail in Ref. [101, 102]. One first writes down the *fermionic* model equivalent to the spin model using the standard Jordan-Wigner transformation and then bosonises that fermionic model to arrive at a solvable (in some limit) bosonic model which can be treated using standard analytical methods, e.g. perturbation theory,[103,104] to calculate both low-energy excitations and correlation functions, valid close to the isotropic point $\epsilon \approx 1$. This approach also[4] results in the same physical picture of confined (at temperature $T < T_N$) spinons and discrete excitation energies effected by the external magnetic field h.

Similarly, at $T < T_N$, the experimental data agrees qualitatively extremely well with both the results from the analytical approaches as well as the DMRG-MPS calculations described in detail above.

To achieve quantitative agreement between experiment and theory, it was however necessary to apply the novel *uniform MPS* techniques developed by Jutho Haegeman et al.[87,105] These techniques work directly in the thermodynamic limit and can calculate the low-energy single-particle excitation spectrum in principle exactly, limited only by the easily-controlled bond dimension of the uMPS ansatz states. With this method, the experimental data can be reproduced excellently apart from a minor overall shift of the energy.[4] Since the actual interactions between the chains of $SrCo_2V_2O_8$ are certain to go beyond the simple mean-field approach used to model these interactions here, the agreement is well within the expected range.

5 The Hubbard Model in Two Dimensions

The Hubbard Hamiltonian[106] is defined as

$$\hat{H} = t \sum_{\langle i,j \rangle} \hat{c}_i^\dagger \cdot \hat{c}_j \Big|_{S=0} + U \sum_i \left(\hat{n}_i^2 - \hat{n}_i \right) \tag{5.0.1}$$

with $\hat{c}_i$ the spinor of $S = 1/2$ Fermion annihilation on site i (cf. Sec. 2.4.2) and $\hat{n}_i$ the particle number operator on site i. The sum $\langle i, j \rangle$ runs over all pairs of nearest-neighbour sites i and j of the underlying lattice and we have already made sure to write it in a $\mathrm{SU}(2)_{\mathrm{Spin}}$-symmetric form. The Hubbard model has a close sibling in the t-J model which precludes double-occupation and models the effective super-exchange interaction[107] of the Hubbard Hamiltonian by an antiferromagnetic spin-spin interaction term between nearest neighbours:

$$\hat{H} = t \sum_{\langle i,j \rangle} \hat{c}_i^\dagger \cdot \hat{c}_j \Big|_{S=0} + J \sum_{\langle i,j \rangle} \left(\hat{s}_i \cdot \hat{s}_j |_{S=0} - \frac{1}{4} \hat{n}_i \hat{n}_j \right) \quad . \tag{5.0.2}$$

By precluding double occupation, this model works directly in the strong-coupling limit of large U and it is generally assumed that the physical properties of the Hubbard and t-J models close to half-filling (hole-doped) $n \lesssim 1$ are nearly identical and both are "equally valid" candidate models to describe the cuprates. The t-J model is sometimes assumed to be easier to treat numerically due to the smaller local Hilbert space of dimension $d = 3$ instead of $d = 4$.

The continuing interest in the Hubbard model lies in its property as being the arguably simplest model containing correlated electrons (at $U \neq 0$) at arbitrary filling factors. As such, the use of the model is not limited to theoretical model condensed matter physics but extends to quantum chemistry and material science, where instead of an infinite lattice with a certain periodicity, very much finite systems are studied with variants of the Hubbard Hamiltonian describing individual orbitals instead of abstract sites of a model Hamiltonian. While there are plenty of exciting applications[85,108–111] in those areas, here we will concentrate on first motivating the use of Hamiltonian (5.0.1) to model solid state systems, review the published literature studying this model Hamiltonian and finally provide some new results on ground-state properties of this Hamiltonian.

5.1 Motivating the Hamiltonian Eq. (5.0.1)

Per se, the study of abstract Hamiltonians is certainly justified and worthwhile. In the present case, however, we have the added benefit that this Hamiltonian is believed to model the physics of correlated electrons on solid state lattices relatively well. Additionally, the physics of novel superconductors are mostly dominated by two-dimensional, weakly-coupled layers of square, periodic lattices.[112,113] This, then, is the primary motivation for the ongoing study of the Hubbard Hamiltonian in particular for two-dimensional systems.

The novel superconductors under discussion were found successively and at least somewhat accidentally following the discovery by Georg Bednorz et. al. in 1986 that compounds $\mathrm{Ba}_x\mathrm{La}_{5-x}\mathrm{Cu}_5\mathrm{O}_{5(3-\mathrm{y})}$ with $x = \{1, 0.75\}$, $y > 0$ become superconducting[114] at $T_c \approx 30\,\mathrm{K}$. Already in 1987, Maw-Kuen Wu et. al. reported[115] a transition temperature of $T_c \approx 93\,\mathrm{K}$ for the compound $(\mathrm{Y}_{1-x}\mathrm{Ba}_x)_2\mathrm{CuO}_4$. Since then, numerous other compounds containing CuO_4 (hence the name *cuprates*) have been found. While a few other classes of high-temperature superconductors (such as iron pnictides[116–118] and high-pressure $\mathrm{H}_2\mathrm{S}$[119]) have been found, the cuprates known since 1986 are entirely sufficient motivation for us.

These cuprates all share the same basic structure of weakly-coupled two-dimensional planes of CuO_2 with the additional elements (such as yttrium and barium) and oxygen ions placed between those planes. The CuO_2 planes then form square rectangular lattices with Cu ions sitting on the vertices of the lattices and O ions on the edges.[112] Electron hopping is then possible between the copper $3d_{x^2-y^2}$ and oxygen $2p_x$ and $2p_y$ orbitals (with the p_z-orbitals oriented out-of-plane). Without additional doping, i.e. for $\mathrm{La}_2\mathrm{CuO}_4$, there is one hole per copper Cu^+ ion,[113] resulting in a half-filled $3d_{x^2-y^2}$ shell and filled $2p_x$ and $2p_y$ orbitals. It is the $3d_{x^2-y^2}$ orbital which imprints its $x-y$ antisymmetry onto the superconducting state[120] and gives it the name *d-wave* superconductor (cf. Ref. [121] for an extensive review).

Additional doping in the atomic limit of small hopping between neighbouring copper and oxygen sites will then result in holes (or electrons) localised to either copper or oxygen ions. The former case can be treated easily by neglecting the oxygen "sites" and arriving at a single-band Hubbard model. In the latter case, Zhang et. al. argue[113] that due to the phase coherence of $2p_{x,y}$ and $3d_{x^2-y^2}$ orbitals, the effective set of low-energy states on each vertex is equally given by a single electronic site, hence mapping the full problem also to an effective single-band Hubbard model.

It is by now generally agreed that the ingredients of superconductivity are not fully captured by the nearest-neighbour Hubbard Hamiltonian (5.0.1) alone but instead one expects to at least need next-nearest-neighbour hopping terms to arrive at a superconducting state. Nevertheless, the complete solution of the "simple" Hamiltonian (5.0.1) is still outstanding and would likely add significant new insights into the physical properties of these materials.[122]

In turn, the two-dimensional Hubbard Hamiltonian has also become a benchmarking model to test new algorithmic approaches by combining difficulties for all existing methods: away from half-filling, the sign problem makes quantum Monte-Carlo simulations difficult while two spatial dimensions are traditionally difficult to capture with tensor network approaches. At the same time, the problem is too strongly correlated and phases are in too fierce competition for the density-matrix embedding theory[110,111] (DMET) to provide notable new insights[122] and the coordinaten number is too small for the dynamical mean-field theory (DMFT),[123,124] exact in the limit of lattices with infinite coordination number,[123] to unfold its full potential.

5.2 Existing Literature on the Hubbard Hamiltonian

Already in 1963, Elliott Lieb and Fa-Yueh Wu solved the one-dimensional case of the Hubbard Hamiltonian in the thermodynamic limit and close to half-filling exactly.[125] According to their analysis, at any non-zero U, the ground state of the Hamiltonian is an insulator while it is conducting in the case of no interactions. Unfortunately, an equally elegant result does not exist for the two-dimensional case. However, both the Hubbard model and the t-J model have been studied extensively.[122,125–161] These studies have largely resulted in four not internally consistent results. In the following, we will briefly categorise the different results.

First, a number of studies[134–139] have mostly restricted themselves to the case of half-filling due to the strong sign problem[134] in the "physically interesting" region of approx. 7/8-ths filling. While there is no perfect consensus on whether the metal-insulator transition at half-filling occurs at $U = 0$ or $U > 0$, all of these studies agree that at half-filling and sufficiently large $U \gtrsim 4$, the ground state is a Mott insulator with strong antiferromagnetic correlations and very small double occupation (the t-J model ground state of course has double occupation equal to zero).

Second, analytical arguments and some numerical evidence point towards a region of macroscopic phase separation[140–150] at least for sufficiently large U or sufficiently large J/t. The existence of this phase separation is well-supported in the t-J model but not entirely clear for the Hubbard model. Even then, the boundaries of the phase-separated region are unclear. However, nearly all of these studies have found some form of d-wave pairing potentially leading to a superfluid or superconducting state.

Third, some studies largely at $U \lesssim 4$ have found some form of uniform ground state away from half-filling. Using a resonanting valence bond ansatz, Ref. [151] finds a d-wave superconductor away from half-filling and a state competitive with antiferromagnetic order at half-filling. In comparison, Ref. [126] claims a paramagnetic metallic state as the ground state away from half-filling while a much more recent study[131] only looks at the region $n < 0.7$ and small $U \lesssim 4$ where they find superfluid states with either d_{xy} (at small $n < 0.6$) or $d_{x^2-y^2}$ (at $0.6 < n < 0.7$) pairing for small U and an additional phase

with p-wave symmetry at very small n for arbitrary U.

Finally, a large number of studies[152–161] consistently find a striped ground state for any filling $n \neq 1$ at most interaction strengths. It should be stressed that the majority of these studies are DMRG[152–156] or iPEPS[158,159] studies. This is problematic for two reasons: First, one may assume that open boundary conditions in DMRG favour some sort of striped ground state on finite systems: even in a uniform state, Friedel oscillations would still result in stripe-like behaviour and the studied systems are usually far too small to reliably extrapolate to the thermodynamic limit. Second, the unit cell configuration supplied to iPEPS precludes some states entirely.[158] There is hence an argument to be made that while the stripes found using DMRG and iPEPS provide a strong argument for a striped ground state or at least a striped low-energy state in close competition with the ground state, the study of larger systems, ideally directly in the thermodynamic limit using some as of now unknown method, is still necessary. A large and recent study[161] attempts to answer this question at the point $U = 8$, $n = 0.875$ using numerous methods and does find a consistently striped ground state but has not been evaluated extensively yet.

While the exact phase boundaries are not yet determined conclusively, nearly all studies so far agree on three primary results. Most conclusively, there is a Mott insulator with strong antiferromagnetic correlations at $n = 1$. Further, some sort of d-wave pairing generally occurs at $n < 1$ with $d_{x^2-y^2}$ pairing more often found close to $n = 1$ and d_{xy} pairing found at $n < 0.6$. Additionally, some kind of phase separation is often found. At large J/t in the t-J model, this is probably some sort of macroscopic phase separation with large regions with and without holes, while there is strong evidence for "mesoscopic", i.e. striped, phase separation also in the Hubbard model. In the following, we will attempt to argue that macroscopic phase separation also occurs at $n \approx 0.9$ in the Hubbard model for relatively modest $U \approx 4 \ldots 6$.

5.3 Hybrid Space Cylinders

Ideally, we would like to solve the two-dimensional Hubbard model directly in the thermodynamic limit of an infinite square lattice. Unfortunately, if we also want to use MPS (and not, say, PEPS), we have to map the two-dimensional problem onto a one-dimensional chain first. Even when using iMPS (or VUMPS), we are hence restricted to a finite length in one direction. However, it can easily be argued[162] that such a system, infinite in one dimension and finite in the other, is essentially a one-dimensional system and that a finite system with a "sane" aspect ratio of width to length is a much better description of an infinite two-dimensional system.

