Lecture notes on DMRG (ISTPC Aussois 2026)

Mi-Song Dupuy

June 24, 2026

Contents

Introduction

These notes constitute a short introduction to the tensor train (TT) decomposition (also called matrix product state - MPS), with a particular focus on solving the many-body electronic Schrödinger equation. The beginning of TT can be tied to the density matrix renormalisation group [Whi92] (DMRG), although the connection with the TT/MPS ansatz has been made a few years later. Originally, DMRG has been applied to one-dimensional statistical physics systems with tremendous success, becoming the state-of-the-art numerical method to compute ground-state and low-excited states properties. It has then been tested for two-dimensional systems, where the question of the geometry of the tensor train, or the ordering of the sites has been difficult to overcome. It has also been successfully applied to quantum chemistry systems -with the name QC-DMRG (quantum chemistry-DMRG)-, where the question of the ordering of the sites is at first glance unclear.

The first part of the lecture notes is devoted to the electronic Schrödinger equation in the second quantisation and the introduction of the tensor to approximate.

The tensor train decomposition [OT09] is presented as a generalisation of the singular value decomposition for matrices, which is central in the characterisation of the low-rank approximation problem.

As quantum entropy is central in the ordering scheme for QC-DMRG [BLMR11], we introduce several notions of the quantum entropy as well as the connection with the TT/MPS approximation.

Finally, we address two points that explain the success of DMRG:

- Hastings area law [Has07] for one-dimensional system which proves that the TT/MPS approximation of a nearest neighbour Hamiltonian is at most *polynomial* in the system size;
- the DMRG algorithm and its polynomial scaling for electronic structure problems.

The content is inspired by the following texts on TT/MPS [Hac12, Hac14, Sch11, BSU16, UV20].

Chapter 1

Tensors in quantum chemistry

1.1 The many-body Schrödinger equation

1.1.1 The equation and its discretisation

Under the Born-Oppenheimer approximation, the ground-state of an electronic system with N electrons and N_{at} atoms with charges Z_K and located at the positions R_K, $1 \leq K \leq N_{\mathrm{at}}$ is given by the lowest eigenvalue

$$H^N \Psi_0^N = E_0^N \Psi_0^N, \tag{1.1.1}$$

where the operator H^N is the many-body electronic Schrödinger operator

$$H^N = \sum_{i=1}^{N} \Big(-\frac{1}{2}\Delta_{r_i} - \sum_{K=1}^{N_{\mathrm{at}}} \frac{Z_K}{|r_i - R_K|} \Big) + \sum_{1 \leq i < j \leq N} \frac{1}{|r_i - r_j|}, \tag{1.1.2}$$

and the wave function Ψ_0^N belongs to $\bigwedge^N L^2(\mathbb{R}^3 \times \mathbb{Z}_2)$.

Using the Rayleigh-Ritz principle, the eigenvalue problem can be rephrased as an optimisation problem

$$E_0^N = \min_{\substack{\Psi \in \bigwedge^N L^2(\mathbb{R}^3 \times \mathbb{Z}_2) \\ \|\Psi\|_{L^2}=1}} \langle \Psi, H^N \Psi \rangle. \tag{1.1.3}$$

A standard way to solve the eigenvalue problem (1.1.1) is to use a Galerkin scheme of the following form. Let $(\phi_i)_{1 \leq i \leq d}$ be an L^2-orthogonal family of $L^2(\mathbb{R}^3 \times \mathbb{Z}_2)$ and let

$$\mathcal{V}_N^d = \mathrm{Span}\Big(\phi_{i_1} \wedge \cdots \wedge \phi_{i_N} = \frac{1}{\sqrt{N!}} \det\big(\phi_{i_j}(x_k)\big), 1 \leq i_1 \leq \cdots \leq i_N \leq d \Big), \tag{1.1.4}$$

then the problem to solve numerically becomes

$$E_{0,L}^N = \min_{\substack{\Psi \in \mathcal{V}_N^d \\ \|\Psi\|_{L^2}=1}} \langle \Psi, H^N \Psi \rangle \geq E_0^N. \tag{1.1.5}$$

The approximate ground-state wave function is given by

$$\Psi_{0,d} = \sum_{1 \leq i_1 < \cdots < i_N \leq d} C_{i_1 \ldots i_N} \phi_{i_1} \wedge \cdots \wedge \phi_{i_N}. \tag{1.1.6}$$

The number of such coefficients is exponential in the number of electrons, hence a clever parametrisation of the coefficients as well as an insightful choice of the Galerkin basis is needed.

Several methods have been tried to sparsely parametrise the coefficients of the ground-state wave function:

- the configuration interaction (CI) method, which is a hierarchical truncation of the coefficients in (1.1.6);
- the coupled-cluster (CC) method, which relies on an intricate parametrisation of the wave function;
- the density matrix renormalisation group (DMRG) which is based on a tensor factorisation of the coefficients in the second quantisation.

1.1.2 Fock space

The discrete Fock space $\mathcal{F}_d$ is defined as the direct sum of the Galerkin spaces $\mathcal{V}_N^d$ given in Eq. (1.1.4)

$$\mathcal{F}_d = \mathcal{V}_0^d \oplus \mathcal{V}_1^d \oplus \cdots \oplus \mathcal{V}_d^d. \tag{1.1.7}$$

A general state $\mathbf{\Psi} \in \mathcal{F}_d$ is the collection $(\Psi^0, \Psi^1, \ldots, \Psi^d)$ where each Ψ^k is a wave function belonging to $\mathcal{V}_k^d$.

To move from $\mathcal{V}_k^d$ to its neighbour, the creation operator $(c_j^\dagger)_{1 \leq j \leq d}$ and the annihilation operator $(c_j)_{1 \leq j \leq d}$ are used. The annihilation operator c_j is the map such that

$$c_j \ : \ \begin{cases} \mathcal{V}_{k+1}^d \to \mathcal{V}_k^d \\ \phi_{i_1} \wedge \cdots \wedge \phi_{i_{k+1}} \mapsto \begin{cases} 0 & \text{if } \forall\, 1 \leq \ell \leq k+1, j \neq i_\ell \\ (-1)^{\ell-1} \phi_{i_1} \wedge \cdots \wedge \phi_{i_{\ell-1}} \wedge \phi_{i_{\ell+1}} \wedge \cdots \wedge \phi_{i_{k+1}}, & \text{if } j = i_\ell. \end{cases} \end{cases} \tag{1.1.8}$$

The annihilation operator c_j destroys a particle in the state ϕ_j if it exists and returns 0 otherwise. Likewise, the creation operator $c_j^\dagger$ is given by

$$c_j^\dagger \ : \ \begin{cases} \mathcal{V}_k^d \to \mathcal{V}_{k+1}^d \\ \phi_{i_1} \wedge \cdots \wedge \phi_{i_k} \mapsto (-1)^\ell \phi_{i_1} \wedge \cdots \wedge \phi_{i_\ell} \wedge \phi_j \wedge \phi_{i_{\ell+1}} \wedge \cdots \wedge \phi_{i_{k+1}}, \end{cases} \tag{1.1.9}$$

with $i_1 < \cdots < i_\ell < j < i_{\ell+1} < \cdots < i_k$. Note that by antisymmetry, if $j = i_\ell$ for some ℓ, then $c_j^\dagger(\phi_{i_1} \wedge \cdots \wedge \phi_{i_k}) = 0$. For the creation operator $c_j^\dagger$, a particle is created in the state ϕ_j.

The creation and annihilation operators satisfy the following anticommutation rules

$$\{c_i^\dagger, c_j^\dagger\} = \{c_i, c_j\} = 0, \quad \text{and} \quad \{c_i, c_j^\dagger\} = \delta_{ij}. \tag{1.1.10}$$

The Hamiltonian is written (in the Mulliken convention)

$$\hat{H} = \sum_{i,j=1}^{d} h_{ij} c_i^\dagger c_j + \frac{1}{2} \sum_{i,j,k,\ell=1}^{d} V_{ijk\ell} c_i^\dagger c_j^\dagger c_\ell c_k, \tag{1.1.11}$$

where $c_i^\dagger$, c_j are the creation and annihilation operators, h is a Hermitian matrix and $(V_{ij,k\ell}) \in \mathbb{C}^{d^2\times d^2}$ is also Hermitian.

In quantum chemistry, given an orthonormal basis $(\phi_i)_{i\in\mathbb{N}}$ of $L^2(\mathbb{R}^3 \times \mathbb{Z}_2)$ the coefficients h_{ij} and $V_{ijk\ell}$ are given by [HJO14, Equation (1.4.40) and (1.4.41)]

$$\begin{cases} h_{ij} = \displaystyle\sum_{s\in\{0,1\}} \int_{\mathbb{R}^3} \phi_i(r,s)^* \big(-\tfrac{1}{2}\Delta + v_{\mathrm{ne}}\big)\phi_j(r,s)\,\mathrm{d}r \\ V_{ijk\ell} = \displaystyle\sum_{s,s'\in\{0,1\}} \int_{\mathbb{R}^3} \frac{\phi_i(r,s)^*\phi_j(r',s')^*\phi_k(r,s)\phi_\ell(r',s')}{|r-r'|}\,\mathrm{d}r\mathrm{d}r'. \end{cases} \tag{1.1.12}$$

The problem to solve is

$$\min\left\{\langle \Psi, \hat{H}\Psi\rangle_{\mathcal{F}}, \Psi \in \mathcal{F}_d, \|\Psi\|_{\mathcal{F}_d} = 1, \hat{N}\Psi = N\Psi\right\}, \tag{1.1.13}$$

where $\hat{N}$ is the particle number operator

$$\hat{N} = \sum_{i=1}^{d} c_i^\dagger c_i. \tag{1.1.14}$$

Remark 1.1.1. *As the constraint on the number of particles can be cumbersome to take into account, an easy alternative is to relax the particle number constraint to a mean-value constraint*

$$\min\left\{\langle \Psi, \hat{H}\Psi\rangle_{\mathcal{F}}, \Psi \in \mathcal{F}_d, \|\Psi\|_{\mathcal{F}_d} = 1, \langle \Psi, \hat{N}\Psi\rangle = N\right\}. \tag{1.1.15}$$

This quadratic constraint can be reformulated as a Lagrange multiplier where we solve

$$\min\left\{\langle \Psi, (\hat{H} - \mu\hat{N})\Psi\rangle_{\mathcal{F}}, \Psi \in \mathcal{F}_d, \|\Psi\|_{\mathcal{F}_d} = 1, \right\}. \tag{1.1.16}$$

for a fixed value $\mu \in \mathbb{R}$. Both minimisation problems (1.1.13) and (1.1.16) are not equivalent in general.

The particle number constraint in the tensor train format can be taken into account directly in the format, as it imposes sparsity patterns in the components of the tensor train decomposition [BGP22].

A minimiser of (1.1.15) or (1.1.16) is of the form

$$\Psi = \sum_{\mu_1,\dots,\mu_d} \Psi_{i_1\dots i_N} c(\phi_{i_1})^\dagger \cdots c(\phi_{i_N})^\dagger |\Omega\rangle. \tag{1.1.17}$$

Another way to parametrise the wave function Ψ is with respect to the occupation number representation where instead of only keeping track of the occupied orbitals, we look at the occupancy of each orbital. More precisely, we define

$$\Phi_{(\mu_1,\dots,\mu_d)} = c(\phi_{i_1})^\dagger \cdots c(\phi_{i_k})^\dagger |\Omega\rangle, \tag{1.1.18}$$

if $i_1 < \cdots < i_k$ and $(i_j)_{1\leq j\leq k}$ are precisely the indices such that $\mu_{i_j} = 1$. The wave function Ψ can then be written

$$\Psi = \sum_{\mu_1,\dots,\mu_d} \Psi_{\mu_1,\dots,\mu_d} \Phi_{(\mu_1,\dots,\mu_d)}. \tag{1.1.19}$$

In DMRG, the tensor $\Psi \in \mathbb{C}^{2^d}$ is expressed as a *matrix product state*, also called a *tensor train* in the mathematical community.

Chapter 2

The low-rank approximation problem for matrices and tensors

2.1 Singular value decomposition and generalisations for tensors

This chapter is devoted to the tensor train decomposition, as a generalisation of the singular value decomposition (SVD) for high-dimensional tensors. The SVD arises in the low-rank approximation of matrices, as such, it is natural to look for generalisation of the SVD for high-dimensional tensors. As it will be mentioned, the historical tensor formats, i.e. the CP decomposition and the Tucker decomposition suffer from drawbacks that the tensor train format does not have.

2.1.1 The low-rank approximation for matrices

The basic tool for the low-rank approximation of matrices is the singular value decomposition (SVD).

Theorem 2.1.1 (Singular value decomposition)**.** *Let $A \in \mathbb{R}^{m \times n}$ be a matrix. There exist orthogonal matrices $U \in \mathbb{R}^{m \times r_A}$ and $V \in \mathbb{R}^{n \times r_A}$, and a diagonal matrix $\Sigma = \mathrm{Diag}(s_1, \dots, s_{r_A})$ with $s_1 \geq \dots \geq s_{r_A} > 0$ such that $A = U \Sigma V^\mathsf{T}$. The triplet of matrices $(U, \Sigma, V^\mathsf{T})$ satisfying these properties is called a singular value decomposition (SVD) of A.*

The SVD given in the above theorem is sometimes called the *compact* SVD of A. Another common definition of the SVD is a decomposition of the matrix $A \in \mathbb{R}^{m \times n}$ is to write the SVD as $A = \mathcal{U} \boldsymbol{\Sigma} \mathcal{V}^\mathsf{T}$ where $\mathcal{U} \in \mathbb{R}^{m \times m}$ and $\mathcal{V} \in \mathbb{R}^{n \times n}$ are orthogonal matrices and $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times n}$ is diagonal. The relationship between this SVD and its compact version is the following

$$\mathcal{U} = \begin{bmatrix} U & 0 \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathcal{V} = \begin{bmatrix} V & 0 \end{bmatrix}.$$

The SVD of A can be derived from the eigenvalue decomposition of the matrices AA^T and AA^T. Indeed, if $A = \mathcal{U}\Sigma\mathcal{V}^\mathsf{T}$ is the SVD of A, then $A^\mathsf{T} = \mathcal{V}\Sigma\mathcal{U}^\mathsf{T}$ so using that $\mathcal{U}$ and $\mathcal{V}$ are orthogonal matrices, we have

$$AA^\mathsf{T} = \mathcal{U}\Sigma\Sigma^\mathsf{T}\mathcal{U}^\mathsf{T} = \mathcal{U}\begin{bmatrix} s_1^2 & & & \\ & \ddots & & \\ & & s_r^2 & \\ & & & 0 \\ & & & & \ddots \end{bmatrix}\mathcal{U}^\mathsf{T}, \quad A^\mathsf{T}A = \mathcal{V}\Sigma^\mathsf{T}\Sigma\mathcal{V}^\mathsf{T} = \mathcal{V}\begin{bmatrix} s_1^2 & & & \\ & \ddots & & \\ & & s_r^2 & \\ & & & 0 \\ & & & & \ddots \end{bmatrix}\mathcal{V}^\mathsf{T}.$$

The singular values of A are simply the eigenvalues of the matrices AA^T and $A^\mathsf{T}A$ and the orthogonal matrices $\mathcal{U}$ and $\mathcal{V}$ the corresponding orthonormal eigenvectors.

From the singular value decomposition -and its connection to the eigenvalue decomposition- we can give another characterisation of the singular values:

$$s_k = \max_{\substack{V_k \subset \mathbb{R}^n \\ \dim V_k = k}} \min_{x \in V_k} \frac{\|Ax\|_2}{\|x\|_2}. \tag{2.1.1}$$

From the SVD, it is possible to directly read the rank of the matrix A. It is simply the number of nonzero singular values.

Singular values are also related to the Frobenius norm of the matrix. In an abuse of notation, viewing A as an element of the vector space $\mathbb{R}^{mn}$, we have by the SVD that

$$A_{ij} = \sum_{k=1}^{r_A} s_k u_{ik} v_{jk} \Rightarrow A = \sum_{k=1}^{r_A} s_k u_k \otimes v_k.$$

Since the vectors (u_k) and (v_k) are orthonormal, it is also the case for $(u_k \otimes v_k)$ thus

$$\|A\|_F^2 = \sum_{k=1}^{r_A} s_k^2.$$

Another important property of the singular value decomposition for the low-rank approximation problem is the following.

Theorem 2.1.2 (Best rank r approximation of a matrix [Sch08]). *Let $A \in \mathbb{R}^{m\times n}$ be a matrix and $(U, \Sigma, V^\mathsf{T})$ an SVD of A. The best rank-r of A in the Frobenius norm is given by*

$$A_r = U_r \Sigma_r V_r^\mathsf{T} = \sum_{k=1}^{r} s_k u_k v_k^\mathsf{T},$$

where $U_r \in \mathbb{R}^{m\times r}$, $\Sigma_r \in \mathbb{R}^{r\times r}$ and $V_r \in \mathbb{R}^{n\times r}$ are the respective truncations of U, Σ and V. The error is given by

$$\|A - A_r\|_F = \Big(\sum_{k\geq r+1} s_k^2\Big)^{1/2}. \tag{2.1.2}$$

The best approximation is unique if $s_r > s_{r+1}$.

