

Computations in Quantum Tensor Networks

T. Huckle^{a,*}, K. Waldherr^a, T. Schulte-Herbrüggen^b

^a*Technische Universität München, Boltzmannstr. 3, 85748 Garching, Germany*

^b*Technische Universität München, Lichtenbergstr. 4, 85748 Garching, Germany*

Abstract

The computation of the ground state (i.e. the eigenvector related to the smallest eigenvalue) is an important task in the simulation of quantum many-body systems. As the dimension of the underlying vector space grows exponentially in the number of particles, one has to consider appropriate subsets promising both convenient approximation properties and efficient computations. The variational ansatz for this numerical approach leads to the minimization of the Rayleigh quotient. The Alternating Least Squares technique is then applied to break down the eigenvector computation to problems of appropriate size, which can be solved by classical methods. Efficient computations require fast computation of the matrix-vector product and of the inner product of two decomposed vectors. To this end, both appropriate representations of vectors and efficient contraction schemes are needed.

Here approaches from many-body quantum physics for one-dimensional and two-dimensional systems (Matrix Product States and Projected Entangled Pair States) are treated mathematically in terms of tensors. We give the definition of these concepts, bring some results concerning uniqueness and numerical stability and show how computations can be executed efficiently within these concepts. Based on this overview we present some modifications and generalizations of these concepts and show that they still allow efficient computations such as applicable contraction schemes. In this context we consider the minimization of the Rayleigh quotient in terms of the PARAFAC (CP) formalism, where we also allow different tensor partitions. This approach makes use of efficient contraction schemes for the calculation of inner products in a way that can easily be extended to the MPS format but also to higher dimensional problems.

Keywords: Quantum many-body systems, Density Matrix Renormalization Group (DMRG), Matrix Product States (MPS), Tensor Trains, Projected Entangled-Pair States, Canonical Decomposition (candecomp or PARAFAC)

*Corresponding author

Email addresses: huckle@in.tum.de (T. Huckle), waldherr@in.tum.de (K. Waldherr), tosh@tum.de (T. Schulte-Herbrüggen)

1. Introduction

Computations with tensors are getting increasingly important in high dimensional problems. In particular in quantum many-body physics, a typical problem amounts to finding an accurate approximation to the smallest eigenvalue (i.e., the ground-state energy) of a hermitian matrix (representing the Hamiltonian) that is larger than one can store even on a powerful computer. To this end, in quantum physics techniques like Matrix Product States (MPS) or Projected Entangled Pair States (PEPS) have been developed for representing vectors, *viz.* eigenstates of quantum systems *efficiently*. In mathematics, besides the Tucker decomposition and the canonical decomposition, concepts like Tensor Trains (TT) were introduced. The examples of MPS or PEPS (in physics) and TT (in mathematics) express a common interest in powerful numerical methods specifically designed for coping with high-dimensional tensor networks. — Unifying variational approaches to ground-state calculations [1] in a *common framework of tensor approximations* will be highly useful, in particular in view of optimizing numerical algorithms [2, 3, 4, 5]. Here it is the goal to cast some of the recent developments in mathematical physics into such a common frame expressed in the terminology of multilinear algebra. Moreover we present numerical results on the Ising-type Hamiltonian underscoring the wealth and the potential of such an approach.

In this paper, we address one-dimensional and two-dimensional methods in a unified frame related to MPS and PEPS. We will introduce some generalization of MPS and the canonical decomposition. Furthermore, we give a short description of tensor-decomposition methods for 2D problems.

Scope and Organization

The paper is organized as follows: Sections 2, 3 and 4 contain an overview of already-known concepts and describe them in the multilinear algebra language: in Section 2 we introduce the physical background of the problem setting and define the matrices involved, in Section 3 we present representation schemes for states in physically motivated 1D and 2D arrays and we show how computations can be performed efficiently and Section 4 finally fixes some basics and notations for tensors and tensor-decomposition schemes.

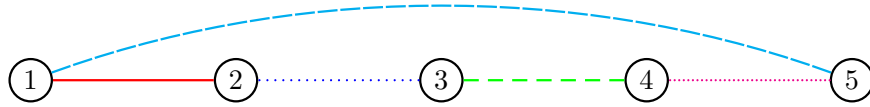
In Section 5 we present new ideas of how to generalize these basic concepts, how to execute calculations efficiently and how to apply them to the ground-state approximation problem. First numerical results will show the benefit of these newly developed concepts.

2. Physical Model Systems

Consider vectors \mathbf{x} in a complex Hilbert space \mathcal{H} representing states of (pure) quantum systems. The differential equation $\dot{\mathbf{x}} = -iH\mathbf{x}$ (Schrödinger's equation of motion) then governs quantum dynamics (neglecting relaxation) with the Hamiltonian H being the generator of unitary time evolution. The Hamiltonian

captures the energies of the constituent subsystems (e.g. spins) as well as the interaction energies between coupled subsystems.

For instance, a linear chain of five spins coupled by nearest neighbor interactions can be depicted as in Figure 1(a). In the case of *open boundary conditions*



(a) 1D system of 5 spins with nearest-neighbor interaction and periodic boundary conditions.

$$\begin{aligned}
 H = & \boxed{P_z \otimes P_z \otimes I \otimes I \otimes I} \\
 & + \boxed{I \otimes P_z \otimes P_z \otimes I \otimes I} \\
 & + \boxed{I \otimes I \otimes P_z \otimes P_z \otimes I} \\
 & + \boxed{I \otimes I \otimes I \otimes P_z \otimes P_z} \\
 & + \boxed{P_z \otimes I \otimes I \otimes I \otimes P_z}
 \end{aligned}$$

(b) Representation of the Hamiltonian related to the physical system illustrated by Figure 1(a).

Figure 1: Example of a linear physical system (a) and the related Hamiltonian (b).

(OBC) there is no coupling interaction between particle 1 and 5, while for *periodic boundary conditions* (PBC) there is a non vanishing coupling interaction between 1 and 5.

2.1. Hamiltonian Representations

For spin $\frac{1}{2}$ particles such as electrons or protons, the spin angular momentum operator describing their internal degree of freedom (i.e. spin-up and spin-down) is usually expressed in terms of the *Pauli matrices*

$$P_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, P_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \text{ and } P_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (1)$$

Being traceless and Hermitian, $\{P_x, P_y, P_z\}$ forms a basis of the Lie algebra $\mathfrak{su}(2)$, while by appending the identity matrix I one obtains a basis of the Lie algebra $\mathfrak{u}(2)$.

Now, spin Hamiltonians are built by summing M terms, each of them representing a physical (inter)action. These terms are themselves tensor products of Pauli matrices or identities

$$H = \sum_{k=1}^M \underbrace{\alpha_k (Q_1^{(k)} \otimes Q_2^{(k)} \otimes \dots \otimes Q_p^{(k)})}_{=: H^{(k)}} = \sum_{k=1}^M H^{(k)}, \quad (2)$$

where $Q_j^{(k)}$ can be P_x, P_y, P_z or I .

In each summand $H^{(k)}$ most of the $Q_j^{(k)}$ are I : *local terms* have just one non-trivial tensor factor, while *pair interactions* have two of them. Higher m -body interactions (with $m > 2$) usually do not occur as physical primitives, but could be represented likewise by m Pauli matrices in the tensor product representing the m -order interaction term¹.

For instance, in the *Ising* (ZZ) model [7] for the 1D chain with p spins and open boundary conditions, the spin Hamiltonian takes the form

$$H = \sum_{k=1}^{p-1} I^{\otimes(k-1)} \otimes (P_z)_k \otimes (P_z)_{k+1} \otimes I^{\otimes(p-k-1)} + \lambda \sum_{k=1}^p I^{\otimes(k-1)} \otimes (P_x)_k \otimes I^{\otimes(p-k)}, \quad (3)$$

where the index k denotes the position in the spin chain and the real number λ describes the ratio of the strengths of the magnetic field and the pair interactions. For simplicity, we will henceforth drop the tensor powers of the identity and tacitly assume appropriate embedding. Then a Hamiltonian for an open-boundary 1D *Heisenberg* (XY) model [8, 9] reads

$$H = \sum_{k=1}^{p-1} \left(J_x \cdot I \otimes (P_x)_k \otimes (P_x)_{k+1} \otimes I + J_y \cdot I \otimes (P_y)_k \otimes (P_y)_{k+1} \otimes I \right) + \lambda \sum_{k=1}^p I \otimes (P_x)_k \otimes I. \quad (4)$$

with real constants J_x, J_y and λ . Models with all coefficients being different are called *anisotropic*.

Being a sum (2) of Kronecker products of structured 2×2 matrices many Hamiltonians have special properties: they can be multilevel-circulant ([10, 11]) or skew-circulant, diagonal or persymmetric ([12]), which can be exploited to derive properties of the respective eigenvalues and eigenvectors.

2.2. Computation of Ground States: Physical Background

A key to understand some of the motivating guidelines lies in the somewhat striking fact that quantum dynamical systems typically evolve in a way that looks non-generic from a mathematical point of view. Yet the very structure of quantum dynamics paves the way to tailored parameterizations based on tensor compression that are efficient in the sense of scaling only polynomially in physical system size. Some of these motivating guidelines established in quantum physics may be sketched as follows: Composing a quantum system

¹For further details, a reader wishing to approach quantum physics from linear and multilinear algebra may refer to [6].

from its components takes a joint Hilbert space that is the tensor product of the individual Hilbert spaces. Likewise a linear operator on the joint Hilbert space can be composed by taking sums (or weighted linear combinations) of tensor factors (like in Eqn. 2). Clearly, in general a linear combination of tensor products does not take the form of a tensor product itself. Thus a quantum state space grows *exponentially* with the number of constituents in contrast to a classical configuration space just growing linearly.

However, correlating quantum interactions typically become smaller and smaller with increasing distance between subsystems (‘particles’): for instance, in Eqn. 3 only nearest-neighbor interactions had to be taken into account. On a general scale, this can be made precise in terms of *area laws*, where the correlations are quantified by a measure termed ‘entanglement entropy’ of ground states [13, 14, 15], see also the recent review in [16]. Remarkably, this entropy of the reduced state of a subregion is not extensive: it typically grows with the boundary region (‘area’) between the subregion and its complement rather than with the volume of the subregion. In one-dimensional systems, a rigorous area law has recently been proven for all systems with a gap between the smallest and the second smallest eigenvalue [17]. Extending the results to two-dimensional lattice systems, however, currently requires stronger assumptions on the eigenvalue distribution [18].

Guided by these area laws, long-distance correlations may be neglected in the sense that eigenvectors (ground states) of physical systems are well approximated within truncated subsets, as has been quantitatively established, e.g., for MPS [19]. Moreover MPS-approximations to ground states can provably be calculated efficiently [20]. Along similar lines, consecutive partitionings have been exploited in unitary and tensor networks addressing ground states and dynamics of large-scale quantum systems. Related techniques for truncating the Hilbert space to pertinent parameterized subsets not only include Matrix Product States (MPS) [21, 22] of Density Matrix Renormalization Groups (DMRG) [23, 24], but also projected entangled pair states (PEPS) [25, 26], weighted graph states (WGS) [27], Multi-scale Entanglement Renormalization Approaches (MERA) [28], string-bond states (SBS) [29] as well as combined methods [1, 30].

To conclude, the evolution of physical systems does not exploit the generic state space (with long-distance many-body interactions), but proceeds via well-defined subspaces of short-distance and mainly pairwise interactions that can be parameterized by data-sparse formats which allow for tensor-contraction schemes.

2.3. Computation of Ground States: Numerical Aspects

The ground state energy of a physical system modeled by the Hamiltonian H corresponds to the smallest eigenvalue of H , which is the minimum of the Rayleigh quotient [31]

$$\min_{\mathbf{x} \in \mathcal{H}} \frac{\mathbf{x}^H H \mathbf{x}}{\mathbf{x}^H \mathbf{x}}. \quad (5)$$

As long as p , the number of particles, is not too large, any standard numerical method for computing eigenvalues and eigenvectors can be used.

But with increasing p , the size of the Hamiltonians grows like 2^p . Thus for $p > 50$ neither matrices nor even vectors of this size can be stored. So, similar to the description of the Hamiltonian (2) we need a sparse approximate representation of eigenvectors. Assume that we have already chosen an appropriate subset $\mathcal{U} \subset \mathcal{H}$, the goal is to find approximations for the eigenvector in this set. Hence we consider the minimization of the Rayleigh quotient (5) only on the subset \mathcal{U} :

$$\min_{\mathbf{x} \in \mathcal{U}} \frac{\mathbf{x}^H H \mathbf{x}}{\mathbf{x}^H \mathbf{x}}. \quad (6)$$

An appropriate subset of vectors should allow for easy computation of $H\mathbf{x}$ and of inner products $\mathbf{y}^H \mathbf{x}$. Therefore we consider vector representations with a less number of coefficients where subsets of the indices can be grouped in partitions corresponding to the binary tensor structure of the Hamiltonian (2). Please note that, in general, the chosen subsets do not form linear subspaces.

2.4. Alternating Least Squares

An important tool for minimizing the Rayleigh quotient for an appropriate subset \mathcal{U} is the Alternating Least Squares approach (ALS), see [32, 33]. Here, all subsets of the partitioning up to one are assumed to be fixed, and then the minimization is reduced to the remaining subset. As introductory example let us look at a set of vectors defined by

$$\mathbf{x} = \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \cdots \otimes \mathbf{x}_p = (x_{1;i_1} \cdot x_{2;i_2} \cdots x_{p;i_p})_{i_1, \dots, i_p} = (x_i)_{i=0, \dots, 2^p-1} \quad (7)$$

with vectors \mathbf{x}_i of length 2, and $i = (i_1, \dots, i_p)_2$ the binary representation of i with $i_j \in \{0, 1\}$. Hence we envisage the vector \mathbf{x} as a p -tensor. So in our example (7) we assume all subsets fixed up to \mathbf{x}_r , and then the minimization is simplified to

$$\begin{aligned} \min_{\mathbf{x}_r} \frac{\mathbf{x}^H H \mathbf{x}}{\mathbf{x}^H \mathbf{x}} &= \min_{\mathbf{x}_r} \frac{(\mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_p)^H \left(\sum_{k=1}^M \alpha_k Q_1^{(k)} \otimes \cdots \otimes Q_p^{(k)} \right) (\mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_p)}{(\mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_p)^H (\mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_p)} \\ &= \min_{\mathbf{x}_r} \frac{\sum_{k=1}^M \alpha_k (\mathbf{x}_1^H Q_1^{(k)} \mathbf{x}_1) \cdots (\mathbf{x}_r^H Q_r^{(k)} \mathbf{x}_r) \cdots (\mathbf{x}_p^H Q_p^{(k)} \mathbf{x}_p)}{(\mathbf{x}_1^H \mathbf{x}_1) \cdots (\mathbf{x}_r^H \mathbf{x}_r) \cdots (\mathbf{x}_p^H \mathbf{x}_p)} \\ &= \min_{\mathbf{x}_r} \frac{\mathbf{x}_r^H \left(\sum_{k=1}^M \alpha_k \beta_k Q_r^{(k)} \right) \mathbf{x}_r}{\mathbf{x}_r^H (\gamma I) \mathbf{x}_r} = \min_{\mathbf{x}_r} \frac{\mathbf{x}_r^H R_r \mathbf{x}_r}{\mathbf{x}_r^H \mathbf{x}_r}, \end{aligned}$$

a standard eigenvalue problem in the effective Hamiltonian $R_r = \sum_{k=1}^M \frac{\alpha_k \beta_k}{\gamma} Q_r^{(k)}$. More generally, if we work with more complex representations such as Matrix Product States, the minimization of the Rayleigh quotient (6) will lead to

$$\min_{\mathbf{x}_r} \frac{\mathbf{x}^H H \mathbf{x}}{\mathbf{x}^H \mathbf{x}} = \min_{\mathbf{x}_r} \frac{\mathbf{x}_r^H R_r \mathbf{x}_r}{\mathbf{x}_r^H N_r \mathbf{x}_r} \quad (8)$$

with an effective Hamiltonian related to the generalized eigenvalue problem in R_r and N_r . So far, Eq. 8 describes the general situation, the particular specification of the matrices R_r and N_r will be given when we consider different representation schemes for the eigenvector. Then, \mathbf{x}_r is set to the eigenvector with smallest eigenvalue. We can repeat this procedure step by step for all \mathbf{x}_j to get approximations for the minimal eigenvalue of H . The main costs are caused by matrix-vector products $H\mathbf{x}$ and inner products $\mathbf{y}^H\mathbf{x}$ plus the solution of the relatively small generalized eigenvalue problem (8). It is therefore important to have efficient schemes for the evaluation of inner products of two vectors out of the chosen subset \mathcal{U} . We emphasize that this approach (and adapted modifications) allows to overcome the curse of dimensionality as it is only polynomial in the maximum length of the small vectors \mathbf{x}_j , in the number of such vectors (which can be upper bounded by p) and in the number of local terms M .