In MPS applications, it is then customary[25,51] to consider a finite system with cylindrical boundary conditions, i.e. open boundary conditions along one axis and periodic boundary conditions along the other (cf. Fig. 5.1). Note that due to the mapping from two to one

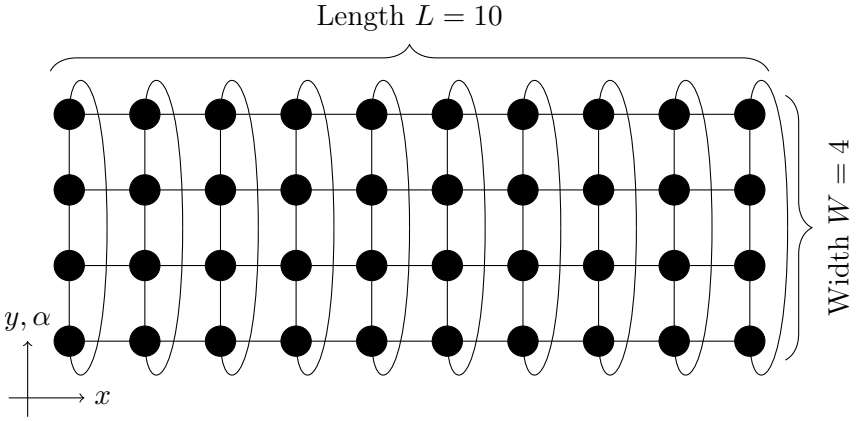


Figure 5.1: The lattice configuration used for our calculations on the Hubbard model. Sites are denoted by black dots while the existing nearest-neighbour interactions are shown via black lines. Note the periodic width-wise boundary conditions. Real-space coordinates x and y label length- and width-wise coordinates while Greek labels $\alpha, \beta, \ldots$ will be used to label hybrid-space legs.

dimensions, two vertical slices of this cylinder are only connected by a single MPS bond. If we assume the area law to hold even in two dimensions, the entanglement transported over such a bond then grows linearly with cylinder width W which unfortunately implies an exponential growth of the bond dimension m with cylinder width W. Therefore, only relatively narrow cylinders of widths $W \lesssim 8$ can be considered.[i] This insufficiency is then partly compensated by selecting periodic boundary conditions along this axis. While those also increase the amount of entanglement in the system, it is generally assumed that the system widths required to obtain valid results with open boundary conditions are much larger (and hence exponentially more expensive) than those needed with periodic boundary conditions (which at most increase m quadratically[21]). On the other hand, increasing the system length L is much easier and typically leads to only a near-linear increase in computational effort, which allows for a decent finite-size analysis in L even with open boundary conditions.[25]

The necessity of the width-wise periodic boundary conditions however also allows employing a Fourier transformation into momentum space along this axis. Under normal circumstances, this would not be beneficial, as a long system with open boundary conditions is typically much nicer behaved than a shorter one with periodic boundary conditions. Furthermore, normally the Fourier transformation introduces unwanted long-range interactions and entanglement. However, both objections are invalid here due to the small cylinder width and the already-existing long-range interactions from the 2D-1D mapping. Additionally, one gains an extra quantum number in the form of the conserved and now-accessible pseudomomentum k in the group $\mathbb{Z}_W$. This additional

[i] If the local dimension is smaller, as for example in spin systems, the maximal width can of course be increased. The same holds if the system contains very little entanglement in the first place.

quantum number is the main benefit of the method: in real-space, it is not possible to assign good pseudomomentum quantum numbers to the local states on each site and hence also not possible to exploit that symmetry within a symmetry-protected tensor network. In momentum space, this is possible and we can use the symmetry to further reduce the effective sizes of our tensors. The effectiveness of this approach has been tested extensively[11,12] and found to be generally satisfactory, typically allowing for much larger bond dimensions and less memory usage. In the following, we will briefly derive the Hubbard Hamiltonian on the hybrid space lattice. Let us split the Hamiltonian (5.0.1) into three parts corresponding to width-wise and length-wise hopping and the on-site interaction U:

$$\hat{H} = t\hat{H}_W + t\hat{H}_L + U\hat{H}_U \quad . \tag{5.3.1}$$

The individual parts are then, with x and y used for real-space length- and width-wise coordinates of a cylindrical lattice with length L and circumference/width W:

$$\hat{H}_W = -\sum_{x=1}^{L}\sum_{y=1}^{W} \hat{c}^\dagger_{x,y} \cdot c_{x,(y+1)} + \hat{c}^\dagger_{x,(y+1)} \cdot c_{x,y} \tag{5.3.2}$$

$$\hat{H}_L = -\sum_{x=1}^{L-1}\sum_{y=1}^{W} \hat{c}^\dagger_{x,y} \cdot \hat{c}_{x+1,y} + \hat{c}^\dagger_{x+1,y} \cdot \hat{c}_{x,y} \tag{5.3.3}$$

$$\hat{H}_U = \frac{1}{2}\sum_{x=1}^{L}\sum_{y=1}^{W} \left(\hat{c}^\dagger_{x,y} \cdot \hat{c}_{x,y} \times \hat{c}^\dagger_{x,y} \cdot \hat{c}_{x,y} - \hat{c}^\dagger_{x,y} \cdot \hat{c}_{x,y}\right) \tag{5.3.4}$$

Here, $\hat{A} \times \hat{B}$ is the standard operator product of two MPOs while $\hat{A}^\dagger \cdot \hat{B}$ is the dot product. The dot product is always taken in the $S = 0$ sector in the following. We can then implement the Fourier transformation along the y-direction into momentum space as

$$\hat{c}^\dagger_{x,y} = \frac{1}{\sqrt{W}}\sum_{\alpha=1}^{W} e^{-\frac{2\pi i}{W}\alpha y}\hat{c}^\dagger_{x,\alpha} \tag{5.3.5}$$

$$\hat{c}_{x,y} = \frac{1}{\sqrt{W}}\sum_{\alpha=1}^{W} e^{\frac{2\pi i}{W}\alpha y}\hat{c}_{x,\alpha} \quad . \tag{5.3.6}$$

We will use Greek letters for momentum-space labels. Note the different sign in the exponent as well as the imaginary unit i: Even if t and U are entirely real, we unfortunately now need complex numbers in our calculation to properly represent the Hamiltonian and its eigenstates. Plugging this transformation into Eqs. (5.3.2) through (5.3.4), we arrive at the expressions

$$\hat{H}'_W = -\sum_{x=1}^{L}\sum_{\alpha=1}^{W} 2\cos\left(2\pi\frac{\alpha}{W}\right)\hat{c}^\dagger_{x,\alpha} \cdot \hat{c}_{x,\alpha} \tag{5.3.7}$$

$$\hat{H}'_L = -\sum_{x=1}^{L-1}\sum_{\alpha=1}^{W}\left(\hat{c}^\dagger_{x,\alpha}\cdot\hat{c}_{x+1,\alpha} + \hat{c}^\dagger_{x,\alpha}\cdot\hat{c}_{x+1,\alpha}\right) \tag{5.3.8}$$

$$\hat{H}'_U = \frac{1}{2}\sum_{x=1}^{L}\sum_{\alpha=1}^{W}\left(\left[\sum_{\beta\gamma=1}^{W}\frac{1}{W}\hat{c}^\dagger_{x,\alpha}\cdot\hat{c}_{x,\beta}\times\hat{c}^\dagger_{x,\gamma}\cdot\hat{c}_{x,\alpha-\beta+\gamma}\right] - \hat{c}^\dagger_{x,\alpha}\cdot\hat{c}_{x,\alpha}\right) \tag{5.3.9}$$

for the Hamiltonian. While $\hat{H}'_U$ is now long-range, this "range" is limited to the width of the cylinder W and hence relatively small. The bond dimension of the MPO only grows linearly in cylinder width while the maximal block size of the contained dense blocks even stays constant due to the large number of available symmetries.[1] When implementing this Hamiltonian, one should take care to properly handle the additional $\mathbb{Z}_W$ pseudomomentum symmetry which in particular results in a non-homogeneous lattice (cf. Sec. 2.4.2): particles on sites (x, α) carry momentum α, hence the creation operator $\hat{c}^\dagger_{x,\alpha}$ has to transform as $S = 1/2$, $n = 1$ and $k = \alpha$.

5.4 Results with DMRG3S

In the following, we will first compare the obtained ground-state energies from our method (DMRG3S) with those obtained by two key references[12,122] which list their obtained energy values at least for some system sizes. Afterwards, we will concentrate on our new results for different filling factors n, which suggest the presence of a phase transition at $n \approx 0.9$.

5.4.1 Convergence and Energy Comparison

As a first check for the validity of our results, it is useful to compare the found ground-state energies with those found by other studies of the 2D Hubbard model. Unfortunately, relatively few such studies have given explicit energies, furthermore, it is difficult to compare the values of methods which work directly in the thermodynamic limit to the values given by DMRG, valid only for a certain cylinder circumference. While one may attempt an extrapolation in the cylinder width, due to the small number of data points (typically $W = 4, 6$), this is rather unreliably. In the following comparison, we will primarily refer to the data from Ref. [12, 122], concentrating on the "hard" case of $n = 0.875$ for various values of the interaction $U = 4, 6, 8$.

The first hurdle towards a useful comparison is the lack of a meaningful truncation error in single-site DMRG. Since the magnitude of the expansion term (3.4.4) is largely arbitrary, no useful information can be extracted from the subsequent truncation. In comparison, two-site DMRG readily provides such a truncation error which allows for a very useful extrapolation of the energy and other observables towards zero truncation error. Typically, values given in the literature are the extrapolated values corresponding to zero truncation error. Ehlers et al.[12] give an excellent example of such a possible

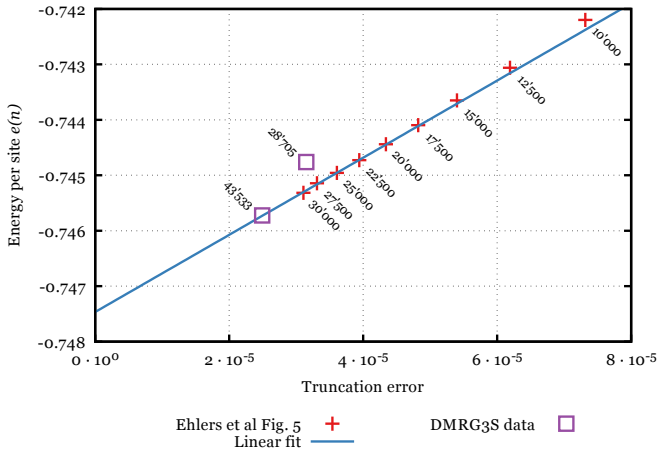


Figure 5.2: Comparison of bare data from DMRG3S with 2DMRG data by Ehlers et al.[12] in the process of extrapolation towards zero truncation error. Placement of the two additional points along the x-axis is estimated given their bond dimensions.

extrapolation. In Fig. 5.2, we have extracted the unextrapolated data from Fig. 5 in Ref. [12] for the case of $L = 16$, $W = 6$, $U = 8$ and $n = 0.875$ and inserted two data points from our calculations, translating the number of $\mathrm{SU}(2)_{\mathrm{Spin}}$-symmetric states $m_S = 10'000$ and $m_S = 15'000$ to the equivalent number of $\mathrm{U}(1)_{S^z}$-symmetric states. As can be seen, while the value at $m_S = 10'000$ is not well converged, the value at $m_S = 15'000$ fits very well with its expected magnitude.

Results for $U = 8$, $n = 0.875$ are most numerous, as we did many calculations for different system sizes at this particular point. The results are compared with the two other recent papers in Fig. 5.3. LeBlanc et al.[122] only give values for $L = \infty$ shown on the y-axis accordingly. Ehlers et al.[12] also provide data for finite systems, but extrapolated to zero truncation error. At $W = 4$, it is easy to match this additional advantage using a higher bond dimension (the red and green crosses hence overlap), at $W = 6$, this extrapolation provides a considerable advantage resulting in lower energies (compare red and green circles). No data was published elsewhere for $W = 8$ (double crosses) and given the behaviour found in our data, this is not surprising: at this width, a very large bond dimension would be required and even with $30'000$ $\mathrm{SU}(2)_{\mathrm{Spin}}$-symmetric states, equivalent to approx. $85'000 - 95'000$ $\mathrm{U}(1)_{S^z}$-symmetric states, we cannot even capture the (expected) lower energy for longer cylinders, as the additional entanglement and error keeps the energy per site approximately constant. Nevertheless, we *do* find an energy per site quite compatible with the previous results for $W = 4$ and $W = 6$, even though evaluation of occupation numbers (e.g. cf. Fig. 5.4) reveals a wildly asymmetric state which is unlikely to be the true ground state of the system.

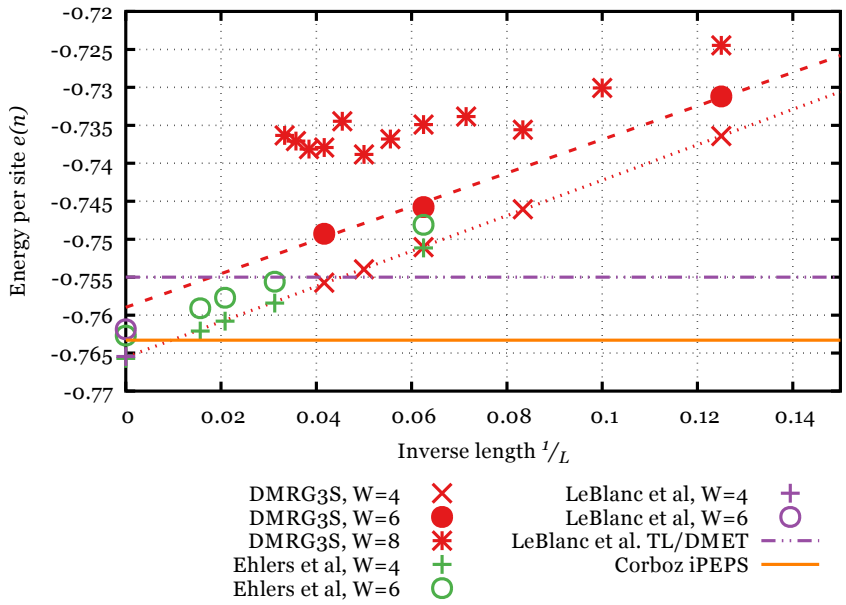


Figure 5.3: Comparison of energies from different DMRG studies at $U = 8$, $n = 0.875$ on cylinders of widths $W = 4$ (crosses), $W = 6$ (circles) and $W = 8$ (double crosses). Dashed lines are linear fits to the DMRG3S data at $W = 4$ and $W = 6$ respectively. Green data points are taken from Ref. [12], Tab. V, violet points are extracted from Ref. [122] Fig. 6 with the 2D thermodynamic limit DMET value as well as the iPEPS 2D thermodynamic limit value[163] included for comparison.