Proof. An upper bound is obtained by a direct computation

$$\|A - A_r\|_F^2 = \Big\| \sum_{j \geq r+1} s_j u_j v_j^\mathsf{T} \Big\|_F^2 = \Big\| \sum_{j \geq r+1} s_j u_j \otimes v_j \Big\|_2^2 = \sum_{j \geq r+1} s_j^2.$$

The lower bound is shown using a bound on the singular values: let $M, N \in \mathbb{R}^{p \times q}$

$$\forall\, 1 \leq i, j \leq \min(p, q), 0 \leq j \leq d - i, s_{i+j-1}(M + N) \leq s_i(M) + s_j(N), \tag{2.1.3}$$

where $(s_k(M))_k, (s_k(N))_k, (s_k(M+N))_k$ are the respective singular values of M, N and $M+N$. This singular value bounds are derived by considering the following subspaces (without loss of generality, we can assume that $q \leq p$):

$$V^{M+N} = \mathrm{Span}\big(v_1^{M+N}, \dots, v_{i+j-1}^{M+N}\big), \quad V^M = \mathrm{Span}\big(v_i^M, \dots, v_q^M\big)$$
$$V^N = \mathrm{Span}\big(v_j^N, \dots, v_q^N\big).$$

By estimating the dimension of the intersection (by using that $\dim V^M + \dim V^N + \dim V^{M+N} = (q - i + 1) + (q - j + 1) + i + j - 1 = 2q + 1$), we deduce that there exists a normalised vector $x \in V^M \cap V^N \cap V^{M+N}$:

$$s_{i+j-1}(M + N) \leq \|(M + N)x\|_2 \leq \|Mx\|_2 + \|Nx\|_2 \leq s_i(M) + s_j(N).$$

Let $\widetilde{A}_r$ be a matrix of rank r. We apply the inequality (2.1.3) with $M = A - \widetilde{A_r}$, $N = \widetilde{A}_r$ and $j = r + 1$. Since $s_{r+1}(\widetilde{A}_r) = 0$, we have

$$\forall\, 1 \leq i \leq q, s_{r+i}(A) \leq s_i(A - \widetilde{A_r}).$$

Hence $\|A - \widetilde{A_r}\|_F^2 = \sum_{i=1}^q s_i(A - \widetilde{A_r})^2 \geq \sum_{i=r+1}^q s_i(A)^2$, which is the result. □

Remark 2.1.3. *A similar approximation result can be written in the matrix norm $\|\cdot\|_2$ subordinate to the vector $\|\cdot\|_2$. In that case, it is straightforward to check that $\|A - A_r\|_2 = \big\| \sum_{j \geq r+1} s_j u_j v_j^\mathsf{T} \big\|_2 = s_{r+1}$. Moreover for a rank-r matrix $\widetilde{A}_r$, by definition, there is a normalised vector $x \in \mathrm{Span}(v_1, \dots, v_{r+1})$ such that $\widetilde{A}_r x = 0$. Thus*

$$\|A - \widetilde{A}_r\|_2 \geq \|(A - \widetilde{A}_r)x\|_2 \geq \|Ax\|_2 \geq s_{r+1}.$$

Another way to phrase the best rank r approximation of a matrix is to take the subspace point of view. A matrix $A \in \mathbb{R}^{m \times n}$ can be viewed as a vector of the product space $\mathbb{R}^m \otimes \mathbb{R}^n$ which is isometrically isomorphic to $\mathbb{R}^{mn}$. The subspace problem is phrased as follows: find subspaces $\mathcal{U} \subset \mathbb{R}^m$ and $\mathcal{V} \subset \mathbb{R}^n$ both of dimension r such that it minimises the distance

$$\mathrm{dist}(A, \mathcal{U} \otimes \mathcal{V}) = \|A - \Pi_{\mathcal{U} \otimes \mathcal{V}} A\| = \min_{\substack{\widetilde{\mathcal{U}} \subset \mathbb{R}^m, \dim \widetilde{\mathcal{U}} = r \\ \widetilde{\mathcal{V}} \subset \mathbb{R}^n, \dim \widetilde{\mathcal{V}} = r}} \|A - \Pi_{\widetilde{\mathcal{U}} \otimes \widetilde{\mathcal{V}}} A\|, \tag{2.1.4}$$

where $\Pi_{\mathcal{W}}$ is the orthogonal projection onto the subspace $\mathcal{W} \subset \mathbb{R}^{mn}$. The SVD gives a characterisation of the solution to the minimisation problem (2.1.4).

Proposition 2.1.4. *Let $A \in \mathbb{R}^{m\times n}$, $(U, \Sigma, V^\mathsf{T})$ its SVD and $r \in \mathbb{N}$. Denote $(u_1, \dots, u_{r_A})$ and $(v_1, \dots, v_{r_A})$ the respective columns of U and V. A solution to the subspace minimisation problem (2.1.4) is given by*

$$\mathcal{U} = \mathrm{Span}(u_1, \dots, u_r), \quad \mathcal{V} = \mathrm{Span}(v_1, \dots, v_r). \tag{2.1.5}$$

The solution is unique if $s_r > s_{r+1}$.

Proof. Let $\widetilde{\mathcal{U}}$ and $\widetilde{\mathcal{V}}$ be respectively subspaces of $\mathbb{R}^m$ and $\mathbb{R}^n$ of dimension r. Let $(\widetilde{u}_i)_{1\leq i\leq r}$ and $(\widetilde{v}_i)_{1\leq i\leq r}$ be ONB of respectively $\widetilde{\mathcal{U}}$ and $\widetilde{\mathcal{V}}$. The minimisation problem (2.1.4) can be rewritten as

$$\min_{\substack{\widetilde{\mathcal{U}}\subset\mathbb{R}^m, \dim\widetilde{\mathcal{U}}=r\\ \widetilde{\mathcal{V}}\subset\mathbb{R}^n, \dim\widetilde{\mathcal{V}}=r}} \|A - \Pi_{\widetilde{\mathcal{U}}\otimes\widetilde{\mathcal{V}}} A\| = \min_{\substack{\widetilde{\mathcal{U}}\subset\mathbb{R}^m, \dim\widetilde{\mathcal{U}}=r\\ \widetilde{\mathcal{V}}\subset\mathbb{R}^n, \dim\widetilde{\mathcal{V}}=r}} \|A - P_{\widetilde{\mathcal{U}}} A P_{\widetilde{\mathcal{V}}}\|_F^2,$$

where $P_{\widetilde{\mathcal{U}}}$ (resp. $P_{\widetilde{\mathcal{V}}}$) is the orthogonal projection onto $\widetilde{\mathcal{U}}$ (resp. $\widetilde{\mathcal{V}}$).

Let $\widetilde{\mathcal{U}}$ and $\widetilde{\mathcal{V}}$ be respectively subspaces of $\mathbb{R}^m$ and $\mathbb{R}^n$ of dimension r. Let $(\widetilde{u}_i)_{1\leq i\leq r}$ and $(\widetilde{v}_i)_{1\leq i\leq r}$ be ONB of respectively $\widetilde{\mathcal{U}}$ and $\widetilde{\mathcal{V}}$. Then we have

$$\begin{aligned}
\|A - P_{\widetilde{\mathcal{U}}} A P_{\widetilde{\mathcal{V}}}\|_F^2 &= \mathrm{Tr}\big((A - P_{\widetilde{\mathcal{U}}} A P_{\widetilde{\mathcal{V}}})^\mathsf{T}(A - P_{\widetilde{\mathcal{U}}} A P_{\widetilde{\mathcal{V}}})\big) \\
&= \mathrm{Tr}\big(A^\mathsf{T} A - P_{\widetilde{\mathcal{V}}} A^\mathsf{T} P_{\widetilde{\mathcal{U}}} A - A^\mathsf{T} P_{\widetilde{\mathcal{U}}} A P_{\widetilde{\mathcal{V}}} + P_{\widetilde{\mathcal{V}}} A^\mathsf{T} P_{\widetilde{\mathcal{U}}} A P_{\widetilde{\mathcal{V}}}\big) \\
&= \mathrm{Tr}\big(A^\mathsf{T} A\big) - \mathrm{Tr}\big(P_{\widetilde{\mathcal{V}}} A^\mathsf{T} P_{\widetilde{\mathcal{U}}} A P_{\widetilde{\mathcal{V}}}\big),
\end{aligned}$$

where we have used that since $P_{\widetilde{\mathcal{V}}}$ is an orthogonal projection, we have $\mathrm{Tr}\big(P_{\widetilde{\mathcal{V}}} A^\mathsf{T} P_{\widetilde{\mathcal{U}}} A\big) = \mathrm{Tr}\big(A^\mathsf{T} P_{\widetilde{\mathcal{U}}} A P_{\widetilde{\mathcal{V}}}\big) = \mathrm{Tr}\big(P_{\widetilde{\mathcal{V}}} A^\mathsf{T} P_{\widetilde{\mathcal{U}}} A P_{\widetilde{\mathcal{V}}}\big)$. We realise that

$$\mathrm{Tr}\big(P_{\widetilde{\mathcal{V}}} A^\mathsf{T} P_{\widetilde{\mathcal{U}}} A P_{\widetilde{\mathcal{V}}}\big) = \sum_{1\leq i,j\leq r} \big|\langle \widetilde{u}_i, A\widetilde{v}_j\rangle\big|^2.$$

Solving the minimisation problem (2.1.4) is equivalent to maximising $\sum_{1\leq i,j\leq r} \big|\langle \widetilde{u}_i, A\widetilde{v}_j\rangle\big|^2$ where $(\widetilde{u}_i)_{1\leq i\leq r}$ and $(\widetilde{v}_i)_{1\leq i\leq r}$ are orthonormal families. Using the max-min characterisation of the singular values (2.1.1), we have $\sum_{1\leq i,j\leq r} \big|\langle \widetilde{u}_i, A\widetilde{v}_j\rangle\big|^2 \leq \sum_{j=1}^r \|A\widetilde{v}_j\|^2 \leq \sum_{j=1}^r s_j^2$. The upper bound is attained for $\widetilde{\mathcal{U}} = \mathrm{Span}(u_1, \dots, u_r)$ and $\widetilde{\mathcal{V}} = \mathrm{Span}(v_1, \dots, v_r)$. □

2.1.2 Generalisations of the SVD for tensors

A tensor $\boldsymbol{u}$ of order $d \in \mathbb{N}$ is a multidimensional array $(\boldsymbol{u}_{i_1 \dots i_d}) \in \mathbb{R}^{n_1\times\cdots\times n_d}$.

For higher-order tensors, different generalisations of the SVD are possible. With the previous discussion, there are two natural options that emerge:

- write the tensor as a sum of rank-1 tensors:

$$\boldsymbol{u} = \sum_{\alpha=1}^{r} u_\alpha^{(1)} \otimes \cdots \otimes u_\alpha^{(d)},$$

where for all $k \in [\![d]\!]$, $u_\alpha^{(k)} \in \mathbb{R}^{n_k}$. This is the *canonical polyadic decomposition* (CP decomposition);

- consider the subspace minimisation problem:

$$\mathrm{dist}(\boldsymbol{u}, \mathcal{U}_1 \otimes \mathcal{U}_2 \otimes \cdots \otimes \mathcal{U}_d) = \min_{\widetilde{\mathcal{U}}_1 \subset \mathbb{R}^{n_1}, \dim \widetilde{\mathcal{U}}_1 = r_1, \dots, \widetilde{\mathcal{U}}_d \subset \mathbb{R}^{n_d}, \dim \widetilde{\mathcal{U}}_d = r_d} \|\boldsymbol{u} - \Pi_{\widetilde{\mathcal{U}}_1 \otimes \cdots \otimes \widetilde{\mathcal{U}}_d} \boldsymbol{u}\|,$$

where $\dim \mathcal{U}_k = r_k$ for all $k \in [\![d]\!]$. This yields the Tucker decomposition.

The canonical decomposition looks the most appealing as it is the most low-complexity way to represent a tensor. It has however one major drawback, being that the best rank r approximation (in the sense of the CP decomposition) is *ill-posed!* [DSL08] Consider noncolinear vectors $a \in \mathbb{R}^n$, $b \in \mathbb{R}^n$ and the tensor

$$\boldsymbol{u} = b \otimes a \otimes a + a \otimes b \otimes a + a \otimes a \otimes b.$$

which is a tensor of canonical rank 3. It can however be approximated as well as we wish by a tensor of canonical rank 2: let $\varepsilon > 0$, then we see that

$$\boldsymbol{u} - \Big(\frac{1}{\varepsilon}(a + \varepsilon b) \otimes (a + \varepsilon b) \otimes (a + \varepsilon b) - \frac{1}{\varepsilon} a \otimes a \otimes a\Big) = \mathcal{O}(\varepsilon). \tag{2.1.6}$$

This example highlights that the set of tensors of canonical rank less than 2 is not closed. Contrary to matrices, the set of tensors of canonical rank less than r is not closed.

Regarding the Tucker decomposition, let $\boldsymbol{u} \in \mathcal{U}_1 \otimes \cdots \otimes \mathcal{U}_d$. Then there is a core tensor $\boldsymbol{c} \in \mathbb{R}^{r_1 \times \cdots \times r_d}$ and matrices $(U_k)_{1 \le k \le d} \in \bigtimes_{k=1}^d \mathbb{R}^{n_k \times r_k}$ such that

$$\forall \mathbf{i} \in [\![\mathbf{n}]\!],\ \boldsymbol{u}_{i_1 \dots i_d} = \sum_{\alpha_1=1}^{r_1} \cdots \sum_{\alpha_d=1}^{r_d} \boldsymbol{c}_{\alpha_1 \dots \alpha_d} (U_1)_{i_1}^{\alpha_1} \cdots (U_d)_{i_d}^{\alpha_d}.$$

The storage cost of the tensor $\boldsymbol{u}$ is still exponential in the order d of the tensor (except if some r_k are equal to 1). As such it is a useful decomposition only for low order tensors. In the following, we will focus on the efficient representation of tensors of order up to a hundred, for which the Tucker decomposition is not suited.

2.2 Tensor train decomposition

2.2.1 Hierarchical SVD and tensor trains

We first define the reshape of a tensor.

Definition 2.2.1 (Reshape of a tensor)**.** *Let $\boldsymbol{u} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ be a tensor. Let $\ell \in [\![d]\!]$. We say that the matrix $\big(\boldsymbol{u}_{i_1 \dots i_\ell; i_{\ell+1} \dots i_d}\big) \in \mathbb{R}^{n_1 \cdots n_\ell \times n_{\ell+1} \cdots n_d}$ is a reshape of $\boldsymbol{u}$ with respect to the modes $[\![\ell]\!]$. This reshape will be denoted by $\boldsymbol{u}^{\le \ell}$ and for $\mathbf{i} \in [\![\mathbf{n}]\!]$ its elements $\boldsymbol{u}_{i_1 \dots i_\ell}^{i_{\ell+1} \dots i_d}$.*

The reshapes $\boldsymbol{u}^{\leq \ell}$ for $1 \leq \ell \leq d-1$ will be of particular interest.

To derive a tensor decomposition generalising the SVD, one can apply the SVD successively. Let $\boldsymbol{u} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ be a tensor and proceed as follows (we use here Einstein convention where we sum over repeated indices)

$$\begin{aligned}
\boldsymbol{u}_{i_1 \dots i_d} &= (\boldsymbol{u}_{i_1}^{i_2 \dots i_d}) && \text{(reshape of } \boldsymbol{u} \text{ to } n_1 \times n_2 \cdots n_d) \\
&= (U_1)_{i_1}^{\alpha_1} (\Sigma_1 V_1^\mathsf{T})_{\alpha_1}^{i_2 \dots i_d} && \text{(SVD)} \\
&= (U_1)_{i_1}^{\alpha_1} (\Sigma_1 V_1^\mathsf{T})_{\alpha_1 i_2}^{i_3 \dots i_d} && \text{(reshape of } \Sigma_1 V_1^\mathsf{T}) \\
&= (U_1)_{i_1}^{\alpha_1} (U_2)_{\alpha_1 i_2}^{\alpha_2} (\Sigma_2 V_2^\mathsf{T})_{\alpha_2}^{i_3 \dots i_d} && \text{(SVD of } \Sigma_1 V_1^\mathsf{T}) \\
&= (U_1)_{i_1}^{\alpha_1} (U_2)_{\alpha_1 i_2}^{\alpha_2} (\Sigma_2 V_2^\mathsf{T})_{\alpha_2 i_3}^{i_4 \dots i_d} && \text{(reshape of } \Sigma_2 V_2^\mathsf{T}),
\end{aligned}$$

where we repeat the process until we get

$$\boldsymbol{u}_{i_1 \dots i_d} = (U_1)_{i_1}^{\alpha_1} (U_2)_{\alpha_1 i_2}^{\alpha_2} \cdots (U_{d-1})_{\alpha_{d-2} i_{d-1}}^{\alpha_{d-1}} (\Sigma_{d-1} V_{d-1}^\mathsf{T})_{\alpha_{d-1}}^{i_d}.$$

The tensors appearing in the decomposition above can be rearranged as below

$$\begin{aligned}
\boldsymbol{u}_{i_1 \dots i_d} &= (U_1)_{i_1}^{\alpha_1} \quad (U_2)_{\alpha_1 i_2}^{\alpha_2} \quad \cdots \quad (U_{d-1})_{\alpha_{d-2} i_{d-1}}^{\alpha_{d-1}} (\Sigma_{d-1} V_{d-1}^\mathsf{T})_{\alpha_{d-1}}^{i_d} \\
&= A_1[i_1]_{\alpha_1} \quad A_2[i_2]_{\alpha_2}^{\alpha_1} \quad \cdots \quad A_{d-1}[i_{d-1}]_{\alpha_{d-1}}^{\alpha_{d-2}} \quad A_d[i_d]^{\alpha_{d-1}}.
\end{aligned}$$

The decomposition above is called the *tensor train* (TT) decomposition [OT09], also called *matrix product state* [KSZ91] in the physics litterature is the simplest instance of a tensor network. This terminology will be clearer when the graphical representations of the tensor formats will be presented in Section 5.1.1.

Definition 2.2.2 ([KSZ91, OT09])**.** *Let $\boldsymbol{u} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ be a tensor. We say that $(A_1, \dots, A_d)$ is a tensor train decomposition of $\boldsymbol{u}$ if we have for all $\mathbf{i} \in [\![\mathbf{n}]\!]$*

$$\begin{aligned}
\boldsymbol{u}_{i_1 \dots i_d} &= A_1[i_1] A_2[i_2] \cdots A_d[i_d] && (2.2.1) \\
&= \sum_{\alpha_1=1}^{r_1} \sum_{\alpha_2=1}^{r_2} \cdots \sum_{\alpha_{d-1}=1}^{r_{d-1}} A_1[i_1]_{\alpha_1} A_2[i_2]_{\alpha_2}^{\alpha_1} \cdots A_d[i_d]^{\alpha_{d-1}}, && (2.2.2)
\end{aligned}$$

where for each $i_k \in [\![n_k]\!]$, $A_k[i_k] \in \mathbb{R}^{r_{k-1} \times r_k}$ $(r_0 = r_d = 1)$. The tensor $A_k \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$ are called the TT cores *and the sizes of the TT cores are the* TT ranks *of $\boldsymbol{u}$.*

Such a representation has a storage cost of $\sum_{k=1}^{d} n_k r_{k-1} r_k$. Provided that the TT ranks do not increase exponentially with the order d of the tensor, the TT decomposition is a sparse representation of the tensor $\boldsymbol{u}$.