The ansatz (8) may cause problems if the denominator matrix N_r is singular. In that case one would apply an orthogonal projection on the nonsingular subspace of N_r .

3. Representations of States for Computing Ground States

Obviously, in general the above simplistic approach based on a single tensor product cannot give good approximations to the eigenvector. Therefore, we have to find a clever combination of such terms. The choice naturally depends on the dimension and the neighborhood relation of the physical setting. So first we consider the 1D linear setting, and in a following section we look at the 2D problem.

3.1. 1D Systems: Approximation by Matrix Product States

The Matrix Product State (MPS) formalism goes back to several sources: early ones are by Affleck, Kennedy, Lieb, and Tasaki [34, 9] including their revival by Fannes, Nachtergaele, and Werner [21, 22], while a more recent treatment is due to Vidal [35]. The application to the eigenvalue problem was discussed by Delgado et al. [36]. A seemingly independent line of thought resorts to the fact that Density Matrix Renormalization Group (DMRG) methods as developed by Wilson and White [37, 38] have a natural interpretation as optimization in the class of MPS states, see, e.g., Refs. [23, 24, 39]. As has been mentioned already, ground states of gapped 1D Hamiltonians are faithfully represented by MPS [19], where the MPS-approximation can be computed efficiently [39, 20], the rationale being an area law [17].

3.1.1. Formalism and Computations

For MPS small $D_j \times D_{j+1}$ -matrices are used for describing vectors in a compact form. The advantage is due to the fact that $D := \max\{D_j\}$ (the *bond dimension*) has to grow only polynomially in the number of particles in order to approximate ground states with a given precision ([19]).

In MPS, the vector components are given by

$$\begin{aligned} x_i = x_{i_1, \dots, i_p} &= \text{tr} \left(A_1^{(i_1)} \cdot A_2^{(i_2)} \cdots A_p^{(i_p)} \right) \\ &= \sum_{m_1=1}^{D_1} \cdots \sum_{m_p=1}^{D_p} a_{1;m_1, m_2}^{(i_1)} \cdot a_{2;m_2, m_3}^{(i_2)} \cdots a_{p;m_p, m_1}^{(i_p)}. \end{aligned} \quad (9)$$

The matrix products lead to indices and summation over m_2, \dots, m_p , and the trace introduces m_1 . The upper (physical or given) indices i_j identify which of the two possible matrices are used at each position, and thereby they determine the vector components. So, e.g., the last component is described by

$$x_{2^p-1} = x_{1, \dots, 1} = \text{tr}(A_1^{(1)} \cdot A_2^{(1)} \cdots A_p^{(1)}) = \sum_{m_1, \dots, m_p} a_{1;m_1, m_2}^{(1)} \cdot a_{2;m_2, m_3}^{(1)} \cdots a_{p;m_p, m_1}^{(1)},$$

where we always choose the matrix index $i_j = 1$. The additional summation indices are called *ancilla* indices.

The above trace form is related to the periodic case. In the open boundary case, there is no connection between first and last particle, and therefore the index m_1 can be neglected.

In this case we have $D_1 = D_{p+1} = 1$ and thus the matrices at the ends are of size $1 \times D_2$ and $D_p \times 1$ respectively.

$$\begin{aligned} x_{i_1, \dots, i_p} &= A_1^{(i_1)} \cdot A_2^{(i_2)} \cdots A_{p-1}^{(i_{p-1})} \cdot A_p^{(i_p)} \\ &= \sum_{m_2=1}^{D_2} \cdots \sum_{m_p=1}^{D_p} a_{1;1, m_2}^{(i_1)} \cdot a_{2;m_2, m_3}^{(i_2)} \cdots a_{p-1;m_{p-1}, m_p}^{(i_{p-1})} \cdot a_{p;m_p, 1}^{(i_p)}. \end{aligned} \quad (10)$$

By introducing the unit vectors $\mathbf{e}_i = \mathbf{e}_{i_1, \dots, i_p} = \mathbf{e}_{i_1} \otimes \cdots \otimes \mathbf{e}_{i_p}$ with unit vectors \mathbf{e}_{i_j} of length 2, another useful representation of the MPS vector is given by

$$\begin{aligned} \mathbf{x} &= \sum_{i_1, \dots, i_p} x_{i_1, \dots, i_p} \mathbf{e}_{i_1, \dots, i_p} = \sum_{i_1, \dots, i_p} \text{tr}(A_1^{(i_1)} \cdots A_p^{(i_p)}) \mathbf{e}_{i_1, \dots, i_p} \\ &= \sum_{i_1, \dots, i_p} \sum_{m_1, \dots, m_p} a_{1;m_1, m_2}^{(i_1)} \cdots a_{p;m_p, m_1}^{(i_p)} \mathbf{e}_{i_1, \dots, i_p} \\ &= \sum_{m_1, \dots, m_p} \left(\sum_{i_1} a_{1;m_1, m_2}^{(i_1)} \mathbf{e}_{i_1} \right) \otimes \cdots \otimes \left(\sum_{i_p} a_{p;m_p, m_1}^{(i_p)} \mathbf{e}_{i_p} \right) \\ &= \sum_{m_1, \dots, m_p} \mathbf{a}_{1;m_1, m_2} \otimes \cdots \otimes \mathbf{a}_{p;m_p, m_1} \end{aligned} \quad (11)$$

with length 2 vectors $\mathbf{a}_{r;m_r, m_{r+1}}$ where the two components are pairwise entries in the matrices $A_r^{(0)}$ and $A_r^{(1)}$ at position (m_r, m_{r+1}) :

$$\mathbf{a}_{r;m_r, m_{r+1}} := \begin{pmatrix} (0) \\ a_{r;m_r, m_{r+1}} \\ (1) \\ a_{r;m_r, m_{r+1}} \end{pmatrix}.$$

Uniqueness of MPS and Normal Forms

In this section, we want to summarize some known results concerning the uniqueness of MPS. For further details, see, e.g., [40]. Obviously the representation of an MPS vector is not unique. So, for a vector with components

$$x_i = x_{i_1, \dots, i_p} = \text{tr} \left(A_1^{(i_1)} \cdot A_2^{(i_2)} \cdot \dots \cdot A_{p-1}^{(i_{p-1})} \cdot A_p^{(i_p)} \right) \quad (12)$$

we can replace the matrices by

$$A_j^{(i_j)} \rightarrow M_j^{-1} A_j^{(i_j)} M_{j+1}, A_1^{(i_1)} \rightarrow A_1^{(i_1)} M_2, A_p^{(i_p)} \rightarrow M_p^{-1} A_p^{(i_p)} \quad (13)$$

with nonsingular matrices $M_j \in \mathbb{C}^{D_j \times D_j}$, $j = 2, \dots, p$.

The absence of uniqueness also causes problems in the solution of the effective generalized eigenvalue problem, because the matrix N_r in Eqn. (8) might be positive semidefinite, but singular. To avoid this problem, we switch from a given MPS representation to a representation based on unitary matrices. To this end, we combine the matrix pair $A_r^{(i_r)}$ for $i_r = 0, 1$ to a rectangular matrix and compute the SVD:

$$\begin{pmatrix} A_r^{(0)} \\ A_r^{(1)} \end{pmatrix} = U_r \cdot \begin{pmatrix} \Lambda_r \\ 0 \end{pmatrix} \cdot V_r = \begin{pmatrix} U_r^{(0)} \\ U_r^{(1)} \end{pmatrix} \cdot (\Lambda_r V_r) \quad (14)$$

where the $U_r^{(i_r)}$ are the left part of U_r . Now we can replace at position r in the MPS vector the matrix pair $A_r^{(i_r)}$ by the pair $U_r^{(i_r)}$ and multiply the remaining SVD factor $\Lambda_r V_r$ from the left to the right neighbor pair $A_{r+1}^{(i_{r+1})}$ without changing the vector:

$$\begin{aligned} & \text{tr} \left(A_1^{(i_1)} \cdot A_2^{(i_2)} \cdot \dots \cdot A_r^{(i_r)} \cdot A_{r+1}^{(i_{r+1})} \cdot \dots \cdot A_{p-1}^{(i_{p-1})} \cdot A_p^{(i_p)} \right) \longrightarrow \\ & \text{tr} \left(A_1^{(i_1)} \cdot A_2^{(i_2)} \cdot \dots \cdot U_r^{(i_r)} \cdot (\Lambda_r V_r) A_{r+1}^{(i_{r+1})} \cdot \dots \cdot A_{p-1}^{(i_{p-1})} \cdot A_p^{(i_p)} \right) . \end{aligned}$$

So we can start from the left, normalizing first $A_1^{(i_1)}$, always moving the remaining SVD part to the right neighbor, until we reach $A_p^{(i_p)}$. During this procedure the MPS matrix pairs $A_j^{(i_j)}$, $j = 1, \dots, p-1$ are replaced by parts of unitary matrices $U_j^{(i_j)}$, which fulfil the *gauge condition*

$$\sum_{i_j} U_j^{(i_j)H} U_j^{(i_j)} = I . \quad (15)$$

In the case of open boundary conditions, the right (unnormalized) matrices $A_p^{(i_p)}$ are column vectors and thus $\sum_{i_p} A_p^{(i_p)H} A_p^{(i_p)}$ is only a scalar γ , which corresponds to the squared norm of the MPS vector:

$$\mathbf{x}^H \mathbf{x} = \sum_{i_1, \dots, i_p} \overline{\left(A_1^{(i_1)} \cdot \dots \cdot A_p^{(i_p)} \right)} \cdot \left(A_1^{(i_1)} \cdot \dots \cdot A_p^{(i_p)} \right)$$

$$\begin{aligned}
&= \sum_{i_1, \dots, i_p} \left(A_1^{(i_1)} \dots A_p^{(i_p)} \right)^H \left(A_1^{(i_1)} \dots A_p^{(i_p)} \right) \\
&= \sum_{i_p} A_p^{(i_p)H} \left(\dots \sum_{i_2} A_2^{(i_2)H} \left(\sum_{i_1} A_1^{(i_1)H} A_1^{(i_1)} \right) A_2^{(i_2)} \dots \right) A_p^{(i_p)} \\
&\stackrel{(15)}{=} \sum_{i_p} A_p^{(i_p)H} A_p^{(i_p)} = \gamma.
\end{aligned}$$

Thus, if \mathbf{x} has norm one, the gauge condition (15) is also fulfilled for $j = p$.

The same procedure can be applied in order to move the remaining SVD part to the left neighbor. To this end, we compute

$$\left(A_r^{(0)} \ A_r^{(1)} \right) = V_r \cdot (\Lambda_r \ 0) \cdot U_r = (V_r \Lambda_r) \left(U_r^{(0)} \ U_r^{(1)} \right). \quad (16)$$

Similarly we can move from right to left and replace the matrix pairs $A_j^{(i_j)}$, $j = p, \dots, 2$ by the unitaries $U_j^{(i_j)}$ until we reach $A_1^{(i_1)}$. Now the gauge conditions take the form

$$\sum_{i_j} U_j^{(i_j)} U_j^{(i_j)H} = I. \quad (17)$$

Analogously, for open boundary conditions the remaining left matrices $A_1^{(i_1)}$ are row vectors and so $\sum_{i_p} A_1^{(i_1)} A_1^{(i_1)H}$ is simply a scalar, which is 1 for a norm 1 vector.

So far, for the normalization process only one matrix pair $A_j^{(i_j)}$ was involved. Similar to the two-site DMRG approach [24], it is also possible to consider the matrices related to two neighboring sites at once [41]. To this end, we consider the two matrix pairs $A_j^{(i_j)} \in \mathbb{C}^{D_j \times D_{j+1}}$ and $A_{j+1}^{(i_{j+1})} \in \mathbb{C}^{D_{j+1} \times D_{j+2}}$. The four matrix products $A_j^{(i_j)} A_{j+1}^{(i_{j+1})} \in \mathbb{C}^{D_j \times D_{j+2}}$ are now re-arranged in matrix notation and an SVD is carried out:

$$\begin{pmatrix} A_j^{(0)} \\ A_j^{(1)} \end{pmatrix} \begin{pmatrix} A_{j+1}^{(0)} & A_{j+1}^{(1)} \end{pmatrix} = \begin{pmatrix} A_j^{(0)} A_{j+1}^{(0)} & A_j^{(0)} A_{j+1}^{(1)} \\ A_j^{(1)} A_{j+1}^{(0)} & A_j^{(1)} A_{j+1}^{(1)} \end{pmatrix} = \begin{pmatrix} U_j^{(0)} \\ U_j^{(1)} \end{pmatrix} \Sigma_j \begin{pmatrix} V_{j+1}^{(0)} & V_{j+1}^{(1)} \end{pmatrix}.$$

If we sweep from left to right, we replace the matrices $A_j^{(i_j)}$ by parts of unitary matrices $U_j^{(i_j)}$, shift the remaining part to the right neighbor, i.e.

$$A_{j+1}^{(i_{j+1})} \leftarrow \Sigma_j V_{j+1}^{(i_{j+1})}$$

and proceed with the adjacent sites $j + 1$ and $j + 2$. Accordingly, if we sweep from right to left, we replace the matrices $A_{j+1}^{(i_{j+1})}$ by the unitaries $V_{j+1}^{(i_{j+1})}$, shift the remaining part to site j , i.e.

$$A_j^{(i_j)} \leftarrow U_j^{(i_j)} \Sigma_j$$

and proceed with the index pair $(j-1, j)$.

There exist even stronger normalization conditions allowing representations which are unique up to permutations and degeneracies in the singular values, see, e.g. [42, 40]. The proof of the existence of such normal forms is based on the SVD of special matricizations of the vector to be represented, see [42, 43]. In the SVD-TT algorithm [44], the same technique is in use. In [43] we present normal forms for MPS which allow for expressing certain symmetry relations.

The presented normalization techniques for MPS vectors have various advantages. They introduce normal forms for MPS vectors which lead to better convergence properties. For the minimization of the Rayleigh quotient (8), the gauge conditions circumvent the problem of bad conditioned N_r matrices in the denominator and therefore approve numerical stability. So far, the presented normalization technique only changes the representation of the vector but does not change the overall vector. However, the SVD can also be used as a truncation technique. This could be interesting if we want to keep the matrix dimensions limited by some $D = D_{\max}$. As an example we mention the PEPS format [45], where such an SVD-based reduction appears, compare Subsection 3.2.