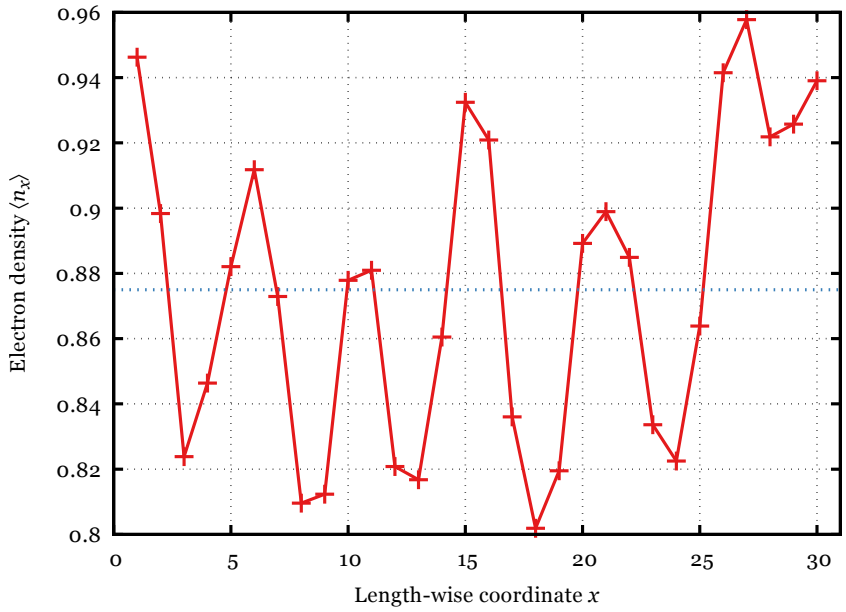


Figure 5.4: Density profile found by DMRG3S for a cylinder of length $L = 30$ with circumference $W = 8$ at $U = 8$, $n = 0.875$. The dashed line denotes the expected average filling. Contrary to our expectations of a ground state, the profile is not at all symmetric, e.g. around its centre $x = 16.5$.

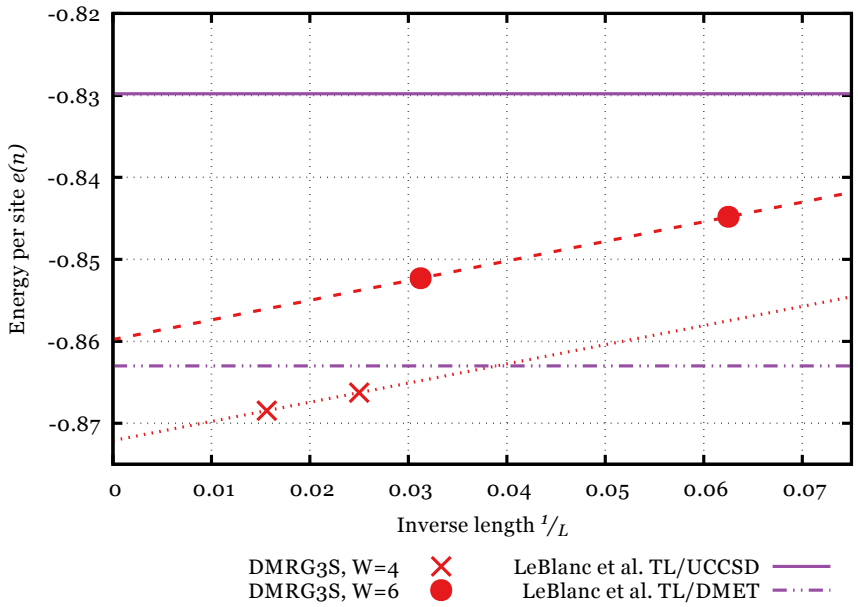


Figure 5.5: Comparison of energies from our DMRG3S method at $U = 6$ with upper and lower estimates in the 2D thermodynamic limit from DMET and UCCSD calculations in Ref. [122] Tab. V. Red dashed lines are linear extrapolations of the two DMRG3S data points each to the $L = \infty$ case.

At $U = 6$, $n = 0.875$, we did relatively few calculations at different system sizes as we concentrated on checking different filling factors n. At the same time, our first reference[12] does not consider the case $U = 6$ at all while our second reference[122] only gives data from DMET and various Monte Carlo calculations but not from DMRG calculations. Our data is plotted in Fig. 5.5. For comparison, we show the two extremal estimates from Ref. [122] from DMET and UCSSD (unconstrainted coupled-cluster theory with single and double excitations) calculations. While our data set is limited, it does not diverge too far from e.g. the DMET results. Assuming a similar behaviour of the truncation extrapolation as for $U = 8$ for $W = 6$ here, which would lower the extrapolated energy for $W = 6$ by a few percent and taking into account the extrapolation into the TL with $W = \infty$, all differences in energy to DMET can be explained.

The case of $U = 4$, $n = 0.875$ is treated in both Ref. [12] and [122]. Our own data on the square lattice is again limited, as we were more interested in results for different values for n, but we also calculated some results on a diagonal lattice.[25,122] All results are combined in Fig. 5.6. Ref. [12] and Ref. [122] agree nearly perfectly on the value at $W = 6$, $L = \infty$, while our data for $W = 4$ agrees perfectly with the data from Ref. [12], the corresponding symbols are placed on top of each other. At $W = 6$, DMRG3S is again restricted due to the lack of a meaningful truncation error, leading to systematically higher energies. The diagonal lattices tested here seem to converge faster with cylinder

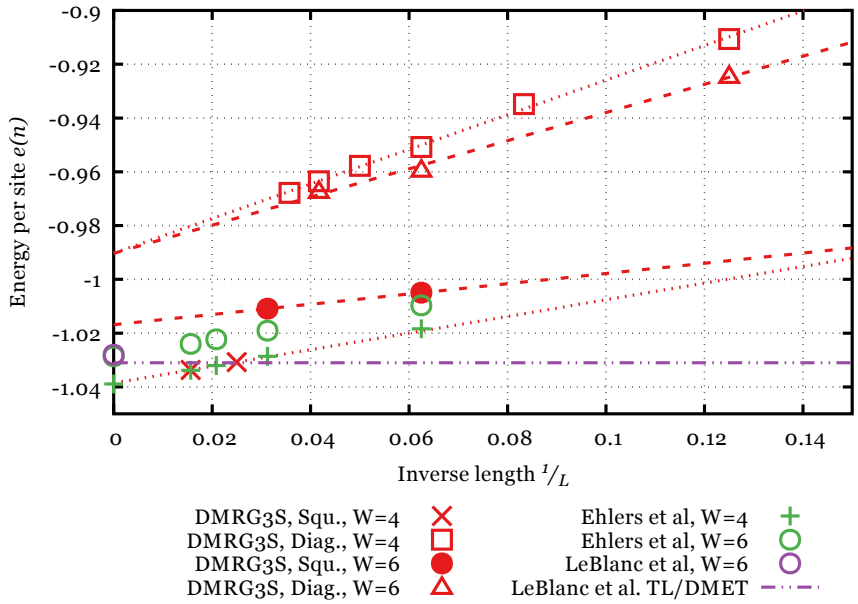


Figure 5.6: Comparison of energies between different DMRG methods at $U = 4$ on cylinders of width 4 (crosses), width 6 (circles), diagonal-width 4 (squares) and diagonal-width 6 (triangles). Red data is from our DMRG3S calculation, green points are from Ehlers et al.[12] and the violet point on the y-axis is from LeBlanc et al.[122]. For comparison, we also included the DMET value from Ref. [122] directly in the 2D thermodynamic limit.

width: For $W = 4$ and $W = 6$, the extrapolations to $L = \infty$ agree nearly perfectly. However, both the raw finite-size and the extrapolated energies are much higher than those obtained from square lattices and also higher than e.g. the DMET comparison value. This may be due to a different ground state being preferred by DMRG on such lattices due to a simpler entanglement structure.

Some remarks regarding this comparison are in order. First, at small cylinder widths, it is obviously possible to reproduce previous results up to a relatively high precision even without extrapolation in the truncation error. This is possible here primarily due to a larger effective bond dimension from using $\mathrm{SU}(2)_{\mathrm{Spin}}$-symmetric states and secondarily due to the faster convergence of DMRG3S with sweeps and, in particular, CPU time.[2]

Second, at larger cylinder widths, the additional gain in energy from the extrapolation is much larger than the gain from using $\mathrm{SU}(2)_{\mathrm{Spin}}$-symmetric states. Assuming a critical ground state (and the observed entanglement spectra do suggest at most a power-law decay of singular values[5]), this is not surprising: unless the additional factor of two to three in the bond dimension from the non-abelian symmetries is sufficient to fully capture the finite-size system, it will only provide a minor correction: truncating after (say) $10'000$ or $30'000$ states on a very slowly decaying spectrum makes a noticeable difference in the truncation error, but both errors will be far away from zero (cf. Fig. 5.2). This problem

is made worse by the fact that the bond dimension itself is not useful as an extrapolation parameter. This has also been realised in other tensor network applications.[163] As such, the only course of action should be to follow up with one or two sweeps of 2DMRG after each stage of DMRG3S has been completed to (relative) convergence: this way, one exploits the faster convergence of DMRG3S while also obtaining a meaningful truncation error. One could even consider restricting the 2DMRG update to e.g. the centre of the chain or a small central part of the system. However, it should be kept in mind that the extrapolation in the truncation error, while well-known and widely used, is *not* a variational extrapolation, i.e. the resulting energy may very well be lower than the true ground-state energy of the finite-size system.

Third, it is still not possible to get results for cylinders of width $W = 8$ in the Hubbard model with DMRG. Other methods (e.g. DMET or iPEPS) do provide results directly in the thermodynamic limit such that no extrapolation in the cylinder width is necessary and while for narrow cylinders, nearly thirty years of continuous improvement have kept DMRG competitive much longer than one would naively expect, it is questionable whether this state will continue into the future.

Fourth and finally, it should be noted that the results obtained here were calculated *without* so-called pinning fields used to stabilise a preferred stripe order elsewhere.[12,122,161] This suggests that such pinning fields are not entirely necessary when calculations are done at sufficiently large bond dimensions.

5.4.2 Hints of Phase Coexistence

Consider the energy per site e as a function of the filling factor n and recall the definition of a convex function $f : A \to B$,

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y) \quad \forall x, y \in A \vee \forall \alpha \in [0, t] \quad . \tag{5.4.1}$$

If we take f as $e(n)$ with $A \equiv [0, 1]$ and $B \equiv \mathbb{R}$, we can take points $x \equiv n - \delta$ and $y \equiv n + \delta$. In this case, the above inequality becomes

$$e(\alpha(n-\delta) + (1-\alpha)(n+\delta)) \leq \alpha e(n-\delta) + (1-\alpha) e(n+\delta) \tag{5.4.2}$$

The right-hand side of this equation is the weighted average energy of two phases at densities $n - \delta$ and $n + \delta$ respectively, while the weighted average density of those two phases is given by the argument $\alpha(n-\delta) + (1-\alpha)(n+\delta) \equiv \bar{n}$ to e on the left-hand side. If the inequality is not fulfilled, it is energetically favourable to split the single phase at density $\bar{n}$ into two co-existing phases of densities $n - \delta$ (with relative occurence α) and $n + \delta$ (with relative occurence $1 - \alpha$). On the other hand, the phase at density $\bar{n}$ is stable if it is not possible to split it into two co-existing phases. It is hence obvious that any physically-realised function $e(n)$ must be convex.

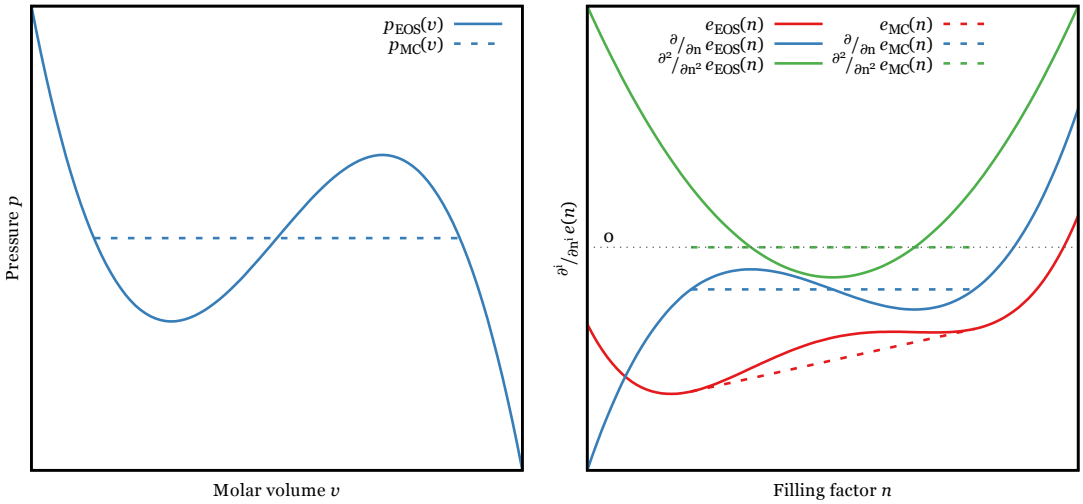


Figure 5.7: Left: Classical Maxwell construction in the pressure-volume diagram where phase coexistence leads to an area of constant pressure not predicted by the equation of state. The dashed curved is constructed by requiring the two enclosed regions to have equal area. Right: Equivalent construction in the energy-filling factor diagram. A concave region in the energy leads to a secant of constant slope being realised physically (red dashed line) which is equivalent to an equal-area-constructed constant segment in the first derivative (blue dashed line) and a clipping of the second derivative (green dashed line) to zero. The $e(n)$ and $\partial/\partial n\, e(n)$ are shifted to avoid clutter.