An exact TT representation of any tensor can be achieved by the algorithm presented at the beginning of Section 2.2.1 and given in Algorithm 2.1. This algorithm is called the hierarchical SVD (HSVD) [Vid03, OT09].

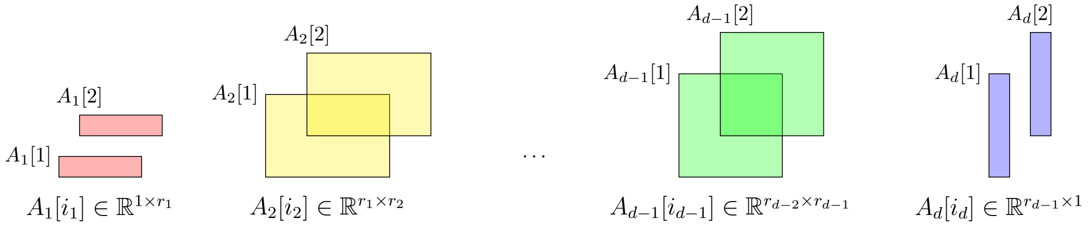


Figure 2.1: Schematic representation of the TT decomposition of a tensor in $\mathbb{R}^{2\times\cdots\times 2}$

It is clear that there is no uniqueness of the TT decomposition. Indeed for a tensor $\boldsymbol{u} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ if $(A_1,\dots,A_d)$ is a tensor train decomposition, then for any invertible matrices $(G_k)_{1\le k\le d-1} \in \bigotimes_{k=1}^{d-1} \mathrm{GL}_{r_k}(\mathbb{R})$, the TT cores $(\widetilde{A}_1,\dots,\widetilde{A}_d)$ defined by

$$\begin{cases} \widetilde{A}_1[i_1] = A_1[i_1]G_1,\ i_1 \in [\![n_1]\!], \quad \widetilde{A}_d[i_d] = G_{d-1}^{-1}A_d[i_d],\ i_d \in [\![n_d]\!] \\ \widetilde{A}_k[i_k] = G_{k-1}^{-1}A_k[i_k]G_k,\ i_k \in [\![n_k]\!],\ k \in [\![2; d-1]\!], \end{cases}$$

is an equivalent TT representation. It is possible to partially lift this gauge freedom. This discussion is postponed to Section 2.2.5.

The history of the TT decomposition dates back to the density-matrix renormalisation group (DMRG) [Whi92] pioneered by White for the computation of properties of one-dimensional statistical physics systems. The connection between DMRG and TT has been realised later [OR95, DMNS98].

Example 2.2.3.

- *a tensor product* $\boldsymbol{u}_{i_1\dots i_d} = u^{(1)}_{i_1}\cdots u^{(d)}_{i_d}$ *is a TT of TT rank 1, as the cores are* $(u^{(k)}_{i_k})_{1\le k\le d, 1\le i_k\le n_k}$.

- *the unnormalised Bell state* $\boldsymbol{b} \in \bigotimes_1^{2d}\mathbb{R}^2$

$$\boldsymbol{b}_{i_1\dots i_{2d}} = \big(\delta_{1,i_1}\delta_{2,i_2} + \delta_{2,i_1}\delta_{1,i_2}\big)\big(\delta_{1,i_3}\delta_{2,i_4} + \delta_{2,i_3}\delta_{1,i_4}\big)\cdots\big(\delta_{1,i_{2d-1}}\delta_{2,i_{2d}} + \delta_{2,i_{2d-1}}\delta_{1,i_{2d}}\big),$$

is a TT of rank 2: let $(B_k)_{1\le k\le 2d}$ *be defined by*

$$B_{2k-1}[i_{2k-1}] = \begin{bmatrix}\delta_{1i_{2k-1}} & \delta_{2i_{2k-1}}\end{bmatrix}, \quad B_{2k}[i_{2k}] = \begin{bmatrix}\delta_{2i_{2k}} \\ \delta_{1i_{2k}}\end{bmatrix}, \quad k = 1,\dots,d. \tag{2.2.3}$$

By a direct calculation, we can check that $\boldsymbol{b}_{i_1\dots i_{2d}} = B_1[i_1]\cdots B_{2d}[i_d]$.

- *for* $d = 2$*, the following reordering of the indices of the Bell state* $\widetilde{\boldsymbol{b}} \in \bigotimes_1^4\mathbb{R}^2$

$$\widetilde{\boldsymbol{b}}_{i_1\dots i_4} = \big(\delta_{1,i_1}\delta_{2,i_3} + \delta_{2,i_1}\delta_{1,i_3}\big)\big(\delta_{1,i_2}\delta_{2,i_4} + \delta_{2,i_3}\delta_{1,i_4}\big)$$

has a TT decomposition of rank 4:

i_k	$\widetilde{B}_1$	$\widetilde{B}_2$	$\widetilde{B}_3$	$\widetilde{B}_4$
1	$\begin{bmatrix}1 & 0\end{bmatrix}$	$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}$	$\begin{bmatrix}0 & 0\\ 0 & 0\\ 1 & 0\\ 0 & 1\end{bmatrix}$	$\begin{bmatrix}0\\ 1\end{bmatrix}$
2	$\begin{bmatrix}0 & 1\end{bmatrix}$	$\begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$	$\begin{bmatrix}1 & 0\\ 0 & 1\\ 0 & 0\\ 0 & 0\end{bmatrix}$	$\begin{bmatrix}1\\ 0\end{bmatrix}$

This elementary example highlights the importance of the ordering of the indices of the tensor for an efficient TT representation. The TT decomposition above can be derived by using that $(AC) \otimes (BD) = (A \otimes B)(C \otimes D)$ for matrices with compatible sizes. Hence the formula for the TT cores of the reordered Bell state is obtained from the TT decomposition (2.2.3) of the Bell state

$$\begin{aligned}\widetilde{\boldsymbol{b}}_{i_1 i_2 i_3 i_4} &= \boldsymbol{b}_{i_1 i_3 i_2 i_4} = B_1[i_1]B_2[i_3]B_3[i_2]B_4[i_4]\\ &= B_1[i_1](\mathrm{id}_2\, B_2[i_3] \otimes B_3[i_2]\,\mathrm{id}_2)B_4[i_4]\\ &= B_1[i_1](\mathrm{id}_2 \otimes B_3[i_2])(B_2[i_3] \otimes \mathrm{id}_2)B_4[i_4].\end{aligned}$$

Remark 2.2.4. *The reordered Bell state example $\widetilde{\boldsymbol{b}} \in \bigotimes_1^{2d} \mathbb{R}^2$*

$$\widetilde{\boldsymbol{b}}_{i_1 \dots i_{2d}} = \prod_{k=1}^{d} \left(\delta_{1,i_k}\delta_{2,i_{k+d}} + \delta_{2,i_k}\delta_{1,i_{k+d}} \right)$$

has a TT decomposition of rank 2^d. The optimality of the ranks is proved by the characterisation of the TT ranks stated in Theorem 2.2.12.

The central tool in the TT decomposition is the HSVD presented earlier and summarised in Algorithm 2.1. From the characterisation of the error in the truncated SVD, it is expected that the HSVD can be used to derive an approximation result by a TT with given TT ranks. This will be treated in Section 2.2.4.

Remark 2.2.5. *It is reasonably clear that such an algorithm extends to the decomposition into a tree tensor network. Indeed, in the HSVD algorithm, we simply partition $\{1, \dots, d\}$ into the sets $(\{1\}, \{2, \dots, d\})$, then $(\{1\}, \{2\}, \{3, \dots, d\})$, and so on so forth. For trees, we choose different partition choices that does not have to reduce to a singleton right away. For tensor networks with loops, there is no equivalent of the HSVD for the construction of a tensor network directly from the tensor. This makes the analysis of such networks much more difficult.*

Algorithm 2.1 Hierarchical SVD

Input: Tensor $\boldsymbol{u} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$
Output: $(A_1, \ldots, A_d)$ TT representation of $\boldsymbol{u}$

function HSVD($\boldsymbol{u}$)
  $(T_0)\alpha_0 i_1{}^{i_2 \ldots i_d} = \boldsymbol{u}_{i_1}^{i_2 \ldots i_d}$ ▷ α_0 dummy index
  for $k = 1, \ldots, d-1$ **do**
    $U_k, \Sigma_k, V_k^\mathsf{T} = \mathtt{svd}\Big((T_{k-1})_{\alpha_{k-1} i_k}^{i_{k+1} \ldots i_d}\Big)$
    $A_k[i_k]_{\alpha_{k-1}}^{\alpha_k} = (U_k)_{\alpha_{k-1} i_k}^{\alpha_k}$
    $(T_{k-1})_{\alpha_k i_{k+1}}^{i_{k+2} \ldots i_d} = \big(\Sigma_k V_k^\mathsf{T}\big)_{\alpha_k}^{i_{k+1} i_{k+2} \ldots i_d}$
  end for
  $A_d[i_d]^{\alpha_{d-1}} = \big(\Sigma_{d-1} V_{d-1}^\mathsf{T}\big)_{\alpha_{d-1}}^{i_d}$
  return $(A_1, \ldots, A_d)$
end function

2.2.2 Algebraic properties of TT and normalisation of TT

The TT decomposition has reasonable algebraic properties as it is stable by multiplication by a scalar and by addition -up to augmentation of the TT ranks.

Proposition 2.2.6 (Algebraic properties of TT)**.** *Let $(A_1, \ldots, A_d)$and $(\widetilde{A}_1, \ldots, \widetilde{A}_d)$ be the respective TT decompositions of the tensors $\boldsymbol{u}, \widetilde{\boldsymbol{u}} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$. Let $\lambda \in \mathbb{R}$. Then*

- *$\lambda \boldsymbol{u}$ has a TT decomposition given by $(B_k)_{k \in [\![d]\!]}$ with $B_k = A_k$ for $k \in [\![d-1]\!]$ and $B_d = \lambda A_d$;*
- *the sum $\boldsymbol{u} + \widetilde{\boldsymbol{u}}$ has a TT decomposition $(S_k)_{k \in [\![d]\!]}$ given by*

$$
\begin{aligned}
S_1[i_1] = \begin{bmatrix} A_1[i_1] & \widetilde{A}_1[i_1] \end{bmatrix}, \quad S_d[i_d] = \begin{bmatrix} A_d[i_d] \\ \widetilde{A}_d[i_d] \end{bmatrix} \\
S_k[i_k] = \begin{bmatrix} A_k[i_k] & 0 \\ 0 & \widetilde{A}_k[i_k] \end{bmatrix}, k \in [\![2; d-1]\!].
\end{aligned}
\tag{2.2.4}
$$

The first item is clear and the proof for the sum consists in expanding the TT decomposition $(S_1, \ldots, S_d)$. The TT decomposition of the sum (2.2.4) is in general not minimal and can be compressed as explained in Section 2.2.4.

Remark 2.2.7. *Since a tensor product $u^{(1)} \otimes \cdots \otimes u^{(d)}$ is a TT of rank 1, we deduce that a CP decomposition of rank r has at most a TT representation of rank r. The TT decomposition is a generalisation of the CP format, with advantageous algebraic and topologic properties.*

The TT decomposition can be seen as a structured low-rank representation of a tensor $\boldsymbol{u} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$. Indeed, if $(A_1, \ldots, A_d)$ is a TT representation of $\boldsymbol{u}$, for any $k \in [\![d-1]\!]$, we can write

$$\boldsymbol{u}^{\leq k} = \underbrace{\begin{bmatrix} A_1[1]A_2[1]\cdots A_k[1] \\ \vdots \\ A_1[n_1]A_2[n_2]\cdots A_k[n_k] \end{bmatrix}}_{\in \mathbb{R}^{n_1\cdots n_k \times r_k}} \underbrace{\begin{bmatrix} A_{k+1}[1] & \cdots & A_d[1] & \ldots & A_{k+1}[n_{k+1}] & \cdots & A_d[n_d] \end{bmatrix}}_{\in \mathbb{R}^{r_k \times n_{k+1}\cdots n_d}},$$

which is a rank-r_k matrix decomposition of the reshape $\boldsymbol{u}^{\leq k} \in \mathbb{R}^{n_1\cdots n_k \times n_{k+1}\cdots n_d}$. This observation will be used in the next sections.

The TT cores $(A_k)_{k\in[\![d]\!]}$ obtained from the HSVD algorithm satisfy a particular property. From the definition of the SVD, we have that the reshaped TT core $((A_k)^{\alpha_k}_{i_k\alpha_{k-1}}) \in \mathbb{R}^{n_k r_{k-1} \times r_k}$ is a partial isometry for any $k \in [\![d-1]\!]$, *i.e.* we have $((A_k)^{\alpha_k}_{i_k\alpha_{k-1}})^\mathsf{T}((A_k)^{\alpha_k}_{i_k\alpha_{k-1}}) = \mathrm{id}_{r_k}$. We say in that case that are *left-orthogonal* or *left-normalised.*

Definition 2.2.8 (TT normalisation)**.** *Let $(A_k)_{k\in[\![d]\!]}$ be a TT decomposition of a tensor $\boldsymbol{u} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ with TT ranks $(r_k)_{k\in[\![0;d]\!]}$. Let $\ell \in [\![d]\!]$. We say that a TT decomposition $(A_1, \ldots, A_d)$ is* ℓ-normalised *or* normalised with root ℓ *if for all $k \in [\![\ell-1]\!]$*

$$\sum_{i_k=1}^{n_k} A_k[i_k]^\mathsf{T} A_k[i_k] = \mathrm{id}_{r_k};$$

and for all $k \in [\![\ell+1;d]\!]$

$$\sum_{i_k=1}^{n_k} A_k[i_k] A_k[i_k]^\mathsf{T} = \mathrm{id}_{r_{k-1}}.$$

If $\ell = 1$, we say that $(A_k)_{k\in[\![d]\!]}$ is left-orthogonal *or* left-normalised*. If $\ell = d$, we say that $(A_k)_{k\in[\![d]\!]}$ is* right-orthogonal *or* right-normalised*.*

The HSVD algorithm described in Algorithm 2.1 yields a left-orthogonal TT decomposition of the tensor $\boldsymbol{u}$. This is because successive SVDs are performed starting from the first index of the tensor $\boldsymbol{u}$. By performing successive SVDs from the "right", *i.e.* by first doing the SVD of $\boldsymbol{u}^{\leq d-1} \in \mathbb{R}^{n_1\cdots n_{d-1}\times n_d} = (U_{d-1}\Sigma_{d-1})^{\alpha_{d-1}}_{i_1\ldots i_{d-1}}(V^\mathsf{T}_{d-1})^{i_d}_{\alpha_{d-1}}$, then the SVD of $(U_{d-1}\Sigma_{d-1})^{i_{d-1}\alpha_{d-1}}_{i_1\ldots i_{d-2}}$ and so on and so forth, one would get a right-orthogonal TT representation of $\boldsymbol{u}$.

TT representations with a specific normalisation have convenient properties.

Proposition 2.2.9. *Let $(A_k)_{k\in[\![d]\!]}$ be a left-orthogonal TT decomposition of a tensor $\boldsymbol{u} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ with TT ranks $(r_k)_{k\in[\![0;d]\!]}$. Then we have that*

- $\|\boldsymbol{u}\| = \|A_d\|$*;*

- *for any $k \in [\![d-1]\!]$, we have that the matrix*

$$\begin{bmatrix} A_1[1]A_2[1]\cdots A_k[1] \\ \vdots \\ A_1[n_1]A_2[n_2]\cdots A_k[n_k] \end{bmatrix} \in \mathbb{R}^{n_1\cdots n_k \times r_k}$$

  is a partial isometry.

Proof. For any $k \in [\![d-1]\!]$ we have

$$\begin{aligned}
&\begin{bmatrix} A_1[1]A_2[1]\cdots A_k[1] \\ \vdots \\ A_1[n_1]A_2[n_2]\cdots A_k[n_k] \end{bmatrix}^\mathsf{T} \begin{bmatrix} A_1[1]A_2[1]\cdots A_k[1] \\ \vdots \\ A_1[n_1]A_2[n_2]\cdots A_k[n_k] \end{bmatrix} \\
&\qquad = \sum_{i_1=1}^{n_1}\cdots\sum_{i_k=1}^{n_k}(A_k[i_k])^\mathsf{T} A_1[i_1]^\mathsf{T}\cdots A_1[i_1]\cdots A_k[i_k] \\
&\qquad = \sum_{i_2=1}^{n_2}\cdots\sum_{i_k=1}^{n_k} A_k[i_k]^\mathsf{T}\cdots A_2[i_2]^\mathsf{T}\underbrace{\Big(\sum_{i_1=1}^{n_1} A_1[i_1]^\mathsf{T} A_1[i_1]\Big)}_{=\mathrm{id}_{r_1}} A_2[i_2]\cdots A_k[i_k] \\
&\qquad = \mathrm{id}_{r_k},
\end{aligned}$$

by left-orthogonality of the TT cores.

For the norm, we have

$$\|\boldsymbol{u}\| = \|\boldsymbol{u}^{\leq d}\| = \left\| \underbrace{\begin{bmatrix} A_1[1]A_2[1]\cdots A_{d-1}[1] \\ \vdots \\ A_1[n_1]A_2[n_2]\cdots A_{d-1}[n_{d-1}] \end{bmatrix}}_{\in\mathbb{R}^{n_1\cdots n_{d-1}\times r_{d-1}}} \underbrace{\begin{bmatrix} A_d[1] & \dots & A_d[n_d] \end{bmatrix}}_{\in\mathbb{R}^{r_{d-1}\times n_d}} \right\| = \|A_d\|,$$

as the first matrix is a partial isometry. □

It is convenient to introduce the $\bowtie$ notation, which simplifies the manipulation of expressions involving partial contractions of tensors of order 3.

