Matrix product states: iterative Schmidt decomposition of a wavefunction

Jeanne Colbois

May 16, 2020

1 Introduction

The aim is to write a quantum ground state wave function as a matrix product state. The main references about this kind of algorithms are [1] (for the idea behind TEBD), [2] (for the idea behind iTEBD), [3], if you need to think about the notions of canonical form of an MPS or contract a tensor using boundary-MPS-like techniques similar to iTEBD but where your gates are not unitary.

2 Writing a 1D quantum state as a matrix product state

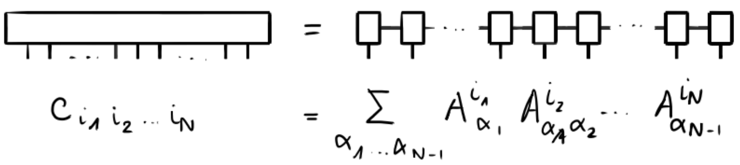
The idead of TEBD, iTEBD and all other tensor networks methods for 1D quantum systems is to approximate the state $|\psi\rangle$ by a Matrix Product State of finite bond dimension [1]. The expression of a wavefunction as a matrix product state was derived in many papers, but for the sake of clarity, let us derive it once more here.

2.1 Definition

Consider a quantum state $|\psi\rangle$. It can be seen as

$$|\psi\rangle = \sum_{i_1,i_2,\ldots} c_{i_1,i_2\ldots} |i_1, i_2, \ldots\rangle \tag{1}$$

where the i_j index the physical state (they span the local Hilbert space) and $c_{i_1,i_2,\ldots}$ is a huge tensor with as many legs as there are sites (or local Hilbert spaces). Typically $i \to \sigma$, i.e. the local Hilbert space corresponds to a spin state. This tensor can be represented as an oblong object with as many legs as there are sites. The aim of our demonstration will be to show that by performing Schmidt decompositions iteratively, one can bring this huge tensor in the form of a product of many much smaller tensors, as depicted in Fig. 1. Here, we follow the derivation by Vidal [2], but we adapt it to compressing the quantum state starting from the left, and we consider open boundary conditions, that is, two virtual legs remain left and right of the tensor.



$c_{i_1 i_2 \cdots i_N}$
$= \sum_{\alpha_1 \cdots \alpha_{N-1}} A^{i_1}_{\alpha_1} A^{i_2}_{\alpha_1 \alpha_2} \cdots A^{i_N}_{\alpha_{N-1}}$

Figure 1: *Matrix product form of a quantum state. The rectangles depict tensors, with as many indices as they have legs. The legs connecting two tensors are summed over. The legs going down denote physical indices $\{i_1, \ldots, i_n\}$. This way, the tensor on the left stands for $c_{i_1,i_2,\ldots}$, while the right hand side corresponds to a product of tensors $\sum_{\alpha_2,\alpha_3,\ldots} A^{(i_1)}_{\alpha_2} A^{(i_2)}_{\alpha_2\alpha_3} \cdots$*

2.2 Tensor notations

There is a very useful notation to work with wavefunctions, tensors and matrices, which is called the tensor notation. In Fig. 2, the main items are identified. In general, a mathematical object is represented as a box, and its indices (each index spanning a dimension) are represented by "sticks", called legs, sticking out of the box. Therefore, a scalar would be just a box with no legs. A vector, described by v_i, that is an object with one index i, is a box with one leg. A matrix $A_{i,j}$ is a box with two legs, and in general, a tensor with k indices (we say as well a tensor of *rank* k, not to be confused with the rank of a matrix) is described by a box with k legs.

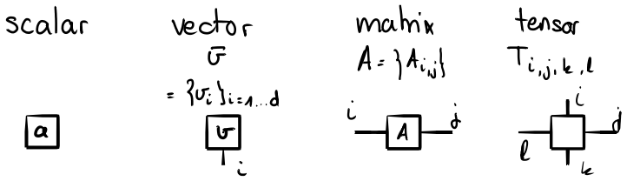


Figure 2: *Elements of tensor notation: any object is a box, and the number of "legs" attached to each box corresponds to the number of indices for the object: 0 is a scalar, 1 is a vector, 2 is a matrix, and k is a rank-k tensor*

The main operation that we use on matrices, tensors or vectors is the multiplication. For instance, the multiplication of a vector v_j by a matrix $A_{i,j}$, denoted

$$x_i := \sum_j A_{i,j} v_j \tag{2}$$

is illustrated in Fig. 3. The sum over the index j which is shared between A and v is denoted by connecting the corresponding legs of the corresponding tensors. A leg can only be connected to one other leg, and they have to have the same dimension (i.e. the same number of degrees of freedom), such that the sum makes sense.

