Title: Date:

10.2 Background & Algorithms

* We know the area law
* You know the concept of the Schmitt-decomposition
* You know what a canonical form of an MPS is.
* You can sketch a DMRG algorithm.
* You can implement a TEBD algorithm.

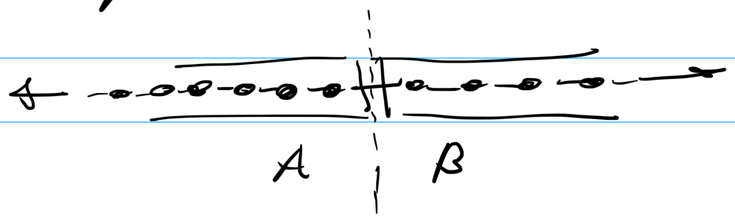
10.2.1 Schmidt decomposition & entanglement

Here, we want to learn that states with (low) area law entanglement can be well represented by matrix product states.

For this, we introduce the concept of entanglement entropy. Let us consider a bi-partition of the Hilbert space

$$\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B .$$

Each Hilbert space $\mathcal{H}_A$ and $\mathcal{H}_B$ have dimension d_A, and d_B, respectively. For one-dimensional systems a typical bi-partition is



Title: Date:

If we introduce a basis for each part $\{|\phi_i^A\rangle\}$ and $\{|\phi_j^B\rangle\}$ we can write any state in $\mathcal{H} = \mathcal{H}^A \otimes \mathcal{H}^B$ as

$$|\psi\rangle = \sum_{ij} \alpha_{ij} |\phi_i^A\rangle |\phi_j^B\rangle.$$

The matrix

$$\alpha = \begin{pmatrix} 0 & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & 0 \end{pmatrix}$$

was an example for $\mathcal{H}^A = \mathbb{C}^2$, $\mathcal{H}^B = \mathbb{C}^2$ and $|\psi\rangle$ the singlet configuration.

The power of the Schmidt decomposition is that we can always write

$$|\psi\rangle = \sum_{\alpha=1}^{r} \sqrt{\lambda_\alpha} \, |\Phi_\alpha^A\rangle |\Phi_\alpha^B\rangle \qquad (1)$$

with $r = \min(d_A, d_B)$. The λ_α are called the Schmidt coefficients. How did we manage to turn the double sum over ij into just one sum over α? Of course this came at the price of choosing new basis vectors

$$|\Phi_\alpha^B\rangle = V_{\alpha j}^{+} |\varphi_j^B\rangle,$$

$$|\Phi_\alpha^A\rangle = U_{\alpha i} |\varphi_i^A\rangle.$$

How did we find V, U and $\{\lambda_\alpha\}_\alpha$? It is nothing but the singular value decomposition (SVD) of α!

Title: Date:

for $d_A > d_B$:

$$\alpha = U \begin{bmatrix} \sqrt{\lambda_1} & & \\ & \ddots & \sqrt{\lambda_r} \\ & 0 & \end{bmatrix} V^+ \qquad \begin{array}{l} V: d_B \times d_B \\ U = d_A \times d_A \end{array}.$$

with $U^+U = \mathbb{1}$ and $V^+V = \mathbb{1}$.

Another way of obtaining the Schmidt coefficients is through

$$\rho = |\psi\rangle\langle\psi|$$

$\Rightarrow$ we trace over the degrees of freedom in $\mathcal{H}_B$

$$\rho_A = \mathrm{tr}_{\mathcal{H}_B}\, \rho \qquad (2)$$

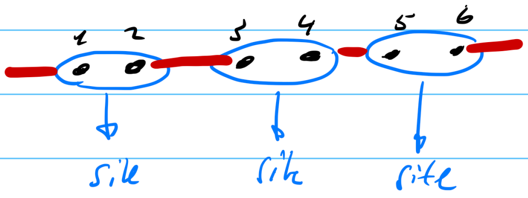
which is called the reduced density matrix if you compare (2) to (1), we immediately see that

$$\rho_A = \sum_\alpha \lambda_\alpha \, |\Phi_\alpha^A\rangle\langle\Phi_\alpha^A| .$$

With other words, we can diagonalize ρ_A to obtain the Schmidt values. We can now calculate the entanglement entropy

$$S = -\mathrm{Tr}_A\, \rho_A \log \rho_A = -\sum_\alpha \lambda_\alpha \log \lambda_\alpha$$

Let us now connect this to our knowledge about MPS. Let us take a generalization of the AKLT construction



But $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3$, etc. is not

Title: Date:

$\mathbb{C}^2$ but $\mathbb{C}^D$. And instead of bond singlets we take maximally entangled states

$$|\omega_D^{23}\rangle = \frac{1}{\sqrt{D}} \sum_{k=1}^{D} |k\rangle_2 |k\rangle_3$$

Clearly we are working towards an MPS with bond dimension $\chi = D$. We again write

$$|\psi\rangle = |\omega_D\rangle^{\otimes L-1} .$$

Let us look at the density matrix

$$\rho^{23} = |\omega_D^{23}\rangle\langle\omega_D^{23}| = \frac{1}{D} \sum_{k,l=1}^{D} |k\rangle_2 \langle l|_2 \otimes |k\rangle_3 \langle l|_3 .$$

Tracing over 3 gives us

$$\rho^2 = \mathrm{Tr}_3 \rho^{23} = \frac{1}{D} \sum_{\alpha=1}^{D} \sum_{k,l=1}^{D} |k\rangle_2 \langle l|_2 \underbrace{\langle\alpha|k\rangle_3}_{\delta_{\alpha k}} \underbrace{{}_3\langle l|\alpha\rangle_3}_{\delta_{l\alpha}}$$

$$= \frac{1}{D} \sum_{\alpha=1}^{D} |\alpha\rangle_2 \langle\alpha|$$

with this we find

$$S^{23} = -\mathrm{Tr}_2 \rho^2 \log \rho^2 =$$

$$= - \sum_k \frac{1}{D} \log \frac{1}{D} = \log D.$$

In other words for the maximally entangled state $|\omega_D\rangle^{\otimes L-1}$ the entanglement entropy is bounded by

Title: Date:

the logarithm of the bond dimension. Turned upside down, this means the states with a finite entanglement entropy are well represented by MPS!

An introduction to MPS by Evert van Nieuwenburg

May 13, 2019

Contents

1

Matrix product states

1.1. GENERAL

The concept of matrix product states (MPS) can be introduced at different levels of complexity. Some are initially more insightful than others, but require a longer exposition to arrive at a working understanding. If all that is needed is being able to work with MPS representations however, many of the underlying elegant details can be left out.

In the following sections, we will provide such different introductions to MPS representations. First, a straightforward version that immediately justifies the *matrix product* nomenclature and provides the necessary details for implementation. Next, MPS representations are introduced by way of a graphical method based on finite state automata. The latter description allows for a deeper insight into the structure and restrictions of matrix product states, but is further removed from a direct implementation. Finally, a more rigorous introduction highlighting the underlying concepts is presented at the end of this section.

Before we begin, a few general remarks are in order. Matrix product states are a way of representing a quantum state of a one dimensional (1D) system. The logic behind this particular representation extends to higher dimensions, for which it is in general referred to as a tensor network. In other words, MPS are the 1D variant of tensor networks. Once we arrive at a diagrammatic representation of an MPS, such conceptual extension to higher dimensions is straightforward. As a matter of fact, one may choose to start the introduction to MPS by beginning with general tensor networks and then specifying to 1D. It should be noted that the actual implementation of algorithms for tensor networks in dimensions larger than one is much more complicated [1].