Definition 2.2.10 (Strong Kronecker product)**.** *For $(B, C) \in \mathbb{R}^{r\times n\times\tilde{r}} \times \mathbb{R}^{\tilde{r}\times\tilde{n}\times\hat{r}}$, the strong Kronecker product denoted by $B \bowtie C \in \mathbb{R}^{r\times n\times\tilde{n}\times\hat{r}}$, is defined by*

$$(B \bowtie C)_{\alpha j\tilde{j}\hat{\alpha}} = \sum_{\tilde{\alpha}=1}^{\tilde{r}} B_{\alpha j\tilde{\alpha}} C_{\tilde{\alpha}\tilde{j}\hat{\alpha}}.$$

Proposition 2.2.11. *Let $(A_k)_{k\in[\![d]\!]}$ is a TT representation of a tensor $\boldsymbol{u}$. Then we have*

(i). $\boldsymbol{u} = A_1 \bowtie \ldots \bowtie A_d$;

(ii). for any $k \in [\![d-1]\!]$, we have that $\boldsymbol{u}^{\leq k} = (A_1 \bowtie \ldots \bowtie A_k)^{\leq k}(A_{k+1} \bowtie \ldots \bowtie A_d)^{\leq 1}$;

(iii). if $(A_k)_{k\in[\![d]\!]}$ is a left-orthogonal TT representation of a tensor, then for any $k \in [\![d-1]\!]$, $(A_1 \bowtie \ldots \bowtie A_k)^{\leq k} \in \mathbb{R}^{n_1\cdots n_k \times r_k}$ is a partial isometry.

2.2.3 Characterisation of exact TT representations

From the hierarchical SVD (Algorithm 2.1), we directly get a characterisation of the TT ranks of the exact TT representation of the tensor.

Theorem 2.2.12 (Characterisation of the TT ranks [HRS12b])**.** *Let $\boldsymbol{u} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ be a tensor. Then the following assertions are true:*

(i). the minimal TT ranks $(r_1, \ldots, r_{d-1})$ is equal to the rank of the reshapes of $\boldsymbol{u}$, i.e.

$$\forall k \in [\![d-1]\!], r_k = \operatorname{Rank}(\boldsymbol{u}^{\leq k}).$$

We will thus call $(\operatorname{Rank}(\boldsymbol{u}^{\leq k}))_{k\in[\![d-1]\!]}$ the TT ranks *of $\boldsymbol{u}$;*

(ii). the HSVD algorithm 2.1 gives a TT decomposition of minimal TT ranks.

This theorem states that the ranks of the optimal TT representation of a tensor $\boldsymbol{u}$ is characterised by its reshapes $(\boldsymbol{u}^{\leq k})_{k\in[\![d-1]\!]}$. Moreover the HSVD algorithm 2.1 produces a TT representation of $\boldsymbol{u}$ with optimal ranks.

Proof. Let $(\widetilde{A}_1, \ldots, \widetilde{A}_d)$ be a TT representation of $\boldsymbol{u}$ of TT ranks $(\widetilde{r}_k)_{k\in[\![0;d]\!]}$. For any $k \in [\![d-1]\!]$, we have

$$(\boldsymbol{u}_{i_1\ldots i_k}^{i_{k+1}\ldots i_d}) = \underbrace{\begin{bmatrix} \widetilde{A}_1[1]\widetilde{A}_2[1]\cdots\widetilde{A}_k[1] \\ \vdots \\ \widetilde{A}_1[n_1]\widetilde{A}_2[n_2]\cdots\widetilde{A}_k[n_k] \end{bmatrix}}_{\in\mathbb{R}^{n_1\cdots n_k\times\widetilde{r}_k}} \underbrace{\begin{bmatrix} \widetilde{A}_{k+1}[1] & \cdots & \widetilde{A}_d[1] & \ldots & \widetilde{A}_{k+1}[n_{k+1}] & \cdots & \widetilde{A}_d[n_d] \end{bmatrix}}_{\in\mathbb{R}^{\widetilde{r}_k\times n_{k+1}\cdots n_d}}.$$

This shows that any TT representation of $\boldsymbol{u}$ has TT ranks at least $(\operatorname{Rank}(\boldsymbol{u}^{\leq k}))_{k\in[\![0;d]\!]}$.

Let $(A_1, \ldots, A_d)$ be the TT cores given by the HSVD algorithm. Using the same notation as in Algorithm 2.1, for any $k \in [\![d-1]\!]$ we have

$$\boldsymbol{u}^{\leq k} = \underbrace{\begin{bmatrix} A_1[1]A_2[1]\cdots A_k[1] \\ \vdots \\ A_1[n_1]A_2[n_2]\cdots A_k[n_k] \end{bmatrix}}_{\in\mathbb{R}^{n_1\cdots n_k\times r_k}} \Sigma_k V_k^{\mathsf{T}}.$$

The first matrix is a partial isometry by Proposition 2.2.9 hence the equation above is an SVD of $\boldsymbol{u}^{\leq k}$. By the properties of the SVD, we have that $r_k = \mathrm{Rank}\,(\boldsymbol{u}^{\leq k})$. □

An important consequence of Theorem 2.2.12 is the closedness of the set with prescribed TT ranks.

Proposition 2.2.13. *Let* $\mathbf{r} \in \mathbb{N}^{d+1}$*. The set of tensor trains with TT rank less that* $\mathbf{r}$

$$\mathscr{M}_{\mathrm{TT}_{\leq \mathbf{r}}} = \left\{ \boldsymbol{u} \mid \exists (A_k)_{k \in [\![d]\!]} \in \bigtimes_{k \in [\![d]\!]} \mathbb{R}^{r_{k-1} \times n_k \times r_k},\ \forall \mathbf{i} \in [\![\mathbf{n}]\!],\ \boldsymbol{u}_{i_1 \dots i_d} = A_1[i_1] \cdots A_d[i_d] \right\},$$

is a closed set.

Proof. The proof follows from the characterisation of the TT ranks given by Theorem 2.2.12: given a tensor $\boldsymbol{u}$, for $1 \leq k \leq d-1$, the minimal TT rank r_k is equal to the rank of the matrix $\boldsymbol{u}_{i_1 \dots i_k}^{i_{k+1} \dots i_d}$. We conclude by recalling that the set of matrices with rank less than r is a closed set. □

This proposition is in stark contrast with the set of tensors with a given canonical rank r

$$\mathscr{M}_{\mathrm{CP}_{\leq r}} = \left\{ \boldsymbol{u} \mid \forall \alpha \in [\![r]\!], \exists\, (v_j^{(\alpha)})_{j \in [\![d]\!]} \in \bigtimes_{j=1}^{d} \mathbb{R}^{n_j}, \boldsymbol{u} = \sum_{\alpha=1}^{r} v_1^{(\alpha)} \otimes \cdots \otimes v_d^{(\alpha)} \right\},$$

as the example exhibited in eq. (2.1.6) shows that the set $\mathscr{M}_{\mathrm{CP}_{\leq r}}$ is not closed if $d \geq 3$ and $r \geq 2$.

Since the set $\mathscr{M}_{\mathrm{TT}_{\leq \mathbf{r}}}$ is closed, we can safely study the question of the best approximation of a tensor with given TT ranks.

2.2.4 Approximation by TT

A natural way to reduce the TT ranks of the TT representation of a tensor is to truncate the SVD at each step of the HSVD algorithm to a tolerance ε:

$$\begin{aligned}
\boldsymbol{u}_{i_1 \dots i_d} &= \boldsymbol{u}_{i_1}^{i_2 \dots i_d} && (\text{reshape of } \boldsymbol{u} \text{ to } n_1 \times n_2 \cdots n_d) \\
&\simeq (U_1)_{i_1}^{\alpha_1} (\Sigma_1^{\varepsilon} V_1^{\mathsf{T}})_{\alpha_1}^{i_2 \dots i_d} && (\text{truncated SVD}) \\
&\simeq (U_1)_{i_1}^{\alpha_1} (\Sigma_1^{\varepsilon} V_1^{\mathsf{T}})_{\alpha_1 i_2}^{i_3 \dots i_d} && (\text{reshape of } \Sigma_1^{\varepsilon} V_1^{\mathsf{T}}) \\
&\simeq (U_1)_{i_1}^{\alpha_1} (U_2)_{\alpha_1 i_2}^{\alpha_2} (\Sigma_2^{\varepsilon} V_2^{\mathsf{T}})_{\alpha_2}^{i_3 \dots i_d} && (\text{truncated SVD of } \Sigma_1^{\varepsilon} V_1^{\mathsf{T}}) \\
&\simeq (U_1)_{i_1}^{\alpha_1} (U_2)_{\alpha_1 i_2}^{\alpha_2} (\Sigma_2^{\varepsilon} V_2^{\mathsf{T}})_{\alpha_2 i_3}^{i_4 \dots i_d} && (\text{reshape of } \Sigma_2^{\varepsilon} V_2^{\mathsf{T}}),
\end{aligned}$$

we repeat the process until we get

$$\boldsymbol{u}_{i_1 \dots i_d} \simeq (U_1)_{i_1}^{\alpha_1} (U_2)_{\alpha_1 i_2}^{\alpha_2} \cdots (U_{d-1})_{\alpha_{d-2} i_{d-1}}^{\alpha_{d-1}} (\Sigma_{d-1}^{\varepsilon} V_{d-1}^{\mathsf{T}})_{\alpha_{d-1}}^{i_d}.$$

Algorithm 2.2 Hierarchical SVD with truncations or TT-SVD

Input: Tensor $\boldsymbol{u} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, tolerance ε
Output: $(A_1, \dots, A_d)$ TT representation of $\boldsymbol{u}$

function TT-SVD($\boldsymbol{u}, \varepsilon$)
  $(T_0)_{\alpha_0 i_1}^{i_2 \dots i_d} = \boldsymbol{u}_{i_1}^{i_2 \dots i_d}$ (α_0 dummy index)
  for $k = 1, \dots, d-1$ **do**
    $U_k, \Sigma_k, V_k^\mathsf{T} = \mathtt{tsvd}\Big((T_{k-1})_{\alpha_{k-1} i_k}^{i_{k+1} \dots i_d}, \varepsilon\Big)$ ▷ Truncated SVD s. t. $\|\mathtt{tsvd}(A) - A\| \leq \varepsilon$
    $A_k[i_k]_{\alpha_{k-1}}^{\alpha_k} = (U_k)_{\alpha_{k-1} i_k}^{\alpha_k}, \quad \forall\, i_k \in [\![n_k]\!], \alpha_{k-1} \in [\![r_{k-1}]\!], \alpha_k \in [\![r_k]\!]$
    $(T_k)_{\alpha_k i_{k+1}}^{i_{k+2} \dots i_d} = \big(\Sigma_k V_k^\mathsf{T}\big)_{\alpha_k}^{i_{k+1} i_{k+2} \dots i_d}, \quad \forall (i_{k+1}, \dots, i_d) \in [\![(n_{k+1}, \dots, n_d)]\!], \alpha_k \in [\![r_k]\!]$
  end for
  $A_d[i_d]^{\alpha_{d-1}} = (T_{d-1})_{\alpha_{d-1}}^{i_d} = \big(\Sigma_{d-1} V_{d-1}^\mathsf{T}\big)_{\alpha_{d-1}}^{i_d}, \quad \forall\, i_d \in [\![n_d]\!], \alpha_{d-1} \in [\![r_{d-1}]\!]$
  return $(A_1, \dots, A_d)$
end function

This algorithm [Ose11] is called the HSVD algorithm with truncations or TT-SVD. It is summarised in Algorithm 2.2.

Truncating the successive SVDs gives an estimate on the best approximation by a tensor train of fixed TT ranks.

Theorem 2.2.14 ([Gra10, Ose11, Hac12, Hac14])**.** *Let $\boldsymbol{u} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, $\tilde{\mathbf{r}} \in \mathbb{N}^{d+1}$ and $\mathscr{M}_{\mathrm{TT} \leq \tilde{\mathbf{r}}}$ be the space of tensor trains of ranks bounded by $\tilde{\mathbf{r}}$. Then we have*

$$\min_{\boldsymbol{v} \in \mathscr{M}_{\mathrm{TT} \leq \tilde{\mathbf{r}}}} \|\boldsymbol{u} - \boldsymbol{v}\| \leq \sqrt{\sum_{k=1}^{d-1} \sum_{\alpha > \tilde{r}_k} {\sigma_\alpha^{(k)}}^2} \leq \sqrt{d-1} \min_{\boldsymbol{v} \in \mathscr{M}_{\mathrm{TT} \leq \tilde{\mathbf{r}}}} \|\boldsymbol{u} - \boldsymbol{v}\|,$$

where for $k \in [\![d-1]\!]$, $(\sigma_\alpha^{(k)})_{\alpha \in [\![r_k]\!]}$ are the singular values of the reshape $(\boldsymbol{u}^{\leq k}) \in \mathbb{R}^{n_1 \cdots n_k \times n_{k+1} \cdots n_d}$.

An important consequence of this theorem is that it is sufficient to derive bounds for the tale of the singular values of each reshape $(\boldsymbol{u}_{i_1 \dots i_k}^{i_{k+1} \dots i_d}) \in \mathbb{R}^{n_1 \cdots n_k \times n_{k+1} \cdots n_d}$ to get bounds on the TT ranks of a TT approximation. This considerably simplifies the question of the approximability by TT of a given tensor, and this characterisation will be used to study the eigenvalue problems of particular operators in Chapter **??**.

The proof of this theorem relies on a close inspection of the HSVD algorithm with truncations 2.2.

Proof of Theorem 2.2.14. For the lower bound on the best approximation $\boldsymbol{u}_{\mathrm{best}} \in \mathscr{M}_{\mathrm{TT} \leq \tilde{\mathbf{r}}}$, we have for each $k \in [\![d-1]\!]$ by definition of the SVD truncation

$$\|\boldsymbol{u}^{\leq k} - \mathtt{tsvd}(\boldsymbol{u}^{\leq k}, \tilde{r}_k)\|^2 \leq \|\boldsymbol{u} - \boldsymbol{u}_{\mathrm{best}}\|^2,$$

as $(\boldsymbol{u}_{\text{best}})_{i_1\ldots i_k}^{i_{k+1}\cdots i_d}$ is a matrix of rank $\tilde{r}_k$. Since $\|\boldsymbol{u}^{\leq k} - \texttt{tsvd}(\boldsymbol{u}^{\leq k}, \tilde{r}_k)\|^2 = \sum_{\alpha > \tilde{r}_k} {\sigma_\alpha^{(k)}}^2$, by summing over $k \in [\![d-1]\!]$ we get the lower bound.

We now prove the upper bound. With the notation in Algorithm 2.2, for $k \in [\![d-1]\!]$, let

$$Q_k = \big(A_1 \bowtie \cdots \bowtie A_k\big)^{\leq k} \in \mathbb{R}^{n_1\cdots n_k \times r_k}.$$

Since by the algorithm $(A_j)_{j\in[\![k]\!]}$ are right-orthogonal, Q_k has orthonormal columns. Let $\Pi_k = Q_k Q_k^\mathsf{T} \in \mathbb{R}^{n_1\cdots n_k \times n_1 \cdots n_k}$. Since Q_k has orthonormal columns, Π_k defines an orthogonal projection. Let $\widetilde{\boldsymbol{u}}_1 = \boldsymbol{u}$, and define recursively the tensor $\widetilde{\boldsymbol{u}}_k \in \mathbb{R}^{n_1\times\cdots\times n_d}$, for $2 \leq k \leq d$, by

$$(\widetilde{\boldsymbol{u}}_k)^{\leq k-1} = \Pi_{k-1}(\widetilde{\boldsymbol{u}}_{k-1})^{\leq k-1}.$$

With the notation in Algorithm 2.2, we have for each $k \in [\![2, d]\!]$,

$$\begin{aligned}
(\widetilde{\boldsymbol{u}}_k)^{\leq k-1} &= \Pi_{k-1}(\widetilde{\boldsymbol{u}}_{k-1})^{\leq k-1} \\
&= Q_{k-1}Q_{k-1}^\mathsf{T}(\widetilde{\boldsymbol{u}}_{k-1})^{\leq k-1} \\
&= \big(A_1 \bowtie \cdots \bowtie A_{k-1}\big)^{\leq k-1}\big((T_{k-1})_{\alpha_{k-1}}^{i_k\ldots i_d}\big).
\end{aligned}$$

As $T_{d-1} = A_d$, by iteration we have that $A_1 \bowtie \cdots \bowtie A_d = \widetilde{\boldsymbol{u}}_d$, and thus $\boldsymbol{u} - \widetilde{\boldsymbol{u}}_d = \sum_{i=1}^{d-1} \widetilde{\boldsymbol{u}}_i - \widetilde{\boldsymbol{u}}_{i+1}$. Now,

$$\begin{aligned}
\|\boldsymbol{u} - \widetilde{\boldsymbol{u}}_d\|_F^2 &= \Big\| \sum_{i=1}^{d-1} \widetilde{\boldsymbol{u}}_i - \widetilde{\boldsymbol{u}}_{i+1} \Big\|_F^2 = \Big\| \sum_{i=1}^{d-1} (\widetilde{\boldsymbol{u}}_i - \widetilde{\boldsymbol{u}}_{i+1})^{\leq 1} \Big\|_F^2 \\
&= \|\widetilde{\boldsymbol{u}}_1 - \widetilde{\boldsymbol{u}}_2\|_F^2 + \Big\| \sum_{i=2}^{d-1} \widetilde{\boldsymbol{u}}_i - \widetilde{\boldsymbol{u}}_{i+1} \Big\|_F^2,
\end{aligned}$$

since the range of $(\widetilde{\boldsymbol{u}}_i - \widetilde{\boldsymbol{u}}_{i+1})^{\leq 1}$ for $2 \leq i \leq d-1$ is spanned by the range of Π_1, and $\widetilde{\boldsymbol{u}}_1^{\leq 1} - \widetilde{\boldsymbol{u}}_2^{\leq 1} = (\mathbf{I}_{n_1} - \Pi_1)\widetilde{\boldsymbol{u}}_1^{\leq 1}$. By repeating the same argument, we have by iteration

$$\|\boldsymbol{u} - \widetilde{\boldsymbol{u}}_d\|_F^2 = \sum_{i=1}^{d-1} \big\|\widetilde{\boldsymbol{u}}_i - \widetilde{\boldsymbol{u}}_{i+1}\big\|_F^2.$$