On the other hand, it may well be the case that an equation of state per-se violates convexity around a phase transition point, the Van-der-Waals equation[164] being a famous example. Such cases necessitate a Maxwell construction[165] which is typically constructed in the pressure p-molar volume v diagram as a line segment of constant pressure between two volumes (cf. Fig. 5.7, left). Translating the Maxwell construction into the energy-filling factor diagram[5,149,166] with correspondence $\partial/\partial n\, e(n) \sim -p$ and $n \sim v^{-1}$ then results in an area of constant slope. This constant slope "shortcuts" through a concave region of the energy function $e_{\mathrm{EOS}}(n)$ derived from the equation of state via a tanget line resulting in the physical energy function $e(n)$ and just barely restores convexity (cf. Fig. 5.7, right). This constant slope secant directly corresponds to the left-hand side of Eq. (5.4.2). Note that if the filling factor n was restricted to the region $n < n'$ with n' the first zero-point of the second derivative, it would not be possible to construct a secant below the energy curve which touches that curve at two points. Hence, even though an area $n \lesssim n'$ is unstable when considering the full range of n, this area would be stable if n was restricted to $n < n'$. For this reason, the Maxwell-constructed dashed lines extend beyond the region where $\partial^2/\partial n^2\, e(n) < 0$.

Therefore an area in $e(n)$ with constant slope might indicate phase coexistence due to a phase transition. The study of physical energy functions $e(n)$ as they result from DMRG, Monte-Carlo[149,166] calculations or analytical arguments[141] can thereby hint at a region of

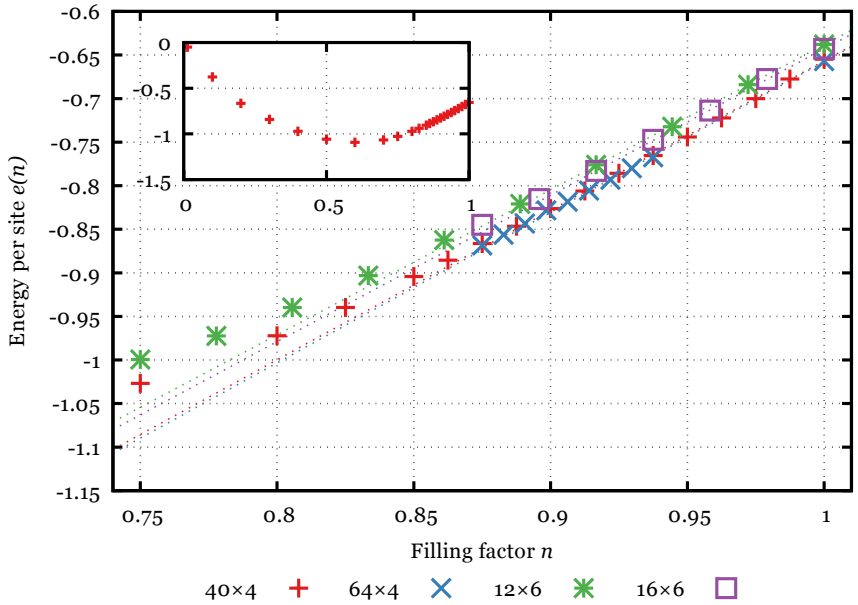


Figure 5.8: Energies per site $e(n)$ at $U = 6$ calculated with DMRG3S at various filling factors for different lattice sizes. The inset shows the overall convex shape of the energy while the dotted lines give linear fits to the data in the range $0.9 \leq n \leq 1$ with sum of residuals $\lesssim 10^{-5}$. All system sizes lead to the same qualitative results with small quantitative differences due to finite-size effects. The linear segment at $0.9 < n < 1$ is relatively well visible, especially compared to the overall convex shape of the curve.

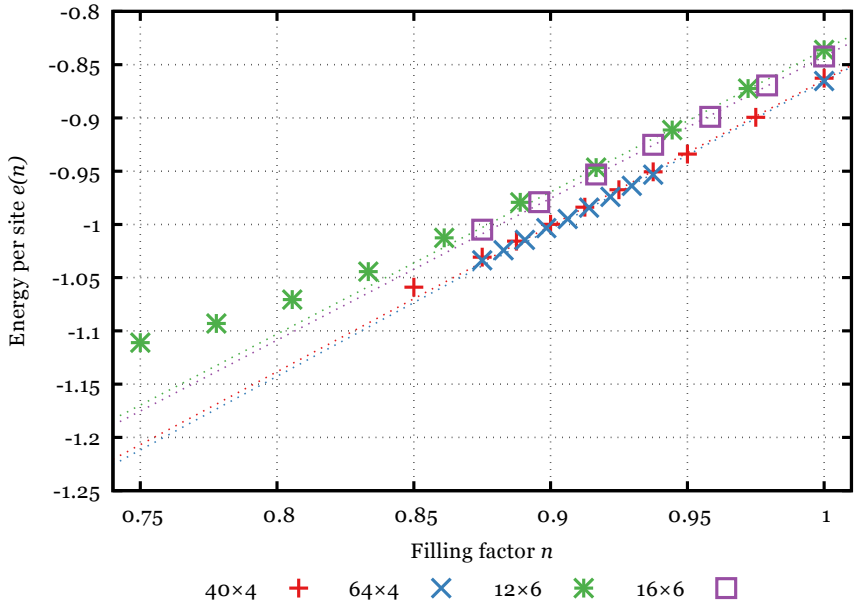


Figure 5.9: Energies per site $e(n)$ at $U = 4$ calculated with DMRG3S at various filling factors for different lattice sizes. The dotted lines give linear fits to the data in the range $0.9 \leq n \leq 1$ with sums of residuals $< 10^{-5}$. The results are qualitatively comparable to those in Fig. 5.8 for $U = 6$.

phase coexistence and hence also phase transition. In the following, some data points of $e(n)$ in addition to the extensive analysis already carried out in Ref. [5] will be presented together with a brief summary of auxiliary arguments supporting the phase transition.

Our study of the phase coexistence region consists of DMRG calculations on lattices of sizes 40×4, 64×4, 12×6 and 16×6. Extensive sweeping over the possible filling factors n was done for the first case, in the other three, we limited ourselves to the physically interesting region $n \approx 0.92$. The values of U are 4 and 6 with little qualitative difference found. Fig. 5.8 gives our results for $U = 6$, Fig. 5.9 those for $U = 4$.[ii] Linear fits have been attempted in the range $0.9 \leq n \leq 1$ with sums of residuals $\lesssim 10^{-5}$ for $U = 6$ and $< 10^{-5}$ for $U = 4$ for all fits. The Figs. 5.8 and 5.9 already show a remarkably linear behaviour in this region with the data noticeably diverging from the linear fit for $n \lesssim 0.85$, see in particular the inset in Fig. 5.8 for the behaviour of $e(n)$ over the full range $0 \leq n \leq 1$. The data is in good agremeent with Fig. 2 of Ref. [149]. Contrary to Fig. 3 of Ref. [149], our data however does not exhibit a minimum in the energy per hole as a function of hole density $e_h(h)$ around a critical hole density $h_c \approx 0.9$. Instead, we find either a very noisy curve for the larger systems or a monotonically increasing $e_h(h)$ for the 40×4 system (not shown). That is, while the raw energy per site appears to behave nicely and nearly converged, the energy per hole is still very noisy. Longer calculations, ideally on shorter and wider systems with many different boundary conditions would likely be necessary to produce more reliable data.

Evaluating the first differences to get the slope of the obtained data is highly sensitive towards the accuracy of the obtained energies and hence difficult, cf. Figs. 5.10 and 5.11. Note especially that these figures would actually suggest a partly concave energy function $e(n)$ due to the numerous slight decreases in the slope $\partial/\partial n\ e(n)$ in the region $n > 0.9$. However, when comparing the region $n > 0.9$ to the nicely-behaved regime $n \lesssim 0.85$, one may find a very noisy relative plateau in the first differences or at least a relatively strong divergence from the previous behaviour of strictly increasing differences. In particular, the 40×4 system (red crosses in Figs. 5.10 and 5.11) exhibits a narrow plateau in the region $0.9 \leq n \leq 0.9375$ and the 12×6 system (green crosses), which was tested down to $n = 0.75$, also shows a marked decrease in the growth of the first differences starting at around $n \approx 0.9$.

Interestingly, this plateau coincides relatively well with a final particle number distribution along the long axis of the cylinder which is much more asymmetric than at $n < 0.85$ and $n > 0.9375$. Collected distributions for 40×4 are plotted in Fig. 5.12. This behaviour is not found in the 12×6 system (cf. Fig. 5.13). This may indicate that while something (such as a phase transition) makes it much harder for DMRG to build a

[ii] A single data point for a single momentum value on a fixed lattice at a specific filling factor and interaction strength requires 4-6 weeks of CPU time on 10 Broadwell-Xeon cores, scanning all possible momenta on the ring multiplies this by a factor of W and even just checking momenta $k = 0$ and $k = \pi$ by a factor of two. Due to this large amount of CPU-time spent, we had to limit most calculations to the area around $n \approx 0.925$.

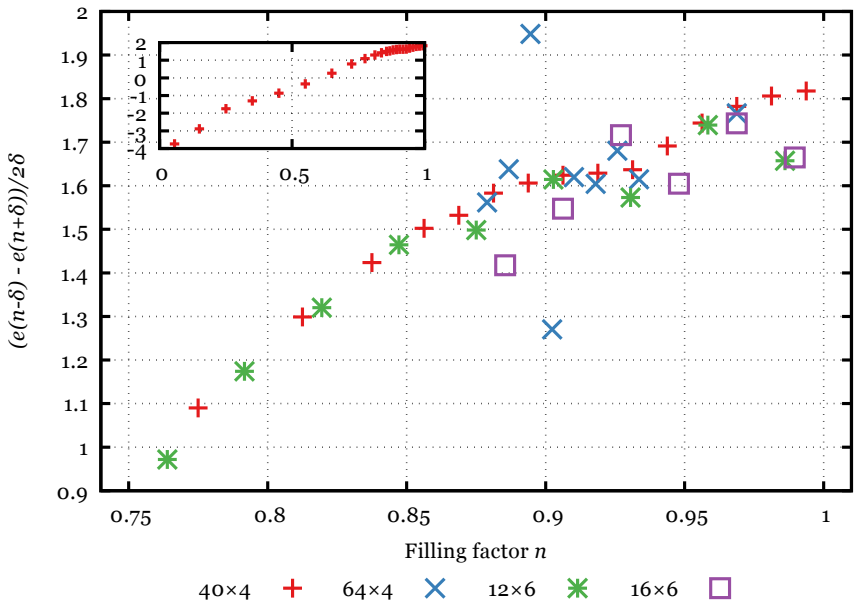


Figure 5.10: Differences in energy evaluated between consecutive points of Fig. 5.8 at $U = 6$. While the data for lattice size 40×4 is relatively smooth and may hint at a region of constant slope between $n \approx 0.9$ and $n \approx 0.94$, the data for the other systems is far too noisy to reliably extract a slope. However, it is likely consistent with a constant slope of $e(n)$. The inset shows the data for the 40×4 system over the full range of $0 \leq n \leq 1$.

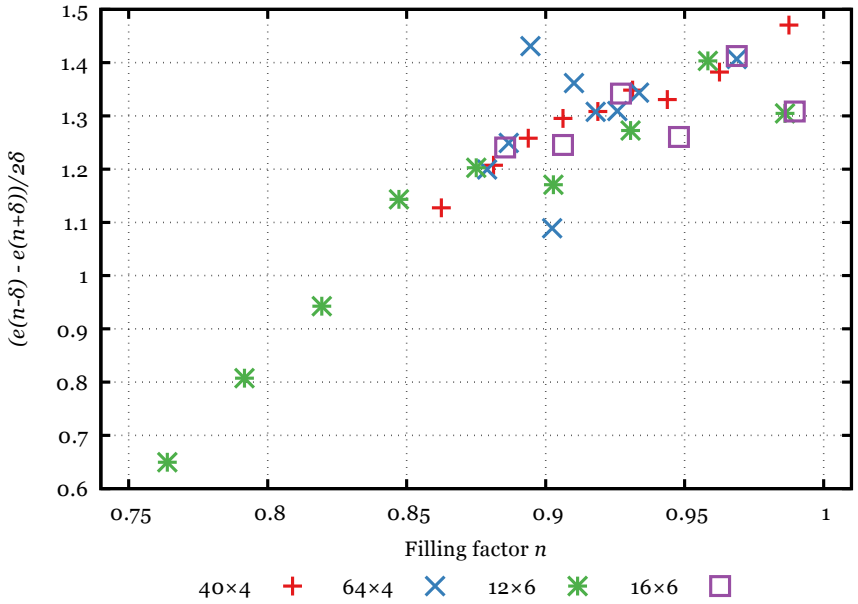


Figure 5.11: Differences in energy evaluated between consecutive points of Fig. 5.9 at $U = 4$. While the data for lattice size 40×4 is relatively smooth and may hint at a region of constant slope at $n \approx 0.9$, the data for the other systems is far too noisy to reliably extract a slope. However, it is likely consistent with a constant slope of $e(n)$.