To set the stage, we consider a general chain of sites as depicted in Figure 1.1. Every site of the 1D

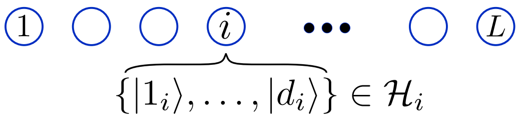


Figure 1.1: A 1D system consisting of sites labeled by an index i, each having associated to it a local Hilbert space $\mathscr{H}_i$ of dimension d_i. The states in this Hilbert space are denoted by $|\, s_i\rangle$, with $s_i = 1 \ldots d_i$.

chain is labeled by an index i and has associated to it a local Hilbert space $\mathscr{H}_i$. The dimension of this local Hilbert space is $\dim(\mathscr{H}_i) = d_i$, such that e.g. for a spin-1 chain we have $d = 3$ for all sites. It is possible to mix different Hilbert spaces for different sites, s.t. models that mix different spin-types or even fermions and bosons can be represented. The possible states for each site are labeled $|\, s_i\rangle$, with $s_i = 1 \ldots d$ (see Fig. 1.1), with which we can form basis states for an L site system by constructing product states $|\, s_1 s_2 \ldots s_L\rangle = |\, s_1\rangle \otimes |\, s_2\rangle \otimes \ldots \otimes |\, s_L\rangle$. An arbitrary pure state wavefunction for this system can be expressed in this basis as

$$|\, \psi\rangle = \sum_{s_1,\ldots s_L} c_{s_1 \ldots s_L} \,|\, s_1 \ldots s_L\rangle. \tag{1.1}$$

Given that the basis states themselves are not time-dependent, the (time-)evolution of this wavefunction can be represented solely by the coefficients $c_{s_1 \ldots s_L}$. The number of coefficients in this wavefunction is d^L, which for a full many-body system quickly becomes impossibly large as a function of the system size. Impossible refers to the possibility of numerically computing this wavefunction. Nowadays, many quantum many-body problems can only be satisfactory dealt with using the help of numerics. This means that the wavefunction has to be stored in memory. As an example consider a chain of two-level systems, s.t. $d = 2$ everywhere. Storing a wavefunction for L sites in memory, requires at the very least 2^L bits (each bit indicating the state of the two level system on a site). For a computer with 64 gigabytes of RAM, this amounts to maximally $L = 40$. It suffices to say that this would already require an optimized algorithm, let alone the required CPU-hours to perform any meaningful calculation with such a wavefunction.

The possibility of the enormous number of coefficients in Eq. (1.1) is not necessarily a problem. Namely, we are usually not after the most general wavefunction of a system. Rather, we tend to target the groundstate and possibly a few other low-energy eigenstates. These states are far from random as far as the coefficients are concerned, and we may wonder if for such states we could do with less than exponentially many coefficients. It is not hard to imagine that low-energy many-body eigenstates have some structure, and that hence many of the coefficients are identical or even zero. We should look for a different way of writing the coefficients that capitalizes on the fact that coefficients are not completely random. Matrix product states turn out to provide this notation, the reasons for which will be made clear in the following sections. To put this into perspective, whereas the exact representation of the general wavefunction in Eq. (1.1) for two-level systems limits us to about $L = 40$, using MPS representations can boost this to 1000+ sites.

1.1.1. HEURISTIC INTRODUCTION

Without loss of generality it is possible to write the general wavefuncion of Eq. 1.1 in a MPS form, by substituting

$$c_{s_1 \dots s_L} \to \mathrm{tr}\big(M^{[1],s_1} \cdots M^{[L],s_L}\big). \tag{1.2}$$

Notice that we have given a 'local' structure to the coefficients: for every site of the chain we have introduced a set of d matrices. The matrix product between all of the local matrices is what gives the MPS its name. The trace over this product of matrices is there to enforce that the result is a scalar. Also notice that we have not thrown away any coefficients; there are still d^L coefficients on the right-hand side, although we now have a way of controlling the number of *distinct* coefficients. This control is hidden in the dimensions of the matrices M, which therefore plays a very important role. We will see later that this also controls the amount of entanglement the wavefunction can have, which already shows that MPS representations work well for states with 'little' (to be made more precise later, in section 1.1.2) entanglement. Letting the dimension of the matrices be $D \times D$, the right-hand side roughly has dLD^2 distinct coefficients. Hence in order to represent a fully random wavefunction in which all of the d^L coefficients are distinct we would need D to scale exponentially in system size. This means that the substitution in Eq. (1.2) is *not* an approximation, as long as we choose the dimensions of the M-matrices large enough. Numerical algorithms based on MPS representations will of course limit the maximum value for D, turning the MPS representation in an approximation to the actual wavefunction.

The class of states with 'little' entanglement mentioned before, are those that can exactly be represented by MPSs with matrix dimensions that scale at most linear in L. In section 1.1.2, we examine this class of states and construct the MPS representation explicitly. For the sake of completeness, we write out a general wavefunction in MPS form:

$$|\psi\rangle = \sum_{s_1,\dots s_L} \mathrm{tr}\big(M^{[1],s_1} \cdots M^{[L],s_L}\big)|s_1 \dots s_L\rangle. \tag{1.3}$$

For most practical purposes, it suffices to start from this representation. As a matter of fact, initializing a new MPS can be done in just a few lines as shown in the `Python` code snippet below. Numerical algorithms such as the *density matrix renormalization group* (DMRG) or *time evolving block decimation* (TEBD) both provide a set of rules on how to update the M-matrices to perform a groundstate search or time evolution, respectively. More on these algorithms follows in section 1.2

```
import numpy as np
# Set the MPS parameters
d = 2; D = 10; L = 40
```

```
# The MPS matrices M are stored here as a set of L tensors,
# each having d matrices of size DxD. We initialize them with zeros,
# but other initializations are also possible.
M = np.zeros( (L,d,D,D ) )
```

1.1.2. RIGOROUS INTRODUCTION

This time, we are motivated by the observation that groundstates and low-energy excited states of various many-body systems tend to have 'little' entanglement. This is a statement that can be made rigorous [2] for 1D systems that have a gap between the groundstate and the first excited state. It has consequences that we will fully exploit to obtain the MPS representation.

First, let us make solid what is meant by entanglement. We consider the so-called *bipartite* entanglement between two parts of a system for any bipartitioning. The two parts are generally refered to as the *left* and *right* half, or as parts A and B, and are associated with two Hilbert spaces $\mathscr{H}^A$ and $\mathscr{H}^B$ of dimensions $\dim\mathscr{H}^{A/B} = d^{A/B}$. These Hilbert spaces depend on where the bipartitioning is made, i.e. at which point the system is separated into two parts. By introducing a basis for parts A and B, denoted $\{|\phi_i^A\rangle\}$ and $\{|\phi_j^B\rangle\}$ respectively, we can write any wavefunction in a productbasis between A and B as follows:

$$|\psi\rangle = \sum_{ij} \alpha_{ij} |\phi_i^A\rangle |\phi_j^B\rangle. \tag{1.4}$$

The beauty of the *Schmidt decomposition*, is that this double sum over the coefficients α_{ij} and basis states can be turned into a single sum

$$|\psi\rangle = \sum_{\alpha=1}^{r} \lambda_\alpha |\Phi_\alpha^A\rangle |\Phi_\alpha^B\rangle, \tag{1.5}$$

where the Schmidt rank r is constrained between 1 and the minimum of the Hilbert space dimensions of parts A and B, i.e. $r = 1\ldots\min(d^A, d^B)$. The coefficients λ_α are known as the Schmidt coefficients. The Schmidt coefficients are real, non-negative, unique (for a given $|\psi\rangle$) and satisfy $\sum_\alpha \lambda_\alpha^2 = 1$. The Schmidt rank is equal to 1 only for a product state, which by definition is not entangled, whereas a Schmidt rank $r > 1$ indicates non-zero entanglement between the two parts. The basis states $|\Phi_\alpha^A\rangle$ and $|\Phi_\alpha^B\rangle$ for the A and B parts are related to the original $|\phi_i^A\rangle$ and $|\phi_j^B\rangle$ by a unitary transformation that depends on the state itself. Namely, in order to get to Eq. 1.5 from Eq. 1.4 we perform a singular value decomposition (SVD) of the coefficient-matrix α_{ij}:

$$\alpha_{ij} = \sum_\alpha U_{i\alpha} S_{\alpha\alpha} V_{\alpha j}^\dagger,$$

where S is a diagonal matrix with entries λ_α, and the matrices U and V have the properties $UU^\dagger = 1$ and $V^\dagger V = 1$ (with $\dagger$ representing the hermitian conjugate). The diagonal entries of S are also called the *singular values*, and give rise to the properties of the λ_α coefficients mentioned before.