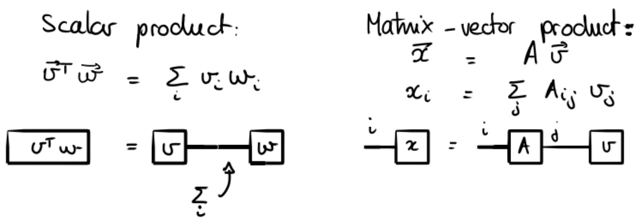


Figure 3: *The basic operations between vectors and matrices, but represented in tensor notation. A leg which is shared between two objects corresponds to summing over the corresponding index. Notice how the scalar product indeed gives a scalar, i.e. a box with no legs, and the matrix-vector product gives a box with one leg, i.e. a vector.*

In general, a multiplication of two tensors such as

$$M_{i,j,k} = \sum_{l,m} A_{i,j,l,m} B_{l,m,k} \tag{3}$$

is called a *tensor contraction.* A tensor contraction can involve more than two tensors, for instance:

$$M_{i,j,k,l} = \sum_{m,n,o} A_{i,j,n,o} B_{n,m,k} C_{m,o,l}, \tag{4}$$

as illustrated in Fig. 4.

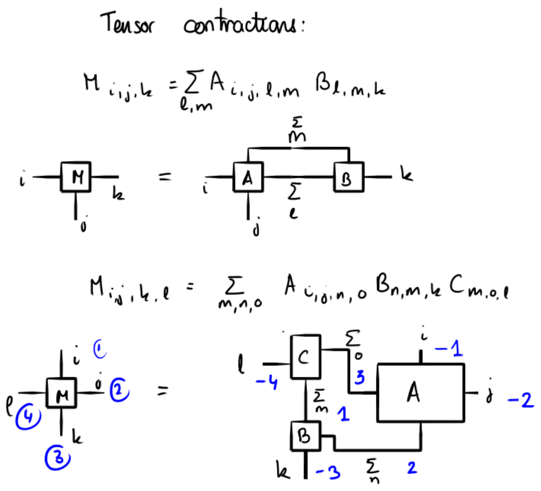


Figure 4: *Examples of tensor contractions, corresponding to Eqs. 3 and 4. In the second case, the blue labels correspond to the entries that have to be given to the* `ncon.m` *function to perform the contraction. The circled indices give the order of the indices in the output of ncon (corresponding to the negative indices on the right of the assignment).*

2.3 Usage of `ncon`

The function `ncon.m`, developed by some of the Tensor Network “kings” Robert N. C. Pfeifer, Glen Evenbly, Sukhwinder Singh and Guifre Vidal [4], is designed to make tensor contraction easy. The basic idea is that one has to give `ncon`

1. A cell array of the tensors to contract (e.g. `{A,B,C}` for the contraction in equation 4)
2. A cell array of legs indices, given by
   a) There has to be one array per tensor to contract (e.g. 3 in our example)
   b) For each tensor, the array has to associate a contraction index to each leg of the tensor (in our example, the first array, associated with A, will contain 4 numbers, the second, associated with B, will have 3 numbers, and the third, associated with C, will have 3 numbers as well)
   c) Two legs that are contracted have to be given the same index (e.g., the 3rd leg of A and the first of B in our example)
   d) An index can only be used for one contraction
   e) A leg which is not contracted has to be given a negative index
   f) The order in which the contractions will be performed is given by the order of the positive indices
   g) The order of the indices for the output tensor is given by the order of the negative indices

   In our example, this would work:
   ```
   M=ncon({A,B,C},{[-1 -2 2 3],[2 1 -3],[1 3 -4]})
   ```

2.4 Writing the wavefunction as a matrix product state

In these notes, we will use extensively the fact that a *Schmidt decomposition* of a wavefunction can be written as a singular value decomposition of an associated tensor. It is therefore not necessary to know about Schmidt decomposition to understand the steps below. However, the proof of the Schmidt decomposition is given at the end of the notes for completeness, sec. 2.5.