Sum of MPS Vectors

Unfortunately, the MPS formalism does not define a linear subspace. According to [41] the sum of two MPS vectors \mathbf{x} and \mathbf{y} , which are both in PBC form, can be formulated as

$$\begin{aligned}
 \mathbf{x} + \mathbf{y} &= \sum_{i_1, \dots, i_p} \text{tr} \left(A_1^{(i_1)} \dots A_p^{(i_p)} \right) \mathbf{e}_{i_1, \dots, i_p} + \sum_{i_1, \dots, i_p} \text{tr} \left(B_1^{(i_1)} \dots B_p^{(i_p)} \right) \mathbf{e}_{i_1, \dots, i_p} \\
 &= \sum_{i_1, \dots, i_p} \text{tr} \left[\begin{pmatrix} A_1^{(i_1)} \dots A_p^{(i_p)} & \\ & B_1^{(i_1)} \dots B_p^{(i_p)} \end{pmatrix} \right] \mathbf{e}_{i_1, \dots, i_p} \\
 &= \sum_{i_1, \dots, i_p} \text{tr} \left[\begin{pmatrix} A_1^{(i_1)} & \\ & B_1^{(i_1)} \end{pmatrix} \dots \begin{pmatrix} A_p^{(i_p)} & \\ & B_p^{(i_p)} \end{pmatrix} \right] \mathbf{e}_{i_1, \dots, i_p} \\
 &= \sum_{i_1, \dots, i_p} \text{tr} \left(C_1^{(i_1)} \dots C_p^{(i_p)} \right) \mathbf{e}_{i_1, \dots, i_p} .
 \end{aligned} \tag{18}$$

In the open boundary case, we can define the interior matrices C_2, \dots, C_{p-1} in the same way. For reasons of consistency, the C matrices at the boundary sites 1 and p have to be specified to be vectors, i.e.

$$C_1^{(i_1)} = \left(A_1^{(i_1)}, B_1^{(i_1)} \right), \quad C_p^{(i_p)} = \begin{pmatrix} A_p^{(i_p)} \\ B_p^{(i_p)} \end{pmatrix} .$$

Hence, the sum of MPS vectors is again an MPS vector, but of larger size. In order to keep the sizes of the MPS matrices limited by a constant D_{\max} , one could apply an SVD-based truncation of the resulting MPS formalism and consider only the D_{\max} dominant terms.

Properties of MPS

Every unit vector can be represented by an MPS with bond dimension $D = 1$: let $j = (j_1, \dots, j_p)$ be the binary form of j , then \mathbf{e}_j can be formulated as an MPS with 1×1 MPS matrices $A_r^{(i_r)} = \delta_{i_r, j_r}$:

$$\mathbf{e}_j = \sum_{i_1, \dots, i_p=0}^1 (\delta_{i_1, j_1} \cdots \delta_{i_p, j_p}) \mathbf{e}_{i_1, \dots, i_p}.$$

In view of Eqn. (18) this observation may be extended as follows: every sparse vector with at most D non zero entries can be written as an MPS with bond dimension D .

Computation of Matrix-Vector Products

To solve the minimization of the Rayleigh quotient (8) in an ALS way as introduced in Subsection 2.4, we have to compute

$$\begin{aligned} \mathbf{y}_k &= H^{(k)} \mathbf{x} = \left(\alpha_k Q_1^{(k)} \otimes \cdots \otimes Q_p^{(k)} \right) \cdot \sum_{m_1, \dots, m_p} \left(\mathbf{a}_{1; m_1, m_2} \otimes \cdots \otimes \mathbf{a}_{p; m_p, m_1} \right) \\ &= \sum_{m_1, \dots, m_p} \left(Q_1^{(k)} \mathbf{a}_{1; m_1, m_2} \right) \otimes \cdots \otimes \left(Q_p^{(k)} \mathbf{a}_{p; m_p, m_1} \right) \\ &= \sum_{m_1, \dots, m_p} \left(\mathbf{b}_{1, k; m_1, m_2} \otimes \cdots \otimes \mathbf{b}_{p, k; m_p, m_1} \right) \end{aligned}$$

and derive a sum of MPS vectors

$$\mathbf{y} = H \mathbf{x} = \sum_{k=1}^M H^{(k)} \mathbf{x} = \sum_{k=1}^M \mathbf{y}_k. \quad (19)$$

Therefore, we can compute all small vectors $Q_j^{(k)} \mathbf{a}_{j; m_j, m_{j+1}}$ in $\mathcal{O}(2pD^2M)$ operations. For reasons of efficiency there are also concepts to express the Hamiltonian in terms of a *Matrix Product Operator* (MPO) as defined in (22). This approach enables short representations for $\text{MPO} \times \text{MPS}$ products.

Computation of Inner Products

Furthermore, we have to compute inner products of two MPS vectors

$$\sum_{i_1, \dots, i_p} \sum_{\substack{m_1, \dots, m_p \\ k_1, \dots, k_p}} \left(\bar{b}_{1; k_1, k_2}^{(i_1)} \cdots \bar{b}_{p; k_p, k_1}^{(i_p)} \right) \cdot \left(a_{1; m_1, m_2}^{(i_1)} \cdots a_{p; m_p, m_1}^{(i_p)} \right). \quad (20)$$

To deal with these sums, it is essential to specify in which order the summations have to be conducted. In the next figures we display such an efficient ordering for the periodic case. To this end, we introduce the following notation. Each box in the following figures describes one factor, e.g., $a_{r; m_r, m_{r+1}}^{(i_r)}$, in these collections of sums. Little lines (legs) describe the connection of two such boxes via a

common index. Hence, the number of legs of a box is exactly the number of indices. So in this case, most of the boxes have three legs. Figure 2(a) shows the sum and the first terms, resp. boxes, with their connections. This also represents a Tensor Network.

Now, in a first step we reorder the sums, and execute a contraction relative to index i_1 . This is given by the partial sum

$$\sum_{i_1} \bar{b}_{1;k_1,k_2}^{(i_1)} a_{1;m_1,m_2}^{(i_1)} = c_{k_1,k_2;m_1,m_2} . \quad (21)$$

This eliminates two boxes, but leads to a new four leg tensor $c_{k_1,k_2;m_1,m_2}$ as shown in 2(b) and 3(a). Now we contract index k_2 , as shown in Figure 3(b),

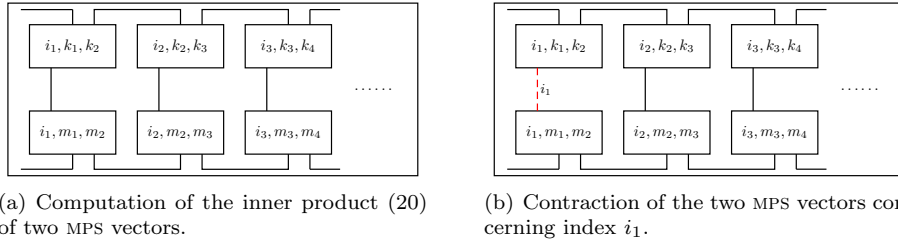


Figure 2: Contraction of the inner product of two MPS vectors. The dashed red line illustrates the index being contracted.

leading to 3(c). In the following step we go from 3(c) to 3(d) by contracting i_2 and m_2 , deriving at Figure 3(d). Now we are in the same situation as in Figure 3(a), and we can proceed exactly in the same way, until all sums are executed, resp. all indices have been contracted.

The main costs depend on the size of the contracted index, e.g. 2 for contracting i_r or D for contracting m_r or k_r , and on the size of the other indices that appear in the contracted boxes. Hence, e.g. the contraction in Figure 3(b) costs $D \cdot 2D^3 = 2D^4$ operations, and Figure 3(c) costs $2D \cdot D^4 = 2D^5$ operations. The total costs for the inner product is therefore less than $4D^5p$, because contractions have to be done until all boxes are removed, that is $2p$ -times. Therefore, the costs for computing $\mathbf{x}^H H \mathbf{x}$ based on (19) are less than $4D^5Mp$.

In the open boundary case, we start at the left or right side and therefore only contractions of boxes of length 3 can occur and thus Figure 3(c) only costs $2D \cdot D^2 = 2D^3$ instead of $2D^5$ in the periodic case. Thus, the overall costs for the inner product is less than $4D^3p$ and for $\mathbf{x}^H H \mathbf{x}$ less than $4D^3Mp$.

Minimization of the Rayleigh Quotient in Terms of MPS

Now, we use the MPS formalism as vector ansatz to minimize the Rayleigh quotient (6). Therefore we start the ALS procedure, updating $A_p^{(i_p)}$ by an improved estimate by solving the generalized effective eigenvalue problem. In addition, we replace $A_p^{(i_p)}$ by a unitary pair via SVD, moving the remaining

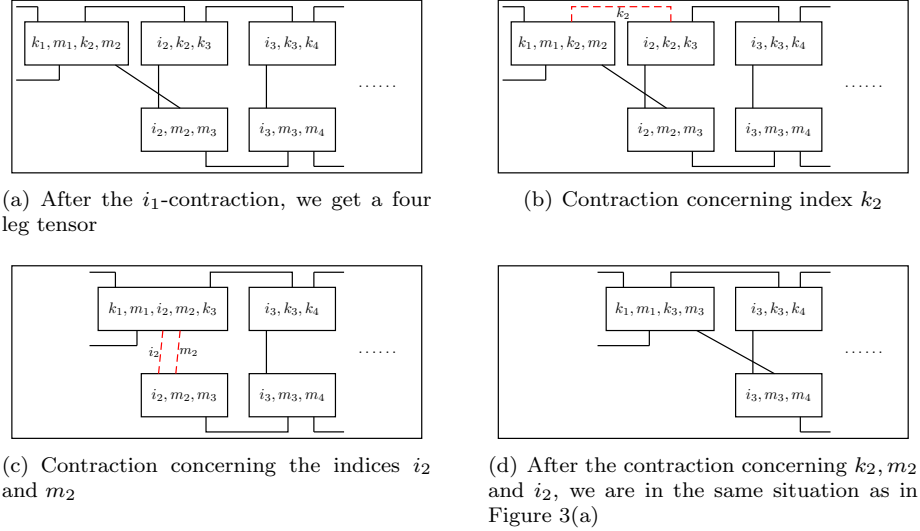


Figure 3: Contraction of the inner product of two MPS vectors.

SVD term to the left neighbor, and so on. A nice side effect here is that in the case of open boundary conditions the matrix N_r is the identity because all the matrix pairs to index different from r are parts of unitary matrices and thus fulfil one of the gauge conditions (15) or (17). Together with the fact that the MPS matrices at both ends are simply vectors we obtain

$$\begin{aligned}
\mathbf{x}^H \mathbf{x} &= \sum_{i_1, \dots, i_p} \text{tr}(\bar{A}_1^{(i_1)} \dots \bar{A}_p^{(i_p)}) \cdot \text{tr}(A_1^{(i_1)} \dots A_p^{(i_p)}) \\
&= \sum_{i_1, \dots, i_p} \text{tr}\left(\left(\bar{A}_1^{(i_1)} \dots \bar{A}_p^{(i_p)}\right) \otimes \left(A_1^{(i_1)} \dots A_p^{(i_p)}\right)\right) \\
&= \sum_{i_1, \dots, i_p} \text{tr}\left(\left(\bar{A}_1^{(i_1)} \otimes A_1^{(i_1)}\right) \dots \left(\bar{A}_p^{(i_p)} \otimes A_p^{(i_p)}\right)\right) \\
&= \text{tr}\left(\sum_{i_1, \dots, i_p} \left(\left(\bar{A}_1^{(i_1)} \otimes A_1^{(i_1)}\right) \dots \left(\bar{A}_p^{(i_p)} \otimes A_p^{(i_p)}\right)\right)\right) \\
&= \text{tr}\left(\prod_j \sum_{i_j} \left(\bar{A}_j^{(i_j)} \otimes A_j^{(i_j)}\right)\right) \\
&= \sum_{i_r} \text{tr}\left(A_r^{(i_r)} A_r^{(i_r)H}\right).
\end{aligned}$$

In the case of periodic boundary conditions the SVD-based normalization can only be performed for all up to one matrix pair $A_r^{(i_r)}$ and so the denominator matrix N_r is non-trivial. However, the normalization is utilized to make the problem numerically stable (see [45]).

The matrix H_r for the eigenvalue problem is given by

$$\begin{aligned}
\mathbf{x}^H H \mathbf{x} &= \\
&= \sum_{\substack{i_1, \dots, i_p \\ i'_1, \dots, i'_p \\ k}} \text{tr}(\bar{A}_1^{(i_1)} \dots \bar{A}_p^{(i_p)}) \cdot \text{tr}(A_1^{(i'_1)} \dots A_p^{(i'_p)}) \cdot (\mathbf{e}_{i'_1}^H Q_1^{(k)} \mathbf{e}_{i_1}) \dots (\mathbf{e}_{i'_p}^H Q_p^{(k)} \mathbf{e}_{i_p}) \\
&= \sum_{\substack{i_1, \dots, i_p \\ i'_1, \dots, i'_p \\ k}} \text{tr}\left((\bar{A}_1^{(i_1)} \dots \bar{A}_p^{(i_p)}) \otimes (A_1^{(i'_1)} \dots A_p^{(i'_p)})\right) \cdot \left(Q_{1;i'_1, i_1}^{(k)} \dots Q_{p;i'_p, i_p}^{(k)}\right) \\
&= \sum_{\substack{i_1, \dots, i_p \\ i'_1, \dots, i'_p \\ k}} \text{tr}\left((\bar{A}_1^{(i_1)} \otimes A_1^{(i'_1)}) \dots (\bar{A}_p^{(i_p)} \otimes A_p^{(i'_p)})\right) \cdot \left(Q_{1;i'_1, i_1}^{(k)} \dots Q_{p;i'_p, i_p}^{(k)}\right) \\
&= \sum_k \text{tr} \left[\sum_{\substack{i_1, \dots, i_p \\ i'_1, \dots, i'_p}} \left(Q_{1;i'_1, i_1}^{(k)} \cdot (\bar{A}_1^{(i_1)} \otimes A_1^{(i'_1)})\right) \dots \left(Q_{p;i'_p, i_p}^{(k)} \cdot (\bar{A}_p^{(i_p)} \otimes A_p^{(i'_p)})\right) \right] \\
&= \sum_k \text{tr} \left(\prod_j \sum_{i'_j, i_j} (Q_{j;i'_j, i_j}^{(k)} \cdot (\bar{A}_j^{(i_j)} \otimes A_j^{(i'_j)})) \right).
\end{aligned}$$

The effective Hamiltonian can be computed by contracting all indices except the indices representing the unknowns in matrix pair $A_r^{(i_r)}$, $i_r, i'_r, m_r, m_{r+1}, k_r, k_{r+1}$ leading to a $2D^2 \times 2D^2$ matrix H_r . So the costs for solving the eigenvalue problem are in this case $O(D^4)$. In the periodic case also a SVD for N_r has to be computed to solve the generalized eigenvalue problem numerically stable, which leads to costs of order D^6 .

3.1.2. Matrix Product Density Operators

The Matrix Product Operator approach extends the idea behind MPS from vectors to operators. Matrix Product Density Operators MPDO [46] have the form

$$\sum_{i, i'} \text{tr} \left(A_1^{(i_1, i'_1)} \dots A_p^{(i_p, i'_p)} \right) \mathbf{e}_i \mathbf{e}_{i'}^T \quad (22)$$

with unit vectors \mathbf{e}_i . Matrix Product Operators MPO [4] read

$$\sum_{i_1, \dots, i_p} \text{tr} \left(A_1^{(i_1)} \dots A_p^{(i_p)} \right) P_{i_1} \otimes \dots \otimes P_{i_p} \quad (23)$$

with 2×2 matrices P , e.g. the Pauli matrices (1). Similarly, the TT format has also been extended to the matrix case ([47]). These MPO concepts may be used for a representation of the Hamiltonian, such that the application of the Hamiltonian on an MPS vector leads to a sum of MPS vectors with less addends.