As a side note, we consider a different route of obtaining the Schmidt coefficients and the vectors $|\Phi_\alpha^{A/B}\rangle$. Starting from the density matrix $\rho = |\psi\rangle\langle\psi|$ of the combined parts A and B, we can obtain both the coefficients and vectors from performing a partial trace of ρ:

$$\rho_A = \mathrm{Tr}_B \rho = \sum_\alpha \lambda_\alpha^2 |\Phi_\alpha^A\rangle\langle\Phi_\alpha^A|$$

and similar for ρ_B. The two reduced density matrices ρ_A and ρ_B share the same spectrum λ_α^2, and the eigenvectors form the orthogonal columns of U and $V^\dagger$.

From the Schmidt values λ_α, or from the reduced density matrices, we may construct a quantity known as the *entanglement entropy*. The entanglement entropy is nothing more than just the regular von Neumann entropy, but evaluated on a subsystem (namely, the reduced density matrix):

$$S = -\mathrm{Tr}\rho_{A/B}\ln\rho_{A/B} = -\sum_\alpha \lambda_\alpha \ln\lambda_\alpha. \tag{1.6}$$

Only for a product state is $S = 0$, hence any nonzero S signals the presence of entanglement [1]. As a remark for later, we should note that the Schmidt values λ_α are also referred to as the entanglement spectrum. Rather, to be precise, it is $-2\ln\lambda_\alpha$ that constitutes the entanglement spectrum, but the term is used interchangeably for both.

We can now come back to explaining what is meant by 'little' entanglement. Consider the entanglement entropy as a function of the size N of the left half of the bipartition, so that $N = 1\dots L/2$. We need only go up to half the system size, since anything beyond that is equivalent to having made the bipartitioning starting from the opposite end. Let us denote this quantity $S(N)$, and ask how it scales as a function of N. This will of course depend on the particular state under consideration, but it is possible to prove that $S(N)$ obeys an *area law* for groundstates of gapped 1D systems. The area law states, independent of dimension, that $S(N)$ is proportional to the size of the boundary that is left after a bipartioning. In 1D systems, the boundary consists of 2 points and hence the entanglement entropy does not scale with N. Near the boundaries of a system there can of course be a dependence on N (boundary effects), but the entanglement entropy quickly satures as is demonstrated in Figure 1.2.

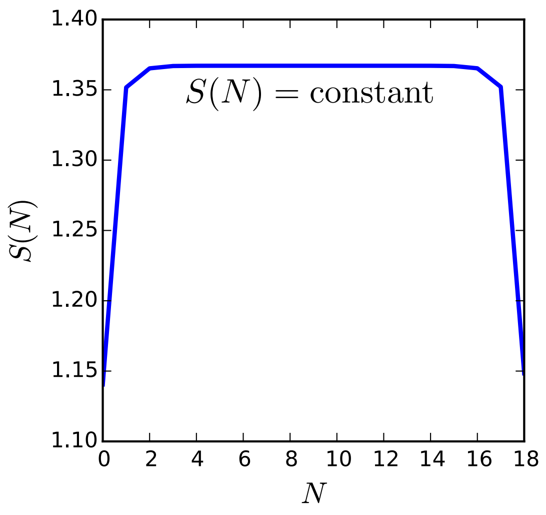


Figure 1.2: The entanglement entropy as a function of bipartition size N for the groundstate of a gapped 1D system with $L = 20$ (the AKLT state) The sites are indexed with labels starting from 0. Apart from the effects at the boundary, it is clearly visible that the entanglement entropy saturates and becomes independent of N (area law).

Let us now make use of the area law, by constructing *explicity* such a state for which $S(N)$ is constant. As a first step, we imagine a chain of sites as depicted in Figure 1.3. Each of the sites is associated with a complex Hilbert space $\mathbb{C}^D$ for some positive $D \geq 1$, and we imagine pairing the sites into groups of $\mathbb{C}^D \times \mathbb{C}^D$ as shown in the figure. We think of these pairs as *one* new site. The pairs are sometimes referred to as auxiliary sites. Next, we construct *bonds* between the pairs, such

[1]Actually, the entanglement entropy also captures classical correlations and may therefore not be the best entanglement 'witness'. Other quantities, such as the logarithmic negativity [3] have been proposed, but finding an optimal witness is a research project by itself.

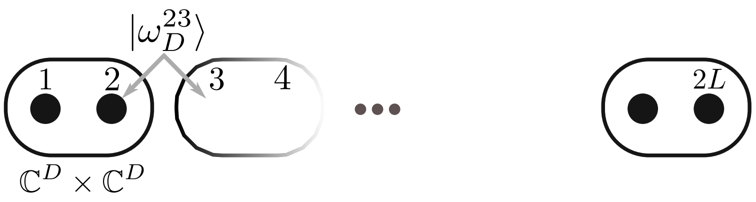


Figure 1.3: Construction of maximally entangled bonds $|\omega_D\rangle$ between 'sites'. Each site consists of two auxiliary sites and is assigned to a space $\mathbb{C}^D \times \mathbb{C}^D$, so that a chain with L sites actually consists of $2L$ auxiliary sites.

that for example the state of auxiliary sites 2 and 3 is given by

$$|\omega_D^{23}\rangle = \frac{1}{\sqrt{D}} \sum_{k=1}^{D} |k_2\rangle|k_3\rangle, \tag{1.7}$$

where $|k_2\rangle$ represents the state of site 2 with $k = 1 \dots D$. The quantity D is referred to as the (maximal) *bond dimension* for the MPS. The full state for the chain is then given by

$$|\psi\rangle = |\omega_D\rangle^{\otimes L-1}, \tag{1.8}$$

where we construct a bond between all of the pairs.