To write a waveunfction as an MPS, we start between sites 1 and 2 and then iterate until the wavefunction is written as a matrix product state. The corresponding steps of the algorithm, explicited in the following equations, are illustrated in tensor notation in Figs. 5, 6, 7.

We consider a wavefunction $|\psi\rangle$ on N sites with open boundary conditions, namely two virtual indices. In the following, we denote physical indices, corresponding to physical degrees of freedom, by i_n (where n goes through the sites indices, and i_n goes through the local Hilbert space basis). Virtual indices are denoted by α_n.

$$|\psi\rangle_{\alpha_1,\alpha_{N+1}} = \sum_{i_1,i_2,\dots} \Psi_{\alpha_1,i_1,i_2,\dots,i_N,\alpha_{N+1}} |i_1,i_2,\dots,i_N\rangle \otimes |\alpha_1,\alpha_{N+1}\rangle \tag{5}$$

$$\overset{\text{reshape}}{=} \sum_{i_1,i_2,\dots} \Psi_{(\alpha_1,i_1),(i_2,\dots,i_N,\alpha_{N+1})} |i_1,i_2,\dots,i_N\rangle \otimes |\alpha_1,\alpha_{N+1}\rangle \tag{6}$$

$$\overset{\text{svd}}{=} \sum_{i_1,i_2,\dots} \sum_{\beta,\alpha_2} U^{(1)}_{(\alpha_1,i_1),\beta} s^{(1)}_{\beta,\alpha_2} V^{(1)}_{\alpha_2,(i_2,\dots,i_N,\alpha_{N+1})} |i_1,i_2,\dots,i_N\rangle \otimes |\alpha_1,\alpha_{N+1}\rangle \tag{7}$$

Notice that we are assuming a finite rank for the Schmidt decomposition. In general, what we will do is put a threshold on the singular values, and drop all the singular values smaller than $1e-14$ (and the corresponding columns and rows in U, s and V, respectively.

We can now perform

$$\tilde{A}^{(1)}_{(\alpha_1,i_1),\alpha_2} := \sum_\beta U^{(1)}_{(\alpha_1,i_1),\beta} s^{(1)}_{\beta,\alpha_2} \tag{8}$$

and then reshape this tensor

$$A^{(1),i_1}_{\alpha_1,\alpha_2} \overset{\text{reshape}}{=} \tilde{A}_{(\alpha_1,i_1),\alpha_2}. \tag{9}$$

$V^{(1)}$ can be reshaped as well, which meas that after this first step we get:

$$|\psi\rangle_{\alpha_1,\alpha_{N+1}} = \sum_{i_1,i_2,\dots} \sum_{\alpha_2} A^{(1),i_1}_{\alpha_1,\alpha_2} V^{(1)}_{\alpha_2,i_2,\dots,i_N,\alpha_{N+1}} |i_1,i_2,\dots\rangle \otimes |\alpha_1,\alpha_{N+1}\rangle\,. \tag{10}$$

Eqs. 5 to 10 are illustrated in tensor notation in Fig. 5.

We can now iterate similar steps on V, saving at step t a new tensor $A^{(t)}$. For simplicity, we show the second step, and then the last step. The second step is:

$$|\psi\rangle_{\alpha_1,\alpha_{N+1}} \overset{\text{reshape } V^{(1)}}{=} \sum_{i_1,i_2,\dots} \sum_{\alpha_2} A^{(1),i_1}_{\alpha_1,\alpha_2} V^{(1)}_{(\alpha_2,i_2),(i_3,\dots,i_N,\alpha_{N+1})} |i_1,i_2,\dots\rangle \otimes |\alpha_1,\alpha_{N+1}\rangle \tag{11}$$

$$\overset{\text{svd } V^{(1)}}{=} \sum_{i_1,i_2,\dots} \sum_{\alpha_2} A^{(1),i_1}_{\alpha_1,\alpha_2} \sum_{\beta',\alpha_3} U^{(2)}_{(\alpha_2,i_2),\beta'} s^{(2)}_{\beta',\alpha_3} V^{(2)}_{\alpha_3,(i_3,\dots,i_N,\alpha_{N+1})} |i_1,i_2,\dots\rangle \otimes |\alpha_1,\alpha_{N+1}\rangle\,, \tag{12}$$