3.2. 2D Systems: Approximation by Projected Entangled Pair States

It is natural to extend the coupling topology of interest from linear chains to 2D arrays representing an entire lattice with open or periodic boundary conditions. To this end, the matrices in MPS are replaced by higher-order tensors thus giving rise to Projected Entangled Pair States (PEPS) [25]. Again, the underlying guideline is an area law [48, 18]. The extension to higher dimensions, however, comes at considerably higher computational cost: calculating expectation values becomes NP-hard (actually the complexity class is $\#P$ -complete) [49]. This is one reason, why computing ground states in two-dimensional arrays remains a major challenge to numerics [5].

For 2D spin systems, the interaction between particles is also of 2D form, e.g., as described by Figure 4. This leads to 2D generalization of MPS using a

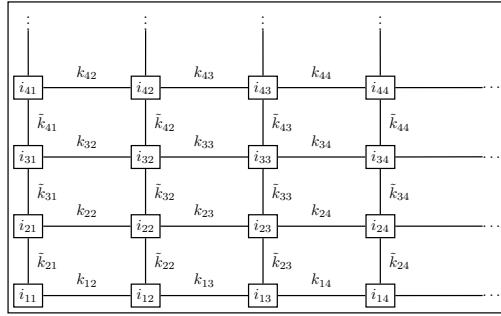


Figure 4: A 2D system in the open boundary case with physical indices i_j and ancilla indices k and \bar{k} .

tensor network with small boxes related to tensor $\left(a_{\bar{k}_r, k_{r+1}; \bar{k}_s, \bar{k}_{s+1}}^{(i_r, s)} \right)$ with 5 legs. Thus, an inner product of two such vectors would look like

$$\sum \bar{b}_{\bar{k}'_r, k'_{r+1}; \bar{k}'_s, \bar{k}'_{s+1}}^{(i_r, s)} a_{\bar{k}_r, k_{r+1}; \bar{k}_s, \bar{k}_{s+1}}^{(i_r, s)} .$$

In a PEPS vector, matrices are replaced by higher order tensors with one physical index related to the given vector \mathbf{x} and the other ancilla indices related to nearest neighbors according to the lattice structure of the tensor network ([50]). A PEPS vector can be formally written as

$$\mathbf{x} = \sum_{I_R} C_R(\{A^{I_R}\}) \mathbf{e}_{I_R} ,$$

where I stands for all \mathbf{x} -indices in Region R , C_R represents the contraction of indices following the nearest neighbor structure.

To compute the inner product of two PEPS vectors, we have to find an efficient ordering of the summations. The related tensor network for the open boundary case is displayed in Figure 5(a).

In a first step all pairwise contractions relative to the vector indices $i_{r,s}$ are computed. Furthermore, in the produced boxes the indices are newly numbered by combining $k_{r,s}$ with $m_{r,s}$ to larger index $k'_{r,s}$. This generates Figure 5(b).

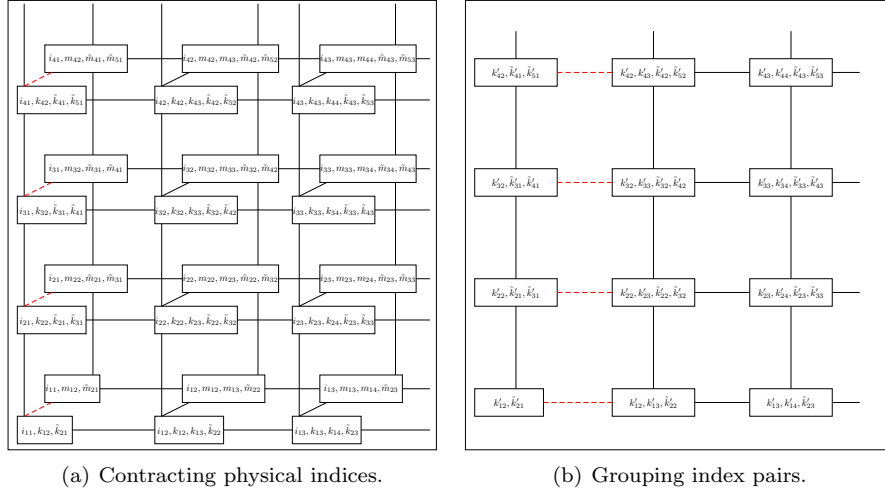


Figure 5: Contraction of two PEPS vectors: After the contraction of the physical indices related to the first column (a) the index pairs $m_{r,s}$ and $k_{r,s}$ are grouped to larger indices $k'_{r,s}$.

Now the first and second column are contracted starting e.g. from the bottom, resulting in the network displayed in Figure 6(a). Unfortunately, the boxes in the newly generated left column have more indices than in the starting column. So we cannot proceed in this way directly. In order to keep the number of indices constant, an approximation step is introduced. The main idea is to reduce the number of indices by considering the first left column as MPS—with indices related to connections with the right neighbors as original physical indices (longer than 2)—and approximating the boxes by little tensors with only one leg instead of two to the vertical neighbors.

Such a reduction can be derived by the following procedure. We want to reduce the rank of size D^2 in index pair $\{k_{2,1}, k_{2,2}\}$ to a new index $k'_{2,1}$ of size D . We can rewrite the whole summation in three parts, where c contains the contracted summation over all indices that are not involved in the actual reduction process. This leads to the sum

$$\sum a_{\{k'_{2,3}, k'_{2,1}\}, \{k'_{3,1}, k'_{3,2}\}} \cdot b_{\{k'_{3,1}, k'_{3,2}\}, \{k'_{3,3}, k'_{4,1}, k'_{4,2}\}} \cdot c_{\{k'_{3,3}, k'_{4,1}, k'_{4,2}\}, \{k'_{2,3}, k'_{2,1}\}} \cdot$$

The entries a in the above sum build the MPS matrices $A_{(k'_{2,1}, k'_{2,2}), (k'_{3,1}, k'_{3,2})}^{(k'_{2,3})}$. Collecting these matrices in a large rectangular matrix, we can apply the SVD on this matrix. Now we can truncate the diagonal part of the SVD to reduce the matrices A from size $D^2 \times D^2$ to size $D^2 \times D$. If we repeat this process along the first column all indices are reduced to length D . Note that this approximation

step is not affected by any accuracy demand but it is performed to keep the number of indices constant, which allows to proceed in an iterative way.

Reducing the rank of the indices in the first column leads to the network illustrated by Figure 6(b) with the same structure as Figure 5(b), so we can contract column by column until we are left with only one column that can be contracted following Figure 7.

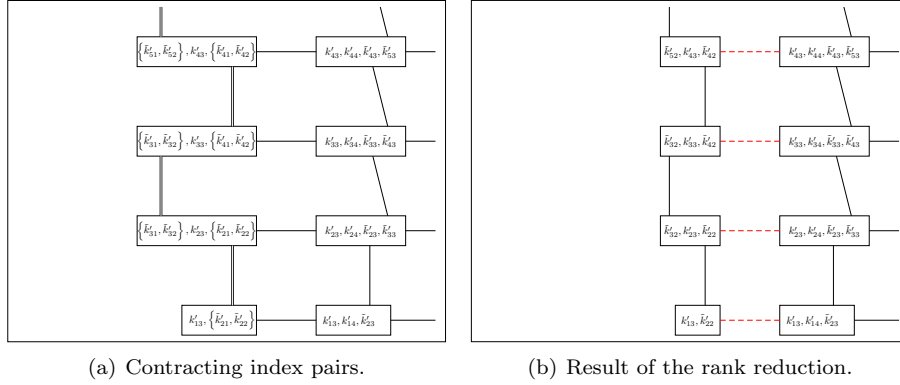


Figure 6: Contraction scheme for PEPS vectors: after the contracting illustrated in Fig. 5(b), The newly generated first column in (a) has sets of indices leading to a more complex contraction process as illustrated by the double lines. After the rank reduction (b), we are in the same situation as in Fig. 5(b) and can thus proceed in the same manner.

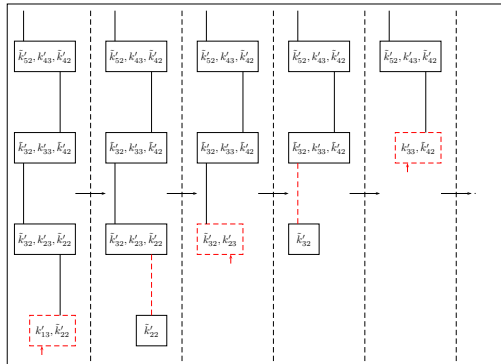


Figure 7: After application of the column-by-column contraction scheme, we end up with the last column, which can be contracted in the illustrated way.

The overall costs depend on D^{10} , resp. D^{20} in the periodic case. The so-called virtual dimension D is usually chosen a priori by the physicists (e.g. $D \leq 5$), such that the computations can still be afforded ([45]).

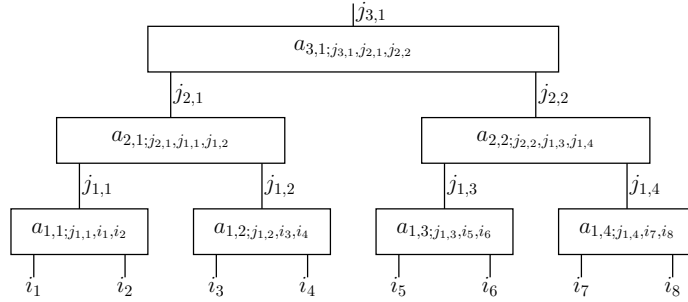


Figure 8: Scheme of the Tree Tensor States (TTS).

3.3. Tree Tensor States and Multi-scale Entanglement Renormalization Ansatz

In MPS the new spin particles are added one by one leading to a new matrix in the tensor ansatz. A modified procedure leads to *Tree Tensor States* (TTS). In this ansatz the particles are grouped pairwise leading to a smaller subspace of the original Hilbert space. This procedure is repeated with the blockpairs until only one block is left. This is displayed in Figure 8 and formula

$$x_{i_1, \dots, i_8} = \sum_{j_{1,1}, \dots, j_{2,2}} (a_{1,1;j_{1,1},i_1,i_2} \cdots a_{1,4;j_{1,4},i_7,i_8}) (a_{2,1;j_{2,1},j_{1,1},j_{1,2}} a_{2,2;j_{2,2},j_{1,3},j_{1,4}}) (a_{3,1;j_{3,1},j_{2,1},j_{2,2}})$$

where each block is isometric:

$$\sum_{i,j} \bar{a}_{k',i,j} \cdot a_{k,i,j} = \delta_{k',k}.$$

Hence, in this scheme the network consists in a binary tree built by small isometric tensors with three indices. It can be shown that all MPS can be represented by TTS (see [51]).

A further generalization of TTS and MPS is given by the *Multi-Scale Entanglement Renormalization Ansatz* (MERA, see [28]). Besides the top tensor t with two indices, the network is built by two types of smaller tensors: three leg isometric tensors (*isometries*) and four leg unitary tensors (*disentangler*s). This formalism is displayed by Figure 9. In connection with eigenvalue approximation for MERA, ALS cannot be used in order to optimize over the isometries that represent the degrees of freedom; instead other optimization methods have to be applied.

4. Representations of Tensors

As any state in a physical system (see Sec. 2) with p particles may be seen as a p -th order tensor, we want to present some basics about tensors within this section. In the next section, we will give some modifications and generalizations.

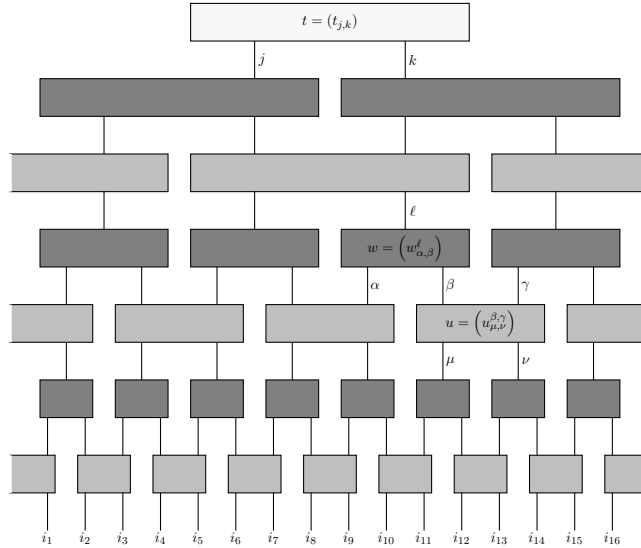


Figure 9: The Multi-scale Entanglement Renormalization Ansatz (MERA) with periodic boundary conditions, consisting of isometries w and unitaries u .

4.1. Some Basics about Tensors

In its most general form, a p th-order tensor $A = (a_{i_1, \dots, i_p}) \in \mathbb{R}^{n_1 \times \dots \times n_p}$ is a multiway array with p indices. A first-order tensor is thus a vector, a second-order tensor corresponds to a matrix. If \mathbf{x} and \mathbf{y} are vectors (i.e. first-order tensors) it is well known that the *outer product* $\mathbf{x} \circ \mathbf{y} := \mathbf{x}\mathbf{y}^T$ is a rank-one matrix. As generalization, if $\mathbf{a}_1, \dots, \mathbf{a}_p$ are vectors, the tensor $A := \mathbf{a}_1 \circ \dots \circ \mathbf{a}_p$, which is defined as $a_{i_1, \dots, i_p} = a_{1; i_1} a_{2; i_2} \dots a_{p; i_p}$ is a *rank-one tensor*. Hence, the application of outer products constructs tensors of higher order. The Kronecker product of matrices just corresponds to this definition of the outer product, but it reshapes the single tensor entries into a matrix of larger size.

To begin, we write a tensor A as a sum of rank-one tensors:

$$A = \sum_{j=1}^R \mathbf{a}_1^{(j)} \circ \mathbf{a}_2^{(j)} \circ \dots \circ \mathbf{a}_p^{(j)}. \quad (24)$$

If R is minimal in the representation (24) of A , we define the *tensor rank* of A as equal to R .

For illustrating tensors, matrices are often in use. One possibility to bring back a general tensor to a matrix is given by the *mode- n -unfolding* (see [52]): When applying this technique, the tensor $A = (a_{i_1, \dots, i_n, \dots, i_p})$ is represented by the matrix

$$A_{(n)} := (a_{i_1, \dots, i_n, \dots, i_p})_{i_n, \{i_1, \dots, i_{n-1}, i_{n+1}, \dots, i_p\}}.$$

The *n-mode tensor matrix product* is given as follows: Let $A = (a_{i_1, \dots, i_n, \dots, i_p})$ be a p -th order tensor and $U = (u_{j, i_n})$ a matrix, then the *mode- n -product* \times_n is defined as

$$A \times_n U = \left(\sum_{i_n} a_{i_1, \dots, i_n} u_{j, i_n} \right)_{i_1, \dots, i_{n-1}, j, i_{n+1}, \dots, i_p} .$$

Beside the total rank of a tensor there also exist a local rank concept: the *mode- n -rank* of a tensor is the rank of the collection of all vectors belonging to index n . If a given tensor A has the n -mode ranks r_n , we define the *multilinear rank* of A as (r_1, \dots, r_p) .

4.2. Decomposition of Tensors

For approximating tensors (x_{i_1, \dots, i_p}) there are two basic methods (see [52]): the *Tucker decomposition* and the *canonical decomposition*. The Tucker decomposition ([53])

$$x_{i_1, \dots, i_p} = \sum_{m_1, \dots, m_p}^D y_{m_1, \dots, m_p} a_{1; i_1, m_1} a_{2; i_2, m_2} \cdots a_{p; i_p, m_p} \quad (25)$$

represents the given tensor by a tensor y_{m_1, \dots, m_p} with less dimension in each direction; D is called the *Tucker rank*, y_{m_1, \dots, m_p} is called *core tensor*. This concept is illustrated in Figure 10(a). In the case of a binary tensor $\mathbf{x} \in \mathbb{R}^{2 \times \dots \times 2}$, this is not meaningful, because then the Tucker rank is already 2.