We will be considering open boundary conditions mostly, and so site number 1 and $2L$ are left unpaired. For periodic boundary conditions, an extra bond between those two sites has to be added. Each of the bonds corresponds to a maximally entangled state, as we can demonstrate on the example bond $|\omega_D^{23}\rangle$. The density matrix for this bond is given by

$$\rho^{23} = |\omega_D^{23}\rangle\langle\omega_D^{23}| = \frac{1}{D} \sum_{k,l=1}^{D} |k_2\rangle\langle l_2| \otimes |k_3\rangle\langle l_3|, \tag{1.9}$$

leading to a reduced density matrix

$$\begin{aligned}
\rho^{(2)} &= \mathrm{Tr}_3 \rho^{23} \\
&= \frac{1}{D} \sum_{\alpha=1}^{D} \sum_{k,l=1}^{D} |k_2\rangle\langle l_2| \underbrace{\langle\alpha_3|k_3\rangle}_{=\delta_{\alpha,k}} \underbrace{\langle l_3|\alpha_3\rangle}_{=\delta_{l,\alpha}}, \\
&= \frac{1}{D} \sum_{\alpha=1}^{D} |\alpha_2\rangle\langle\alpha_2|.
\end{aligned} \tag{1.10}$$

The entanglement entropy is then easily found to be

$$\begin{aligned}
S^{23} &= -\mathrm{Tr}(\rho^{(2)} \ln \rho^{(2)}) \\
&= -\sum_k \frac{1}{D} \ln \frac{1}{D} \\
&= \ln D,
\end{aligned} \tag{1.11}$$

and hence for any bipartitioning, the state $|\psi\rangle$ in Eq. (1.8) has entropy equal to $\ln D$. Notice here that we do not consider bipartitionings in between the bonds. The two sites in a pair are regarded as *one* physical site that cannot be cut in half. It is clear here that the bond dimension sets the maximal entanglement entropy that can exist across any bond. Apart from the constant entanglement entropy, the current state is actually not very interesting. To turn it into an interesting state, we introduce maps $P^{[i]}$ from the pairs in $\mathbb{C}^D \times \mathbb{C}^D$ to the 'physical' Hilbert space $\mathbb{C}^d \sim \mathscr{H}_i$, as shown in Fig. 1.4. We obtain a new state by pairwise application of these maps:

$$|\tilde{\psi}\rangle = P^{[1]} \otimes \cdots \otimes P^{[N]} |\psi\rangle = P^{[1]} \otimes \cdots \otimes P^{[N]} |\omega_D\rangle^{\otimes L-1}$$

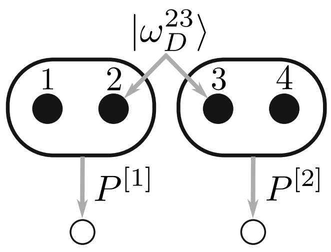


Figure 1.4: We construct maps $P^{[i]} : \mathbb{C}^D \times \mathbb{C}^D \to \mathbb{C}^d$, mapping the auxiliary sites to physical sites. These maps map the auxiliary wavefunction in Eq. 1.8 to a physical wavefunction that has more interesting correlations, but is still a maximally entangled state across each bond.

From the field of quantum information theory it is known that local operators (such as our P) do not change the entanglement. For an operator to change the entanglement between two parts of a bipartition, it needs to act on both parts simultaneously. Our P-maps never act across a bond that can be cut. In other words, we still have a state that is maximally entangled across each bond, but we can add structure (correlations) to the state via our P-maps.

Let us try to write an explicit form for the P-maps. The basis for the local Hilbert spaces is denoted $|s\rangle$ as before, but we additionally introduce basis states $|\alpha\beta\rangle$ for the pairs living in $\mathbb{C}^D \times \mathbb{C}^D$ (so that α and β take values $1\dots D$). For a given P mapping to site i, we then write

$$P^{[i]} = \sum_{\alpha,\beta=1}^{D} \sum_{s=1}^{d} |i\rangle\langle\alpha\beta|\, M^{[i],s}_{\alpha,\beta}, \tag{1.12}$$

in which the coefficients are written compactly via the tensors $M^{[i],s}_{\alpha,\beta}$. Hence for a given site i we have d matrices $\left(M^{[i],1},\dots,M^{[i],d}\right)$, and we associate $P^{[i]}$ to this set of matrices $M^{[i]}$. Notice in particular that we only have to specify ND^2d parameters to completely determine all of the P-maps. Next, instead of a single site, let us look at a two-site case explicitly. We consider the situation shown in Fig. 1.4, in which we have two physical sites that come from four auxiliary sites connected by a maximally entangled bond $|\omega_D^{23}\rangle$. We act with the P-maps:

$$\begin{aligned} P^{[1]} \otimes P^{[2]} |\omega_D^{23}\rangle &= \sum_{s_1,\alpha,\beta} \sum_{s_2,\gamma,\delta} |s_1 s_2\rangle\langle\alpha\beta\gamma\delta|\, M^{[1],s_1}_{\alpha,\beta} M^{[2],s_2}_{\gamma,\delta} \sum_k \frac{1}{\sqrt{D}} |k\rangle_2 |k\rangle_3 \\ &= \sum_{s_1,s_2,\alpha,\beta,\gamma,\delta} \frac{1}{\sqrt{D}} M^{[1],s_1}_{\alpha,\beta} M^{[2],s_2}_{\gamma,\delta}\, |s_1 s_2\rangle\langle\alpha\delta|\, \underbrace{\langle\beta\gamma|kk\rangle}_{\delta_{\beta,k}\delta_{\gamma,k}} \\ &= \sum_{s_1,s_2,\alpha,\delta,k} M^{[1],s_1}_{\alpha,k} M^{[2],s_2}_{k,\delta}\, |s_1 s_2\rangle\langle\alpha\delta|. \end{aligned} \tag{1.13}$$

In the last line, we absorbed the $1/\sqrt{D}$ factor into the M-matrices. What we learn from this example is that when the matrices $M^{[1],s_1}$ and $M^{[2],s_2}$ are associated to $P^{[1]}$ and $P^{[2]}$ separately, we can associate the matrix product $M^{[1],s_1} \cdot M^{[2],s_2}$ to the combination $P^{[12]} = P^{[1]} \otimes P^{[2]}$. By iterating this procedure, we will arrive at

$$|\tilde{\psi}\rangle = \sum_{s_1,\dots,s_L,\alpha,\delta} M^{[1],s_1} \cdot \ldots \cdot M^{[L],s_L} |s_1 \dots s_L\rangle\langle\alpha\delta|. \tag{1.14}$$

Boundary condititions

This expression is basically the MPS representation of Eq. 1.3, except for the $\langle\alpha\delta|$ term. By inspecting Eq. 1.13, we see that the indices α and δ belong to the first and last M-matrices, respectively.

They therefore determine the boundary conditions on the MPS, and choosing them is an important part of MPS representations.

To get to a fully physical wavefunction, we need to eliminate the $\langle\alpha\delta|$ term. There are two main ways of achieving this, keeping in mind that the final product of the M-matrices should result in a scalar (they are, after all, the coefficients to the basis states).

The first method, often chosen for *open* boundary conditions, is to take the inner product of Eq. 1.14 with $|1,1\rangle$ on the first and last auxiliary indices. This leads to fixing $\alpha = \delta = 1$, and thereby turns the first and last matrix into a row and column vector respectively. Since the coefficients themselves have no idea about the boundary conditions, it is not entirely true that this choice is valid only for open boundary conditions. For periodic boundary conditions the restriction of α and δ to 1 will also work, but the entanglement across the periodic bond would have to be encoded throughout all of the other M-matrices. This most likely means that the bond dimension D has to be a (very) large number. Instead, it is more efficient for periodic boundary conditions to keep the first and last M as matrices instead of vectors, and turn the matrix product into a scalar by taking the trace. Instead of multiplying by $|1,1\rangle$, we multiply by $|\gamma,\gamma\rangle$ and sum over γ. This procedure will lead to exactly the formulation of an MPS state as in Eq. 1.3. For notational convenience, we will always write the MPS using the trace form, even for open boundary conditions in which we turn the first and last M into row and column vectors. The trace will then simply act on a scalar and change nothing. In the following section, we will introduce the diagrammatic way of working with MPS representations. It is here where the difference between periodic and open boundaries can be clearly seen.