For $i \in [\![d-1]\!]$, we have

$$\big\|\widetilde{\boldsymbol{u}}_i - \widetilde{\boldsymbol{u}}_{i+1}\big\|_F^2 = \big\|\widetilde{\boldsymbol{u}}_i^{\leq i} - \Pi_i \widetilde{\boldsymbol{u}}_i^{\leq i}\big\|_F^2 = \sum_{\alpha_i > r_i} \sigma_{\alpha_i}\big(\widetilde{\boldsymbol{u}}_i^{\leq i}\big)^2.$$

It remains to show that $\sum_{\alpha_i > r_i} \sigma_{\alpha_i}\big(\widetilde{\boldsymbol{u}}_i^{\leq i}\big)^2 \leq \sum_{\alpha_i > r_i} \sigma_{\alpha_i}\big(\boldsymbol{u}^{\leq i}\big)^2$. We have for all $k \in [\![2, d]\!]$

$$(\widetilde{\boldsymbol{u}}_k)^{\leq k-1} = \Pi_{k-1}(\widetilde{\boldsymbol{u}}_{k-1})^{\leq k-1},$$

which can be rewritten

$$\widetilde{\boldsymbol{u}}_k = \big(\Pi_{k-1} \otimes \mathrm{id}_{k:d}\big)\widetilde{\boldsymbol{u}}_{k-1}.$$

Thus by iteration, for any $k \in [\![2, d]\!]$ we have

$$\begin{aligned}\widetilde{\boldsymbol{u}}_k &= \big(\Pi_{k-1} \otimes \mathrm{id}_{k:d}\big)\widetilde{\boldsymbol{u}}_{k-1} = \big(\Pi_{k-1} \otimes \mathrm{id}_{k:d}\big)\big(\Pi_{k-2} \otimes \mathrm{id}_{\mathrm{id}_{k-1:d}}\big)\widetilde{\boldsymbol{u}}_{k-2} \\ &= \big(\Pi_{k-1} \otimes \mathrm{id}_{k:d}\big)\big(\Pi_{k-2} \otimes \mathrm{id}_{\mathrm{id}_{k-1:d}}\big)\cdots\big(\Pi_1 \otimes \mathrm{id}_{2:d}\big)\boldsymbol{u},\end{aligned}$$

using that $\widetilde{\boldsymbol{u}}_1 = \boldsymbol{u}$. This means that

$$\widetilde{\boldsymbol{u}}_k^{\leq k} = \big(\Pi_{k-1} \otimes \mathrm{id}_k\big)\big(\Pi_{k-2} \otimes \mathrm{id}_{k-1:k}\big)\cdots\big(\Pi_1 \otimes \mathrm{id}_{2:k}\big)\boldsymbol{u}^{\leq k}.$$

As $(\Pi_j)_{j \in [\![k-1]\!]}$ are orthogonal projectors, the matrices $(\Pi_j \otimes \mathrm{id}_{j+1:k})$ are also orthogonal projectors, thus the operator norm of $\big(\Pi_{k-1} \otimes \mathrm{id}_k\big)\big(\Pi_{k-2} \otimes \mathrm{id}_{k-1:k}\big)\cdots\big(\Pi_1 \otimes \mathrm{id}_{2:k}\big)$ is bounded by 1. Using the variational characterisation of the singular values (2.1.1), we have that

$$\begin{aligned}\sigma_\alpha\big(\widetilde{\boldsymbol{u}}_k^{\leq k}\big) = \sigma_\alpha\big((\Pi_{k-1} \otimes \mathrm{id}_k)\cdots(\Pi_1 \otimes \mathrm{id}_{2:k})\boldsymbol{u}^{\leq k}\big) &\leq \big\|(\Pi_{k-1} \otimes \mathrm{id}_k)\cdots(\Pi_1 \otimes \mathrm{id}_{2:k})\big\|_{\mathrm{op}}\sigma_\alpha(\boldsymbol{u}^{\leq k}) \\ &\leq \sigma_\alpha(\boldsymbol{u}^{\leq k}).\end{aligned}$$

This finishes the proof of the theorem. □

A drawback of the HSVD algorithm or its truncated version is that it requires to handle the full tensor. This means that the cost of the HSVD algorithm with truncations is still exponential in the number of modes d.

If the tensor is already in a TT format, it is possible to reduce the cost of this truncation, provided that the TT cores have the right normalisation. Let $(A_1, \dots, A_d)$ be a right-orthogonal TT representation of the tensor $\boldsymbol{u} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$. The first reshape is

$$\boldsymbol{u}^{\leq 1} = \begin{bmatrix} A_1[1] \\ \vdots \\ A_1[n_1] \end{bmatrix} \begin{bmatrix} A_2[1]\cdots A_d[1] & \dots & A_2[n_2]\cdots A_d[n_d] \end{bmatrix},$$

and since the TT cores $(A_2, \dots, A_d)$ are right-orthogonal, the matrix $V_2 = \begin{bmatrix} A_2[1]\cdots A_d[1] & \dots & A_2[n_2]\cdots A_d[n_d] \end{bmatrix}$ satisfies $V_2V_2^\mathsf{T} = \mathrm{id}_{r_1}$. Hence the first step of the HSVD truncation can be reduced to the SVD of the reshape of A_1. The same would hold for the next step of the HSVD truncation, hence the total cost of the TT compression of $\boldsymbol{u}$ in a TT format is reduced to $\mathcal{O}(dnr^3)$ where $r = \max(r_k)$ and $n = \max(n_k)$.

The algorithm is summarised in Algorithm 2.3, starting with a right-orthogonal TT representation of the input tensor. It is often called *TT rounding* – as it is conceptually the same operation as rounding a float – or *TT compression*. In practice, the TT to compress is not given in a right-orthogonal representation. In that case, successive LQ orthogonalisations need to be performed in order to retrieve a right-orthogonal TT representation. This step can be costly in practice as the successive LQ factorisations do not decrease the rank of the TT cores.

Algorithm 2.3 TT rounding algorithm

Input: $(A_1, \dots, A_d)$ right-orthogonal TT representation, $\varepsilon > 0$ tolerance
Output: $(A_1^\varepsilon, \dots, A_d^\varepsilon)$ TT representation such that $\|\mathrm{TT}(A_i^\varepsilon) - \mathrm{TT}(A_i)\| \leq \sqrt{d-1}\,\varepsilon$

function TT-ROUNDING($(A_1, \dots, A_d), \varepsilon$)
  for $k = 1, \dots, d-1$ **do**
    $U_k, \Sigma_k, V_k^\mathsf{T} = \mathtt{tsvd}\left(\begin{bmatrix} A_k[1] \\ \vdots \\ A_k[n_k] \end{bmatrix}, \varepsilon\right)$ ▷ Truncated SVD s. t. $\|\mathtt{tsvd}(A) - A\| \leq \varepsilon$
    $r_k = \mathtt{size}(\Sigma_k)$
    $(A_k^\varepsilon)_{i_k \alpha_{k-1}}^{\alpha_k} = (U_k)_{i_k \alpha_{k-1}}^{\alpha_k}, \quad \forall\, i_k \in [\![n_k]\!], \alpha_{k-1} \in [\![r_{k-1}]\!], \alpha_k \in [\![r_k]\!]$
    $A_{k+1}[i_{k+1}] = \Sigma_k V_k^\mathsf{T} A_{k+1}[i_{k+1}], \quad \forall\, i_{k+1} \in [\![n_{k+1}]\!]$ ▷ Root shifting
  end for
  $A_d^\varepsilon = A_d$
  return $(A_1^\varepsilon, \dots, A_d^\varepsilon)$
end function

2.2.5 Gauge fixing

Such normalisations turn out to be convenient for the computation of the norm a tensor, for example to get the normalisation of the tensor, as stated in Proposition 2.2.9. Another instance where the choice of the normalisation is crucial is in solving eigenvalue problems in DMRG (see Chapter 5). In this section, we give results on the gauge remaining from the TT normalisation introduced in Section 2.2.2. We also present algorithms to obtain a TT decomposition with a prescribed normalisation.

Previously, we have seen the left and right orthogonalisation, but it is also possible to interpolate between both. In that case, the norm of the tensor is carried by the TT core that is not normalised:

$$\begin{aligned}
\|\boldsymbol{u}\|^2 &= \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} A_d[i_d]^\mathsf{T} \cdots A_1[i_1]^\mathsf{T} A_1[i_1] \cdots A_d[i_d] \\
&= \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} \operatorname{Tr}\left(A_d[i_d]^\mathsf{T} \cdots A_1[i_1]^\mathsf{T} A_1[i_1] \cdots A_d[i_d]\right) \\
&= \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} \operatorname{Tr}\left(A_{\ell+1}[i_{\ell+1}] \cdots A_d[i_d] A_d[i_d]^\mathsf{T} \cdots A_1[i_1]^\mathsf{T} A_1[i_1] \cdots A_\ell[i_\ell]\right) \\
&= \sum_{i_\ell=1}^{n_\ell} \operatorname{Tr}\left(A_\ell[i_\ell]^\mathsf{T} A_\ell[i_\ell]\right).
\end{aligned}$$

A representation of this type can be obtained by slightly modifying the hierarchical SVD described earlier. Instead of performing SVDs from left to right, one stops the SVDs from the left to the cut ℓ and does the SVDs from the right. For example for $\ell = 2$, we have (using again Einstein convention)

$$\begin{aligned}
\boldsymbol{u}_{i_1\dots i_d} &= (\boldsymbol{u}_{i_1}^{i_2\dots i_d}) && \text{(reshape of } \boldsymbol{u} \text{ to } n_1 \times n_2 \cdots n_d)\\
&= (U_1)_{i_1}^{\alpha_1} (\Sigma_1 V_1^\mathsf{T})_{\alpha_1}^{i_2\dots i_d} && \text{(SVD)}\\
&= (U_1)_{i_1}^{\alpha_1} (\Sigma_1 V_1^\mathsf{T})_{\alpha_1 i_2\dots i_{d-1}}^{i_d} && \text{(reshape of } \Sigma_1 V_1^\mathsf{T})\\
&= (U_1)_{i_1}^{\alpha_1} (U_{d-1}\Sigma_{d-1})_{\alpha_1 i_2\dots i_{d-1}}^{\alpha_{d-1}} (V_{d-1}^\mathsf{T})_{\alpha_{d-1}}^{i_d} && \text{(SVD of } \Sigma_1 V_1^\mathsf{T})\\
&= (U_1)_{i_1}^{\alpha_1} (U_{d-1}\Sigma_{d-1})_{\alpha_1 i_2\dots i_{d-2}}^{\alpha_{d-1} i_{d-1}} (V_{d-1}^\mathsf{T})_{\alpha_{d-1}}^{i_d} && \text{(reshape of } U_{d-1}\Sigma_{d-1}),
\end{aligned}$$

where we repeat the process until we get

$$\boldsymbol{u}_{i_1\dots i_d} = (U_1)_{i_1}^{\alpha_1} (U_2\Sigma_2)_{\alpha_1 i_2}^{\alpha_2} \cdots (V_{d-2}^\mathsf{T})_{\alpha_{d-2} i_{d-1}}^{\alpha_{d-1}} (V_{d-1}^\mathsf{T})_{\alpha_{d-1}}^{i_d}.$$

The TT decomposition then reads as

$$\begin{aligned}
\boldsymbol{u}_{i_1\dots i_d} &= (U_1)_{i_1}^{\alpha_1} \quad (U_2\Sigma_2)_{\alpha_1 i_2}^{\alpha_2} \quad \cdots \quad (V_{d-2}^\mathsf{T})_{\alpha_{d-2} i_{d-1}}^{\alpha_{d-1}} \quad (V_{d-1}^\mathsf{T})_{\alpha_{d-1}}^{i_d}\\
&= A_1[i_1]_{\alpha_1} \quad A_2[i_2]_{\alpha_2}^{\alpha_1} \quad \cdots \quad A_{d-1}[i_{d-1}]_{\alpha_{d-1}}^{\alpha_{d-2}} \quad A_d[i_d]^{\alpha_{d-1}},
\end{aligned}$$

where $(A_1, \dots, A_{\ell-1})$ are left-orthogonal and $(A_{\ell+1}, \dots, A_d)$ are right-orthogonal.

Conversion between left and right orthogonal TT representations

By successive LQ decompositions, it is possible to transform a left-orthogonal to a right orthogonal TT decomposition. Let $(A_1, \dots, A_d)$ be a left-orthogonal TT decomposition of $\boldsymbol{u} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$. Then we have

$$\begin{aligned}
\boldsymbol{u}_{i_1\dots i_d} &= A_1[i_1] \cdots A_d[i_d]\\
&= A_1[i_1]^{\alpha_1} A_2[i_2]_{\alpha_1}^{\alpha_2} \cdots A_{d-1}[i_{d-1}]_{\alpha_{d-2}}^{\alpha_{d-1}} (A_d)_{\alpha_{d-1}}^{i_d}\\
&= A_1[i_1]^{\alpha_1} A_2[i_2]_{\alpha_1}^{\alpha_2} \cdots A_{d-1}[i_{d-1}]_{\alpha_{d-2}}^{\alpha_{d-1}} (L_d)_{\alpha_{d-1}}^{\beta_{d-1}} (Q_d)_{\beta_{d-1}}^{i_d}\\
&= A_1[i_1]^{\alpha_1} A_2[i_2]_{\alpha_1}^{\alpha_2} \cdots A_{d-2}[i_{d-2}]_{\alpha_{d-3}}^{\alpha_{d-2}} (A_{d-1}L_d)_{\alpha_{d-2}}^{i_{d-1}\beta_{d-1}} (Q_d)_{\beta_{d-1}}^{i_d}\\
&= A_1[i_1]^{\alpha_1} A_2[i_2]_{\alpha_1}^{\alpha_2} \cdots A_{d-2}[i_{d-2}]_{\alpha_{d-3}}^{\alpha_{d-2}} (L_{d-1})_{\alpha_{d-2}}^{\beta_{d-2}} (Q_{d-1})_{\beta_{d-2}}^{i_{d-1}\beta_{d-1}} (Q_d)_{\beta_{d-1}}^{i_d},
\end{aligned}$$

we repeat this process until we reach

$$\begin{aligned}
\boldsymbol{u}_{i_1\dots i_d} &= (A_1 L_2)^{i_1\beta_1} \quad (Q_2)_{\beta_1}^{i_2\beta_2} \quad \cdots \quad (Q_{d-1})_{\beta_{d-2}}^{i_{d-1}\beta_{d-1}} \quad (Q_d)_{\beta_{d-1}}^{i_d}\\
&= B_1[i_1]_{\beta_1} \quad B_2[i_2]_{\beta_2}^{\beta_1} \quad \cdots \quad B_{d-1}[i_{d-1}]_{\beta_{d-1}}^{\beta_{d-2}} \quad B_d[i_d]^{\beta_{d-1}}.
\end{aligned}$$

We simply need to check that the TT cores $B_2, \dots, B_d$ are right-orthogonal: for any $k \in [\![2; d]\!]$, we have

$$\sum_{i_k=1}^{n_k} \left(B_k[i_k] B_k[i_k]^\mathsf{T}\right)_{\alpha_{k-1}\tilde{\alpha}_{k-1}} = \sum_{i_k=1}^{n_k} \sum_{\alpha_k=1}^{r_k} (Q_k)^{i_k \alpha_k}_{\alpha_{k-1}} (Q_k)^{i_k \alpha_k}_{\tilde{\alpha}_{k-1}} = \delta_{\alpha_{k-1}\tilde{\alpha}_{k-1}}.$$

The whole process is summarised in the Algorithm 2.4.

Algorithm 2.4 Right orthogonalisation algorithm

Input: $(A_1, \dots, A_d)$ TT representation of a tensor $\boldsymbol{u}$
Output: $(B_1, \dots, B_d)$ right-orthogonal TT representation of $\boldsymbol{u}$

function RightOrthogonalisation($(A_1, \dots, A_d)$)
  for $k = d, \dots, 2$ **do**
    $L_k, Q_k = \mathtt{lq}\Big(\begin{bmatrix} A_k[1] & \dots & A_k[n_k] \end{bmatrix}, \varepsilon \Big)$ ▷ LQ factorisation
    $(B_k)^{i_k \alpha_k}_{\alpha_{k-1}} = (Q_k)^{i_k \alpha_k}_{\alpha_{k-1}}, \quad \forall\, i_k \in [\![n_k]\!], \alpha_{k-1} \in [\![r_{k-1}]\!], \alpha_k \in [\![r_k]\!]$
    $A_{k-1}[i_{k-1}] = A_{k-1}[i_{k-1}] L_k, \quad \forall\, i_{k-1} \in [\![n_{k-1}]\!]$
  end for
  return $(B_1, \dots, B_d)$
end function

These normalisations have the advantage of reducing the gauge freedom in the TT representation.

Proposition 2.2.15 (Gauge freedom of left-orthogonal TT decompositions [HRS12b])**.** *A left-orthogonal TT representation of minimal TT rank $(r_1, \dots, r_{d-1})$ is unique up to the insertion of orthogonal matrices,* i.e. *if $(A_1, \dots, A_d)$ and $(B_1, \dots, B_d)$ are left-orthogonal TT representations of the same tensor $\boldsymbol{u}$, then there are orthogonal matrices $(Q_k)_{1 \leq k \leq d-1}$, $Q_k \in \mathbb{R}^{r_k \times r_k}$ such that for all $1 \leq i_k \leq n_k$ we have*

$$\begin{aligned} A_1[i_1] Q_1 = B_1[i_1], \quad Q_{d-1}^\mathsf{T} A_d[i_d] = B_d[i_d] \\ Q_{k-1}^\mathsf{T} A_k[i_k] Q_k = B_k[i_k], \text{ for } k = 2, \dots, d-1. \end{aligned} \tag{2.2.5}$$

Similar statements would be true for other types of normalisations.