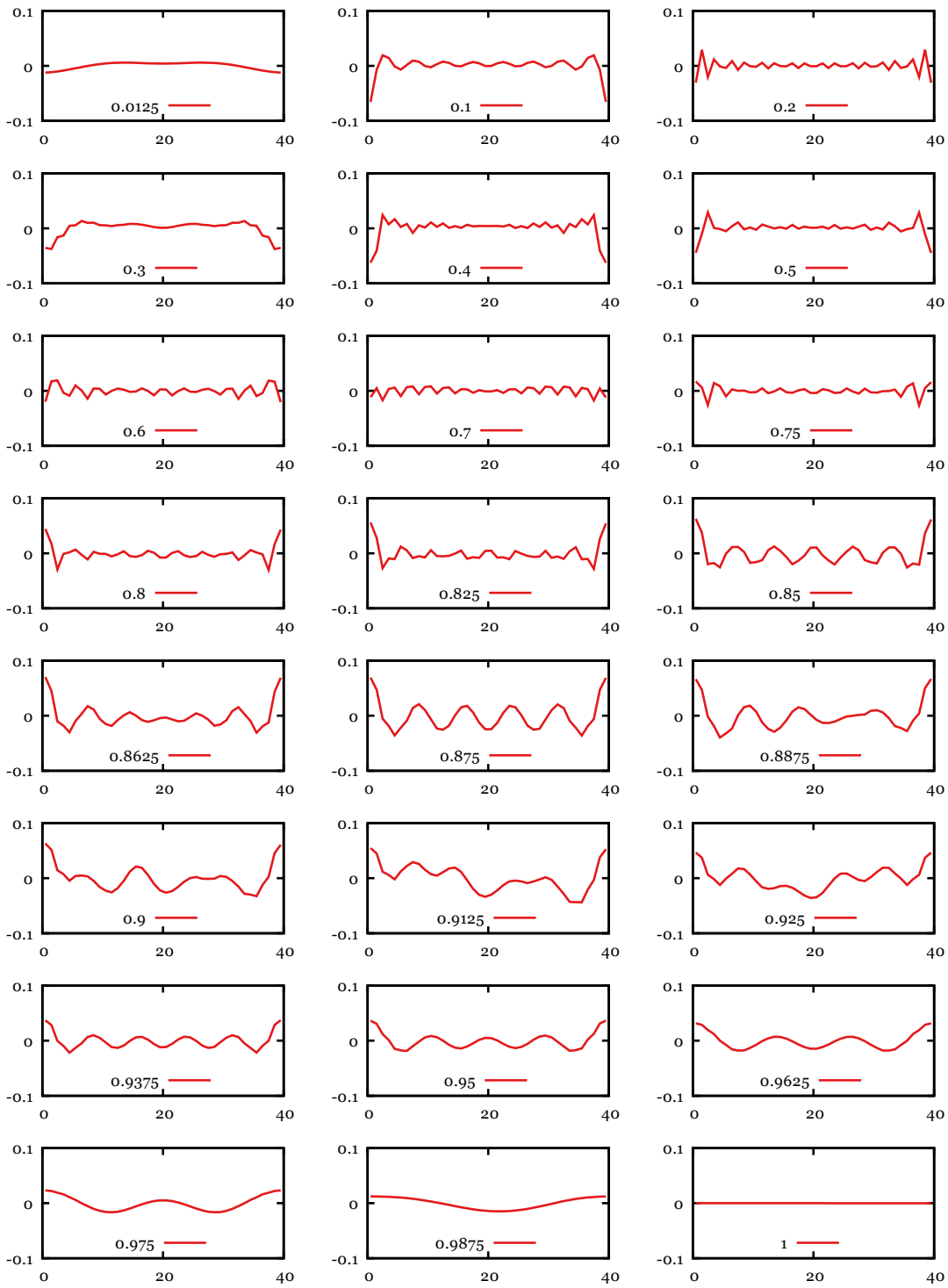


Figure 5.12: Relative occupation profiles collected for the 40×4 system at $U = 6$ along the long axis x of the cylinder. Plotted is $n_{x,y} - n$ over x (cf. Fig. 5.1). Occupation numbers on each site y of a single ring x are identical due to the hybrid-space formulation. The distributions for $n \leq 0.875$ and $n \geq 0.9375$ are very symmetric, while those in-between display large asymmetries.

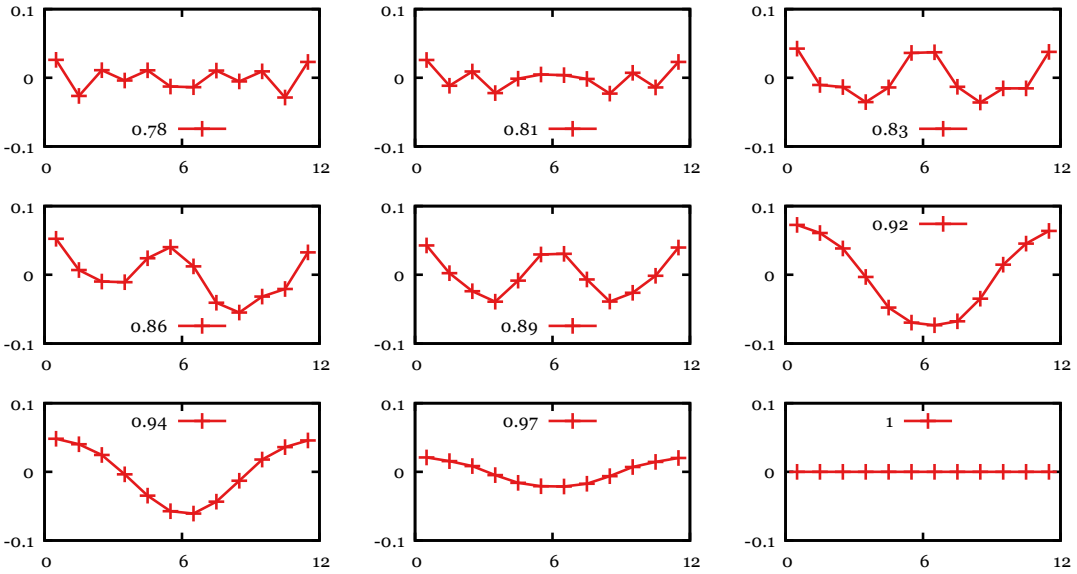


Figure 5.13: Relative occupation profiles collected for the 12×6 system at $U = 6$ along the long axis x of the cylinder. Plotted is $n_{x,y} - n$ over x (cf. Fig. 5.1). Occupation numbers on each site y of a single ring x are identical due to the hybrid-space formulation. Most of the calculated distributions are very symmetric with only a single deviation at $n = 0.86\bar{1}$.

symmetric ground-state for the long cylinder, this is much easier on the shorter cylinder of just twelve sites. Note that even a macroscopic phase transition does not imply a visible change in the density profile, since multiple ground states with different profiles may overlap to result in one very symmetric overall density profile. In this respect, while the irregularities observed around $n \approx 0.9$ for the 40×4 system may hint at a problematic situation for DMRG, they are not a required condition for phase coexistence. On the other hand, they may simply be an artefact of convergence failure, though it is unclear for what other physical reason DMRG should fail to converge at precisely those densities.

Outlook

In this section as well as the more extensive analysis of (parts of) the current data presented in Ref. [5] we have shown that at least something is happening at around $n \approx 0.9$ in the Hubbard model at medium interaction strengths $U \approx 4 \ldots 6$. This by itself is not surprising: We know that the Hubbard model at half-filling exhibits a Mott insulator with strong antiferromagnetic correlations and, at lower filling factors, likely a striped state with properties, though they have not yet been conclusively determined, at least different from those of a Mott insulator. As such, a phase transition and potential phase coexistence is not surprising but rather expected. However, our data, as well as the previous results[149,166] suggest a critical filling factor $n_c \gtrsim 0.9$, slightly above the value $n = 0.875$ usually taken to approximate cuprate physics. It may therefore be interesting

to first understand the full phase diagram of the standard Hubbard model, including the determination of the critical filling factor n_c before attempting to introduce more complications necessary to approximate the physical cuprate systems.

Such "complications" will likely include next-nearest neighbour hopping already investigated in some cases,[122] multi-band models currently only investigated with more approximate methods such as DMFT,[71,167,168] or longer-range interactions to more closely model the existing Coulomb repulsion. While tensor network methods also provide for excellent impurity solvers to be employed by embedding methods such as DMFT or DMET, it will also be useful to solve (relatively small) systems directly, as the wealth of information available from the tensor network wavefunction is unparalleled by embedding methods.

In this context, it will most likely be necessary to abandon the 2D-1D mappings currently used and focus on either full-2D networks such as iPEPS or intermediates such as tree tensor states where each ring of the cylinder acts as a branch of the tree tensor. The latter may be a reasonable choice as they reduce the number of large tensors from $L \cdot W$ to L and intra- not inter-ring entanglement is highest in our current calculations: at equal accuracy, the intra-ring bonds are approximately 40% larger than the inter-ring bonds.

Acknowledgements

The work presented in this thesis would not have been possible without the help and support from numerous people. In particular, I would like to thank

- Ulrich Schollwöck, Ian McCulloch, Alexander Wolf and Vincenzo Alba for extensive mentoring and advice, useful pointers and many helpful discussions.
- Norbert Schuch for co-examining this thesis.
- Cordula Weber for her support with administrative tasks and her very successful shielding of day-to-day tasks from the external bureaucracy.
- Ralph Simmler, Klaus Steinberger, Felix Rauscher, Daniela Aldea and Ralph Heuer for making sure that computers could compute.
- Fabian Heidrich-Meisner and Thomas Schulte-Herbrueggen for coordinating the many interesting seminars both at the Schollwöck Chair and ExQM.
- Leo Stenzel, Andreas Swoboda, Nils-Oliver Linden, Felix Lachenmaier, Moritz Hahn, Sam Mardazad and Maximilian Lechner for their close collaboration, apparent trust in my code and exciting projects.
- Fabian H. L. Essler, Anup Kumar Bera, Zi Cai, Teresa Reinhard, Uliana Mordovina, Florian Buchholz, Heiko Appel, Ilia Khait, Nicola Lanatà, Tsung-Han Lee, Yongxin Yao and Philipp Snajberk for our ongoing and successful collaborations.
- Lode Pollet, Florian Dorfner, Jonas Greitemann, Stephen Inglis, Dennis Schimmel, Lukas Weidinger, Jan Stolpp and Fabian Kugler for many discussions not only over lunch.
- The ExQM Graduate School and the Nanosystem Initiative Munich for funding during the past four years.
- Anja Metelmann and Tobias Brandes for mentoring during my Bachelor thesis and an initial introduction to the field of numerical nanophysics.
- Lena Schneiderheinze, Christoph Hubig and Christiane Messerschmidt-Hubig for their ongoing support and motivation and Sebastian Sester for the many useful comments throughout the years regarding the quality of my code.

List of Algorithms

List of Figures

List of Tables

Bibliography

[1] C. Hubig, I. P. McCulloch and U. Schollwöck. "Generic construction of efficient matrix product operators". *Phys. Rev. B* 95 (3 2017), p. 035129. DOI: 10.1103/PhysRevB.95.035129.

[2] C. Hubig, I. P. McCulloch, U. Schollwöck and F. A. Wolf. "Strictly single-site DMRG algorithm with subspace expansion". *Phys. Rev. B* 91 (15 2015), p. 155115. DOI: 10.1103/PhysRevB.91.155115.

[3] A. Swoboda, N.-O. Linden, U. Schollwöck and C. Hubig. "Time evolution for matrix-product states with the Krylov subspace method". In preparation.

[4] A. K. Bera, B. Lake, F. H. L. Essler, L. Vanderstraeten, C. Hubig, U. Schollwöck, A. T. M. N. Islam, A. Schneidewind and D. L. Quintero-Castro. "Spinon confinement in a quasi-one-dimensional anisotropic Heisenberg magnet". *Phys. Rev. B* 96 (5 2017), p. 054423. DOI: 10.1103/PhysRevB.96.054423.

[5] L. Stenzel. "A DMRG study of the Fermi-Hubbard model in hybrid space". Master thesis, LMU Munich. 2017.

[6] F. Lachenmaier. "Parallelized Time Evolution on the Heisenberg Spin Chain with Matrix Product States". Bachelor thesis, TU Munich. 2016.

[7] M. Hahn. "Contractor Renormalization und das Kagomé Gitter". Master thesis, LMU Munich. 2017.