followed by:

$$A^{(2),i_2}_{\alpha_2,\alpha_3} \overset{\text{reshape}}{=} \sum_{\beta'} U^{(2)}_{(\alpha_2,i_2),\beta'} s^{(2)}_{\beta',\alpha_3}, \tag{13}$$

which gives:

$$|\psi\rangle_{\alpha_1,\alpha_{N+1}} \overset{\text{reshape } V^{(2)}}{=} \sum_{i_1,i_2,\dots} \sum_{\alpha_2,\alpha_3} A^{(1),i_1}_{\alpha_1,\alpha_2} A^{(2),i_2}_{\alpha_2,\alpha_3} V^{(2)}_{\alpha_3,i_3,\dots,i_N,\alpha_{N+1}} |i_1,i_2,\dots,i_N\rangle \otimes |\alpha_1,\alpha_{N+1}\rangle\,. \tag{14}$$

Equations 11 to 14 are depicted in Fig. 6.

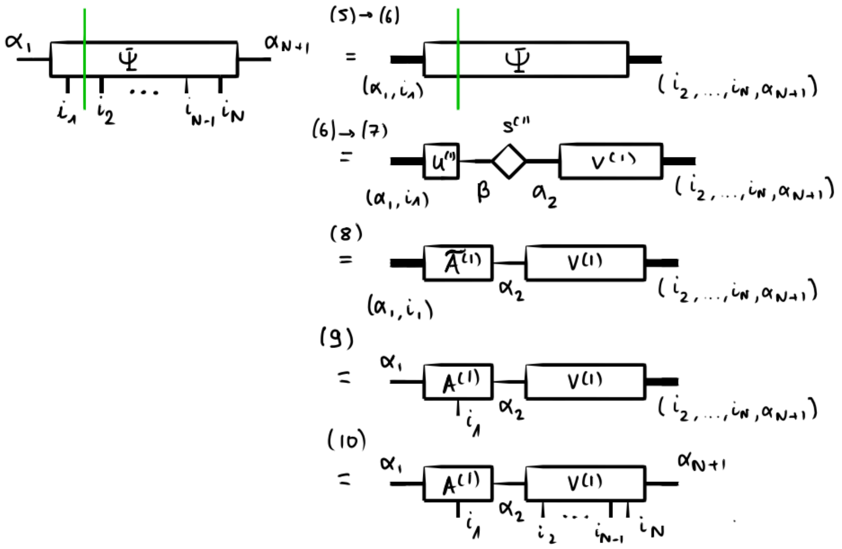


Figure 5: *The first step in the procedure for the iterative Schmidt decomposition. The green line shows where the SVD decomposition is going to be performed next. Bold legs illustrate grouped indices, as written next. The diamond describes the diagonal singular values matrix. The numbers above the equalities correspond to the equations in the main text.*

Iterating this, at the previous to last step, one gets:

$$|\psi\rangle_{\alpha_1,\alpha_{N+1}} = \sum_{i_1,i_2,\ldots} \sum_{\alpha_2,\alpha_3\ldots,\alpha_{N-1}} A^{(1),i_1}_{\alpha_1,\alpha_2} A^{(2),i_2}_{\alpha_2,\alpha_3} \cdots$$
$$\cdots A^{(N-2),i_{N-2}}_{\alpha_{N-2},\alpha_{N-1}} V^{(N-2)}_{\alpha_{N-1},i_{N-1},i_N,\alpha_{N+1}} |i_1, i_2, \ldots, i_N\rangle \otimes |\alpha_1, \alpha_{N+1}\rangle \,. \quad (15)$$

The last step is still very similar, the only difference being that one will have to save the last V tensor:

$$|\psi\rangle_{\alpha_1,\alpha_{N+1}} \overset{\text{reshape } V^{(N-2)}}{=} \sum_{i_1,i_2,\ldots} \sum_{\alpha_2,\alpha_3\ldots,\alpha_{N-1}} A^{(1),i_1}_{\alpha_1,\alpha_2} A^{(2),i_2}_{\alpha_2,\alpha_3} \cdots$$
$$\cdots A^{(N-2),i_{N-2}}_{\alpha_{N-2},\alpha_{N-1}} V^{(N-2)}_{(\alpha_{N-1},i_{N-1}),(i_N,\alpha_{N+1})} |i_1, i_2, \ldots, i_N\rangle \otimes |\alpha_1, \alpha_{N+1}\rangle \quad (16)$$