Proof. The proof relies on the following observation: let $M_1, N_1 \in \mathbb{R}^{p \times r}$ and $M_2, N_2 \in \mathbb{R}^{r \times q}$ be matrices of rank r such that

$$M_1 M_2 = N_1 N_2 \quad \text{and} \quad M_1^\mathsf{T} M_1 = N_1^\mathsf{T} N_1 = \mathrm{id}_r,$$

there is an orthogonal matrix $Q \in \mathbb{R}^{r \times r}$ such that

$$M_1 = N_1 Q \quad \text{and} \quad M_2 = Q^\mathsf{T} N_2.$$

The proof of this result is straightforward. We have $N_2 = N_1^\top M_1 M_2 = N_1^\top M_1 M_1^\top N_1 N_2$. Since N_2 is full-rank, it shows that $N_1^\top M_1$ is an orthogonal matrix. Denote this matrix Q. Hence $N_2 = QM_2$ and $M_1 N_1^\top N_1 = M_1$ thus, $N_1 = M_1 Q^\top$.

The proof then goes by iteration. We have

$$\begin{aligned} \big(A_1[i_1]\big)\big(A_2[i_2]\cdots A_d[i_d]\big) &= \big(B_1[i_1]\big)\big(B_2[i_2]\cdots B_d[i_d]\big) \\ \sum_{i_1=1}^{n_1} A_1[i_1]^\top A_1[i_1] &= \sum_{i_1=1}^{n_1} B_1[i_1]^\top B_1[i_1] = \mathrm{id}_{r_1}\,. \end{aligned}$$

Since $\big(A_1[i_1]\big), \big(A_2[i_2]\cdots A_d[i_d]\big), \big(B_1[i_1]\big)$ and $\big(B_2[i_2]\cdots B_d[i_d]\big)$ have rank r_1, by applying the previous result, there is an orthogonal matrix $Q_1 \in \mathbb{R}^{r_1\times r_1}$ such that

$$\begin{aligned} A_1[i_1]Q_1 &= B_1[i_1] \\ Q_1^\top\big(A_2[i_2]\cdots A_d[i_d]\big) &= \big(B_2[i_2]\cdots B_d[i_d]\big). \end{aligned}$$

For the next iteration, we have

$$\begin{aligned} \big(Q_1^\top A_2[i_2]\big)\big(A_3[i_3]\cdots A_d[i_d]\big) &= \big(B_2[i_2]\big)\big(B_3[i_3]\cdots B_d[i_d]\big) \\ \sum_{i_2=1}^{n_2} A_2[i_2]^\top Q_1 Q_1^\top A_2[i_2] &= \sum_{i_2=1}^{n_2} B_2[i_2]^\top B_2[i_2] = \mathrm{id}_{r_1}\,. \end{aligned}$$

Applying again the lemma, we have

$$\begin{aligned} Q_1^\top A_2[i_2]Q_2 &= B_2[i_2] \\ Q_2^\top\big(A_3[i_3]\cdots A_d[i_d]\big) &= \big(B_3[i_3]\cdots B_d[i_d]\big). \end{aligned}$$

By iteration, we prove the proposition. □

The Vidal representation

A convenient - albeit numerically unstable - way to convert easily between left-orthogonal and right-orthogonal TT representations is to use the Vidal representation [Vid03].

Definition 2.2.16 (Vidal representation [Vid03]). *Let $\boldsymbol{u} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ be a tensor. We say that $(\Gamma_k)_{1\leq k\leq d}$, $(\Sigma_k)_{1\leq k\leq d-1}$ is a Vidal representation if Σ_k are diagonal matrices with positive diagonal entries, for all $\mathbf{i} \in [\![\mathbf{n}]\!]$,*

$$\boldsymbol{u}_{i_1,\ldots,i_d} = \Gamma_1[i_1]\Sigma_1\Gamma_2[i_2]\Sigma_2\cdots\Sigma_{d-1}\Gamma_d[i_d], \tag{2.2.6}$$

and for all $k \in [\![d]\!]$, the matrices $\Gamma_k \in \mathbb{R}^{n_k\times r_{k-1}\times r_k}$ satisfy

$$\sum_{i_1=1}^{n_1}\Gamma_1[i_1]^\top\Gamma_1[i_1] = \mathrm{id}_{r_1}, \quad \sum_{i_d=1}^{n_d}\Gamma_d[i_d]\Gamma_d[i_d]^\top = \mathrm{id}_{r_{d-1}} \tag{2.2.7}$$

$$\forall\, k = 2,\ldots,d-1, \;\sum_{i_k=1}^{n_k}\Gamma_k[i_k]^\top\Sigma_{k-1}^2\Gamma_k[i_k] = \mathrm{id}_{r_k}, \quad \sum_{i_k=1}^{n_k}\Gamma_k[i_k]\Sigma_k^2\Gamma_k[i_k]^\top = \mathrm{id}_{r_{k-1}}\,. \tag{2.2.8}$$

The Vidal representation directly gives left and right orthogonal TT decompositions:

(i). $(A_1, \ldots, A_d)$ left-orthogonal TT representation

$$A_1[i_1] = \Gamma_1[i_1], \quad A_d[i_d] = \Sigma_{d-1}\Gamma_d[i_d]$$
$$A_k[i_k] = \Sigma_{k-1}\Gamma_k[i_k], \quad k \in [\![2; d-1]\!];$$

(ii). $(B_1, \ldots, B_d)$ right-orthogonal TT representation

$$B_1[i_1] = \Gamma_1[i_1]\Sigma_1, \quad B_d[i_d] = \Gamma_d[i_d]$$
$$B_k[i_k] = \Gamma_k[i_k]\Sigma_k, \quad k \in [\![2; d-1]\!].$$

The conversion from left (or right) orthogonal decomposition to a Vidal representation is more involved [Sch11, Section 4.6]. Let $(A_k)_{1\leq k\leq d}$ be the TT components of a left-orthogonal TT representation. Then we have

$$\boldsymbol{u}_{i_1\ldots i_k}^{i_{k+1}\ldots i_d} = \underbrace{\begin{bmatrix} A_1[1]A_2[1]\cdots A_k[1] \\ \vdots \\ A_1[n_1]A_2[n_2]\cdots A_k[n_k] \end{bmatrix}}_{=:M_k\in\mathbb{R}^{n_1\cdots n_k\times r_k}} \underbrace{\begin{bmatrix} A_{k+1}[i_{k+1}] & \cdots & A_d[i_d] \end{bmatrix}}_{\in\mathbb{R}^{r_k\times n_{k+1}\cdots n_d}}$$

Because (A_k) are left-orthogonal, then for all $k \in [\![d-1]\!]$, $M_k^\mathsf{T} M_k = \mathrm{id}_{r_k}$, hence the singular values of the reshaped tensor are exactly the singular values of the right matrix.

With this remark, we can now write the iterative algorithm to get the Vidal representation of the tensor.

Algorithm 2.5 Left-orthogonal to Vidal representation

Input: $(A_1, \ldots, A_d)$ left-orthogonal TT representation
Output: $(\Gamma_1, \ldots, \Gamma_d), (\Sigma_1, \ldots, \Sigma_{d-1})$ Vidal representation

```
function LeftToVidal((A_1, ..., A_d))
    U_{d-1}, Σ_{d-1}, V_d^T = svd( [A_d[1] A_d[2] ··· A_d[n_d]] )
    [Γ_d[1] ··· Γ_d[n_d]] = V_d^T
    for k = d-1, ..., 1 do
        U_{k-1}, Σ_{k-1}, V_k^T = svd( [A_k[1]U_kΣ_k ··· A_k[n_k]U_kΣ_k] ).
        Γ_k solution to V_k^T = [Γ_k[1]Σ_k ··· Γ_k[n_k]Σ_k]
    end for
    return (Γ_1, ..., Γ_d), (Σ_1, ..., Σ_{d-1}).
end function
```

By induction, one can show that the singular values of the successive SVD in the previous algorithm are indeed the singular values of the tensor reshape.

Proposition 2.2.17. *Let* $(\Gamma_k)_{1\leq k\leq d}$, $(\Sigma_k)_{1\leq k\leq d-1}$ *be a Vidal representation of* $\boldsymbol{u} \in \mathbb{R}^{n_1\times\cdots\times n_d}$. *Then* Σ_k *is the matrix of the singular values of the reshape* $\boldsymbol{u}_{i_1\ldots i_k}^{i_{k+1}\ldots i_d} \in \mathbb{R}^{n_1\cdots n_k \times n_{k+1}\cdots n_d}$.

Proof. By definition of the SVD, the Vidal TT components Γ_k satisfy

$$\sum_{i_k=1}^{n_k} \Gamma_k[i_k]\Sigma_k^2\Gamma_k[i_k]^\top = \mathrm{id}_{r_{k-1}} .$$

We also have

$$\begin{bmatrix} A_k[1]U_k & \cdots A_k[n_k]U_k \end{bmatrix} = \begin{bmatrix} U_{k-1}\Sigma_{k-1}\Gamma_k[1] & \cdots & U_{k-1}\Sigma_{k-1}\Gamma_k[n_k] \end{bmatrix} .$$

Thus

$$\begin{aligned} \sum_{i_k}^{n_k} \Gamma_k[i_k]^\top \Sigma_{k-1}^2 \Gamma_k[i_k] &= \sum_{i_k}^{n_k} \Gamma_k[i_k]^\top \Sigma_{k-1} U_{k-1}^\top U_{k-1} \Sigma_{k-1}\Gamma_k[i_k] \\ &= \sum_{i_k}^{n_k} U_k^\top A_k[i_k]^\top A_k[i_k] U_k \\ &= \mathrm{id}_{r_k} . \end{aligned}$$

□

2.3 Manifold of tensor trains

Proposition 2.3.1. *The set of tensor trains with TT rank* $\mathbf{r} = (r_1, \ldots, r_{d-1})$

$$\mathscr{M}_{\mathrm{TT}_\mathbf{r}} = \Big\{ \boldsymbol{u} \mid \ \exists (A_k)_{k\in[\![d]\!]} \in \bigtimes_{k\in[\![d]\!]} \mathbb{R}^{r_{k-1}\times n_k\times r_k},\ \forall\, \mathbf{i} \in [\![\mathbf{n}]\!], \boldsymbol{u}_{i_1\ldots i_d} = A_1[i_1]\cdots A_d[i_d], \\ \forall\, k \in [\![d-1]\!],\ \mathrm{rank}(\boldsymbol{u}_{i_1\ldots i_k}^{i_{k+1}\ldots i_d}) = r_k \Big\},$$

is a manifold of dimension

$$\dim \mathscr{M}_{\mathrm{TT}_\mathbf{r}} = \sum_{i=1}^{d} r_{i-1}n_i r_i - \sum_{i=1}^{d-1} r_i^2. \tag{2.3.1}$$

Proof. Two TT representations $(A_1, \ldots, A_d)$ and $(\tilde{A}_1, \ldots, \tilde{A}_d)$ of a same tensor are related by a gauge $(G_1, \ldots, G_{d-1}) \in \mathrm{GL}_{r_1}(\mathbb{R}) \times \cdots \mathrm{GL}_{r_{d-1}}(\mathbb{R})$

$$\forall\, 1 \leq i_k \leq n_k, A_k[i_k] = G_{k-1}\tilde{A}_k[i_k]G_k, \quad k = 1, \ldots, d, \quad (G_0 = G_d = 1).$$

The dimension of $\mathrm{GL}_{r_k}(\mathbb{R})$ is r_k^2, hence the dimension of $\mathscr{M}_{\mathrm{TT}_\mathbf{r}}$ is

$$\dim \mathscr{M}_{\mathrm{TT}_\mathbf{r}} = \sum_{i=1}^{d} r_{i-1} n_i r_i - \sum_{i=1}^{d-1} r_i^2.$$

□

Proposition 2.3.2 (Tangent space of $\mathscr{M}_{\mathrm{TT}_\mathbf{r}}$ [HRS12b]). *Let $A \in \mathscr{M}_{\mathrm{TT}_\mathbf{r}}$ and $(A_1, \dots, A_d)$ be a left-orthogonal TT representation of A. Let $\delta A \in \mathcal{T}_A \mathscr{M}_{\mathrm{TT}_\mathbf{r}}$.*

There are unique components $(W_k)_{1\leq k\leq d} \in \bigotimes_{k=1}^d \mathbb{R}^{r_{k-1}\times n_k \times r_k}$ such that

$$\delta A = \sum_{k=1}^{d} \delta A^{(k)}, \tag{2.3.2}$$

with

$$\delta A^{(k)}_{i_1\dots i_d} = A_1[i_1]\cdots A_{k-1}[i_{k-1}] W_k[i_k] A_{k+1}[i_{k+1}]\cdots A_d[i_d], \tag{2.3.3}$$

and where for $k \in [\![d-1]\!]$ we have

$$\sum_{i_k=1}^{n_k} A_k[i_k]^\mathsf{T} W_k[i_k] = \mathbf{0}_{r_k\times r_k}. \tag{2.3.4}$$

Proof. By definition of the tangent space $\mathcal{T}_A \mathscr{M}_{\mathrm{TT}_\mathbf{r}}$, the tangent vectors are given by the derivatives $\dot\Gamma$ of the differentiable curves $\Gamma : \mathbb{R} \to \mathscr{M}_{\mathrm{TT}_\mathbf{r}}$ such that $\Gamma(0) = A$.

For all $t \in \mathbb{R}$, since $\Gamma(t) \in \mathscr{M}_{\mathrm{TT}_\mathbf{r}}$, we can choose a left-orthogonal TT representation of $\Gamma(t)$ such that

$$\Gamma(t)_{i_1\dots i_d} = \Gamma_1^{(t)}[i_1]\cdots \Gamma_d^{(t)}[i_d],$$

where for all $1 \leq k \leq d$, $t \mapsto \Gamma_k^{(t)} \in \mathbb{R}^{n_k\times r_{k-1}\times r_k}$ is differentiable and $\Gamma_k^{(0)} = A_k$.

Since for $k \in [\![d-1]\!]$, $\sum_{i_k=1}^{n_k} \Gamma_k^{(t)}[i_k]^\mathsf{T} \Gamma_k^{(t)}[i_k] = \mathrm{id}_{r_k}$, there is a differentiable function $t \mapsto U_k(t) \in \mathcal{O}_{n_k r_{k-1}}(\mathbb{R})$ such that

$$\begin{bmatrix} \Gamma_k^{(t)}[1] \\ \vdots \\ \Gamma_k^{(t)}[n_k] \end{bmatrix} = U_k(t) \begin{bmatrix} A_k[1] \\ \vdots \\ A_k[n_k] \end{bmatrix}.$$

This implies that $\begin{bmatrix} \dot\Gamma_k^{(0)}[1] \\ \vdots \\ \dot\Gamma_k^{(0)}[n_k] \end{bmatrix} = S_k \begin{bmatrix} A_k[1] \\ \vdots \\ A_k[n_k] \end{bmatrix}$ for some antisymmetric matrix $S_k \in \mathbb{R}^{n_k r_{k-1}\times n_k r_{k-1}}$.

Let

$$\begin{bmatrix} W_k[1] \\ \vdots \\ W_k[n_k] \end{bmatrix} = S_k \begin{bmatrix} A_k[1] \\ \vdots \\ A_k[n_k] \end{bmatrix}.$$

Then

$$\sum_{i_k=1}^{n_k} A_k[i_k]^\mathsf{T} W_k[i_k] = \begin{bmatrix} A_k[1]^\mathsf{T} & \dots & A_k[n_k]^\mathsf{T} \end{bmatrix} S_k \begin{bmatrix} A_k[1] \\ \vdots \\ A_k[n_k] \end{bmatrix},$$

which is a symmetric and an antisymmetric matrix, hence it is zero.

The tangent vectors are hence necessarily of the form given by eq. (2.3.2)-(2.3.4). Dimension counting and invoking Proposition 2.3.1 show the uniqueness of the representation. □

TODO : adapt the proof of [LOV15] p. 922

Chapter 3

Reduced density matrix, block entropy and tensor trains

3.1 Reduced density matrix and the quantum entropy

The main idea of the Fiedler order is to minimise a proxy of the decay of the singular values, which is the block entropy. There are several possible choices of the entropy that can be minimised. In the litterature, the two main choices have been the von Neumann entropy and the Rényi entropy. We will first shortly review both concepts of entropy. A good reference on this topic is Carlen's notes [CL14] and for other discussions on matrix identities and inequalities, refer to Tropp's book [T+15].

3.1.1 Reduced density matrix

For a given normalised tensor $\Psi \in \mathbb{C}^{2^d}$, we define the *k-orbital reduced density matrix* (k-RDM) $\rho_{1:k} \in \mathbb{C}^{2^k \times 2^k}$ the matrix

$$\left(\rho_{1:k}\right)_{\mu_1\cdots\mu_k}^{\nu_1\cdots\nu_k} = \sum_{\mu_{k+1}\cdots\mu_d} \left(\Psi_{\mu_1\cdots\mu_k}^{\mu_{k+1}\cdots\mu_d}\right)^* \Psi_{\mu_{k+1}\cdots\mu_d}^{\nu_1\cdots\nu_k}. \tag{3.1.1}$$

Note that the eigenvalues of the k-RDM are squares of the singular values of the reshaped tensor $(\Psi_{\mu_1\cdots\mu_k}^{\mu_{k+1}\cdots\mu_d})$ which monitor the approximability of Ψ by TT.

More generally, for a subset $A \subset \{1, \dots, d\}$, we define the RDM $\rho_A \in \mathbb{C}^{2^{|A|} \times 2^{|A|}}$ by

$$\left(\rho_A\right)_{\boldsymbol{\mu}_i, i\in A}^{\boldsymbol{\nu}_i, i\in A} = \sum_{\boldsymbol{\mu}_j, j\notin A} \left(\Psi_{\boldsymbol{\mu}_i}^{\boldsymbol{\mu}_j}\right)^* \Psi_{\boldsymbol{\mu}_j}^{\boldsymbol{\nu}_i}. \tag{3.1.2}$$

Note that since Ψ is normalised, we have

$$\operatorname{Tr} \rho_A = 1. \tag{3.1.3}$$

By definition, RDM are Hermitian and semi-positive definite matrices.