The canonical decomposition (candecomp), which is also known as Parallel Factorization (PARAFAC), has the form

$$x_{i_1, \dots, i_p} = \sum_{s=1}^D a_{1; i_1}^{(s)} a_{2; i_2}^{(s)} \cdots a_{p; i_p}^{(s)} . \quad (26)$$

Hence, the PARAFAC decomposes the given tensor into a sum of rank one tensors (see [54, 55]), which is illustrated by Figure 10(b). If we think of \mathbf{x} as a vector this is equivalent to

$$\mathbf{x} = \sum_{s=1}^D \mathbf{a}_1^{(s)} \otimes \cdots \otimes \mathbf{a}_p^{(s)}$$

with tensor products of smaller vectors. One often finds the normalized form

$$\mathbf{x} = \sum_{s=1}^D \lambda_s \mathbf{a}_1^{(s)} \otimes \cdots \otimes \mathbf{a}_p^{(s)} \quad (27)$$

with vectors $\mathbf{a}_i^{(s)}$ of norm 1.

If D is minimal in the representation (26) of \mathbf{x} , it is called the *canonical rank*.

Tensor Train Schemes

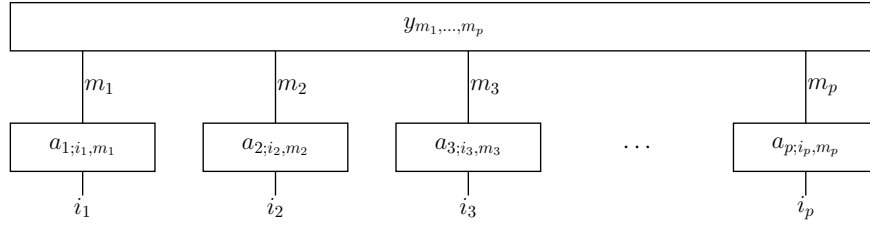
Application of the concepts (25) or (26) would be a generalization of SVD that allows a substantial reduction in number of parameters for deriving a good approximation for a given tensor. Unfortunately, these decompositions have disadvantages like still exponential growth, lack of robust algorithms for rank reduction. Oseledets and Tyrtyshnikov ([56]) proposed the following Tensor Train scheme (TT) as an attempt to overcome these difficulties. In a first step a dyadic decomposition is introduced by cutting the index set and the tensor in two parts introducing an additional summation with a newly introduced ancilla index m :

$$x_{i_1, \dots, i_k; i_{k+1}, \dots, i_p} = \sum_m a_{1; i_1, \dots, i_k, m} \cdot a_{2; i_{k+1}, \dots, i_p, m}. \quad (28)$$

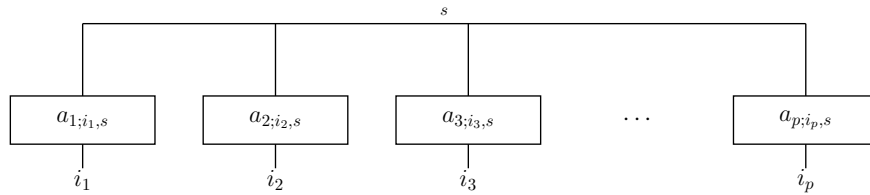
This is the first step of the Tree Tucker [55] decomposition. Now we may apply this process recursively to a_1 and a_2 . If we always use the index partitioning $i_j; i_{j+1}, \dots, i_p$, we arrive at the TT format

$$x_{i_1, \dots, i_p} = \sum_{m_2, \dots, m_p} a_{1; i_1, m_2} \cdot a_{2; i_2, m_2, m_3} \cdot \dots \cdot a_{p; i_p, m_p}. \quad (29)$$

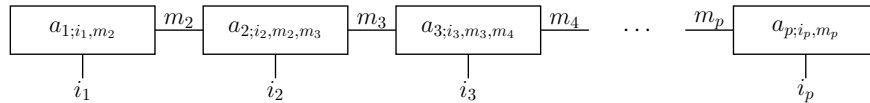
Again we can distinguish between given, physical indices i_j and ancilla indices. Figure 10(c) illustrates the TT decomposition concept.



(a) The Tucker decomposition scheme.



(b) The canonical decomposition scheme.



(c) The Tensor Train decomposition scheme

Figure 10: Tensor decomposition schemes

The TT scheme (29) is exactly the MPS form (9) for open boundary conditions:

$$x_{i_1, \dots, i_p} = G_1^{(i_1)} G_2^{(i_2)} \dots G_p^{(i_p)} \quad (30)$$

with matrices $G_j^{(i_j)}$ of size $D_j \times D_{j+1}$, ($D_1 = D_{p+1} = 1$), where the matrix sizes differ and are called the *compression rank*.

In [57], Khoromskij generalizes the TT concept to *Tensor Chains* via the definition

$$x_{i_1, \dots, i_p} = \sum_{m_1, \dots, m_p} a_{1; i_1, m_1, m_2} a_{2; i_2, m_2, m_3} \dots a_{p; i_p, m_p, m_1}, \quad (31)$$

which corresponds to MPS with periodic boundary conditions, i.e.

$$x_{i_1, \dots, i_p} = \text{tr} \left(G_1^{(i_1)} G_2^{(i_2)} \dots G_p^{(i_p)} \right). \quad (32)$$

Starting with Formula (28) this step can also be applied recursively to the two newly introduced tensors (a_1) and (a_2) with any cutting point. Cutting a tensor in two parts should introduce as many ancilla indices as described by the tensor network connecting the two parts. So in 1D there is only one neighboring connection that is broken up and introduces 1 additional index.

In the 2D network described in Figure 4 we could apply this method by cutting in a first step the down left knot i_{11} , thus introducing two ancilla indices labeled with $(i_{1,1}, i_{1,2})$, resp. $(i_{1,1}, i_{2,1})$, representing the broken connections to the neighbors of $i_{1,1}$. Proceeding in this way knot by knot leads to the PEPS form. But generalizing the approach we can also consider general cuts of region R of the tensor network in two regions R_1 and R_2 , replacing the given tensor by a sum over a tensor product of two smaller tensors with indices related to R_1 , resp. R_2 , introducing as many ancilla indices as broken connections in the cut.

5. Modifications of PARAFAC and MPS Representations

In this section we want to introduce some own ideas concerning the generalization and modification of both the PARAFAC decomposition scheme (see 4.2) and the MPS (TT) format. These newly presented formats will be used as vector ansatz for the minimization of the Rayleigh quotient (6). As we have pointed out in Section 2.3, we are looking for representation schemes that allow both proper approximation properties and efficient computations such as fast contractions for inner product calculations and cheap matrix-vector evaluations. In view of these requirements we present both theoretical results concerning the computational complexity and numerical results showing the benefit of our concepts.

At this point we want to emphasize that our problem does not consist in the decomposition of a known tensor but to find an appropriate ansatz for vectors and to formulate algorithms for the ground state computation working

on such vector representations. Hence, in this context we focus on algorithmic considerations and investigate, which modifications and generalizations of the presented concepts are still affordable to solve physical problems, where we a priori work with approximations with ranks which are given by the physicists. It turns out that the usage of low-rank approaches suffices to give proper results, compare, e.g., [58].

5.1. PARAFAC Formats

Any state $\mathbf{x} \in \mathbb{C}^{2^p}$ of a physical system can be tensorized artificially in several ways. In the easiest way we may rewrite \mathbf{x} as a p th ordered binary tensor of the form

$$\mathbf{x} = (x_{i_1, \dots, i_p})_{i_\ell=0,1} . \quad (33)$$

But we may also define blockings of larger size, which will follow the interactions of the physical system (e.g. nearest-neighbor interaction). These blocking concepts introduce formats with a larger number of degree of freedoms promising better approximation properties but still allow efficient computations of matrix-vector and inner products. For reproducing the physical structure such as internal and external interactions we will also allow formats with different overlapping blockings in the addends, compare Subsection 5.3.

Such blockings of indices can be seen as tensorizations of lower order $q \leq p$:

$$\mathbf{x} = (x_{(i_1, \dots, i_{s_1}), \dots, (i_{s_{q-1}+1}, \dots, i_{s_q})}) = (x_{j_1, \dots, j_q}) . \quad (34)$$

where the k th mode combines $t_k := s_k - s_{k-1}$ binary indices and has therefore index $j_k = (i_{s_{k-1}+1}, \dots, i_{s_k})$ of size 2^{t_k} (for reasons of consistency we define $s_0 = 0$ and $s_q = p$).

One way to find a convenient representation is to consider appropriate decompositions of the tensorization (34). In this context we don't consider the Tucker format (25) as it is only meaningful to decompose very large mode sizes (meaning large sets of blocked indices).

Hence, from now on we consider the PARAFAC concept and choose the ansatz vector to be a sum of rank-1 tensors (see Figure 11). In the simplest case (33), the PARAFAC decomposition takes the form

$$\mathbf{x} = \sum_{s=1}^D \mathbf{x}_1^{(s)} \otimes \dots \otimes \mathbf{x}_p^{(s)} = \sum_{s=1}^D \begin{pmatrix} \alpha_1^{(s)} \\ \beta_1^{(s)} \end{pmatrix} \otimes \dots \otimes \begin{pmatrix} \alpha_p^{(s)} \\ \beta_p^{(s)} \end{pmatrix} , \quad (35)$$

a sum of tensor products of length 2-vectors. In view of (11), the decomposition (35) can be seen as a special MPS form. Indeed, every PARAFAC representation (35) with D addends corresponds to an MPS term with $D \times D$ diagonal matrices. This fact becomes clear from the construction

$$A_r^{(0)} = \begin{pmatrix} \alpha_r^{(1)} & & & \\ & \ddots & & \\ & & \alpha_r^{(D)} & \end{pmatrix}, \quad A_r^{(1)} = \begin{pmatrix} \beta_r^{(1)} & & & \\ & \ddots & & \\ & & & \beta_r^{(D)} \end{pmatrix} .$$

More generally, the PARAFAC scheme for the tensorization (34) leads to the ansatz

$$\mathbf{x} = \sum_{\ell=1}^D \mathbf{x}_1^{(\ell)} \otimes \cdots \otimes \mathbf{x}_q^{(\ell)}. \quad (36)$$

with vectors $\mathbf{x}_j^{(\ell)}$ of moderate size 2^{t_j} . Figure 11 illustrates such a decomposition.

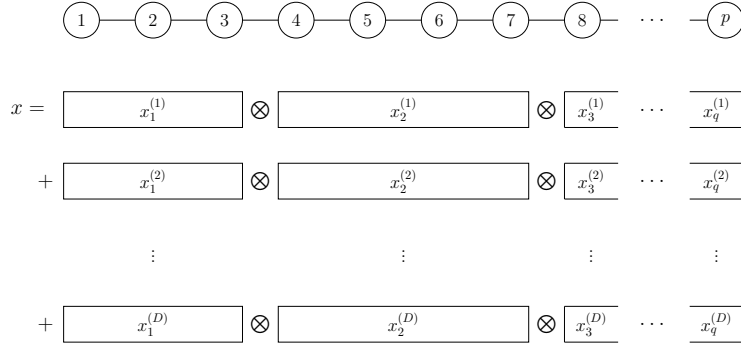


Figure 11: PARAFAC ansatz for a chosen tensorization of the vector \mathbf{x} to be represented.

Computational Costs

Inner product calculations of two representations of the form (36) reduce to the inner product of the block vectors:

$$\mathbf{y}^H \mathbf{x} = \sum_{\ell, \ell'=1}^D (\mathbf{y}_1^{(\ell)H} \mathbf{x}_1^{(\ell')}) (\mathbf{y}_2^{(\ell)H} \mathbf{x}_2^{(\ell')}) \cdots (\mathbf{y}_q^{(\ell)H} \mathbf{x}_q^{(\ell')}). \quad (37)$$

Therefore, the total costs for each of the D^2 inner products are

$$2(2^{t_1} + 2^{t_2} + \cdots + 2^{t_q}) + q \leq q(2l + 1) \leq p(2l + 1)$$

where $t := \max\{t_1, t_2, \dots, t_q\}$ is the maximum number of grouped indices and thus $l = 2^t$ denotes the largest block size in \mathbf{x} . If all q index sets have the same size t (i.e. $t = p/q$), the costs can be bounded by $q(2 \cdot 2^{p/q} + 1)$.

To compute the matrix-vector product efficiently, we group the Hamiltonian H (2) in the same way, i.e.

$$\begin{aligned} H &= \sum_{k=1}^M \alpha_k (Q_1^{(k)} \otimes \cdots \otimes Q_{s_1}^{(k)}) \otimes \cdots \otimes (Q_{s_{q-1}+1}^{(k)} \otimes \cdots \otimes Q_p^{(k)}) \\ &= \sum_{k=1}^M \alpha_k H_1^{(k)} \otimes \cdots \otimes H_q^{(k)}. \end{aligned}$$

For the matrix-vector product we thus obtain

$$\begin{aligned}
H\mathbf{x} &= \left(\sum_{k=1}^M \alpha_k H_1^{(k)} \otimes \cdots \otimes H_q^{(k)} \right) \left(\sum_{\ell=1}^D \mathbf{x}_1^{(\ell)} \otimes \cdots \otimes \mathbf{x}_q^{(\ell)} \right) \\
&= \sum_{k=1}^M \sum_{\ell=1}^D \left(\alpha_k H_1^{(k)} \mathbf{x}_1^{(\ell)} \right) \otimes \cdots \otimes \left(H_q^{(k)} \mathbf{x}_q^{(\ell)} \right) \\
&= \sum_{k=1}^M \sum_{\ell=1}^D \mathbf{y}_1^{(k,\ell)} \otimes \cdots \otimes \mathbf{y}_q^{(k,\ell)}.
\end{aligned} \tag{38}$$

However, typically we do not need the matrix-vector products (38) explicitly, we only require them for inner products of the form $\mathbf{y}^H(H\mathbf{x})$ (compare the nominator of the Rayleigh quotient)

$$\mathbf{y}^H H\mathbf{x} = \sum_{k=1}^M \alpha_k \sum_{\ell, \ell'=1}^D \left(\mathbf{y}_1^{(\ell')}^H H_1^{(k)} \mathbf{x}_1^{(\ell)} \right) \cdots \left(\mathbf{y}_q^{(\ell')}^H H_q^{(k)} \mathbf{x}_q^{(\ell)} \right). \tag{39}$$

The products $H_i^{(k)} \mathbf{x}_i^{(\ell)}$ can be computed implicitly without constructing the matrices $H_i^{(k)}$ explicitly. Each of these small matrix-vector products can be computed linearly in the size of $H_i^{(k)}$ (i.e. 2^{t_i}). Thus, the total costs for each addend in the inner product (39) are $3(2^{t_1} + 2^{t_2} + \cdots + 2^{t_q}) + q$ and can again be bounded by $q(3 \cdot 2^{p/q} + 1)$ in the case of equal block sizes. Hence, the costs for (39) are $2MD^2q(3 \cdot 2^{p/q} + 1)$.

Using PARAFAC for the Ground State Problem

Let us now apply the PARAFAC approach (36) as ansatz for the Rayleigh quotient minimization. Then, Eq. 6 reads

$$\min \frac{\left(\sum_{\ell=1}^D \mathbf{x}_1^{(\ell)} \otimes \cdots \otimes \mathbf{x}_q^{(\ell)} \right)^H H \left(\sum_{\ell=1}^D \mathbf{x}_1^{(\ell)} \otimes \cdots \otimes \mathbf{x}_q^{(\ell)} \right)}{\left(\sum_{\ell=1}^D \mathbf{x}_1^{(\ell)} \otimes \cdots \otimes \mathbf{x}_q^{(\ell)} \right)^H \left(\sum_{\ell=1}^D \mathbf{x}_1^{(\ell)} \otimes \cdots \otimes \mathbf{x}_q^{(\ell)} \right)}. \tag{40}$$

This minimization task can be realized by an ALS-based procedure.