then

$$|\psi\rangle_{\alpha_1,\alpha_{N+1}} \overset{\text{svd } V^{(N-2)}}{=} \sum_{i_1,i_2,\ldots} \sum_{\alpha_2,\alpha_3\ldots,\alpha_{N-1}} \Big(A^{(1),i_1}_{\alpha_1,\alpha_2} A^{(2),i_2}_{\alpha_2,\alpha_3} \cdots A^{(N-2),i_{N-2}}_{\alpha_{N-2},\alpha_{N-1}}$$
$$\cdot \sum_{\beta,\alpha_N} U^{(N-1)}_{(\alpha_{N-1},i_{N-1}),\beta} s^{(N-1)}_{\beta,\alpha_N} V^{(N-1)}_{\alpha_N,(i_N,\alpha_{N+1})} |i_1, i_2, \ldots, i_N\rangle \otimes |\alpha_1, \alpha_{N+1}\rangle \Big) \,. \quad (17)$$

Defining $A^{(N-1)}$ similarly as before

$$A^{(N-1),i_{N-1}}_{\alpha_{N_1},\alpha_N} \overset{\text{reshape}}{=} \sum_{\beta} U^{(N-1)}_{(\alpha_{N-1},i_{N-1}),\beta} s^{(N-1)}_{\beta,\alpha_N} \,, \quad (18)$$

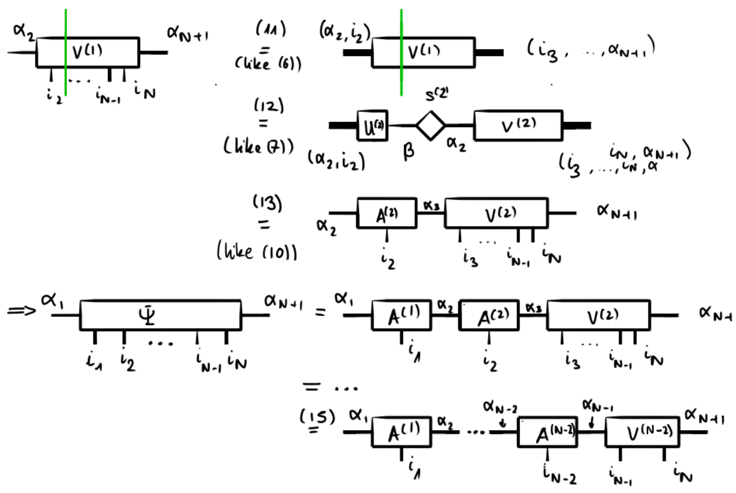


Figure 6: *The second step in the procedure for the iterative Schmidt decomposition. The input is the $V^{(1)}$ tensor obtained at the first step. The green line shows where the SVD decomposition is going to be performed next. The numbers above the equalities correspond to the equations in the main text.*

one gets

$$
\begin{aligned}
|\psi\rangle_{\alpha_1,\alpha_{N+1}} = \sum_{i_1,i_2,\ldots} \sum_{\alpha_2,\alpha_3\ldots,\alpha_N} & A^{(1),i_1}_{\alpha_1,\alpha_2} A^{(2),i_2}_{\alpha_2,\alpha_3} \cdots \\
& \cdots A^{(N-2),i_{N-2}}_{\alpha_{N-2},\alpha_{N-1}} A^{(N-1),i_{N-1}}_{\alpha_{N_1},\alpha_N} V^{(N-1)}_{\alpha_N,(i_N,\alpha_{N+1})} |i_1, i_2, \ldots, i_N\rangle \otimes |\alpha_1, \alpha_{N+1}\rangle \,. \quad (19)
\end{aligned}
$$

This is shown in Fig. 7.