[8] M. B. Hastings. "An area law for one-dimensional quantum systems". *J. Stat. Mech.* 2007.08 (2007), P08024. DOI: 10.1088/1742-5468/2007/08/P08024.

[9] C. Holzhey, F. Larsen and F. Wilczek. "Geometric and renormalized entropy in conformal field theory". *Nucl. Phys. B* 424.3 (1994), pp. 443–467. DOI: 10.1016/0550-3213(94)90402-2.

[10] G. Ehlers, J. Sólyom, Ö. Legeza and R. M. Noack. "Entanglement structure of the Hubbard model in momentum space". *Phys. Rev. B* 92 (23 2015), p. 235116. DOI: 10.1103/PhysRevB.92.235116.

[11] J. Motruk, M. P. Zaletel, R. S. K. Mong and F. Pollmann. "Density matrix renormalization group on a cylinder in mixed real and momentum space". *Phys. Rev. B* 93 (15 2016), p. 155139. DOI: 10.1103/PhysRevB.93.155139.

[12] G. Ehlers, S. R. White and R. M. Noack. "Hybrid-space density matrix renormalization group study of the doped two-dimensional Hubbard model". *Phys. Rev. B* 95 (12 2017), p. 125125. DOI: 10.1103/PhysRevB.95.125125.

[13] S. R. White. "Density matrix formulation for quantum renormalization groups". *Phys. Rev. Lett.* 69 (19 1992), pp. 2863–2866. DOI: 10.1103/PhysRevLett.69.2863.

[14] U. Schollwöck. "The density-matrix renormalization group". *Rev. Mod. Phys.* 77 (1 2005), pp. 259–315. DOI: 10.1103/RevModPhys.77.259.

[15] S. Rommer and S. Östlund. "Class of ansatz wave functions for one-dimensional spin systems and their relation to the density matrix renormalization group". *Phys. Rev. B* 55 (4 1997), pp. 2164–2181. DOI: 10.1103/PhysRevB.55.2164.

[16] 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". *EPL* 43.4 (1998), p. 457. DOI: 10.1209/epl/i1998-00381-x.

[17] I. V. Oseledets. "Tensor-Train Decomposition". *SIAM J. Sci. Comput.* 33.5 (2011), pp. 2295–2317. DOI: 10.1137/090752286.

[18] R. J. Baxter. "Variational approximations for square lattice models in statistical mechanics". *J. Stat. Phys.* 19.5 (1978), pp. 461–478. DOI: 10.1007/BF01011693.

[19] I. Affleck, T. Kennedy, E. H. Lieb and H. Tasaki. "Rigorous results on valence-bond ground states in antiferromagnets". *Phys. Rev. Lett.* 59 (7 1987), pp. 799–802. DOI: 10.1103/PhysRevLett.59.799.

[20] M. Fannes, B. Nachtergaele and R. F. Werner. "Finitely correlated states on quantum spin chains". *Comm. Math. Phys.* 144.3 (1992), pp. 443–490. DOI: 10.1007/BF02099178.

[21] U. Schollwöck. "The density-matrix renormalization group in the age of matrix product states". *Ann. Phys.* 326.1 (2011), pp. 96–192. DOI: 10.1016/j.aop.2010.09.012.

[22] I. P. McCulloch. "From density-matrix renormalization group to matrix product states". *J. Stat. Mech.* 2007.10 (2007), P10014. DOI: 10.1088/1742-5468/2007/10/P10014.

[23] I. P. McCulloch. "Infinite size density matrix renormalization group, revisited" (2008). arXiv: 0804.2509 [cond-mat.str-el].

[24] V. Zauner-Stauber, L. Vanderstraeten, M. T. Fishman, F. Verstraete and J. Haegeman. "Variational optimization algorithms for uniform matrix product states" (2017). arXiv: 1701.07035 [quant-ph].

[25] E. Stoudenmire and S. R. White. "Studying Two-Dimensional Systems with the Density Matrix Renormalization Group". *Annu. Rev. Condens. Matter Phys.* 3.1 (2012), pp. 111–128. DOI: 10.1146/annurev-conmatphys-020911-125018.

[26] F. Verstraete and J. I. Cirac. "Renormalization algorithms for quantum-many body systems in two and higher dimensions" (2004). arXiv: cond-mat/0407066 [cond-mat].

[27] P. Corboz. "Variational optimization with infinite projected entangled-pair states". *Phys. Rev. B* 94 (3 2016), p. 035133. DOI: 10.1103/PhysRevB.94.035133.

[28] H. N. Phien, J. A. Bengua, H. D. Tuan, P. Corboz and R. Orús. "Infinite projected entangled pair states algorithm improved: Fast full update and gauge fixing". *Phys. Rev. B* 92 (3 2015), p. 035142. DOI: 10.1103/PhysRevB.92.035142.

[29] J. Jordan, R. Orús, G. Vidal, F. Verstraete and J. I. Cirac. "Classical Simulation of Infinite-Size Quantum Lattice Systems in Two Spatial Dimensions". *Phys. Rev. Lett.* 101 (25 2008), p. 250602. DOI: 10.1103/PhysRevLett.101.250602.

[30] P. Corboz, R. Orús, B. Bauer and G. Vidal. "Simulation of strongly correlated fermions in two spatial dimensions with fermionic projected entangled-pair states". *Phys. Rev. B* 81 (16 2010), p. 165104. DOI: 10.1103/PhysRevB.81.165104.

[31] I. P. McCulloch. "Collective Phenomena in Strongly Correlated Electron Systems". PhD thesis. Australian National University, 2002. URL: http://physics.uq.edu.au/people/ianmcc/aachen/public_html/thesis.pdf.

[32] I. P. McCulloch and M. Gulácsi. "The non-Abelian density matrix renormalization group algorithm". *EPL* 57.6 (2002), p. 852. DOI: 10.1209/epl/i2002-00393-0.

[33] B. Bauer, P. Corboz, R. Orús and M. Troyer. "Implementing global Abelian symmetries in projected entangled-pair state algorithms". *Phys. Rev. B* 83 (12 2011), p. 125106. DOI: 10.1103/PhysRevB.83.125106.

[34] S. Singh, R. N. C. Pfeifer and G. Vidal. "Tensor network decompositions in the presence of a global symmetry". *Phys. Rev. A* 82 (5 2010), p. 050301. DOI: 10.1103/PhysRevA.82.050301.

[35] 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 (11 2011), p. 115125. DOI: 10.1103/PhysRevB.83.115125.

[36] S. Singh and G. Vidal. "Tensor network states and algorithms in the presence of a global SU(2) symmetry". *Phys. Rev. B* 86 (19 2012), p. 195114. DOI: 10.1103/PhysRevB.86.195114.

[37] A. Weichselbaum. "Non-abelian symmetries in tensor networks: A quantum symmetry space approach". *Ann. Phys.* 327.12 (2012), pp. 2972–3047. DOI: 10.1016/j.aop.2012.07.009.

[38] H. Q. Lin. "Exact diagonalization of quantum-spin models". *Phys. Rev. B* 42 (10 1990), pp. 6561–6567. DOI: 10.1103/PhysRevB.42.6561.

[39] P. Ramond. *Group Theory*. Cambridge University Press, 2010. URL: http://www.cambridge.org/9780521896030.

[40] N. Schuch, M. M. Wolf, F. Verstraete and J. I. Cirac. "Entropy Scaling and Simulability by Matrix Product States". *Phys. Rev. Lett.* 100 (3 2008), p. 030504. DOI: 10.1103/PhysRevLett.100.030504.

[41] J. Eisert, M. Cramer and M. B. Plenio. "Colloquium: Area laws for the entanglement entropy". *Rev. Mod. Phys.* 82 (1 2010), pp. 277–306. DOI: 10.1103/RevModPhys.82.277.

[42] F. Verstraete, D. Porras and J. I. Cirac. "Density Matrix Renormalization Group and Periodic Boundary Conditions: A Quantum Information Perspective". *Phys. Rev. Lett.* 93 (22 2004), p. 227205. DOI: 10.1103/PhysRevLett.93.227205.

[43] P. Pippan, S. R. White and H. G. Evertz. "Efficient matrix-product state method for periodic boundary conditions". *Phys. Rev. B* 81 (8 2010), p. 081103. DOI: 10.1103/PhysRevB.81.081103.

[44] A. Ekert and P. L. Knight. "Entangled quantum systems and the Schmidt decomposition". *Am. J. Phys.* 63.5 (1995), pp. 415–423. DOI: 10.1119/1.17904.

[45] F. Verstraete and J. I. Cirac. "Matrix product states represent ground states faithfully". *Phys. Rev. B* 73 (9 2006), p. 094423. DOI: 10.1103/PhysRevB.73.094423.

[46] J. L. Lagrange. *Mécanique Analytique*. Ed. by J. P. M. Binet and J. G. Garnier. Ve Courcier, 1811.

[47] 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". *Comp. Phys. Comm.* 185.12 (2014), pp. 3430–3440. DOI: 10.1016/j.cpc.2014.08.019.

[48] O. Kahn, Y. Pei, M. Verdaguer, J. P. Renard and J. Sletten. "Magnetic ordering of manganese(II) copper(II) bimetallic chains; design of a molecular based ferromagnet". *J. Am. Chem. Soc.* 110.3 (1988), pp. 782–789. DOI: 10.1021/ja00211a017.

[49] S. K. Pati, S. Ramasesha and D. Sen. "Low-lying excited states and low-temperature properties of an alternating spin-1 - spin-1/2 chain: A density-matrix renormalization-group study". *Phys. Rev. B* 55 (14 1997), pp. 8894–8904. DOI: 10.1103/PhysRevB.55.8894.

[50] W. P. Su, J. R. Schrieffer and A. J. Heeger. "Solitons in Polyacetylene". *Phys. Rev. Lett.* 42 (25 1979), pp. 1698–1701. DOI: 10.1103/PhysRevLett.42.1698.

[51] S. Depenbrock, I. P. McCulloch and U. Schollwöck. "Nature of the Spin-Liquid Ground State of the $S = 1/2$ Heisenberg Model on the Kagome Lattice". *Phys. Rev. Lett.* 109 (6 2012), p. 067201. DOI: 10.1103/PhysRevLett.109.067201.

[52] E. M. Stoudenmire and S. R. White. "Minimally entangled typical thermal state algorithms". *New J. Phys.* 12.5 (2010), p. 055026. DOI: 10.1088/1367-2630/12/5/055026.

[53] I. P. McCulloch. *The Matrix Product Toolkit*. MPS/DMRG Software Suite. 2000-2017. URL: https://people.smp.uq.edu.au/IanMcCulloch/mptoolkit/.

[54] F. Fröwis, V. Nebendahl and W. Dür. "Tensor operators: Constructions and applications for long-range interaction systems". *Phys. Rev. A* 81 (6 2010), p. 062337. DOI: 10.1103/PhysRevA.81.062337.

[55] G. K.-L. Chan, A. Keselman, N. Nakatani, Z. Li and S. R. White. "Matrix product operators, matrix product states, and ab initio density matrix renormalization group algorithms". *J. Chem. Phys* 145.1 (2016), p. 014102. DOI: 10.1063/1.4955108.

[56] S. Keller, M. Dolfi, M. Troyer and M. Reiher. "An efficient matrix product operator representation of the quantum chemical Hamiltonian". *J. Chem. Phys* 143.24, 244118 (2015). DOI: 10.1063/1.4939000.

[57] E. R. Davidson. "The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices". *J. Comp. Phys.* 17.1 (1975), pp. 87–94. DOI: 10.1016/0021-9991(75)90065-0.

[58] E. Tirrito, S.-J. Ran, A. J. Ferris, I. P. McCulloch and M. Lewenstein. "Efficient perturbation theory to improve the density matrix renormalization group". *Phys. Rev. B* 95 (6 2017), p. 064110. DOI: 10.1103/PhysRevB.95.064110.

[59] R.-Z. Huang, H.-J. Liao, Z.-Y. Liu, H.-D. Xie, Z.-Y. Xie, H.-H. Zhao, J. Chen and T. Xiang. "A generalized Lanczos method for systematic optimization of tensor network states" (2016). arXiv: 1611.09574 [cond-mat.str-el].

[60] E. M. Stoudenmire and S. R. White. "Real-space parallel density matrix renormalization group". *Phys. Rev. B* 87 (15 2013), p. 155137. DOI: 10.1103/PhysRevB.87.155137.

[61] S. Depenbrock. "Tensor networks for the simulation of strongly correlated systems". PhD thesis. LMU Munich, 2013. URL: http://nbn-resolving.de/urn:nbn:de:bvb:19-159631.

[62] M. F. Dolfi. "Dilute systems with the density matrix renormalization group: from continuum to lattice models". PhD thesis. ETH Zurich, 2016. DOI: 10.3929/ethz-a-010735455.

[63] F. Dorfner and F. Heidrich-Meisner. "Properties of the single-site reduced density matrix in the Bose-Bose resonance model in the ground state and in quantum quenches". *Phys. Rev. A* 93 (6 2016), p. 063624. DOI: 10.1103/PhysRevA.93.063624.

[64] S. R. White. "Density matrix renormalization group algorithms with a single center site". *Phys. Rev. B* 72 (18 2005), p. 180403. DOI: 10.1103/PhysRevB.72.180403.

[65] Ö. Legeza and G. Fáth. "Accuracy of the density-matrix renormalization-group method". *Phys. Rev. B* 53 (21 1996), pp. 14349–14358. DOI: 10.1103/PhysRevB.53.14349.