In a first way we could think about the following proceeding: We start with a PARAFAC representation of rank 1 ($D = 1$), optimize it via ALS and then we successively add one summand and optimize it in an ALS-based way. In the first stage we would have to minimize

$$\min_{\mathbf{x}} \frac{\mathbf{x}^H H \mathbf{x}}{\mathbf{x}^H \mathbf{x}} = \min_{\mathbf{x}_1, \dots, \mathbf{x}_q} \frac{(\mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \cdots \otimes \mathbf{x}_q)^H H (\mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \cdots \otimes \mathbf{x}_q)}{(\mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \cdots \otimes \mathbf{x}_q)^H (\mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \cdots \otimes \mathbf{x}_q)}.$$

This optimization problem can be solved via ALS: considering all of the \mathbf{x}_j up to \mathbf{x}_i as fixed, we obtain

$$\min_{\mathbf{x}_i} \frac{(\mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_i \otimes \cdots \otimes \mathbf{x}_q)^H H (\mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_i \otimes \cdots \otimes \mathbf{x}_q)}{(\mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_i \otimes \cdots \otimes \mathbf{x}_q)^H (\mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_i \otimes \cdots \otimes \mathbf{x}_q)}. \tag{41}$$

Following (39) and (37), we may contract all indices up to i and obtain

$$\begin{aligned} & \min_{\mathbf{x}_i} \frac{\sum_{k=1}^M \alpha_k (\mathbf{x}_1^H H_1^{(k)} \mathbf{x}_1) \cdots (\mathbf{x}_i^H H_i^{(k)} \mathbf{x}_i) \cdots (\mathbf{x}_q^H H_q^{(k)} \mathbf{x}_q)}{(\mathbf{x}_1^H \mathbf{x}_1) \cdots (\mathbf{x}_i^H \mathbf{x}_i) \cdots (\mathbf{x}_q^H \mathbf{x}_q)} \\ &= \min_{\mathbf{x}_i} \frac{\sum_{k=1}^M \alpha_k \beta_k \mathbf{x}_i^H H_i^{(k)} \mathbf{x}_i}{\gamma (\mathbf{x}_i^H \mathbf{x}_i)} = \min_{\mathbf{x}_i} \frac{\mathbf{x}_i^H \left(\sum_{k=1}^M \frac{\alpha_k \beta_k}{\gamma} H_i^{(k)} \right) \mathbf{x}_i}{\mathbf{x}_i^H \mathbf{x}_i}, \end{aligned}$$

a standard eigenvalue problem for a matrix of size $2^{t_i} \times 2^{t_i}$ that can be solved via classical iterative methods which only require the computation of matrix vector products. These products can be executed implicitly without constructing the matrices explicitly.

Now we suppose that we have already optimized $D - 1$ addends in the representation (36) and we want to find the next optimal addend. This means that the ansatz vector is now of the form

$$\mathbf{x} = \underbrace{\mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_q}_{= \bigotimes_{j=1}^q \mathbf{x}_j} + \sum_{\ell=1}^{D-1} \underbrace{\mathbf{y}_1^{(\ell)} \otimes \cdots \otimes \mathbf{y}_q^{(\ell)}}_{=: \mathbf{y}^{(\ell)}}.$$

with already optimized $\mathbf{y}_j^{(\ell)}$ -terms and vectors \mathbf{x}_i that have to be optimized via ALS. Contracting over all terms up to \mathbf{x}_i (compare Eq. 39) we obtain

$$\mathbf{x}^H H \mathbf{x} = \mathbf{x}_i^H H_i \mathbf{x}_i + \mathbf{u}_i^H \mathbf{x}_i + \mathbf{x}_i^H \mathbf{u}_i + \beta,$$

where \mathbf{u}_i and β comprise the contractions with the \mathbf{y}_j terms. For the denominator we analogously obtain

$$\mathbf{x}^H H \mathbf{x} = \mathbf{x}_i^H \gamma I \mathbf{x}_i + \mathbf{v}_i^H \mathbf{x}_i + \mathbf{x}_i^H \mathbf{v}_i + \rho.$$

Altogether we have to solve the generalized eigenvalue problem

$$\min_{\mathbf{x}_i} \frac{\begin{pmatrix} \mathbf{x}_i^H & 1 \end{pmatrix} \begin{pmatrix} H_i & \mathbf{u}_i \\ \mathbf{u}_i^H & \beta \end{pmatrix} \begin{pmatrix} \mathbf{x}_i \\ 1 \end{pmatrix}}{\begin{pmatrix} \mathbf{x}_i^H & 1 \end{pmatrix} \begin{pmatrix} \gamma I & \mathbf{v}_i \\ \mathbf{v}_i^H & \rho \end{pmatrix} \begin{pmatrix} \mathbf{x}_i \\ 1 \end{pmatrix}} \quad (42)$$

It turns out that the denominator matrix can be factorized via a Cholesky factorization. Therefore we have to solve a standard eigenvalue problem of moderate size.

However, the proposed procedure may cause problems which result from the fact that, for general tensors, the best rank- D approximation does not have to comprise the best rank- $(D - 1)$ approximation, see, e.g., [52]. Our numerical results (Figure 12) will approve this fact.

For this reason we now consider a technique where we use ALS to optimize all the blocks related to the same mode i simultaneously. The minimization task takes the form

$$\min_{\mathbf{x}_i} \frac{\left(\sum_{\ell=1}^D \mathbf{x}_1^{(\ell)} \otimes \cdots \otimes \mathbf{x}_i^{(\ell)} \otimes \cdots \otimes \mathbf{x}_q^{(\ell)} \right)^H H \left(\sum_{\ell=1}^D \mathbf{x}_1^{(\ell)} \otimes \cdots \otimes \mathbf{x}_i^{(\ell)} \otimes \cdots \otimes \mathbf{x}_q^{(\ell)} \right)}{\left(\sum_{\ell=1}^D \mathbf{x}_1^{(\ell)} \otimes \cdots \otimes \mathbf{x}_i^{(\ell)} \otimes \cdots \otimes \mathbf{x}_q^{(\ell)} \right)^H \left(\sum_{\ell=1}^D \mathbf{x}_1^{(\ell)} \otimes \cdots \otimes \mathbf{x}_i^{(\ell)} \otimes \cdots \otimes \mathbf{x}_q^{(\ell)} \right)}$$

Contracting all terms in the nominator (compare Eq. 39) up to the $\mathbf{x}_i^{(\ell)}$ vectors results in

$$\begin{aligned} & \sum_{k=1}^M \alpha_k \sum_{\ell, \ell'=1}^D \left(\mathbf{x}_1^{(\ell')}^H H_1^{(k)} \mathbf{x}_1^{(\ell)} \right) \cdots \left(\mathbf{x}_i^{(\ell')}^H H_i^{(k)} \mathbf{x}_i^{(\ell)} \right) \cdots \left(\mathbf{x}_q^{(\ell')}^H H_q^{(k)} \mathbf{x}_q^{(\ell)} \right) \\ &= \sum_{\ell, \ell'=1}^D \mathbf{x}_i^{(\ell')}^H \tilde{H}_i^{(\ell', \ell)} \mathbf{x}_i^{(\ell)}. \end{aligned}$$

For the denominator, we analogously obtain

$$\sum_{\ell, \ell'=1}^D \mathbf{x}_i^{(\ell')}^H (\beta_{\ell', \ell} I) \mathbf{x}_i^{(\ell)}.$$

Altogether, the approach leads to the minimization problem

$$\min_{\mathbf{x}_i} \frac{\begin{pmatrix} \mathbf{x}_i^{(1)H}, \dots, \mathbf{x}_i^{(D)H} \end{pmatrix} \begin{pmatrix} \tilde{H}_i^{(1,1)} & \dots & \tilde{H}_i^{(1,D)} \\ \vdots & \ddots & \vdots \\ \tilde{H}_i^{(D,1)} & \dots & \tilde{H}_i^{(D,D)} \end{pmatrix} \begin{pmatrix} \mathbf{x}_i^{(1)} \\ \vdots \\ \mathbf{x}_i^{(D)} \end{pmatrix}}{\begin{pmatrix} \mathbf{x}_i^{(1)H}, \dots, \mathbf{x}_i^{(D)H} \end{pmatrix} \begin{pmatrix} \beta_{1,1} I & \dots & \beta_{1,D} I \\ \vdots & \ddots & \vdots \\ \beta_{D,1} I & \dots & \beta_{D,D} I \end{pmatrix} \begin{pmatrix} \mathbf{x}_i^{(1)} \\ \vdots \\ \mathbf{x}_i^{(D)} \end{pmatrix}}, \quad (43)$$

a generalized eigenvalue problem of size $D2^{t_i} \times D2^{t_i}$. This approach requires the solution of larger eigenproblems, but by increasing the set of variables to be optimized it turns out that we require less optimization steps and obtain better convergence results (see Figure 12).

Formulation of an Algorithm

As possibility to choose the initial guesses, we propose to set $\{\mathbf{x}_i^{(1)}, \dots, \mathbf{x}_i^{(D)}\}$ to D linearly independent eigenvectors of

$$H_i = \sum_{k=1}^M Q_{s_{i-1}+1}^{(k)} \otimes \cdots \otimes Q_{s_i}^{(k)}.$$

Algorithm 1 Rayleigh Quotient Minimization in the PARAFAC model

- 1: Provide initial guess for all vectors $\mathbf{x}_i^{(\ell)}$ ($i = 1, \dots, q$ and $\ell = 1, \dots, D$)
 - 2: **while** Not yet converged **do**
 - 3: **for** $i = 1, \dots, q$ **do**
 - 4: Compute all contractions in $\mathbf{x}^H H \mathbf{x}$ and $\mathbf{x}^H \mathbf{x}$ up to index i
 - 5: Solve the generalized eigenvalue problem (43)
 - 6: **end for**
 - 7: Normalize each addend in the PARAFAC representation (27)
 - 8: **end while**
-

As stopping criterion we may define a maximum number of iterations or we could specify a stopping criterion based on the degree of improvement. One possibility to accelerate the ALS-based optimization procedure is to apply the Enhanced Linesearch method [59], which is not considered here.

Numerical Results

After these technical derivations, we want to show first numerical results. We computed the ground state of a 10 spin and a 12 spin Ising-type Hamiltonian (3) using different blocking schemes $[t_1, \dots, t_q]$. The ALS-based Rayleigh quotient minimization was performed in two different ways: a one-by-one minimization following (42) and a simultaneous minimization described by Algorithm 1. It turns out that the results are getting better when using larger vector blocks. This consideration is based on the fact that using larger blocks leads to a larger number of degrees of freedom. Figure 12 depicts the effect of using different PARAFAC representations and shows the benefit resulting from the simultaneous optimization.

5.2. Block MPS

In the block MPS ansatz we group single particles together and apply MPS concepts to these grouped blocks. The MPS ansatz (9) is based on using 2 matrices $A_j^{(i_j)}$ at position j . A first generalization of MPS vectors can be derived by blocking indices in subgroups. So we can combine indices (i_1, \dots, i_r) to index j_1 , and indices (i_{r+1}, \dots, i_p) to index j_2 . Then the MPS vector to this blocking with open boundary conditions would give

$$\mathbf{x}_i = \mathbf{x}_{i_1, \dots, i_p} = \mathbf{x}_{\{i_1, \dots, i_r\}, \{i_{r+1}, \dots, i_p\}} = \mathbf{x}_{j_1, j_2} = A_1^{(j_1)} \cdot A_2^{(j_2)}$$

or vector

$$\mathbf{x} = \sum_{m_2=1}^D \mathbf{a}_{1;1,m_2} \otimes \mathbf{a}_{2;m_2,1} \quad (44)$$

with D pairs of short vectors of length 2^r , resp. 2^{p-r} . Compared to standard MPS we have different numbers for the degree of freedom (DOF), and different costs for inner products. MPS combines in each term p vectors of length 2 in a

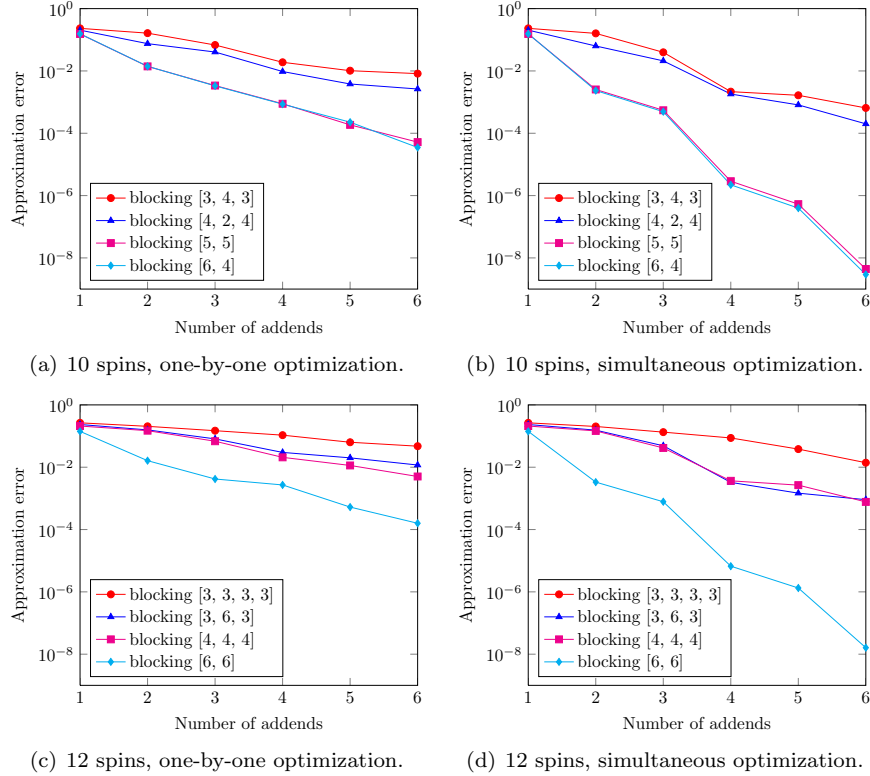


Figure 12: Approximation error for the computation of the ground state energy of a 10 spin and 12 spin Ising-type Hamiltonian.

tensor product and combines D^p vectors; thus, it has $2pD^2$ DOF's and $2pD^3$ costs for inner products. The 2-term block coarse grain MPS form (44) combines in each term 2 vectors of total length 2^r and 2^{p-r} . Therefore, it has $D(2^r + 2^{p-r})$ DOF's and the estimate for the inner product costs is $D^2(2^r + 2^{p-r})$. The computation of the full inner product reduces to D^2 inner products of short vectors at position 1, resp. 2:

$$\begin{aligned}
 \mathbf{x}^H \mathbf{x} &= \sum_{m'_2}^D \mathbf{a}_{1;1,m'_2}^H \otimes \mathbf{a}_{2;m'_2,1}^H \cdot \sum_{m_2}^D \mathbf{a}_{1;1,m_2} \otimes \mathbf{a}_{2;m_2,1} \\
 &= \sum_{m_2, m'_2} (\mathbf{a}_{1;1,m'_2}^H \mathbf{a}_{1;1,m_2}) \cdot (\mathbf{a}_{2;m'_2,1}^H \mathbf{a}_{2;m_2,1}) .
 \end{aligned}$$

For using three blocks we derive

$$x_i = x_{\{i_1, \dots, i_{r_1}\}, \{i_{r_1+1}, \dots, i_{r_2}\}, \{i_{r_2+1}, \dots, i_p\}} = x_{j_1, j_2, j_3} = A_1^{(j_1)} A_2^{(j_2)} A_3^{(j_3)}$$

or vector

$$\mathbf{x} = \sum_{m_2, m_3}^D \mathbf{a}_{1;1,m_2} \otimes \mathbf{a}_{2;m_2,m_3} \otimes \mathbf{a}_{3;m_3,1}$$

with $D2^{t_1} + D^22^{t_2} + D2^{t_3}$ DOF's, $t_1 = r_1, t_2 = r_2 - r_1, t_3 = p - r_2$, combining D^2 long vectors, and costs of order D^4 for an inner product in view of

$$\mathbf{x}^H \mathbf{x} = \sum_{\substack{m_2, m_3 \\ m'_2, m'_3}} (\mathbf{a}_{1;1,m'_2})^H \mathbf{a}_{1;1,m_2} \cdot (\mathbf{a}_{2;m'_2,m'_3})^H \mathbf{a}_{2;m_2,m_3} \cdot (\mathbf{a}_{3;m'_3,1})^H \mathbf{a}_{3;m_3,1}.$$

By using the MPS contraction scheme (Figures 2(a) – 3(d)), the costs for the inner product can be reduced to $D^3(2^{t_1} + 2^{t_2} + 2^{t_3})$.