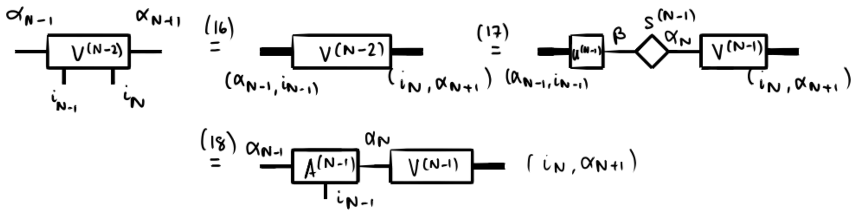


Figure 7: *The previous to last step in the procedure for the iterative Schmidt decomposition. The input is the $V^{(N-2)}$ tensor obtained at the step before that. The numbers above the equalities correspond to the equations in the main text.*

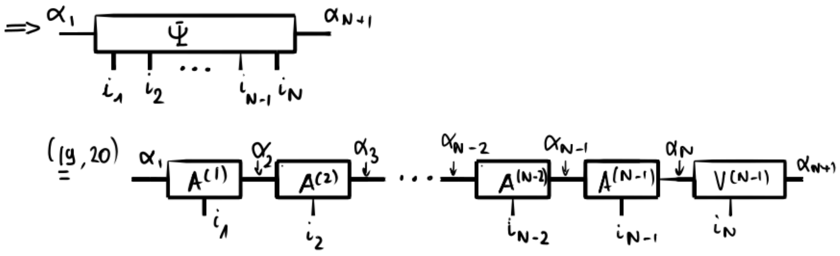
Finally, reshaping

$$
V^{(N-1)}_{\alpha_N,(i_N,\alpha_{N+1})} \to V^{(N-1),i_N}_{\alpha_N,\alpha_{N+1}} \quad (20)
$$

one get

$$|\psi\rangle_{\alpha_1,\alpha_{N+1}} = \sum_{i_1,i_2,\dots} \sum_{\alpha_2,\alpha_3\dots,\alpha_N} A^{(1),i_1}_{\alpha_1,\alpha_2} A^{(2),i_2}_{\alpha_2,\alpha_3} \cdots$$
$$\cdots A^{(N-2),i_{N-2}}_{\alpha_{N-2},\alpha_{N-1}} A^{(N-1),i_{N-1}}_{\alpha_{N_1},\alpha_N} V^{(N-1),i_N}_{\alpha_N,\alpha_{N+1})} |i_1,i_2,\dots,i_N\rangle \otimes |\alpha_1,\alpha_{N+1}\rangle\,, \quad (21)$$

which is a matrix product state form for the wavefunction $|\psi\rangle_{\alpha_1,\alpha_{N+1}}$, as illustrated in Fig. 8.



=>
α_1
Ψ
α_{N+1}
i_1 i_2 ... i_{N-1} i_N
(19, 20)
=
α_1 α_2 α_3 α_{N-2} α_{N-1} α_N α_{N+1}
$A^{(1)}$ $A^{(2)}$ ... $A^{(N-2)}$ $A^{(N-1)}$ $V^{(N-1)}$
i_1 i_2 i_{N-2} i_{N-1} i_N

Figure 8: *The last step in the procedure for the iterative Schmidt decomposition. In this last step, $V^{(N-1)}$ is simply reshaped. The numbers above the equalities correspond to the equations in the main text.*

2.5 Schmidt decomposition

Theorem 2.1 (Schmidt decomposition)**.** *Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be Hilbert spaces of finite dimensions n_1 and n_2 (w.l.o.g. $n_1 \geq n_2$). Then for any $|\varphi\rangle$ in $\mathcal{H}_1 \otimes \mathcal{H}_2$, there exist orthonormal sets $\{|\Phi^{\triangleleft}_1\rangle,\dots,|\Phi^{\triangleleft}_{n_2}\rangle\} \subset \mathcal{H}_1$ and $\{|\Phi^{\triangleright}_1\rangle,\dots,|\Phi^{\triangleright}_{n_2}\rangle\} \subset \mathcal{H}_2$ such that*

$$|\varphi\rangle = \sum_{i=1}^{n_2} s_i\, |\Phi^{\triangleleft}_i\rangle \otimes |\Phi^{\triangleright}_i\rangle \quad (22)$$

where $s_i \in \mathbb{R}$, $s_i \geq 0$, and $\{s_i\}_{1,n_2}$ is uniquely determined by $|\varphi\rangle$.