[66] M. Dolfi, B. Bauer, M. Troyer and Z. Ristivojevic. "Multigrid Algorithms for Tensor Network States". *Phys. Rev. Lett.* 109 (2 2012), p. 020604. DOI: 10.1103/PhysRevLett.109.020604.

[67] S. Dolgov and D. Savostyanov. "Alternating Minimal Energy Methods for Linear Systems in Higher Dimensions". *SIAM J. Sci. Comput.* 36.5 (2014), A2248–A2271. DOI: 10.1137/140953289.

[68] D. Kressner, M. Steinlechner and B. Vandereycken. "Low-rank tensor completion by Riemannian optimization". *BIT Num. Math.* 54.2 (2014), pp. 447–468. DOI: 10.1007/s10543-013-0455-z.

[69] C. Brockt, F. Dorfner, L. Vidmar, F. Heidrich-Meisner and E. Jeckelmann. "Matrix-product-state method with a dynamical local basis optimization for bosonic systems out of equilibrium". *Phys. Rev. B* 92 (24 2015), p. 241106. DOI: 10.1103/PhysRevB.92.241106.

[70] F. Dorfner. "Numerical methods for strongly correlated many-body systems with bosonic degrees of freedom". PhD thesis. LMU Munich, 2017. URL: http://nbn-resolving.de/urn:nbn:de:bvb:19-205528.

[71] F. A. Wolf. "Solving dynamical mean-field theory using matrix product states". PhD thesis. LMU Munich, 2015. URL: http://nbn-resolving.de/urn:nbn:de:bvb:19-189370.

[72] G. Vidal. "Efficient Simulation of One-Dimensional Quantum Many-Body Systems". *Phys. Rev. Lett.* 93 (4 2004), p. 040502. DOI: 10.1103/PhysRevLett.93.040502.

[73] A. J. Daley, C. Kollath, U. Schollwöck and G. Vidal. "Time-dependent density-matrix renormalization-group using adaptive effective Hilbert spaces". *J. Stat. Mech.* 2004.04 (2004), P04005. DOI: 10.1088/1742-5468/2004/04/P04005.

[74] S. R. White and A. E. Feiguin. "Real-Time Evolution Using the Density Matrix Renormalization Group". *Phys. Rev. Lett.* 93 (7 2004), p. 076401. DOI: 10.1103/PhysRevLett.93.076401.

[75] M. L. Wall and L. D. Carr. "Out-of-equilibrium dynamics with matrix product states". *New J. Phys.* 14.12 (2012), p. 125015. DOI: 10.1088/1367-2630/14/12/125015.

[76] M. P. Zaletel, R. S. K. Mong, C. Karrasch, J. E. Moore and F. Pollmann. "Time-evolving a matrix product state with long-ranged interactions". *Phys. Rev. B* 91 (16 2015), p. 165112. DOI: 10.1103/PhysRevB.91.165112.

[77] J. J. García-Ripoll. "Time evolution of Matrix Product States". *New J. Phys.* 8.12 (2006), p. 305. DOI: 10.1088/1367-2630/8/12/305.

[78] M. Suzuki. "Generalized Trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems". *Comm. Math. Phys.* 51.2 (1976), pp. 183–190. DOI: 10.1007/BF01609348.

[79] N. Hatano and M. Suzuki. "Finding Exponential Product Formulas of Higher Orders". *Quantum Annealing and Other Optimization Methods*. Ed. by A. Das and B. K. Chakrabarti. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 37–68. DOI: 10.1007/11526216_2.

[80] M. Urbanek and P. Soldán. "Parallel implementation of the time-evolving block decimation algorithm for the Bose–Hubbard model". *Comp. Phys. Comm.* 199 (2016), pp. 170–177. DOI: 10.1016/j.cpc.2015.10.016.

[81] D. Bauernfeind, M. Zingl, R. Triebl, M. Aichhorn and H. G. Evertz. "Fork Tensor Product States - Efficient Three Orbital Real Time DMFT Solver" (2016). arXiv: 1612.05587 [cond-mat.str-el].

[82] J. Haegeman, C. Lubich, I. Oseledets, B. Vandereycken and F. Verstraete. "Unifying time evolution and optimization with matrix product states". *Phys. Rev. B* 94 (16 2016), p. 165116. DOI: 10.1103/PhysRevB.94.165116.

[83] Y. Saad. *Iterative Methods for Sparse Linear Systems*. 2nd ed. SIAM, 2003. URL: http://www-users.cs.umn.edu/~saad/IterMethBook_2ndEd.pdf.

[84] M. Hochbruck and C. Lubich. "On Krylov Subspace Approximations to the Matrix Exponential Operator". *SIAM J. Numer. Anal.* 34.5 (1997), pp. 1911–1925. DOI: 10.1137/S0036142995280572.

[85] F. A. Wolf, I. P. McCulloch, O. Parcollet and U. Schollwöck. "Chebyshev matrix product state impurity solver for dynamical mean-field theory". *Phys. Rev. B* 90 (11 2014), p. 115124. DOI: 10.1103/PhysRevB.90.115124.

[86] J. C. Halimeh, F. Kolley and I. P. McCulloch. "Chebyshev matrix product state approach for time evolution". *Phys. Rev. B* 92 (11 2015), p. 115130. DOI: 10.1103/PhysRevB.92.115130.

[87] J. Haegeman, T. J. Osborne and F. Verstraete. "Post-matrix product state methods: To tangent space and beyond". *Phys. Rev. B* 88 (7 2013), p. 075133. DOI: 10.1103/PhysRevB.88.075133.

[88] U. Schollwöck, J. Richter, D. Farnell and R. Bishop, eds. *Quantum Magnetism*. Vol. 645. Lecture Notes in Physics. Springer, 2004. DOI: 10.1007/b96825.

[89] T. D. Kühner and S. R. White. "Dynamical correlation functions using the density matrix renormalization group". *Phys. Rev. B* 60 (1 1999), pp. 335–343. DOI: 10.1103/PhysRevB.60.335.

[90] P. E. Dargel, A. Wöllert, A. Honecker, I. P. McCulloch, U. Schollwöck and T. Pruschke. "Lanczos algorithm with matrix product states for dynamical correlation functions". *Phys. Rev. B* 85 (20 2012), p. 205119. DOI: 10.1103/PhysRevB.85.205119.

[91] T. Barthel, U. Schollwöck and S. R. White. "Spectral functions in one-dimensional quantum systems at finite temperature using the density matrix renormalization group". *Phys. Rev. B* 79 (24 2009), p. 245101. DOI: 10.1103/PhysRevB.79.245101.

[92] T. Barthel. "Precise evaluation of thermal response functions by optimized density matrix renormalization group schemes". *New Journal of Physics* 15.7 (2013), p. 073010. DOI: 10.1088/1367-2630/15/7/073010.

[93] F. J. Harris. "On the use of windows for harmonic analysis with the discrete Fourier transform". *Proc. IEEE* 66.1 (1978), pp. 51–83. DOI: 10.1109/PROC.1978.10837.

[94] S. G. Johnson. *The NLopt nonlinear-optimization package*. URL: http://ab-initio.mit.edu/nlopt.

[95] D. Kraft. "Algorithm 733: TOMP–Fortran modules for optimal control calculations". *ACM Trans. Math. Software* 20.3 (1994), pp. 262–281. DOI: 10.1145/192115.192124.

[96] M. Oshikawa and I. Affleck. "Field-Induced Gap in $S = 1/2$ Antiferromagnetic Chains". *Phys. Rev. Lett.* 79 (15 1997), pp. 2883–2886. DOI: 10.1103/PhysRevLett.79.2883.

[97] I. Affleck and M. Oshikawa. "Field-induced gap in Cu benzoate and other $S = \frac{1}{2}$ antiferromagnetic chains". *Phys. Rev. B* 60 (2 1999), pp. 1038–1056. DOI: 10.1103/PhysRevB.60.1038.

[98] H. Shiba. "Quantization of Magnetic Excitation Continuum Due to Interchain Coupling in Nearly One-Dimensional Ising-Like Antiferromagnets". *Progress of Theoretical Physics* 64.2 (1980), p. 466. DOI: 10.1143/PTP.64.466.

[99] N. Ishimura and H. Shiba. "Dynamical Correlation Functions of One-Dimensional Anisotropic Heisenberg Model with Spin 1/2. I: Ising-Like Antiferromagnets". *Progress of Theoretical Physics* 63.3 (1980), p. 743. DOI: 10.1143/PTP.63.743.

[100] S. Coleman. "Quantum sine-Gordon equation as the massive Thirring model". *Phys. Rev. D* 11 (8 1975), pp. 2088–2097. DOI: 10.1103/PhysRevD.11.2088.

[101] T. Giamarchi. *Quantum Physics in One Dimension*. International Series of Monographs in Physics. Clarendon Press, 2004.

[102] A. Gogolin, A. Nersesyan and A. Tsvelik. *Bosonization and Strongly Correlated Systems*. Cambridge University Press, 2004.

[103] G. Delfino and G. Mussardo. "Non-integrable aspects of the multi-frequency sine-Gordon model". *Nucl. Phys. B* 516.3 (1998), pp. 675–703. DOI: 10.1016/S0550-3213(98)00063-7.

[104] I. Affleck. "Soliton Confinement and the Excitation Spectrum of Spin-Peierls Antiferromagnets". *Dynamical properties of unconventional magnetic systems*. Ed. by A. Skjeltorp and D. Sherrington. NATO ASI. 1998. URL: http://arxiv.org/abs/cond-mat/9705127v1.

[105] J. Haegeman, B. Pirvu, D. J. Weir, J. I. Cirac, T. J. Osborne, H. Verschelde and F. Verstraete. "Variational matrix product ansatz for dispersion relations". *Phys. Rev. B* 85 (10 2012), p. 100408. DOI: 10.1103/PhysRevB.85.100408.

[106] J. Hubbard. "Electron Correlations in Narrow Energy Bands". *Proc. R. Soc. A* 276.1365 (1963), pp. 238–257. DOI: 10.1098/rspa.1963.0204.

[107] P. W. Anderson. "New Approach to the Theory of Superexchange Interactions". *Phys. Rev.* 115 (1 1959), pp. 2–13. DOI: 10.1103/PhysRev.115.2.

[108] K. H. Marti, I. M. Ondík, G. Moritz and M. Reiher. "Density matrix renormalization group calculations on relative energies of transition metal complexes and clusters". *J. Chem. Phys* 128.1, 014104 (2008). DOI: 10.1063/1.2805383.

[109] M. A. Martín-Delgado, J. Rodriguez-Laguna and G. Sierra. "Density-matrix renormalization-group study of excitons in dendrimers". *Phys. Rev. B* 65 (15 2002), p. 155116. DOI: 10.1103/PhysRevB.65.155116.

[110] G. Knizia and G. K.-L. Chan. "Density Matrix Embedding: A Simple Alternative to Dynamical Mean-Field Theory". *Phys. Rev. Lett.* 109 (18 2012), p. 186404. DOI: 10.1103/PhysRevLett.109.186404.

[111] G. Knizia and G. K.-L. Chan. "Density Matrix Embedding: A Strong-Coupling Quantum Embedding Theory". *J. Chem. Theory Comput.* 9.3 (2013). PMID: 26587604, pp. 1428–1432. DOI: 10.1021/ct301044e.

[112] B. Keimer, S. A. Kivelson, S. Uchida and J. Zaanen. "From quantum matter to high-temperature superconductivity in copper oxides". *Nature* 518 (2015), pp. 179–186. DOI: 10.1038/nature14165.

[113] F. C. Zhang and T. M. Rice. "Effective Hamiltonian for the superconducting Cu oxides". *Phys. Rev. B* 37 (7 1988), pp. 3759–3761. DOI: 10.1103/PhysRevB.37.3759.

[114] J. G. Bednorz and K. A. Müller. "Possible high-T_c superconductivity in the Ba–La–Cu–O system". *Z. Phys. B* 64.2 (1986), pp. 189–193. DOI: 10.1007/BF01303701.

[115] M. K. Wu, J. R. Ashburn, C. J. Torng, P. H. Hor, R. L. Meng, L. Gao, Z. J. Huang, Y. Q. Wang and C. W. Chu. "Superconductivity at 93 K in a new mixed-phase Y-Ba-Cu-O compound system at ambient pressure". *Phys. Rev. Lett.* 58 (9 1987), pp. 908–910. DOI: 10.1103/PhysRevLett.58.908.

[116] Y. Kamihara, H. Hiramatsu, M. Hirano, R. Kawamura, H. Yanagi, T. Kamiya and H. Hosono. "Iron-Based Layered Superconductor: LaOFeP". *J. Am. Chem. Soc.* 128.31 (2006). PMID: 16881620, pp. 10012–10013. DOI: 10.1021/ja063355c.

[117] Y. Kamihara, T. Watanabe, M. Hirano and H. Hosono. "Iron-Based Layered Superconductor $LaO_{1-x}F_xFeAs$ with $T_c = 26$ K". *J. Am. Chem. Soc.* 130.11 (2008). PMID: 18293989, pp. 3296–3297. DOI: 10.1021/ja800073m.