As a final note, there also exist infinite MPS (iMPS) representations. In this case, we consider a unit cell of L sites, and impose periodic boundary conditions. A crucial difference to a finite periodic case, is the way in which observables are evaluated. More on this will be highlighted in the following sections, but it is good to be aware of the existence of iMPS already now.

1.1.3. MPS DIAGRAMS

Formulating operations on MPS states can be done very efficiently by introducing a diagrammatic representation for the M-matrices. The simplest object we consider is exactly such a single M-matrix, and is shown below in Fig. 1.5. The lines in this diagram indicate the indices of the M-

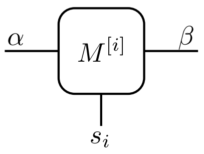


Figure 1.5: MPS diagram for a single M-tensor. The number of 'legs' is the number of indices of the tensor, which is three for a single M-tensor. The downward pointing leg (convention) is chosen to be the physical leg, having the index corresponding to the physical Hilbert space. The left and right legs are auxiliary legs.

tensor, also referred to as its 'legs'. The diagram hence has as many legs as the object it represents has indices. The downward pointing leg is called the physical leg, since it contains the indices belonging to the physical basis states. The other two legs, simply called the left and right legs, represent the auxiliary indices running from 1 up to D. With this representation, we can re-express common operations on the tensors in a graphical way. The most basic and most important operation one can perform is a *contraction* of two tensors over one or multiple of their indices. An example of this is shown in Fig. 1.6. A contraction of two tensors over one or multiple legs can be implemented straightforwardly in `Python`. The code snippet below and performs the contraction shown in Fig. 1.6, albeit with tensors that have been initialized to zero.

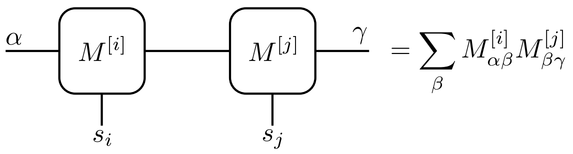


Figure 1.6: Contracting two (or more) legs of tensors is a graphical way of representing a summation over the corresponding indices of the tensors. Only legs with the same dimension can be contracted over. In this example, the resulting object is a tensor with 4 open indices, and hence would be drawn as an object with 4 legs.

```
# ...
M = np.zeros( (L,d,D,D ) )
# Contract MPS matrices of site 4 and 5, using the `right' leg of
# site 4 (index 2) and the `left' leg of site 5 (index 1). Notice
# that the resulting object has four legs, the first index being
# the physical leg of site 4, the second being the left leg of site
# 4, then the physical leg and right leg of site 5.
newTensor = np.tensordot( M[3], M[4], axes=(2,1) )
```

A general MPS as in Eq. 1.3, reproduced below for convenience, can now be represented diagrammatically. As alluded to in the previous section, it is here where we can distinguish clearly between open and periodic boundary conditions. As is shown in Fig. 1.7, for open boundary conditions the first and last M-matrices are vectors and hence have one index less than do the matrices[2]. In the case of periodic boundaries, the trace amounts to a sum over the α and δ indices in Eq. 1.14 and is represented by the line connecting the first and last MPS matrix.

$$|\psi\rangle = \sum_{s_1,\ldots s_L} \mathrm{tr}\big(M^{[1],s_1} \cdots M^{[L],s_L}\big)|s_1 \ldots s_L\rangle.$$

Notice that some of the labels have been left out on purpose, but will be put back whenever more detail is needed. We will not consider periodic boundaries on MPS in the rest of this section, so the diagrams will from now on only consist of those with open boundaries.

The overlap between two wavefunctions $|\psi\rangle$ and $|\phi\rangle$, with MPS matrices denoted by A and B respectively, can then be represented as shown in Fig. 1.8. The MPS for $|\phi\rangle$ has to be conjugated (to form $\langle\phi|$), and is drawn with physical indices pointing up. Let us directly turn to the question of how to measure observables. We consider the *transfer matrix*, $\mathbb{E}^{[i]}$, shown in Fig. 1.9. It maps auxiliary indices $(\alpha,\alpha') \to (\beta,\beta')$, and can be expressed in terms of the MPS matrices as

$$\begin{aligned}\mathbb{E}^{[i]}_{(\alpha\alpha'),(\beta\beta')} &= \sum_{s_i} M^{[i],s_i}_{\alpha\beta}\left(M^{[i],s_i}_{\alpha'\beta'}\right)^* \\ &= \sum_{s_i} M^{[i],s_i} \otimes \left(M^{[i],s_i}\right)^*. \qquad (1.15)\end{aligned}$$

With this notation, we can express the normalization of a wavefunction $|\psi\rangle$ as

$$\langle\psi|\psi\rangle = \mathrm{tr}\left(\mathbb{E}^{[1]} \ldots \mathbb{E}^{[L]}\right),$$

where we understand that $\mathbb{E}^{[1]}$ and $\mathbb{E}^{[L]}$ for the first and last sites are slightly different from the transfer matrix shown in Fig. 1.9 in that they have only one set of auxiliary indices. That also means

[2]In principle, we could use the same diagram also for the vectors, but restricting the legs to running only over one value. But that means there is nothing to contract anyway, so we may as well drop the leg.

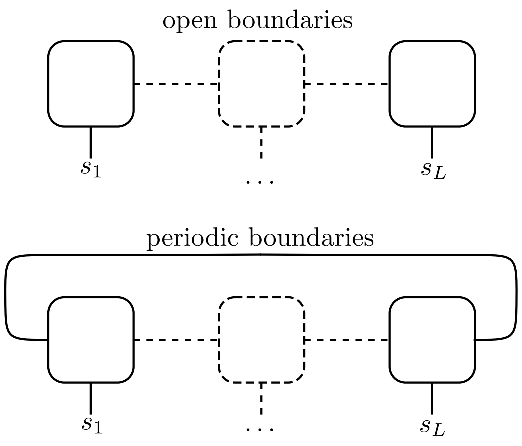


Figure 1.7: A general MPS wavefunction is determined fully by the coefficients in a given basis. These coefficients can be represented in an MPS form, for which the diagram is shown here. Hence we think of this diagram as representing the wavefunction (in a corresponding basis). The top figure shows the diagram for an MPS with open boundary conditions, whereas the periodic connection in the lower figure indicates periodic boundary conditions.

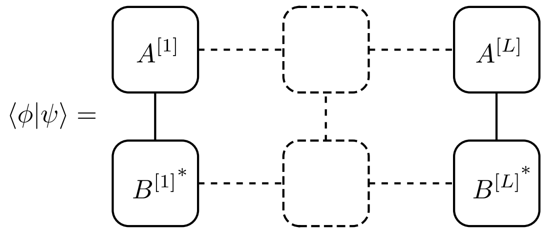


Figure 1.8: Overlap of two MPSs $|\psi\rangle$ and $|\phi\rangle$, with matrices labeled by A and B respectively. Notice that the B matrices have their physical indices pointing *upward*, meaning that their matrix entries have to be complex conjugated.

that the trace is redundant, but we keep the notation compatible with periodic boundaries. By a slight extension of the transfer matrix, we also introduce $\mathbb{E}_X^{[i]}$:

$$\mathbb{E}_X^{[i]} = \sum_{s_1 s_1'} M^{[i],s_1} X_{s_1,s_1'} \left(M^{[i],s_1'}\right)^*, \tag{1.16}$$

so that $\mathbb{E}^{[i]} = \mathbb{E}_{\hat{1}}^{[i]}$, with $\hat{1}$ the identity operator. We are then capable of expressing an expectation value such as:

$$\langle\psi|X_i \otimes Y_j|\psi\rangle = \mathrm{tr}\Big(\mathbb{E}^{[1]} \cdots \mathbb{E}_X^{[i]} \cdots \mathbb{E}_Y^{[j]} \cdots \mathbb{E}^{[L]}\Big). \tag{1.17}$$

In a diagram, such an expectation value would look like that shown in Fig. 1.10. This is a good point to take a step back and evaluate the efficiency of the computation of an expectation value as in Eq. 1.17. When contracting such a full network, it pays to consider attention to the order in which the contractions are carried out. When we perform the contraction shown in Fig. 1.11, we have the option of first contracting the pair of legs **A** and then **B**, or vice versa. If we contract the pair of legs indicated by the letter **A** first, we effectively perform multiplication of two matrices of sizes $D^2 \times d$ and $d \times D^2$. Since the complexity of matrix mulitplication of matrices of shapes $n \times m$ and $m \times p$ is $\mathcal{O}(nmp)$, this route costs $\mathcal{O}(dD^4)$. Via the outher route, by contracting the **B** pair first, we multiply two matrices of size $dD \times D$ and $D \times dD$, leading to a complexity $\mathcal{O}(d^2D^3)$. In the limit of very large D, the second route is much better than the first.