Proof. The Schmidt decomposition can be seen as an application of the Singular Value Decomposition (SVD) in the context of finite-dimensional Hilbert spaces. To be able to use this, one defines orthonormal bases $\{|\varepsilon^{\triangleleft}_1\rangle,\dots,|\varepsilon^{\triangleleft}_{n_1}\rangle\} \subset \mathcal{H}_1$ and $\{|\varepsilon^{\triangleright}_1\rangle,\dots,|\varepsilon^{\triangleright}_{n_2}\rangle\} \subset \mathcal{H}_2$. Then, by definition of the tensor product, it exists $m_{i,k}$ such that

$$|\varphi\rangle = \sum_{i=1,k=1}^{n_1,n_2} m_{i,k}\, |\varepsilon^{\triangleleft}_i\rangle \otimes |\varepsilon^{\triangleright}_k\rangle\,. \quad (23)$$

This allows to describe $|\varphi\rangle$ through the $n_1 \times n_2$ matrix $M = \{m_{i,k}\}$. Using the SVD result, we are going to prove the Schmidt decomposition. We know that it exists two unitary matrices ($n_1 \times n_1$ U and $n_2 \times n_2$ V) such that $M = U\Sigma V^\dagger$ where Σ is an $n_1 \times n_2$ diagonal non-negative matrix. Let $\{s_i\}_{i=1,\dots,n_2}$ bet the diagonal elements of Σ.

Thus we can write

$$|\varphi\rangle = \sum_{i=1,k=1}^{n_1,n_2} m_{i,k} \, |\varepsilon_i^{\triangleleft}\rangle \otimes |\varepsilon_k^{\triangleright}\rangle \tag{24}$$

$$= \sum_{i=1,k=1,\alpha}^{n_1,n_2,n_2} U_{i,\alpha} s_\alpha (V^\dagger)_{\alpha,k} \, |\varepsilon_i^{\triangleleft}\rangle \otimes |\varepsilon_k^{\triangleright}\rangle \, . \tag{25}$$

Defining

$$|\Phi_\alpha^{\triangleleft}\rangle := \sum_{i=1}^{n_1} U_{i,\alpha} \, |\varepsilon_i^{\triangleleft}\rangle \qquad \alpha = 1 \dots n_2 \tag{26}$$

and

$$|\Phi_\alpha^{\triangleright}\rangle := \sum_{k=1}^{n_2} (V^\dagger)_{\alpha,k} \, |\varepsilon_k^{\triangleright}\rangle \qquad \alpha = 1 \dots n_2, \tag{27}$$

we get

$$|\varphi\rangle = \sum_{i=1}^{n_2} s_i \, |\Phi_i^{\triangleleft}\rangle \otimes |\Phi_i^{\triangleright}\rangle \, . \tag{28}$$

It is easy to see that

$$\left\langle \Phi_\alpha^{\triangleright} \middle| \Phi_\beta^{\triangleright} \right\rangle = \sum_{i=1,k=1}^{n_2} V_{i,\alpha} (V^\dagger)_{\beta,k} \, \langle \varepsilon_i^{\triangleright} | \varepsilon_k^{\triangleright} \rangle \tag{29}$$

$$= (V^\dagger V)_{\beta,\alpha} = \delta_{\alpha,\beta}, \tag{30}$$

and similarly $\left\langle \Phi_\alpha^{\triangleleft} \middle| \Phi_\beta^{\triangleleft} \right\rangle = \delta_{\alpha,\beta}$. We can thus conclude that the orthonormal sets exists, which ends the proof of the Schmidt decomposition theorem. □

References

[1] Guifré Vidal. "Efficient Classical Simulation of Slightly Entangled Quantum Computations". en. In: *Physical Review Letters* 91.14 (Oct. 2003). ISSN: 0031-9007, 1079-7114. DOI: `10.1103/PhysRevLett.91.147902`.

[2] G. Vidal. "Classical Simulation of Infinite-Size Quantum Lattice Systems in One Spatial Dimension". In: *Phys. Rev. Lett.* 98.7 (Feb. 2007), p. 070201. DOI: `10.1103/PhysRevLett.98.070201`.

[3] R. Orús and G. Vidal. "Infinite time-evolving block decimation algorithm beyond unitary evolution". In: *Phys. Rev. B* 78.15 (Oct. 2008), p. 155117. DOI: `10.1103/PhysRevB.78.155117`.

[4] Robert N. C. Pfeifer et al. *NCON: A tensor network contractor for MATLAB*. 2014. arXiv: `1402.0939 [physics.comp-ph]`.