[118] A. Subedi, L. Zhang, D. J. Singh and M. H. Du. "Density functional study of FeS, FeSe, and FeTe: Electronic structure, magnetism, phonons, and superconductivity". *Phys. Rev. B* 78 (13 2008), p. 134514. DOI: 10.1103/PhysRevB.78.134514.

[119] A. P. Drozdov, M. I. Eremets, I. A. Troyan, V. Ksenofontov and S. I. Shylin. "Conventional superconductivity at 203 kelvin at high pressures in the sulfur hydride system". *Nature* 525 (2015), pp. 73–76. DOI: 10.1038/nature14964.

[120] D. A. Wollman, D. J. Van Harlingen, W. C. Lee, D. M. Ginsberg and A. J. Leggett. "Experimental determination of the superconducting pairing state in YBCO from the phase coherence of YBCO-Pb dc SQUIDs". *Phys. Rev. Lett.* 71 (13 1993), pp. 2134–2137. DOI: 10.1103/PhysRevLett.71.2134.

[121] C. C. Tsuei and J. R. Kirtley. "Pairing symmetry in cuprate superconductors". *Rev. Mod. Phys.* 72 (4 2000), pp. 969–1016. DOI: 10.1103/RevModPhys.72.969.

[122] J. P. F. LeBlanc et al. "Solutions of the Two-Dimensional Hubbard Model: Benchmarks and Results from a Wide Range of Numerical Algorithms". *Phys. Rev. X* 5 (4 2015), p. 041041. DOI: 10.1103/PhysRevX.5.041041.

[123] W. Metzner and D. Vollhardt. "Correlated Lattice Fermions in $d = \infty$ Dimensions". *Phys. Rev. Lett.* 62 (3 1989), pp. 324–327. DOI: 10.1103/PhysRevLett.62.324.

[124] A. Georges and G. Kotliar. "Hubbard model in infinite dimensions". *Phys. Rev. B* 45 (12 1992), pp. 6479–6483. DOI: 10.1103/PhysRevB.45.6479.

[125] E. H. Lieb and F. Y. Wu. "Absence of Mott Transition in an Exact Solution of the Short-Range, One-Band Model in One Dimension". *Phys. Rev. Lett.* 20 (25 1968), pp. 1445–1448. DOI: 10.1103/PhysRevLett.20.1445.

[126] J. E. Hirsch. "Two-dimensional Hubbard model: Numerical simulation study". *Phys. Rev. B* 31 (7 1985), pp. 4403–4419. DOI: 10.1103/PhysRevB.31.4403.

[127] H. Yokoyama and H. Shiba. "Variational Monte-Carlo Studies of Hubbard Model. II". *J. Phys. Soc. Jpn.* 56.10 (1987), pp. 3582–3592. DOI: 10.1143/JPSJ.56.3582.

[128] H. Yokoyama and H. Shiba. "Variational Monte-Carlo Studies of Hubbard Model. I". *J. Phys. Soc. Jpn.* 56.4 (1987), pp. 1490–1506. DOI: 10.1143/JPSJ.56.1490.

[129] C. Gros. "Physics of projected wavefunctions". *Ann. Phys.* 189.1 (1989), pp. 53–88. DOI: 10.1016/0003-4916(89)90077-8.

[130] E. Dagotto. "Correlated electrons in high-temperature superconductors". *Rev. Mod. Phys.* 66 (3 1994), pp. 763–840. DOI: 10.1103/RevModPhys.66.763.

[131] Y. Deng, E. Kozik, N. V. Prokof'ev and B. V. Svistunov. "Emergent BCS regime of the two-dimensional fermionic Hubbard model: Ground-state phase diagram". *EPL* 110.5 (2015), p. 57001. DOI: 10.1209/0295-5075/110/57001.

[132] J. Gukelberger, L. Huang and P. Werner. "On the dangers of partial diagrammatic summations: Benchmarks for the two-dimensional Hubbard model in the weak-coupling regime". *Phys. Rev. B* 91 (23 2015), p. 235114. DOI: 10.1103/PhysRevB.91.235114.

[133] H. Shi, S. Chiesa and S. Zhang. "Ground-state properties of strongly interacting Fermi gases in two dimensions". *Phys. Rev. A* 92 (3 2015), p. 033603. DOI: 10.1103/PhysRevA.92.033603.

[134] S. R. White, D. J. Scalapino, R. L. Sugar, E. Y. Loh, J. E. Gubernatis and R. T. Scalettar. "Numerical study of the two-dimensional Hubbard model". *Phys. Rev. B* 40 (1 1989), pp. 506–516. DOI: 10.1103/PhysRevB.40.506.

[135] N. Trivedi and D. M. Ceperley. "Ground-state correlations of quantum antiferromagnets: A Green-function Monte Carlo study". *Phys. Rev. B* 41 (7 1990), pp. 4552–4569. DOI: 10.1103/PhysRevB.41.4552.

[136] M. Capello, F. Becca, S. Yunoki and S. Sorella. "Unconventional metal-insulator transition in two dimensions". *Phys. Rev. B* 73 (24 2006), p. 245116. DOI: 10.1103/PhysRevB.73.245116.

[137] J. P. F. LeBlanc and E. Gull. "Equation of state of the fermionic two-dimensional Hubbard model". *Phys. Rev. B* 88 (15 2013), p. 155108. DOI: 10.1103/PhysRevB.88.155108.

[138] T. Schäfer, F. Geles, D. Rost, G. Rohringer, E. Arrigoni, K. Held, N. Blümer, M. Aichhorn and A. Toschi. "Fate of the false Mott-Hubbard transition in two dimensions". *Phys. Rev. B* 91 (12 2015), p. 125109. DOI: 10.1103/PhysRevB.91.125109.

[139] T. Schäfer, A. Toschi and K. Held. "Dynamical vertex approximation for the two-dimensional Hubbard model". *J. of Magn. and Magn. Mater.* 400 (2016). Proceedings of the 20th International Conference on Magnetism (Barcelona) 5-10 July 2015, pp. 107–111. DOI: 10.1016/j.jmmm.2015.07.103.

[140] D. A. Huse. "Ground-state staggered magnetization of two-dimensional quantum Heisenberg antiferromagnets". *Phys. Rev. B* 37 (4 1988), pp. 2380–2382. DOI: 10.1103/PhysRevB.37.2380.

[141] V. J. Emery, S. A. Kivelson and H. Q. Lin. "Phase separation in the t-J model". *Phys. Rev. Lett.* 64 (4 1990), pp. 475–478. DOI: 10.1103/PhysRevLett.64.475.

[142] M. A. Baranov and M. Y. Kagan. "D-wave pairing in the two-dimensional Hubbard model with low filling". *Z. Phys. B* 86.2 (1992), pp. 237–239. DOI: 10.1007/BF01313830.

[143] A. V. Chubukov and J. P. Lu. "Pairing instabilities in the two-dimensional Hubbard model". *Phys. Rev. B* 46 (17 1992), pp. 11163–11166. DOI: 10.1103/PhysRevB.46.11163.

[144] W. O. Putikka, M. U. Luchini and T. M. Rice. "Aspects of the phase diagram of the two-dimensional t-J model". *Phys. Rev. Lett.* 68 (4 1992), pp. 538–541. DOI: 10.1103/PhysRevLett.68.538.

[145] D. Zanchi and H. J. Schulz. "Superconducting instabilities of the non-half-filled Hubbard model in two dimensions". *Phys. Rev. B* 54 (13 1996), pp. 9509–9519. DOI: 10.1103/PhysRevB.54.9509.

[146] F. Becca, M. Capone and S. Sorella. "Spatially homogeneous ground state of the two-dimensional Hubbard model". *Phys. Rev. B* 62 (19 2000), pp. 12700–12706. DOI: 10.1103/PhysRevB.62.12700.

[147] C. J. Halboth and W. Metzner. "d-wave Superconductivity and Pomeranchuk Instability in the Two-Dimensional Hubbard Model". *Phys. Rev. Lett.* 85 (24 2000), pp. 5162–5165. DOI: 10.1103/PhysRevLett.85.5162.

[148] M. Lugas, L. Spanu, F. Becca and S. Sorella. "Finite compressibility in the low-doping region of the two-dimensional $t-J$ model". *Phys. Rev. B* 74 (16 2006), p. 165122. DOI: 10.1103/PhysRevB.74.165122.

[149] C.-C. Chang and S. Zhang. "Spatially inhomogeneous phase in the two-dimensional repulsive Hubbard model". *Phys. Rev. B* 78 (16 2008), p. 165101. DOI: 10.1103/PhysRevB.78.165101.

[150] J. Otsuki, H. Hafermann and A. I. Lichtenstein. "Superconductivity, antiferromagnetism, and phase separation in the two-dimensional Hubbard model: A dual-fermion approach". *Phys. Rev. B* 90 (23 2014), p. 235132. DOI: 10.1103/PhysRevB.90.235132.

[151] C. Gros. "Superconductivity in correlated wave functions". *Phys. Rev. B* 38 (1 1988), pp. 931–934. DOI: 10.1103/PhysRevB.38.931.

[152] 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 (6 1998), pp. 1272–1275. DOI: 10.1103/PhysRevLett.80.1272.

[153] S. R. White and D. J. Scalapino. "Energetics of Domain Walls in the 2D t-J Model". *Phys. Rev. Lett.* 81 (15 1998), pp. 3227–3230. DOI: 10.1103/PhysRevLett.81.3227.

[154] S. R. White and D. J. Scalapino. "Phase separation and stripe formation in the two-dimensional t-J model: A comparison of numerical results". *Phys. Rev. B* 61 (9 2000), pp. 6320–6326. DOI: 10.1103/PhysRevB.61.6320.

[155] S. R. White and D. J. Scalapino. "Checkerboard patterns in the t-J model". *Phys. Rev. B* 70 (22 2004), p. 220506. DOI: 10.1103/PhysRevB.70.220506.

[156] S. R. White and D. J. Scalapino. "Pairing on striped t-t′-J lattices". *Phys. Rev. B* 79 (22 2009), p. 220504. DOI: 10.1103/PhysRevB.79.220504.

[157] C.-C. Chang and S. Zhang. "Spin and Charge Order in the Doped Hubbard Model: Long-Wavelength Collective Modes". *Phys. Rev. Lett.* 104 (11 2010), p. 116402. DOI: 10.1103/PhysRevLett.104.116402.

[158] P. Corboz, S. R. White, G. Vidal and M. Troyer. "Stripes in the two-dimensional t-J model with infinite projected entangled-pair states". *Phys. Rev. B* 84 (4 2011), p. 041108. DOI: 10.1103/PhysRevB.84.041108.

[159] P. Corboz, T. M. Rice and M. Troyer. "Competing States in the *t-J* Model: Uniform *d*-Wave State versus Stripe State". *Phys. Rev. Lett.* 113 (4 2014), p. 046402. DOI: 10.1103/PhysRevLett.113.046402.

[160] R. Rodríguez-Guzmán, C. A. Jiménez-Hoyos and G. E. Scuseria. "Variational description of the ground state of the repulsive two-dimensional Hubbard model in terms of nonorthogonal symmetry-projected Slater determinants". *Phys. Rev. B* 90 (19 2014), p. 195110. DOI: 10.1103/PhysRevB.90.195110.

[161] B.-X. Zheng, C.-M. Chung, P. Corboz, G. Ehlers, M.-P. Qin, R. M. Noack, H. Shi, S. R. White, S. Zhang and G. Kin-Lic Chan. "Stripe order in the underdoped region of the two-dimensional Hubbard model" (2017). arXiv: 1701.00054 [cond-mat.str-el].

[162] A. W. Sandvik. "Finite-size scaling and boundary effects in two-dimensional valence-bond solids". *Phys. Rev. B* 85 (13 2012), p. 134407. DOI: 10.1103/PhysRevB.85.134407.

[163] P. Corboz. "Improved energy extrapolation with infinite projected entangled-pair states applied to the two-dimensional Hubbard model". *Phys. Rev. B* 93 (4 2016), p. 045116. DOI: 10.1103/PhysRevB.93.045116.

[164] J. D. van der Waals. "Over de Continuiteit van den Gas- en Vloeistoftoestand". PhD thesis. Universiteit Leiden, 1873. URL: http://www.scs.illinois.edu/~mainzv/exhibit/vanderwaals.htm.

[165] J. C. Maxwell. "On the dynamical evidence of the molecular constitution of bodies". *Nature* 11 (1875), p. 357. URL: https://www.nature.com/nature/journal/v11/n279/pdf/011357a0.pdf.

[166] F. I. Šimkovic, Y. Deng, N. V. Prokof'ev, B. V. Svistunov, I. Tupitsyn and E. Kozik (2016). private communication.

[167] K. M. Stadler, Z. P. Yin, J. von Delft, G. Kotliar and A. Weichselbaum. "Dynamical Mean-Field Theory Plus Numerical Renormalization-Group Study of Spin-Orbital Separation in a Three-Band Hund Metal". *Phys. Rev. Lett.* 115 (13 2015), p. 136401. DOI: 10.1103/PhysRevLett.115.136401.

[168] F. A. Wolf, A. Go, I. P. McCulloch, A. J. Millis and U. Schollwöck. "Imaginary-Time Matrix Product State Impurity Solver for Dynamical Mean-Field Theory". *Phys. Rev. X* 5 (4 2015), p. 041032. DOI: 10.1103/PhysRevX.5.041032.