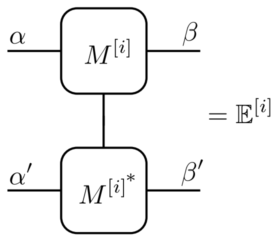


Figure 1.9: Mapping auxiliary indices from $(\alpha, \alpha') \to (\beta, \beta')$ by using the transfer matrix $\mathbb{E}^{[i]}$.

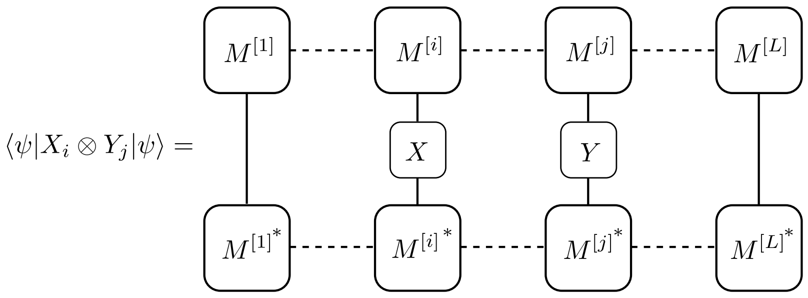


Figure 1.10: Computing the expectation value of the operator $X_i \otimes Y_j$ represented in diagrammatical form.

CANONICAL FORM

There is a much more important point that has to be made when we talk about measuring operators. If we want to measure the expectation value of a local operator on a system of length L, we would have to perform the full contraction of all tensors as in Fig. 1.10. Instead, there is a degree of freedom in the choice of the MPS matrices that allows us to evaluate local operators by only using the MPS matrices for the site we consider. This degree of freedom is sometimes called a 'gauge' degree of freedom, and amounts to the following transformation on the M-matrices:

$$M^{[i],s_i} \to X_i M^{[i],s_i} \left(X_{i+1}\right)^{-1}.$$

The matrices X_i and X_{i+1} are unitary matrices, and one can easily convince oneself – by inserting the transformation into Eq. 1.3 – that this transformation leaves the MPS unchanged.

A particular choice of the unitary matrices X is given by the 'isometric gauge'. Before going into detail on how to explicitly get the matrices in this form, let us highlight what it will do for us. When chosen correctly, this gauge will bring our M matrices into *canonical form*. In matrix notation, this means we can choose the MPS matrices to obey

$$\sum_{s_i,\gamma} M^{[i],s_i}_{\alpha\gamma} \left(M^{[i],s_i}_{\gamma\beta}\right)^* = \lambda \delta_{\alpha\beta},, \qquad |\lambda| = 1.. \tag{1.18}$$

Equivalently, canonical form means that $\mathbb{E}^{[i]}$ has as its dominant left eigenmatrix the identity matrix with unit-modulus eigenvalue, i.e.:

$$\sum_{\beta,\beta'} \mathbb{E}^{[i]}_{(\alpha\alpha'),(\beta\beta')} \delta_{\alpha\beta} = \lambda \delta_{\alpha'\beta'}, \qquad |\lambda| = 1. \tag{1.19}$$

If the above conditions are fulfilled, the MPS is called left-canonical. Similary, by straightforward comparison to the left-canonical case, an MPS can also be in a right-canonical form. Diagrammatically, the left-canonical form can be represented as shown in Fig. 1.12. Now imagine we are trying to compute the expectation value of an operator X on site i, denoted as $\langle\psi|X_i|\psi\rangle$. By making sure

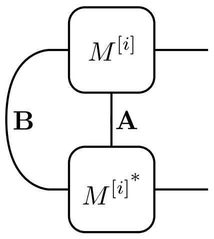


Figure 1.11: The order of contracting, i.e. first the pair of legs indicated by **A** or first the pair indicated by **B**, can make a big difference in efficiency. If one contracts **A** first and then **B**, the cost of the full contraction scales as $\mathcal{O}(dD^4)$, whereas the opposite order only costs $\mathcal{O}(d^2D^3)$.

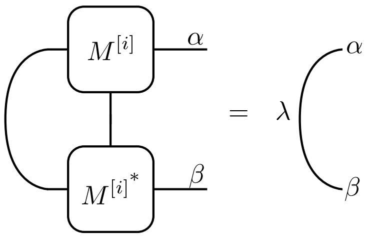


Figure 1.12: Left-canonical form means that the contraction of the two MPS matrices over their physical and left indices can be reduced, effectively eliminating the need to actually perform the contraction numerically.

that all the MPS matrices to the left of site i are in left-canonical form, and those to the right are all in right-canonical form, we are able to measure a local observable by contracting only three objects as shown in Fig. 1.13. This is quite a simplification over a diagram such as the one in Fig. 1.10. Prac-

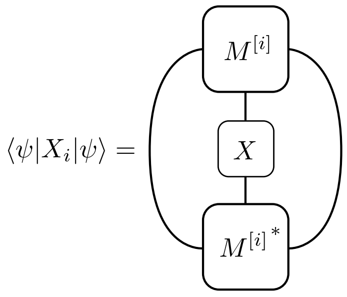


Figure 1.13: When the MPS is in canonical form, we no longer need to contract all L sites but can instead focus on just the sites involved in the operator we are measuring.

tically, let us look at how the left- or right-canonical form can be obtained, starting from a given set of MPS matrices M. A canonical MPS can be obtained directly from the coefficients $c_{s_1\dots s_L}$ in any basis too, but the details of this are left to the extensive MPS review by U. Schollwöck [4]. For convenience and for fixing notation for the rest of this section, we write here a general MPS with all the indices shown:

$$|\psi\rangle = \sum_{s_1,\dots,s_L}\sum_{\alpha_1,\dots} M^{[1],s_1}_{1,\alpha_1} M^{[2],s_2}_{\alpha_1,\alpha_2} M^{[3],s_3}_{\alpha_2,\alpha_3}\dots |s_1\dots s_L\rangle. \tag{1.20}$$