In general, the Block MPS ansatz with k blocks gives

$$x_i = x_{i_1, \dots, i_p} = x_{j_1, \dots, j_k} = A_1^{(j_1)} \cdot A_2^{(j_2)} \cdot \dots \cdot A_{k-1}^{(j_{k-1})} \cdot A_k^{(j_k)} \quad (45)$$

or vector

$$\mathbf{x} = \sum_{m_2, \dots, m_k} \mathbf{a}_{1;1,m_2} \otimes \mathbf{a}_{2;m_2,m_3} \otimes \dots \otimes \mathbf{a}_{k-1;m_{k-1},m_k} \otimes \mathbf{a}_{k;m_k,1}. \quad (46)$$

DOF's: $D(2^{t_1} + 2^{t_k}) + D^2(2^{t_2} + \dots + 2^{t_{k-1}})$ for $t_1 = r_1, t_k = r_k - r_{k-1}$.

Costs: $D^3(2^{t_1} + \dots + 2^{t_k})$ using the MPS contraction scheme,

Combination of D^{k-1} full vectors.

Altogether, the Block MPS ansatz causes greater effort for the computation of inner products, but the increasing number of degrees of freedom may allow better approximation properties. The Block MPS format corresponds to the Tensor Train format with mode sizes $2^{t_i}, i = 1, \dots, k$.

In summary, MPS can be considered as a fine grain method, using many vectors, while the coarse grain MPS uses only a sum of tensor products of a few large vectors respectively. The fine grain MPS uses at each position s a matrix pair $A_s^{(i_s)}, i_s = 0, 1$, while the k -form (45) uses 2^{t_s} matrices $A_s^{(j_s)}, j_s = 0, 1, \dots, 2^{t_s} - 1$. In the same way the length of vector $\mathbf{a}_{s;m_s,m_{s+1}}$ is 2^{t_s} .

5.3. Mixed Blockings in the PARAFAC Ansatz

In this section, we want to analyze mixed tensor representations similar to PARAFAC concepts presented in Section 5.1 but allowing different blockings in the addends (see Figure 13). This may also be interesting for mixing, e.g., Block-MPS vectors with different block structure (compare Subsection 5.4).

Reasoning for mixed blockings: In connection with PARAFAC or MPS, approximations of a given vector as a sum of tensor products of smaller vectors (such as (11) or (50)) which are of different sizes might give different approximation qualities. E.g. using a partitioning with two vectors is related to a certain matricization of the given vector. The matricization leads to an SVD with related singular values of certain decay behavior and therefore a certain truncation error depending on the number of singular values that are used for

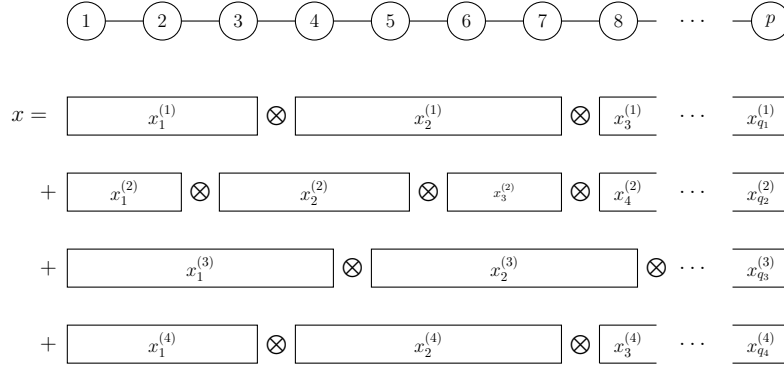


Figure 13: A PARAFAC-like representation. In contrast to the decomposition scheme illustrated by Figure 11 different blockings are possible.

the approximation. Hence, it might be useful to combine different matricizations in PARAFAC allowing tensor products of different partitionings.

In the following we want to discuss the question which blockings allow efficient contractions similar to the case of uniform blockings. Using such mixed schemes for finding ground states also leads to generalized eigenproblems similar to (42,43), but now the computation of inner products is generally more costly. In contrast, the application of the Hamiltonian (2) to a mixed vector representation can again be computed efficiently because the Hamiltonians are given by Kronecker products of 2×2 matrices. The question here is, what type of blocking allows fast and efficient contractions in inner products for vectors with different tensor blocking.

In the following we will consider the problem what tensor aware permutations can be combined that allow fast and efficient contractions. Such permutations allow a generalization of PARAFAC, where in each approximation step a new tensor aware permutation can be applied before the new term is computed.

So in the first, elementary case we consider

$$\begin{aligned}
 \mathbf{z} = (z_{i_1, \dots, i_p}) = & (x_{1; i_1, \dots, i_{r_1}}) \otimes \dots \otimes (x_{k; i_{r_{k-1}+1}, \dots, i_p}) \\
 & + (y_{1; i_1, \dots, i_{s_1}}) \otimes \dots \otimes (y_{m; i_{s_{m-1}+1}, \dots, i_p})
 \end{aligned} \tag{47}$$

where the vector \mathbf{z} with 2^p components is decomposed in two vectors that are Kronecker products of smaller vectors. The length of the k vectors $\mathbf{x}_j = (x_{j; i_{r_{j-1}+1}, \dots, i_{r_j}})$ is given by $2^{r_j - r_{j-1}}$ and similarly the m vectors \mathbf{y}_j are of length $2^{s_j - s_{j-1}}$. The blocking of \mathbf{x} and \mathbf{y} may be different. Such forms can be seen as generalizations of PARAFAC, or the coarse grain MPS form.

The interesting question is whether it is possible to introduce efficient contraction schemes for such sums. So in the following we want to analyze the costs for inner products of two vectors that are built as tensor products, but with different block structure as displayed in Figures 14 and 15.

The One-Dimensional Case

If both vectors have the same block structure, the inner product reduces to the inner product of the block vectors, compare (37). The costs can then be bounded by $3k2^{p/k}$.

The more interesting case appears if we allow different block sizes. First, let us consider the open boundary case where the first block in both vectors starts with i_1 and the last block ends with i_p , compare Figure 14. The contraction

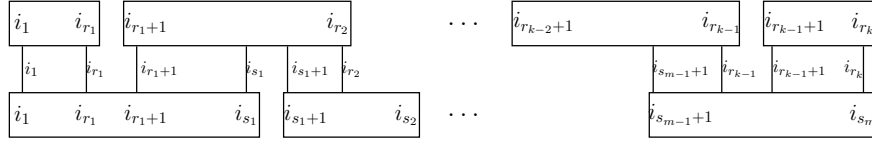


Figure 14: Contraction scheme for the calculation of the inner product of two vectors, which are formed by tensor products of smaller vectors, in the open boundary case.

starts with the left block computing the summation over i_1, \dots, i_{r_1} :

$$\sum_{i_1, \dots, i_{r_1}} x_{1;i_1, \dots, i_{r_1}} \cdot y_{1;i_1, \dots, i_{r_1}, \dots, i_{s_1}} \rightarrow y_{1;i_{r_1+1}, \dots, i_{s_1}}. \quad (48)$$

This leads to an update of $\bar{\mathbf{y}}_1$ which is replaced by a new vector to indices $i_{r_1+1}, \dots, i_{s_1}$. The costs are 2^{r_1} for one summation, and the summation has to be done for all indices $i_{r_1+1}, \dots, i_{s_1}$, hence $2^{s_1-r_1}$ times, leading to total costs of 2^{s_1} for this step. Now we have to repeat this procedure for the pair $x_{2;i_{r_1+1}, \dots, i_{r_2}}$ and the updated $y_{1;i_{r_1+1}, \dots, i_{s_1}}$. Let us assume that all blocks have size bounded by 2^r . In each step the costs are bounded by the size of the longest block, and the block size of the intermediate generated vectors is smaller than the size of the previous vectors. Hence in total the costs are bounded by the number of necessary contractions $= k + m$ times the maximum block size: $2^r \cdot (k + m)$.

In the periodic case, the first and the last blocks are allowed to overlap as displayed in Figure 15. In this case the first contraction looks like

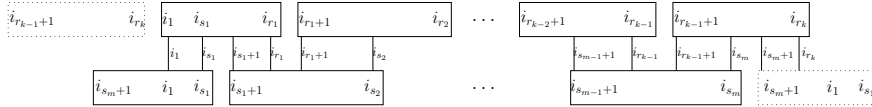


Figure 15: Contraction scheme in the periodic boundary case. Differing to open boundary case illustrated in Figure 14, the first and last blocks also overlap.

$$\sum_{i_1 \dots i_{s_1}} x_{1;i_1, \dots, i_{s_1}, \dots, i_{r_1}} \cdot y_{m;i_{s_m+1}, \dots, i_p, i_1, \dots, i_{s_1}} \rightarrow z_{1;i_{s_m+1}, \dots, i_p, i_{s_1+1}, \dots, i_{r_1}}. \quad (49)$$

The costs are given by the length of the contraction times the number of times it has to be executed, that is $2^{s_1} \cdot 2^{p-s_m} \cdot 2^{r_1-s_1}$. In the worst case the summation

part could be only one index which leads to 2^{2r-1} costs for maximum block size 2^r , where $r := \max\{r_1, r_2 - r_1, \dots, r_k - r_{k-1}, s_1, s_2 - s_1, \dots, s_m - s_{m-1}\}$. But in each step we can choose a contraction where the summation part is larger than the remaining indices. Then, the costs for one contraction are bounded by $2^{r_1+(p-s_m)} \leq 2^{3r/2}$. Therefore, in total the costs are bounded by $2^{3r/2}(m+k)$.

5.4. Mixed Blockings for Block MPS Vectors

The contraction schemes (48) presented in the previous subsection may also be applied to calculate inner products of Block MPS vectors (45) as introduced in Section 5.2. To this end we have to append the ancilla indices resulting from the matrix products to the physical indices. For contracting the inner product of the Block MPS vectors

$$\begin{aligned} \mathbf{x} &= \sum_{i_1, \dots, i_p} A_1^{(i_1, \dots, i_{r_1})} A_2^{(i_{r_1+1}, \dots, i_{r_2})} \dots A_k^{(i_{r_{k-1}+1}, \dots, i_p)}, \\ \mathbf{y} &= \sum_{i_1, \dots, i_p} B_1^{(i_1, \dots, i_{s_1})} B_2^{(i_{s_1+1}, \dots, i_{s_2})} \dots B_m^{(i_{s_{m-1}+1}, \dots, i_p)}, \end{aligned} \quad (50)$$

we can proceed in the same way as for standard MPS vectors (see Figures 2, 3), but instead of contracting over only one physical index i_j per step, we have to sum over all overlapping indices as illustrated in Figure 16.

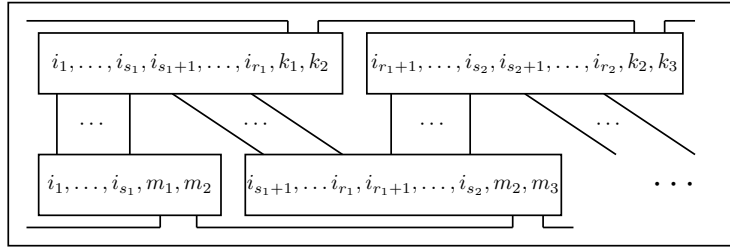


Figure 16: Contraction scheme for the inner product of the block MPS vectors (50).

5.5. The PARAFAC Model for the Two-Dimensional Case

For a 2D tensor network, we introduce another block structure that also allows fast contractions. As example we consider a 2D physical system with nearest-neighbor interaction in both directions as displayed in Figure 17.

In a first step we group r indices together to larger subblocks taking into account the neighborhood relations, see Figure 18(a).

We denote the newly introduced blocks by indices j_s . In the above example each j block contains 4 physical i indices (Figure 18(b)).

Based on the introduced coarse numbering, we define a final collection of blocks by building pairs of subblocks. We allow four different patterns of superblocks (see Figure 19), which enable efficient contractions in the calculation of mixed inner products.

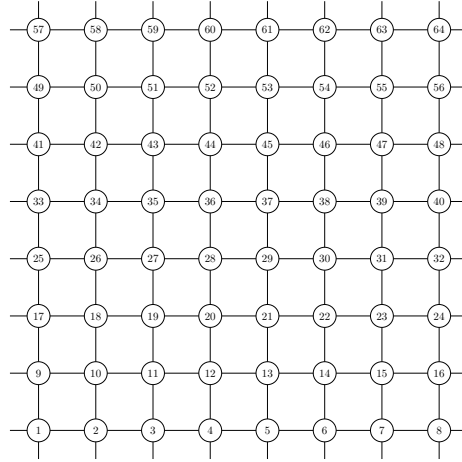


Figure 17: A 2D physical system with nearest neighbor interaction in both directions and periodic boundary conditions.

These four patterns are related to four different vector types

$$\begin{aligned}
 \mathbf{x}^1 &= \mathbf{a}_{j_1, j_2} \otimes \mathbf{a}_{j_3, j_4} \otimes \mathbf{a}_{j_5, j_6} \otimes \cdots \otimes \mathbf{a}_{j_{13}, j_{14}} \otimes \mathbf{a}_{j_{15}, j_{16}} , \\
 \mathbf{x}^2 &= \mathbf{a}_{j_4, j_1} \otimes \mathbf{a}_{j_2, j_3} \otimes \mathbf{a}_{j_8, j_5} \otimes \cdots \otimes \mathbf{a}_{j_{16}, j_{13}} \otimes \mathbf{a}_{j_{14}, j_{15}} , \\
 \mathbf{x}^3 &= \mathbf{a}_{j_1, j_5} \otimes \mathbf{a}_{j_2, j_6} \otimes \mathbf{a}_{j_3, j_7} \otimes \cdots \otimes \mathbf{a}_{j_{11}, j_{15}} \otimes \mathbf{a}_{j_{12}, j_{16}} , \\
 \mathbf{x}^4 &= \mathbf{a}_{j_{13}, j_1} \otimes \mathbf{a}_{j_{14}, j_2} \otimes \mathbf{a}_{j_{15}, j_3} \otimes \cdots \otimes \mathbf{a}_{j_7, j_{11}} \otimes \mathbf{a}_{j_8, j_{12}} .
 \end{aligned} \tag{51}$$

As ansatz vector we may consider combinations of the four vector patterns:

$$\mathbf{x} = \sum_i \alpha_i \mathbf{x}_i^1 + \sum_j \beta_j \mathbf{x}_j^2 + \sum_\ell \gamma_\ell \mathbf{x}_\ell^3 + \sum_\kappa \delta_\kappa \mathbf{x}_\kappa^4 .$$

Contractions between vectors of type 1–4 can be done in $\mathcal{O}(2^{3r})$ operations. Each contraction considers a block of vector \mathbf{x}_s combined with a block of vector \mathbf{x}_t that has an index j_k in common. Therefore, the costs are the subblocksize to the power of three. The result of such a contraction is a new block consisting of the pair of indices that were left over. Hence the total costs are the total numbers of blocks times the subblocksize to the power of three.