3.1.2 Quantum entropy

Definition 3.1.1 (Von Neumann and Rényi entropies)**.** *Let* $\rho \in \mathbb{C}^{n\times n}$ *be a Hermitian, semi-positive definite matrix such that* $\operatorname{Tr}\rho = 1$*. The von Neumann entropy is defined by*

$$S(\rho) = -\operatorname{Tr}(\rho \log \rho). \tag{3.1.4}$$

The Rényi entropy of parameter $\alpha \in (0,\infty)$*,* $\alpha \neq 1$ *is defined by*

$$S_\alpha(\rho) = \frac{1}{1-\alpha}\log(\operatorname{Tr}(\rho^\alpha)). \tag{3.1.5}$$

For pure states, *i.e.* when ρ is up to a scalar factor a projector, one can check that the entropy of ρ is 0. As pure states can be written as TT of TT rank 1, this motivates the further investigation of the entropy as a proxy for the approximability of Ψ by TT. This suggests that states with a low quantum entropy are easily approximable by TT. As we are going to highlight, although in practice the quantum entropy is a fair indicator for the approximation problem, there are counter-examples of states that have an exponentially complex TT representation but a low quantum entropy [SWVC08].

Another desirable property of the entropy is the additivity - also sometimes called the extensivity - and the subadditivity:

- additivity: we say that an entropy S is additive if for all RDM ρ_A and ρ_B we have

$$S(\rho_A \otimes \rho_B) = S(\rho_A) + S(\rho_B); \tag{3.1.6}$$

- subadditivity: we say that an entropy S is subadditive if for all RDM ρ_{AB} defined on a tensor space $\mathcal{H}_A \otimes \mathcal{H}_B$, $\rho_A = \operatorname{Tr}_{\mathcal{H}_B}\rho_{AB}$ and $\rho_B = \operatorname{Tr}_{\mathcal{H}_A}\rho_{AB}$, we have

$$S(\rho_{AB}) \leq S(\rho_A) + S(\rho_B). \tag{3.1.7}$$

We are going to review few important properties of the von Neumann and Rényi entropies.

Remark 3.1.2. *The Rényi and von Neumann entropies are closely related as*

$$\lim_{\alpha\to 1} S_\alpha(\rho) = S(\rho). \tag{3.1.8}$$

Proposition 3.1.3 (Schur concavity)**.** *The von Neumann and the Rényi entropies are Schur concave,* i.e. *if* ρ_α *and* ρ_β *are RDM with respective eigenvalues* $(\alpha_i)_{1\leq i\leq n}$ *and* $(\beta_i)_{1\leq i\leq n}$ *such that for all* $1 \leq k \leq n$

$$\sum_{i=1}^k \alpha_i \leq \sum_{i=1}^k \beta_i, \tag{3.1.9}$$

then $S(\rho_\alpha) \geq S(\rho_\beta)$*.*

From the Schur concavity, we deduce that

- a pure state ρ with eigenvalues $(1, 0, \dots, 0)$ majorises any sequence, hence for any $\tilde{\rho}$, $S(\tilde{\rho}) \geq S(\rho) = 0$;
- a state with maximal entanglement *i.e.* with eigenvalues $(\frac{1}{n}, \dots, \frac{1}{n})$ is majorised by any sequence, hence it is the state with maximal entropy.

Proof. We simply need to use the concavity of the map $g : x \mapsto -x \log x$ or $g : x \mapsto \frac{x^\alpha}{1-\alpha}$. Let us prove the result for $n = 2$. By assumption on the eigenvalues of ρ_1 and ρ_2, we have

$$\begin{cases} \alpha_1 \geq \beta_1 \\ \alpha_1 + \alpha_2 = \beta_1 + \beta_2, \end{cases} \tag{3.1.10}$$

hence there exists $\lambda \in [0, 1]$ such that

$$\begin{cases} \beta_1 = \lambda\alpha_1 + (1-\lambda)\alpha_2 \\ \beta_2 = (1-\lambda)\alpha_1 + \lambda\alpha_2. \end{cases} \tag{3.1.11}$$

We have then

$$\begin{aligned} S(\rho_\beta) &= g(\beta_1) + g(\beta_2) \\ &= g(\lambda\alpha_1 + (1-\lambda)\alpha_2) + g((1-\lambda)\alpha_1 + \lambda\alpha_2) \\ &\geq \lambda g(\alpha_1) + (1-\lambda)g(\alpha_2) + (1-\lambda)g(\alpha_1) + \lambda g(\alpha_2) = g(\alpha_1) + g(\alpha_2) = S(\rho_\alpha). \end{aligned}$$

□

It turns out that additivity holds for the von Neumann and the Rényi entropies but subadditivity - and strong subadditivity that is introduced further down - only holds for the von Neumann entropy [LMW13].

Proposition 3.1.4 (Additivity of the Rényi and von Neumann entropies)**.** *The von Neumann and the Rényi entropies are additive,* i.e. *for all RDM ρ_A, ρ_B respectively defined on $\mathcal{H}_A$ and $\mathcal{H}_B$, then we have*

$$S(\rho_A \otimes \rho_B) = S(\rho_A) + S(\rho_B). \tag{3.1.12}$$

Proof. The proof follows from a direct calculation of $S(\rho_A \otimes \rho_B)$. Let (λ_i) and (μ_j) be the eigenvalues of ρ_A and ρ_B, then we have

$$\begin{aligned} S_\alpha(\rho_A \otimes \rho_B) &= \frac{1}{1-\alpha} \log\left(\mathrm{Tr}(\rho^\alpha)\right) \\ &= \frac{1}{1-\alpha} \log\left(\sum_{ij} \lambda_i^\alpha \mu_j^\alpha\right) \\ &= \frac{1}{1-\alpha} \log\left(\sum_i \lambda_i^\alpha \sum_j \mu_j^\alpha\right) \\ &= S_\alpha(\rho_A) + S_\alpha(\rho_B). \end{aligned}$$

The proof is the same with the von Neumann entropy. □

Proposition 3.1.5 (Subadditivity of the von Neumann entropy). *The von Neumann entropy is subadditive,* i.e.*for all RDM ρ_{AB} defined on a tensor space $\mathcal{H}_A \otimes \mathcal{H}_B$, $\rho_A = \operatorname{Tr}_{\mathcal{H}_B} \rho_{AB}$ and $\rho_B = \operatorname{Tr}_{\mathcal{H}_A} \rho_{AB}$, we have*

$$S(\rho_{AB}) \leq S(\rho_A) + S(\rho_B). \tag{3.1.13}$$

Before giving the proof, we state the Klein's inequality.

Lemma 3.1.6 (Klein's inequality). *Let f be a convex function, A and B be Hermitian matrices such that $f(A)$ and $f(B)$ are well-defined. Then the following inequality holds*

$$\operatorname{Tr}(f(A) - f(B) - (A - B)f'(B)) \geq 0. \tag{3.1.14}$$

If f is strictly convex, we have equality if and only if $A = B$.

Proof. We first write the spectral decomposition of A and B

$$\begin{cases} A = \displaystyle\sum_{i=1}^{n} \alpha_i |a_i\rangle\langle a_i| \\ B = \displaystyle\sum_{i=1}^{n} \beta_i |b_i\rangle\langle b_i|. \end{cases} \tag{3.1.15}$$

Then we have

$$\begin{aligned} \operatorname{Tr}(f(A) - f(B) - (A - B)f'(B)) &= \sum_{i=1}^{n} f(\alpha_i) - f(\beta_i) - \sum_{j=1}^{n} |\langle a_i, \beta_j\rangle|^2 \alpha_i f'(\beta_j) \\ &= \sum_{i,j=1}^{n} |\langle a_i, \beta_j\rangle|^2 \big(f(\alpha_i) - f(\beta_j) + (\beta_j - \alpha_i) f'(\beta_j)\big), \end{aligned}$$

where we have used that $\sum_{i=1}^{n} |\langle a_i, \beta_j\rangle|^2 = \sum_{j=1}^{n} |\langle a_i, \beta_j\rangle|^2 = 1$. We conclude using the convexity of f.

The equality case follows from the strict convexity and the properties of the scalar product. □

We can now prove the subadditivity of the von Neumann entropy.

Proof of Proposition 3.1.5. By additivity, we have

$$\begin{aligned} S(\rho_A) + S(\rho_B) - S(\rho_{AB}) &= \operatorname{Tr}(\rho_{AB} \log(\rho_{AB})) - \operatorname{Tr}(\rho_A \otimes \mathrm{id}_B \log(\rho_A \otimes \mathrm{id}_B)) \\ &\quad - \operatorname{Tr}(\mathrm{id}_A \otimes \rho_B \log(\mathrm{id}_A \otimes \rho_B)) \\ &= \operatorname{Tr}(\rho_{AB}(\log(\rho_{AB}) - \log(\rho_A \otimes \mathrm{id}_B) - \log(\mathrm{id}_A \otimes \rho_B))) \\ &= \operatorname{Tr}(\rho_{AB}(\log(\rho_{AB}) - \log(\rho_A \otimes \mathrm{id}_B + \mathrm{id}_A \otimes \rho_B))) \end{aligned}$$

where we have used that $\operatorname{Tr}(\rho_{AB}\log(\rho_A\otimes \mathrm{id}_B)) = \operatorname{Tr}(\rho_A\otimes \mathrm{id}_B \log(\rho_A\otimes \mathrm{id}_B))$. It remains to show that for positive semi-definite matrices M, N such that $\operatorname{Tr} M = \operatorname{Tr} N$, we have

$$\operatorname{Tr}(M(\log(M) - \log(N))) \geq 0. \tag{3.1.16}$$

The map $x \mapsto x\log x$ is convex hence by Klein's inequality 3.1.6, we have

$$\operatorname{Tr}(M\log(M) - N\log N) \geq \operatorname{Tr}(M - N + (M-N)\log(N)) \geq \operatorname{Tr}((M-N)\log(N)),$$

by simplifying $\operatorname{Tr}(N\log N)$ on both sides, we get (3.1.16) and this finishes the proof. □

Remark 3.1.7. *Note that we have proved that the relative entropy - also called the Kullback-Leibler divergence -*

$$d(\rho_1,\rho_2) = \operatorname{Tr}(\rho_1(\log(\rho_1) - \log(\rho_2))), \tag{3.1.17}$$

is always non negative for RDM.

The von Neumann has an additional property that is useful to have better bounds of the entropy of a larger blocks. This property is the *strong subadditivity*.

Proposition 3.1.8 (Strong subadditivity of the von Neumann entropy)**.** *The von Neumann entropy is strongly subadditive,* i.e. *for all RDM ρ_{ABC} on the tensor space $\mathcal{H}_A\otimes\mathcal{H}_B\otimes\mathcal{H}_C$, ρ_{AB},ρ_{BC} and ρ_B the corresponding partial traces, we have*

$$S(\rho_{ABC}) \leq S(\rho_{AB}) + S(\rho_{BC}) - S(\rho_B). \tag{3.1.18}$$

The proof of the strong subadditivity follows the line of the subadditivity of the von Neumann entropy [AL70].

A summary of the properties satisfied by the Von Neumann and the Rényi entropies is provided in Table 3.1.

	Additivity	Schur concavity	Subadditivity	Strong subadditivity
Von Neumann entropy	YES	YES	YES	YES
Rényi entropy	YES	YES	NO	NO

Table 3.1: Properties of the Von Neumann and Rényi entropies

3.1.3 Relationship between TT approximability and entropy scaling

For the Rényi entropy, for $\alpha < 1$ a bound on the entropy is sufficient to bound the TT error.

Proposition 3.1.9 ([VC06])**.** *Let $\Psi \in \mathbb{C}^{2^d}$ be a normalised state and $\epsilon_j(r)$ the L^2 local error by its TT approximation of TT rank r. Then the error is bounded by*

$$\log_2(\epsilon_j(r)^2) \leq \frac{1-\alpha}{\alpha}(S_\alpha(\rho_{1:j}) - \log_2(\tfrac{r}{1-\alpha})). \tag{3.1.19}$$

These results are tied to the concept of *area laws* that states that the entropy is bounded regarless of the system size d. This is proved for 1D Hamiltonian with nearest neighbour interaction [Has07] under the assumption that the system is gapped. Indeed, if the entropy $S_\alpha(\rho_{1:j})$ for $\alpha < 1$ is bounded independently of j and d, then the local error is polynomial in the truncation rank r.

Numerous extensions of this result have been shown to include interactions with longer interactions. For higher dimensions, area laws have not been proved although it is generally accepted that the same statements should remain valid, in which case, the best tensor network to approximate the ground-state of such systems will not be TT but projected entangled pair states (PEPS) [VC04].

Proof. We use the Schur convexity of the entropy and find a sequence that majorises the singular values of Ψ. □

For $\alpha > 1$, the Rényi entropy provides a lower bound of the TT error. This gives a sufficient condition for nonapproximability results.

Proposition 3.1.10 ([SWVC08]). *Let $\Psi \in \mathbb{C}^{2^d}$ be a normalised state and $\epsilon_j(r)$ the L^2 local error by its TT approximation of TT rank r. Then the error is bounded by*

$$S_\alpha(\rho_{1:j}) \geq \frac{1-\alpha}{\alpha} \log_2(1 - \epsilon_j(r)^2) + \log_2(r). \tag{3.1.20}$$

Proof. We use the Schur convexity of the entropy and find a sequence that majorises the singular values of Ψ. □

For the von Neumann entropy, we have the same result in the nonapproximable case.

However, counterexamples of states exist where the von Neumann entropy is bounded but the state is not approximable by TT.

Proposition 3.1.11 ([SWVC08]). *Let $\Psi_{2N} \in \mathbb{C}^{3^{2N}}$ be the state defined by*

$$\Psi_{2N} = \sqrt{1-p_N}|2\rangle^{\otimes 2N} + \sqrt{\tfrac{p_N}{2^N}} \sum_{x \in \{0,1\}^N} |x\rangle \otimes |x\rangle, \tag{3.1.21}$$

with p_N and

$$|\chi_M\rangle = |\Psi_{2N}\rangle^{\otimes N^2}. \tag{3.1.22}$$

Proof. For the TT truncation error, for $r < 2^N$ the truncation error on Ψ_{2N} is bounded by

$$\|\Psi_{2N} - \mathrm{TT}_r \Psi_{2N}\| \leq p_N, \tag{3.1.23}$$

thus the truncation error on χ_M is bounded by

$$\|\chi_M - \mathrm{TT}_r \chi_M\| \leq N^2 \|\Psi_{2N} - \mathrm{TT}_r \Psi_{2N}\| \leq N^2 p_N. \tag{3.1.24}$$

For the entropy, let $L \leq 2N^3$ then the d-orbital RDM of χ_M is bounded by the ℓ-orbital RDM of Ψ_{2N} where $\ell = L \mod 2N$

$$\begin{aligned} S(\rho_d(|\chi_M\rangle)) &\leq S(\rho_\ell(\Psi_{2N})) \\ &\leq S(\rho_{2N}^{(\ell)}) + p_N, \end{aligned}$$

with $\rho_{2N}^{\ell} = (1-p_N)|2\rangle^{\otimes \ell}\langle 2|^{\otimes \ell} + \frac{p_N}{2^\ell}\sum_{x\in\{0,1\}^\ell}$. The von Neumann entropy of $\rho_{2N}^{(\ell)}$ is explicit

$$\begin{aligned} S(\rho_{2N}^{(\ell)}) &= -(1-p_N)\log(1-p_N) - p_N \log\left(\frac{p_N}{2^\ell}\right) \\ &= -(1-p_N)\log(1-p_N) - p_N\log(p_N) + \ell p_N. \end{aligned} \tag{3.1.25}$$

We want the state to have a bounded entropy and not being approximable by a TT, hence we need to pick p_N such that

- ℓp_N bounded ;
- $N^2 p_N \to \infty$

thus $p_N = \mathcal{O}(\frac{1}{N^\beta})$, with $1 \leq \beta < 1$ will do. □

The different results regarding the boundedness of the entropy and the TT approximability are gathered in Table 3.2.

	Bounded entropy	Unbounded entropy
Rényi $\alpha < 1$	TT approximable	?
von Neumann	?	Non TT approximable
Rényi $\alpha > 1$	?	Non TT approximable

Table 3.2: TT approximability and entropy. Question marks mean that both can happen.

Although this counterexample seems to indicate that the von Neumann entropy is not suited to assess the TT approximability of a state, results that are obtained using the Fiedler order are generally satisfactory.

A heuristic argument relying on the strong subadditivity of the von Neumann entropy suggests that the Fiedler order is a first order optimiser of the block entropy. However as shown by the counterexample, it is not sufficient to argue that the state is easier to approximate by a TT. This suggests that such counterexamples are scarce.

3.2 Mutual information and the Fiedler order

The Fiedler order has been introduced in [LRH03] inspired by the notions of quantum entanglement in quantum information theory. It is the default ordering method in the major DMRG codes [LRH03, CHG02, RNW06, FKK+18].

For a quantum state $\Psi \in \mathbb{C}^{2^d}$, we introduce the *quantum mutual information matrix* (QMI) $I_{ij} \in \mathbb{R}^{d \times L}$ by

$$I_{ij} = \begin{cases} S(\rho_i) + S(\rho_j) - S(\rho_{i,j}), i \neq j \\ 0, i = j \end{cases} \tag{3.2.1}$$

The QMI is exactly the Kullback-Leibler divergence between ρ_{ij} and $\rho_i \otimes \rho_j$. As such, by Equation (3.1.17), the QMI has nonnegative entries.

The motivation of the Fiedler order scheme relies on the following proposition.

Proposition 3.2.1 ([Ali21]). *Let $\Psi \in \mathbb{C}^{2^d}$ be a normalised state. Let $1 \leq j \leq L-1$ and $1 \leq \delta \leq \frac{j}{2}$. Then we have*

$$S(\rho_{1:j}) \leq \sum_{k=1}^{j} S(\rho_k) - I_{k,k+\delta}. \tag{3.2.2}$$

To minimise the block entropy $S(\rho_{1:j})$, assuming that the one-site entropies are all of the same order, it is reasonable to maximise the QMI of neighbouring sites. This is the main idea of the Fiedler order. Before sketching the algorithm, we will prove the previous proposition.