Notice that the first index of the first M-matrix is fixed to taking only one single value, reflecting our use of open boundary conditions. In order for us to construct a left-canonical MPS, we start from the first site and consider $M^{[1],s_1}_{1,\alpha_1}$. We reshape the left and physical indices of this tensor into a single index, and denote the resulting tensor $M^{[1]}_{(s_1,1),\alpha_1}$. This tensor is now a matrix (it has only two indices), and can be decomposed using a singular value decomposition (SVD):

$$M^{[1]}_{(s_1,1),\alpha_1} = \sum_{\sigma_1} U^{[1]}_{(s_1,1),\sigma_1} S_{\sigma_1,\sigma_1} V^{\dagger}_{\sigma_1,\alpha_1},$$

for which S is a diagonal matrix containing the singular values, and U and $V^\dagger$ are unitary matrices obeying $UU^\dagger = 1$ and $V^\dagger V = 1$. We insert this decomposition into Eq. 1.20 and perform the sum over α_1 to obtain a new effective MPS matrix $\tilde{M}$ for site 2:

$$\begin{aligned}|\psi\rangle &= \sum_{s_1,\dots,s_L}\sum_{\alpha_1,\dots} M^{[1],s_1}_{1,\alpha_1} M^{[2],s_2}_{\alpha_1,\alpha_2} M^{[3],s_3}_{\alpha_2,\alpha_3}\dots |s_1\dots s_L\rangle \\ &= \sum_{s_1,\dots,s_L}\sum_{\alpha_2,\dots}\sum_{\sigma_1} U^{[1]}_{(s_1,1),\sigma_1}\Big(\sum_{\alpha_1} S_{\sigma_1,\sigma_1} V^{\dagger}_{\sigma_1,\alpha_1} M^{[2],s_2}_{\alpha_1,\alpha_2}\Big) M^{[3],s_3}_{\alpha_2,\alpha_3}\dots |s_1\dots s_L\rangle \\ &= \sum_{s_1,\dots,s_L}\sum_{\alpha_2,\dots}\sum_{\sigma_1} A^{[1],s_1}_{1,\sigma_1} \tilde{M}^{[2],s_2}_{\sigma_1,\alpha_2} M^{[3],s_3}_{\alpha_2,\alpha_3}\dots |s_1\dots s_L\rangle.\end{aligned}$$

Note that we have also renamed and reshaped $U^{[1]}_{(s_1,1),\sigma_1} \to A^{[1],s_1}_{1,\sigma_1}$ in the last line. This automatically guarantees that $A^{[1]}$ is left-canonical, c.f. Eq. 1.18. This procedure can now be repeated for the next matrix, $\tilde{M}^{[2]}$. It is first reshaped into a matrix, by grouping the left and physical indices: $\tilde{M}^{[2],s_2}_{\sigma_1,\alpha_2} \to \tilde{M}^{[2]}_{(s_2,\sigma_1),\alpha_2}$. Next, we perform a SVD of this matrix, multiply $S_{\sigma_2,\sigma_2}V^{\dagger}_{\sigma_2,\alpha_2}$ onto $M^{[3]}$ to obtain $\tilde{M}^{[3]}$ and reshape $U^{[2]}_{(s_2,\sigma_1),\sigma_2}$ back into $A^{[2],s_2}_{\sigma_1,\sigma_2}$. Iterating this procedure until the very last site brings the full MPS into left-canonical form. When SVD'ing the last site, the $S_{1,1}V^{\dagger}_{1,1}$ is simply a scalar since $\tilde{M}^{[L]}$ is a column vector. It determines the normalization of the wavefunction, and hence should be equal to 1. The reshaping of the tensors and the SVD can also be done easily in `Python`, shown below in the code snippet.

```python
# ...
# We are going to SVD the fourth MPS tensor, getting it into
# a left-canonical form. It is important here that the
# legs to be reshaped are neighboring indices.
reshapedM = np.reshape( M[3], (d*D, D) )
# We force the SVD to keep the full matrices for simplicity here.
# In an actual implementation, truncating the matrices is a
# beneficial thing to do.
U,S,Vdag  = np.linalg.svd( reshapedM, full_matrices=1 )
# The new M[3] is now set to be the reshaped U matrix, auto-
# matically being in left-canonical form.
M[3]      = np.reshape( U, (d,D,D) )
# We should now multiply S and Vdag, and contract the result
# with tensor M[4]. Then we repeat these steps.
```

If we want to generate a right-canonical MPS, we start from the last site and regroup the *right* index with the physical index before performing the singular value decomposition. The new matrices are then formed from the $V^\dagger$ matrices of the SVD, automatically guaranteeing the right-canonical form.

Before discussing algorithms that can be based on MPS representations, it is worth introducing a slightly different notation for the MPS matrices that facilitates working with the left- and right-canonical form. In this new notation, the MPS is written as

$$|\psi\rangle = \sum_{s_1,\dots,s_L} \Gamma^{[1],s_1}\Lambda^{[1]}\Gamma^{[2],s_2}\Lambda^{[2]}\dots\Gamma^{[L-1],s_{L-1}}\Lambda^{[L-1]}\Gamma^{[L],s_L}|s_1\dots s_L\rangle. \tag{1.21}$$

The $\Lambda^{[i]}$ matrices are diagonal with entries corresponding to the Schmidt values for a bipartitioning after site i. These are of course exactly the singular values obtained from the SVD of matrix $M^{[i]}$. In fact, the easiest way of obtaining this representation is by first constructing a fully left-canonical MPS, keeping track of the singular values and storing them as $\Lambda^{[i]}$. Then we may insert the identities $\Lambda^{[i]}\left(\Lambda^{[i]}\right)^{-1}$ in between the $A^{[i]}$ and $A^{[i+1]}$ matrices, absorbing the inverse matrix into $A^{[i+1]}$ and renaming it $\Gamma^{[i+1]}$. Notice though, that we have left out the final $\Lambda^{[L]}$. This is because, as mentioned before, the last tensor is a column vector and hence $\Lambda^{[L]}$ is simply a scalar. For periodic boundary conditions this would not be the case and we benefit from including a $\Lambda[L]$ matrix.

The elegancy of this notation is that we can easily convert it to left- or right-canonical form as follows. We have already seen, by construction, that the left-canonical matrices can be formed via $A^{[i],s_i} = \Gamma^{[i],s_i}\Lambda^{[i]}$. In order to construct right-canonical matrices, we simply have to contract in the opposite direction: $B^{[i],s_i} = \Lambda^{[i-1]}\Gamma^{[i],s_i}$.

TRUNCATION OF AN MPS

So far, all of the above discussion has been about matrix product states with a general bond dimension D. We have also seen that groundstates of gapped 1D Hamiltonians are efficiently represented by MPS states, meaning that the bond dimension needed does not scale exponentially with the system size (area law). In practice, however, even though the bond dimension has a favourable scaling we will most likely want to limit it to a maximum value. For a given wavefunction that would require a bond dimension D to be represented exactly, we therefore ask how big of an error we make by limiting the allowed bond dimension to $\chi < D$. The answer is given by the singular values obtained from bringing the state into canonical form. If the MPS is in canonical form, the singular values correspond to the Schmidt coefficients, which are empirically known to fall of exponentially in amplitude for groundstate wavefunctions. Hence truncating the bond dimension can be done in a controlled fashion, by e.g. deciding on a threshold below which the singular values are discarded. A typical threshold number is 10^{-8}, although even smaller threshold values may be taken as well. The truncation error is defined by summing all of the discarded singular values:

$$\epsilon_{\text{trunc}} = \sum_{\lambda_i < 10^{-8}} \lambda_i .$$

If one is more flexible in terms of time and memory requirements of numerical simulations, an alternative route for simulations is fixing the maximal truncation error and adjusting the bond dimension to reach this goal.

INFINITE MPS

There is a variant of matrix product states that will allow us to simulate systems directly in the thermodynamic limit of infinite system size. For this we have to assume translational invariance by introducing a unit cell of size L. For simplicity and for use in the (i)TEBD algorithm, we choose $L = 2$. The MPS wavefunction then reads

$$|\psi\rangle = \sum_{s_1,s_2} \text{Tr}\left(\Gamma^{[1],s_1}\Lambda^{[1]}\Gamma^{[2],s_2}\Lambda^{[2]}\right)|s_1 s_2\rangle.$$

The difference to a 2-site periodic system lies in the canonical form. If we choose the MPS matrices such that Eq. 1.19 is fulfilled, measurements performed on this system can be thought of as having been contracted ‘from infinity’. Seen in the opposite way, the fact that the dominant eigenvalue of the transfer matrix is unity means we could add it as many times as we want to construct a larger and larger system.