Remark: Similar to the generalization of MPS to Block MPS with mixed blockings as introduced in Section 5.4, the mixed blocking formats for 2D systems could also be applied to modify the PEPS ansatz (see Section 3.2). For efficient computations of inner products we may generalize the contraction scheme (Figures 5 and 6) to sets of overlapping physical indices.

6. Conclusions

This work combines similar concepts proposed in quantum information and linear algebra communities and presents them from a (multi-) linear algebraic

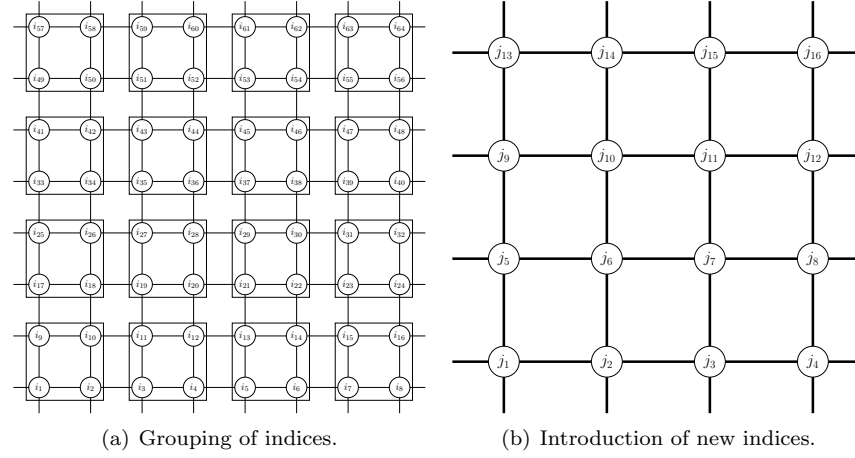


Figure 18: The 2D physical system as introduced in Figure 17, but now adjacent indices are grouped together and denoted by new indices j . The original neighborhood relations from Figure 17 are still taken into account.

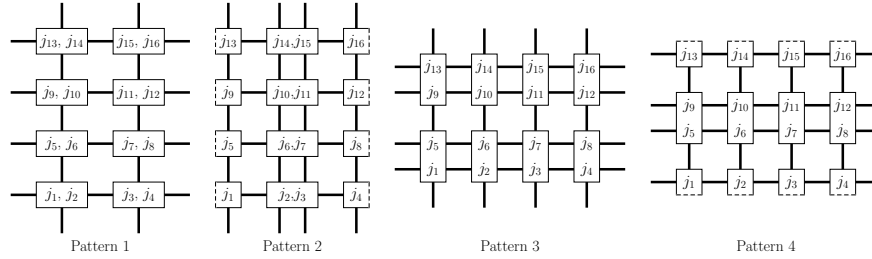


Figure 19: Four different patterns of subblocks related to the four vector types defined in (51)

point of view in a unifying mathematical and computational framework. In this context we are mainly interested in algorithmic considerations.

For our main task, the computation of ground states of physical quantum systems, we have to minimize the Rayleigh quotient over an exponentially growing vector space. To overcome the curse of dimensionality we have to find appropriate formats which allow cheap computations of matrix products and inner products, but still guarantee proper approximation properties. These formats are structured in such a way that they enable relaxation methods such as Alternating Least Squares.

Physicists have developed concepts like Matrix Product States, which correspond to Tensor Trains, for linear one-dimensional problems or Projected Entangled Pair States for two-dimensional problems. We present these concepts from a mathematical point of view and show how computations such as contractions of inner products can be performed efficiently.

As an ansatz format to minimize the Rayleigh quotient we also consider

the PARAFAC format, which allows modifications in several directions, such as different blocking structures. These generalizations, which can also be applied to modify, e.g., the MPS or PEPS ansatz, are constructed in such a way that

1. the physical (inter)actions (e.g. nearest neighbor interactions) can be reproduced properly,
2. representations are only polynomial in the system size p ,
3. computations such as inner product contractions can still be performed efficiently.

Our mixed blocking ansatz in PARAFAC and MPS is an efficient and flexible way to represent states of physical systems and can easily be extended to higher dimensions or systems with more complex interactions. It is, however, an open question, how the difference in the blockings can be constructed automatically and how efficient they can be for general problems.

Acknowledgements

This work roots back in a presentation at the 26th GAMM Seminar on Tensor Approximations at the Max-Planck-Institute for Mathematics in the Sciences in Leipzig, February 2010. It was supported in part by the Bavarian excellence network ENB via the International Doctorate Program of Excellence *Quantum Computing, Control, and Communication* (QCCC) as well as by the EU programmes QAP, Q-ESSENCE, and the exchange with COQUIT. — We wish to thank Norbert Schuch for many helpful comments.

- [1] C. Dawson, J. Eisert, T. J. Osborne, Unifying Variational Methods for Simulating Quantum Many-Body Systems, *Phys. Rev. Lett.* 100 (2008) 130501.
- [2] ALPS Consortium, The ALPS project (Algorithms and Libraries for Physics Simulations) is an open source effort aiming at providing high-end simulation codes for strongly correlated quantum mechanical systems as well as *C++* libraries for simplifying the development of such code, <http://alps.comp-phys.org/mediawiki/index.php/Main-Page>.
- [3] J. J. García-Ripoll, Time Evolution of Matrix Product States, *New J. Phys.* 8 (2006) 305.
- [4] B. Pirvu, V. Murg, J. I. Cirac, F. Verstraete, Matrix Product Operator Representations, *New J. Phys.* 12 (2010) 025012.
URL <http://dx.doi.org/10.1088/1367-2630/12/2/025012>
- [5] F. Verstraete, I. Cirac, V. Murg, Matrix Product States, Projected Entangled Pair States, and Variational Renormalization Group Methods for Quantum Spin Systems, *Adv. Phys.* 57 (2008) 143–224.

- [6] M. Nakahara, T. Ohmi, *Quantum Computing: From Linear Algebra to Physical Realizations*, CRC Press, 2008.
- [7] P. Pfeuty, The One-Dimensional Ising Model with a Transverse Field, *Ann. Phys.* 57 (1970) 79–90.
- [8] E. Lieb, T. Schultz, D. Mattis, Two Soluble Models of an Antiferromagnetic Chain, *Ann. Phys.* 16 (1961) 407–466.
- [9] I. Affleck, T. Kennedy, E. H. Lieb, H. Tasaki, Rigorous Results on Valence-Bond Ground States in Antiferromagnets, *Phys. Rev. Lett.* 59 (1987) 799–802.
- [10] P. J. Davis, *Circulant Matrices*, AMS Chelsea Publishing, 1994.
- [11] S. S. Capizzano, E. Tyrtyshnikov, Any Circulant-Like Preconditioner for Multilevel Matrices Is Not Superlinear, *SIAM J. Matrix Anal. Appl.* 21 (2000) 431–439.
- [12] A. Cantoni, P. Butler, Properties of the eigenvectors of persymmetric matrices with applications to communication theory, *IEEE Trans. Commun.* 24 (1976) 804–809.
- [13] M. Plenio, J. Eisert, J. Dreissig, M. Cramer, Entropy, Entanglement, and Area: Analytical results for Harmonic Lattice Systems, *Phys. Rev. Lett.* 94 (2005) 060503.
- [14] M. Cramer, J. Eisert, M. Plenio, J. Dreissig, Entanglement-Area Law for General Bosonic Harmonic Lattice Systems, *Phys. Rev. A* 73 (2006) 012309.
- [15] M. Wolf, F. Verstraete, M. B. Hastings, I. Cirac, Area Laws in Quantum Systems: Mutual Information and Correlations, *Phys. Rev. Lett.* 100 (2008) 070502.
- [16] J. Eisert, M. Cramer, M. B. Plenio, Area Laws for the Entanglement Entropy, *Rev. Mod. Phys.* 82 (2010) 277–306.
- [17] M. B. Hastings, An area law for one-dimensional quantum systems, *J. Stat. Mech. Theor. Exp.* 2007 (2007) P08024.
- [18] M. B. Hastings, Entropy and Entanglement in Quantum Ground States, *Phys. Rev. B* 76 (2007) 035144.
- [19] F. Verstraete, J. I. Cirac, Matrix Product States Represent Ground States Faithfully, *Phys. Rev. B* 73 (2006) 094423.
- [20] N. Schuch, I. Cirac, Matrix Product State and Mean Field Solutions for One-Dimensional Systems can be Found Efficiently, *Phys. Rev. A* 82 (2010) 012314.

- [21] M. Fannes, B. Nachtergaele, R. Werner, Abundance of Translation Invariant Pure States on Quantum Spin Chains, *Lett. Math. Phys.* 25 (1992) 249–258.
- [22] M. Fannes, B. Nachtergaele, R. F. Werner, Finitely Correlated States on Quantum Spin Chains, *Commun. Math. Phys.* 144 (1992) 443–490.
- [23] I. Peschel, X. Wang, M. Kaulke, K. Hallberg (Eds.), *Density-Matrix Renormalization: A New Numerical Method in Physics*, Lecture Notes in Physics Vol. 528, Springer, Berlin, 1999.
- [24] U. Schollwöck, The Density-Matrix Renormalization Group, *Rev. Mod. Phys.* 77 (2005) 259–315.
- [25] F. Verstraete, J. I. Cirac, Valence-Bond States for Quantum Computation, *Phys. Rev. A* 70 (2004) 060302.
- [26] F. Verstraete, J. I. Cirac, Renormalization Algorithms for Quantum-Many Body Systems in Two and Higher Dimensions, <http://arXiv.org:cond-mat/0407066> (2004).
- [27] S. Anders, M. B. Plenio, W. Dür, F. Verstraete, H. J. Briegel, Ground-State Approximation for Strongly Interacting Spin Systems in Arbitrary Spatial Dimension, *Phys. Rev. Lett.* 97 (2006) 107206.
- [28] G. Vidal, Entanglement Renormalization, *Phys. Rev. Lett.* 99 (2007) 220405.
- [29] N. Schuch, M. Wolf, F. Verstraete, I. Cirac, Strings, Projected Entangled Pair States, and Variational Monte Carlo Methods, *Phys. Rev. Lett.* 100 (2008) 040501.
- [30] R. Hübner, C. Kruszynska, L. Hartmann, W. Dür, F. Verstraete, J. Eisert, M. Plenio, Renormalisation Algorithm with Graph Enhancement, *Phys. Rev. A* 79 (2009) 022317.
- [31] R. Horn, C. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1987.
- [32] J. Carroll, J. J. Chang, Analysis of Individual Differences in Multidimensional Scaling via an n-Way Generalization of Eckart-Young Decomposition, *Psychometrika* 35 (1970) 283–319.
- [33] A. Smilde, R. Bro, P. Geladi, *Multi-Way Analysis. Applications in the Chemical Sciences*, Wiley, 2004.
- [34] I. Affleck, Large- n limit of $su(n)$ quantum "spin" chains, *Phys. Rev. Lett.* 54 (1985) 966–969.
- [35] G. Vidal, Efficient Classical Simulation of Slightly Entangled Quantum Computations, *Phys. Rev. Lett.* 91 (2003) 147902.

- [36] M. A. Martín-Delgado, M. Roncaglia, G. Sierra, Stripe Ansätze from Exactly Solved Models, *Phys. Rev. B* 64 (2001) 075117.
- [37] K. G. Wilson, The Renormalization Group: Critical Phenomena and the Kondo Problem, *Rev. Mod. Phys.* 47 (1975) 773–840.
- [38] S. R. White, Density Matrix Formulation for Quantum Renormalization Groups, *Phys. Rev. Lett.* 69 (1992) 2863–2866.
- [39] N. Schuch, I. Cirac, F. Verstraete, Computational Difficulty of Finding Matrix Product Ground States, *Phys. Rev. Lett.* 100 (2008) 250501.
- [40] D. Pérez-García, F. Verstraete, M. M. Wolf, J. I. Cirac, Matrix-Product State Representations, *Quant. Inf. Comput.* 7 (2007) 401–430.
- [41] U. Schollwöck, The Density-Matrix Renormalization Group in the Age of Matrix Product States, *Ann. Phys.* 326 (2011) 96–192.
- [42] M. G. Eckholt-Perotti, Matrix product formalism, Master’s thesis, Technische Universität München and Max-Planck-Institut für Quantenoptik (2005).
- [43] T. Huckle, K. Waldherr, T. Schulte-Herbrüggen, Exploiting Matrix Symmetries and Physical Symmetries in Matrix Product States and Tensor Trains, Unpublished results, Submitted (2011).
- [44] I. V. Oseledets, Tensor-Train Decomposition, *SIAM J. Sci. Comput.* 33 (2011) 2295–2317.
- [45] F. Verstraete, D. Porras, I. Cirac, DMRG and Periodic Boundary Conditions: A Quantum Information Perspective, *Phys. Rev. Lett.* 93 (2004) 227205.
- [46] F. Verstraete, J. J. García-Ripoll, J. I. Cirac, Matrix Product Density Operators: Simulation of Finite-Temperature and Dissipative Systems, *Phys. Rev. Lett.* 93 (2004) 207204.
- [47] I. V. Oseledets, Approximation of $2^d \times 2^d$ matrices using tensor decomposition, *SIAM J. Matrix Anal. Appl.* 31 (2010) 2130–2145.
- [48] F. Verstraete, M. M. Wolf, D. Pérez-García, I. Cirac, Criticality, the Area Law, and the Computational Power of Projected Entangled Pair States, *Phys. Rev. Lett.* 96 (2006) 220601.
- [49] N. Schuch, M. M. Wolf, F. Verstraete, I. Cirac, Computational Complexity of Projected Entangled Pair States, *Phys. Rev. Lett.* 98 (2007) 140506.
- [50] M. Aguado, J. I. Cirac, G. Vidal, Topology in Quantum States. PEPS Formalism and beyond, *J. Phys. Conf. Ser.* 87 (2007) 012003.

- [51] F. Verstraete, J. I. Cirac, Renormalization and Tensor Product States in Spin Chains and Lattices, *Phys. Rev. A* 42 (2010) 504004.
- [52] T. G. Kolda, B. W. Bader, Tensor Decompositions and Applications, *SIAM Rev.* 51 (2009) 455–500.
- [53] L. de Lathauwer, B. de Moor, J. Vandewalle, On the Best Rank-1 and Rank- (R_1, R_2, \dots, R_n) Approximation of Higher-Order Tensors, *SIAM J. Matrix Anal. Appl.* 21 (2000) 1324–1342.
- [54] I. V. Oseledets, D. V. Savostyanov, E. E. Tyrtyshnikov, Linear Algebra for Tensor Problems, *Computing* 85 (2009) 169–188.
- [55] I. V. Oseledets, E. E. Tyrtyshnikov, Breaking the Curse of Dimensionality, or How to Use SVD in Many Dimensions, *SIAM J. Sci. Comput.* 31 (2009) 3744–3759.
- [56] I. V. Oseledets, E. E. Tyrtyshnikov, Recursive decomposition of multidimensional tensors, *Doklady Math.* 80 (2009) 460–462.
- [57] B. Khoromskij, $\mathcal{O}(d \log N)$ -Quantics Approximation of $N - d$ Tensors in High-Dimensional Numerical Modeling, *Constructive Approximation* 34 (2011) 1–24.
- [58] F. Verstraete and J. I. Cirac, Matrix Product States Represent Ground States Faithfully, *Phys. Rev. B* 73 (2006) 094423.
- [59] M. Rajih, P. Comon, R. A. Harshman, Enhanced Line Search: A Novel Method to Accelerate Parafac, *SIAM J. Matrix Anal. Appl.* 30 (2008) 1128–1147.