1.2. MPS ALGORITHMS

1.2.1. DMRG

The density matrix renormalization group (DMRG) algorithm was initially not formulated using matrix product states. It lends itself very well for such a formulation however. In this section we will shortly highlight the essence of the DMRG algorithm, but will not go into details of how the actual underlying implementation works. The main algorithm that has been used for the projects described in this thesis is the (i)TEBD algorithm, that will be discussed in more detail next.

The goal of the DMRG algorithm is to find the MPS with the lowest energy for a given local Hamiltonian $H = \sum_i h_{i,i+1}$. Each of the operators $h_{i,i+1}$ is a two-site operator. The algorithm can be extended easily to operators involving more than two sites, at the cost of computational complexity. As a three-phase process, DMRG can be summarized as follows:

1. Start with a randomly initialized (but normalized) MPS wavefunction
$$|\psi\rangle = |\psi[M^{[1]}, M^{[2]}, \ldots, M^{[L]}]\rangle$$
In this notation, we emphasise that the wavefunction is a function of the M-matrices.
2. We choose a site j, and replace the MPS matrix on that site by a new one labeled X:
$$|\psi[X]\rangle = |\psi[M^{[1]}, \ldots, M^{[j-1]}, X, M^{[j+1]}, \ldots M^{[L]}]\rangle.$$
We then evaluate the energy of this state:
$$E = \frac{\langle\psi[X]|H|\psi[X]\rangle}{\langle\psi[X]|\psi[X]\rangle}.$$
At this point, the DMRG algorithm provides a direct way of getting the matrix X_{min} that minimizes the energy. We then set $M^{[s]} = X_{\text{min}}$.
3. Instead of choosing a random site s, we perform a ‘sweep’: start at site 1 and optimize the MPS matrix. Then move to site 2 and continue all the way up to site L. Then move back all the way to the first site, and we have completed one sweep. This sweeping is then iterated until the energy has converged.

The main part in this algorithm is evidently the second step, i.e. how does one find X_{min}? Consider first the normalization of $|\psi[X]\rangle$. We make sure the MPS is in canonical form, so that the normalization can be computed from X only:

$$\langle\psi[X]|\psi[X]\rangle = \sum_{s_j \alpha\beta} X^{[j],s_j}_{\alpha\beta} \left(X^{[j],s_j}_{\alpha\beta}\right)^* = \mathbf{x}\cdot\mathbf{x},$$

where $\mathbf{a}\cdot\mathbf{b} = \sum_i a_i b_i^*$. This is a useful shorthand notation for the normalization. Next, we turn to evaluating the Hamiltonian:

$$\langle\psi[X]|H|\psi[X]\rangle = \sum_i \langle\psi[X]|h_{i,i+1}|\psi[X]\rangle.$$

If we have replaced site s by the matrix X, there are four general cases that occur for the evaluation of the Hamiltonian. In the first case, i and $i+1$ are both to the left of site j as depicted in Fig. 1.14. We can collect the whole left hand side into an object referred to as the *left environment* of site s, denoted $L_{\alpha\alpha'}$. This contraction can be written out as follows:

$$\text{case 1} = \sum_{s_j,\beta,\alpha,\alpha'} X^{[j],s_j}_{\alpha\beta} L_{\alpha\alpha'} \left(X^{[j],s_j}_{\alpha'\beta}\right)^* = \sum_{s_j,\beta,\alpha,\alpha'} X^{[j],s_j}_{\alpha\beta} \left(\tilde{X}^{[j],s_j}_{\alpha'\beta}\right)^* = \mathbf{x}\cdot\tilde{\mathbf{x}}. \qquad (1.22)$$

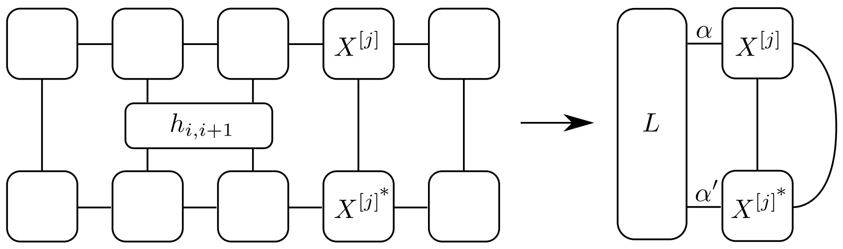


Figure 1.14: Definition of the left environment in evaluating the Hamiltonian on an MPS.

This can be further simplified by realizing that $\tilde{\mathbf{x}} = \mathbf{M}\mathbf{x}^*$, in which $\mathbf{M} = \hat{1} \otimes L_{\alpha\alpha'} \otimes \hat{1}$. The first identity operator acts on the physical index of X, whereas the last acts on the right leg. The full expression then turns into

$$\text{case } 1 = \mathbf{x} \cdot \mathbf{M}^* \mathbf{x}.$$

For the other three cases, corresponding to $i = j$, $i + 1 = j$ or both i and $i + 1$ on the right side of j can be treated similarly. Those cases also lead to a matrix $\mathbf{M}$, and the results can be summed into $\mathbf{M}_{\text{tot}}$. In the end then, we are trying to compute $\min_{|\mathbf{x}|=1} \mathbf{x} \cdot \mathbf{M}^*_{\text{tot}} \mathbf{x}$, which is solved by nothing but the smallest eigenvalue of $\mathbf{M}_{\text{tot}}$ (by the Min-Max theorem). So once we know how to compute $\mathbf{M}_{\text{tot}}$, we know how to get X.

1.2.2. TEBD AND ITEBD

The time evolving block decimation (TEBD) algorithm is an algorithm that was designed for performing time evolution of a wavefunction written in MPS representation [5–7]. Roughly speaking, it is simply an application of the fact that any wavefunction can be brought into an MPS form by repeated singular value decompositions. TEBD provides an efficient way of simulating the time evolution given by the Schrödinger equation:

$$\frac{\partial}{\partial t} |\psi\rangle(t) = -iH|\psi\rangle(t),$$

where, for simplicity, we consider a time-independent Hamiltonian. The solution to this equation is formally given by

$$|\psi\rangle(t) = e^{-iHt} |\psi\rangle(0),$$

from which we define the time evolution operator $U = \exp -iHt$. Since we know how to apply operators to an MPS representation (by contraction over the physical indices), we already have the neccessary ingredients for implementing TEBD:

- Apply the time evolution operator (for an infinitesimal amount of time) to the MPS to generate a new MPS.
- This new MPS may not be in a canonical form, and may require a larger bond dimension. So we perform an SVD to truncate the resulting MPS, and repeat.

REFERENCES

[1] R. Orús, Annals of Physics **349**, 117 (2014).

[2] J. Eisert, M. Cramer, and M. B. Plenio, Rev. Mod. Phys. **82**, 277 (2010), URL.

[3] M. B. Plenio, Phys. Rev. Lett. **95**, 090503 (2005), URL.

[4] U. Schollwöck, Annals of Physics **326**, 96 (2011), URL.

[5] G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003), URL.

[6] G. Vidal, Phys. Rev. Lett. **98**, 070201 (2007), URL.

[7] R. Orús and G. Vidal, Phys. Rev. B **78**, 155117 (2008), URL.