Proof. By the strong subadditivity of the von Neumann entropy 3.1.8, for all $1 \leq k \leq j-\delta$ we have

$$S(\rho_{k:j}) + S(\rho_{k+\delta}) \leq S(\rho_{k,k+\delta}) + S(\rho_{k+1:j}). \tag{3.2.3}$$

Summing these equations, we obtain

$$\begin{aligned} S(\rho_{1:j}) + \sum_{k=1}^{j-\delta} S(\rho_{k+\delta}) &\leq \sum_{k=1}^{j-\delta} S(\rho_{k,k+\delta}) + S(\rho_{j-\delta+1:j}) \\ S(\rho_{1:j}) &\leq \sum_{k=1}^{j-\delta} S(\rho_k) - \sum_{k=1}^{j-\delta} I_{k,k+\delta} + S(\rho_{j-\delta+1:j}) \\ &\leq \sum_{k=1}^{j} S(\rho_k) - I_{k,k+\delta}, \end{aligned}$$

where we have used the definition of the QMI and the additivity of the entropy. □

For the Fiedler order [LRH03, BLMR11], the function that is minimised is the total entanglement

$$I_{\text{dist}}(\pi) = \sum_{i,j=1}^{d} I_{i,j} |\pi(i) - \pi(j)|^2, \tag{3.2.4}$$

over the set of permutations $\pi \in \mathcal{P}_N$.

As it is a combinatorial problem, it is necessary to resort to an approximation in practice. In that case, the problem that is solved is the minimisation of

$$\tilde{I}_{\text{dist}}(x) = \sum_{i,j=1}^{d} I_{i,j}|x_i - x_j|^2, \tag{3.2.5}$$

with $x \in \mathbb{R}^d$ under the constraint that $\sum_i x_i = 0$ and $\|x\|_2 = 1$. Introducing the graph Laplacian $L_{ij} = D_{ij} - I_{ij}$ where D is the diagonal matrix with diagonal entries $D_{ii} = \sum_{j=1}^{d} I_{ij}$, we see that

$$x^T L x = \sum_{i,j=1}^{d} I_{i,j}|x_i - x_j|^2. \tag{3.2.6}$$

The solution to this minimisation problem under the constraint that $\sum_i x_i = 0$ and $\|x\|_2 = 1$ is given by the second eigenvector of d (the lowest eigenvalue is 0 by construction of d), which is called the *Fiedler vector* . The Fiedler order consists in ordering the sites according to the magnitude of the entries of the Fiedler vector.

Indeed the Fiedler vector is related to the problem of *graph partitioning* or in our case the *min-cut* of the graph. The QMI matrix can be seen as a weight on the graph with d vertices, for which we need to determine a partition of the vertices into two distinct sets A and $\{1, \dots, L\} \setminus A$ such that it minimises

$$\sum_{i \in A, j \notin A} I_{ij}. \tag{3.2.7}$$

For simple cases, it can be proved that the Fiedler vector solves this problem by considering $A = \{i \mid x_i > 0\}$. It is generally believed that for weighted graphs, the Fiedler vector is a good approximation to the min-cut problem.

3.3 An example: minimal-basis H_2

To illustrate the different ordering methods, we now apply them to the minimal-basis H_2 wave-function

$$\Psi = \Big|(c\varphi_A + s\varphi_B) \uparrow, \, (c'\varphi_A + s'\varphi_B) \downarrow \Big\rangle, \tag{3.3.1}$$

where φ_A and φ_B are respectively the bonding and antibonding orbitals. In the occupation representation, the state Ψ has the following form

$$\Psi = cc'\Phi_{(1100)} + cs'\Phi_{(1001)} - sc'\Phi_{(0110)} + ss'\Phi_{(0011)} \in \bigotimes_{i=1}^{4} \mathbb{C}^2. \tag{3.3.2}$$

We will compute the singular values of the matrix reshape $\Psi^{\mu_1\mu_2}_{\mu_3\mu_4} \in \mathbb{R}^{2^2 \times 2^2}$, for the different orderings of the basis set delivered by all the above ordering schemes. To avoid degenerate

cases we assume that all coefficients c, s, c', s' in (3.3.1) are nonzero.
Canonical order. We abbreviate the single-particle basis states as $\{A\uparrow, A\downarrow, B\uparrow, B\downarrow\}$. Directly from (3.3.2) we see that with respect to the canonical order in which the bonding orbital with either spin comes first,

$$A\uparrow\ A\downarrow\ B\uparrow\ B\downarrow, \tag{3.3.3}$$

the reshape $\Psi^{\mu_1\mu_2}_{\mu_3\mu_4}$ is

$\mu_1\mu_2 \backslash \mu_3\mu_4$	00	01	10	11
00				ss'
01			$-sc$	
10		cs'		
11	cc'			

The singular values are

$$(cc')^2,\ (cs')^2,\ (sc')^2,\ (ss')^2$$

and the rank of the matrix reshape is 4.

Fiedler order. We begin by working out the one- and two-orbital density matrices and the corresponding entropies. The one-orbital quantities are elementary to compute, they are

$$\rho^{(1)}_{A\uparrow} = \begin{bmatrix} s^2 & 0 \\ 0 & c^2 \end{bmatrix},\quad \rho^{(1)}_{A\downarrow} = \begin{bmatrix} s'^2 & 0 \\ 0 & c'^2 \end{bmatrix},\quad \rho^{(1)}_{B\uparrow} = \begin{bmatrix} c^2 & 0 \\ 0 & s^2 \end{bmatrix},\quad \rho^{(1)}_{B\uparrow} = \begin{bmatrix} c'^2 & 0 \\ 0 & s'^2 \end{bmatrix}.$$

It follows that

$$\begin{aligned} s^{(1)}_{A\uparrow} = s^{(1)}_{B\uparrow} &= -c^2\log c^2 - s^2 \log s^2 =: s_\uparrow \in (0,1], \\ s^{(1)}_{A\downarrow} = s^{(1)}_{B\downarrow} &= -c'^2\log c'^2 - s'^2 \log s'^2 =: s_\downarrow \in (0,1]. \end{aligned}$$

As regards the two-orbital density matrices, we find after some calculation that

$$\rho^{(2)}_{A\uparrow A\downarrow} = \begin{bmatrix} s^2s'^2 & & & \\ & s^2c'^2 & & \\ & & c^2s'^2 & \\ & & & c^2c'^2 \end{bmatrix},\ \rho^{(2)}_{B\uparrow B\downarrow} = \begin{bmatrix} c^2c'^2 & & & \\ & c^2s'^2 & & \\ & & s^2c'^2 & \\ & & & s^2s'^2 \end{bmatrix},\ \rho^{(2)}_{A\uparrow B\downarrow} = \begin{bmatrix} s^2c'^2 & & & \\ & s^2s'^2 & & \\ & & c^2c'^2 & \\ & & & c^2s'^2 \end{bmatrix},\ \rho^{(2)}_{A\downarrow B\uparrow} = \begin{bmatrix} c^2c'^2 & & & \\ & c^2s'^2 & & \\ & & s^2s'^2 & \\ & & & s^2c'^2 \end{bmatrix}.$$

It follows that $S^{(2)} = -\mathrm{tr}\,\rho^{(2)}\log\rho^{(2)} =: S_{\uparrow\downarrow}$ is the same for all four matrices. Moreover writing out the above trace and using $c^2+s^2 = c'^2+s'^2 = 1$ we find that

$$S_{\uparrow\downarrow} = s_\uparrow + s_\downarrow. \tag{3.3.4}$$

The two remaining two-orbital RDMs contain off-diagonal terms

$$\rho^{(2)}_{A\uparrow B\uparrow} = \begin{bmatrix} 0 & & & \\ & s^2 & -cs(c'^2-s'^2) & \\ & -cs(c'^2-s'^2) & c^2 & \\ & & & 0 \end{bmatrix},\quad \rho^{(2)}_{A\downarrow B\downarrow} = \begin{bmatrix} 0 & & & \\ & s'^2 & c's'(c^2-s^2) & \\ & c's'(c^2-s^2) & c'^2 & \\ & & & 0 \end{bmatrix}.$$

We denote the associated entropies by $S^{(2)}_{A\uparrow B\uparrow} =: S_{\uparrow\uparrow}$, $S^{(2)}_{A\downarrow B\downarrow} =: S_{\downarrow\downarrow}$. The mutual information matrix and graph Laplacian are thus, using the vanishing of all nearest-neighbour elements of I by (3.3.4) and denoting $a := 2s_\uparrow - S_{\uparrow\uparrow}$, $b := 2s_\downarrow - S_{\downarrow\downarrow}$,

$I =$

	$A\uparrow$	$A\downarrow$	$B\uparrow$	$B\downarrow$
$A\uparrow$	0	0	a	0
$A\downarrow$	0	0	0	b
$B\uparrow$	a	0	0	0
$B\downarrow$	0	b	0	0

$L =$

	$A\uparrow$	$A\downarrow$	$B\uparrow$	$B\downarrow$
$A\uparrow$	a	0	-a	0
$A\downarrow$	0	b	0	-b
$B\uparrow$	-a	0	a	0
$B\downarrow$	0	-b	0	b

To determine the Fiedler ordering we need to find the second eigenvector of the graph Laplacian, alias Fiedler vector. The first eigenvector is always, by construction, the constant vector, with eigenvalue 0. For the above L, by inspection the remaining eigenvalues are 0, $2a > 0$, $2b > 0$, with eigenvectors $(1, -1, 1, -1)$, $(1, 0, -1, 0)$, $(0, 1, 0, -1)$. The second eigenvector is thus $(1, -1, 1, -1)$. It follows that the Fiedler ordering is

$$A\uparrow \; B\uparrow \; A\downarrow \; B\downarrow \tag{3.3.5}$$

(up to re-ordering the orbitals in the left block, re-ordering the orbitals in the right block, and flipping the two blocks; none of this affects the singular values). The matrix reshape $\Psi^{\mu_1\mu_2}_{\mu_3\mu_4}$ with respect to this ordering is

$\mu_1\,\mu_2$ \ $\mu_3\,\mu_4$	00	01	10	11
00	0			
01		ss'	sc'	
10		cs'	cc'	
11				0

Since the middle block is the rank-1 matrix $\begin{bmatrix} c \\ s \end{bmatrix} \begin{bmatrix} c' & s' \end{bmatrix}$, the singular values are

$$1, \, 0, \, 0, \, 0$$

and the rank of the matrix reshape is 1. We see that the Fiedler order has dramatically improved the decay of the singular values.

Chapter 4

Area laws for one-dimensional systems

Area laws have first been stated rigorously for ground-state of one-dimensional gapped systems with nearest neighbour interactions (NNI) by Hastings [Has07]. Later on, another proof using approximate ground-state projector has been discovered yielding better bounds [AKLV13]. For both proofs, the goal is to bound the Rényi entropy of the RDM $\rho_{j:j+\ell-1}$ by a constant S independent of ℓ and of the size of the system

$$S_\alpha(\rho_{j:j+\ell-1}) \leq S. \tag{4.0.1}$$

By Proposition 3.1.9, this implies that there is a TT approximation of the ground-state with TT ranks bounded by $2^{\frac{1-\alpha}{\alpha}S}$.

4.1 Hastings area law

4.1.1 Hamiltonian with nearest neighbour interactions

The NNI Hamiltonian considered is of the form

$$\boldsymbol{\mathcal{H}}^{(d)} = \sum_{j=1}^{d-1} \boldsymbol{W}_j, \tag{4.1.1}$$

where $\boldsymbol{\mathcal{H}}^{(d)}$ is an operator acting on $\bigotimes_{j=1}^d \mathbb{R}^n$ and $\boldsymbol{W}_j$ is a two-body operator of the form $\mathrm{id}_{[\![j-1]\!]} \otimes W \otimes \mathrm{id}_{[\![j+2,d]\!]}$.

Assumption 4.1.1. *We assume that for each d, the many-body Hamiltonian $\boldsymbol{\mathcal{H}}^{(d)}$ has a unique ground-state $\boldsymbol{\Psi}_0^{(d)}$ with eigenvalue 0 and a spectral gap $\gamma > 0$ independent of d.*

If the gap closes not too fast, it is possible to still get a polynomial bound on the TT approximation of the ground-state instead of an exponential one.

Remark 4.1.2. *Hastings' proof also holds if we relax the form of the two-body operators* $\boldsymbol{W}_j$ *to be such that* $\mathrm{id}_{[\![j-1]\!]} \otimes W_j \otimes \mathrm{id}_{[\![j+2,d]\!]}$ *with* W_j *acting on* $\mathbb{R}^n \otimes \mathbb{R}^n$*. In that case, if the operators* $\boldsymbol{W}_j$ *satisfy the following conditions*

- *the operators* W_j *are uniformly bounded, i.e. there is a constant* C *such that for all* $j, \|W_j\| \leq C$*;*
- *the commutators are uniformly bounded, i.e. there is a constant* J *such that for all* j*,* $\|[W_j, W_{j+1}]\| \leq J$*.*

The first assumption can actually be lifted and is taken for simplicity. As long as the commutators $[\widetilde{h}_j, \widetilde{h}_{j+1}]$ *are uniformly bounded, the proof can be adapted to unbounded operators (see [Ali21]).*

4.1.2 Lieb-Robinson bounds

An essential ingredient of the area law by Hastings is the repeated use of the Lieb-Robinson bound for NNI Hamiltonians. This bound describes how the correlation evolves for local operators.

Proposition 4.1.3 (Lieb-Robinson bound [NS06])**.** *Let* $\boldsymbol{A} \in \mathscr{L}(\bigotimes_{i\in I} \mathbb{R}^n)$ *and* $\boldsymbol{B} \in \mathscr{L}(\bigotimes_{j\in J} \mathbb{R}^n)$ *be two operators with* $I \cap J = \varnothing$*. Let* $\boldsymbol{A}(t) = e^{i\boldsymbol{\mathcal{H}}^{(d)}t} \boldsymbol{A} \otimes \mathrm{id}_{X^c} e^{-i\boldsymbol{\mathcal{H}}^{(d)}t}$ *with* $\boldsymbol{\mathcal{H}}^{(d)}$ *given by* (4.1.1)*. Then there are constants* $c, a, v > 0$ *independent of* $\boldsymbol{A}$*,* $\boldsymbol{B}$ *or* d *such that*

$$\|[\boldsymbol{A}(t), \mathrm{id}_I \otimes \boldsymbol{B}]\| \leq c|I||J|\|\boldsymbol{A}\|\|\boldsymbol{B}\| \exp(-a(d(I,J) - v|t|)), \tag{4.1.2}$$

where $d(I,J) = \min_{i\in I, j\in J} |i-j|$*.*

The Lieb-Robinson bound is stated here in the special case of a one-dimensional NNI Hamiltonian but it holds for more general local interactions types [NS06]. In that case, the distance d is replaced by the natural distance of the interaction picture.

The Lieb-Robinson bound enables to state that the evolution of a local operator remains local by the next lemma.

Lemma 4.1.4. *Let* $\boldsymbol{A} \in \mathscr{L}(\mathrm{X} \otimes \mathrm{Y})$*. We assume that* Y *is finite-dimensional. Suppose there is* $\varepsilon > 0$ *such that for all* $\boldsymbol{B} \in \mathscr{L}(\mathrm{Y})$*, we have*

$$\|[\boldsymbol{A}, \mathrm{id}_{\mathrm{X}} \otimes \boldsymbol{B}]\| \leq \varepsilon \|\boldsymbol{B}\|. \tag{4.1.3}$$

Then there is an operator $\boldsymbol{A}_1 \in \mathscr{L}(\mathrm{X})$ *such that*

$$\|\boldsymbol{A} - \boldsymbol{A}_1 \otimes \mathrm{id}_{\mathrm{Y}}\| \leq \varepsilon. \tag{4.1.4}$$

Moreover, if $\boldsymbol{A}$ *is self-adjoint, then* $\boldsymbol{A}_1$ *can also be chosen self-adjoint.*

Proof of Lemma 4.1.4. The operator $\boldsymbol{A}_1$ is explicitely constructed: take $\boldsymbol{A}_1 = \frac{1}{\dim \mathrm{Y}} \mathrm{Tr}_{\mathrm{Y}} \boldsymbol{A} = \int_{\mathscr{U}(\mathrm{Y})} \mathrm{id}_{\mathrm{X}} \otimes U^* \boldsymbol{A} \, \mathrm{id} \otimes U \, \mathrm{d}U$ where $\mathrm{d}U$ is the uniform Haar measure on the unitary matrices of Y. Then we have

$$\|\boldsymbol{A} - \boldsymbol{A}_1 \otimes \mathrm{id}_{\mathrm{Y}}\| = \left\| \int_{\mathscr{U}(\mathrm{Y})} \mathrm{id}_{\mathrm{X}} \otimes U^* [\boldsymbol{A}, \mathrm{id} \otimes U] \, \mathrm{d}U \right\| \leq \varepsilon.$$

□

Corollary 4.1.5. *Let $\boldsymbol{A} \in \mathscr{L}(\bigotimes_{i \in I} \mathbb{R}^n)$, $\ell > 0$ and $\widetilde{I} = \{\tilde{i} \mid \exists i \in I, |i - \tilde{i}| \leq \ell\}$. Let $\boldsymbol{A}(t) = e^{\mathrm{i}\boldsymbol{\mathcal{H}}^{(d)} t} \boldsymbol{A} \otimes \mathrm{id}_{I^c} e^{-\mathrm{i}\boldsymbol{\mathcal{H}}^{(d)} t}$ with $\boldsymbol{\mathcal{H}}^{(d)}$ given by (4.1.1). Then for all $t \in \mathbb{R}$, there is an operator $\boldsymbol{A}_\ell(t) \in \mathscr{L}(\bigotimes_{i \in \widetilde{I}} \mathbb{R}^n)$ such that*

$$\|\boldsymbol{A}(t) - \boldsymbol{A}_\ell(t) \otimes \mathrm{id}_{\widetilde{I}^c}\| \leq d|I| \|\boldsymbol{A}\| \exp(-a(\ell - v|t|)). \tag{4.1.5}$$

If $\boldsymbol{A}$ is self-adjoint, then $\boldsymbol{A}_\ell(t)$ is self-adjoint for all t.

Proof. Combining Lemma 4.1.4 with the Lieb-Robinson bonud (4.1.2), we directly get the result. □

4.1.3 Main theorem and Hastings area law

The main result in Hastings seminal paper states that the ground-state projector can be exponentially well approximated using an almost tensor product of operators with an overlapping domain of size ℓ independent of the size of the system.

Theorem 4.1.6. *Let $\boldsymbol{\mathcal{H}}^{(d)}$ be the Hamiltonian defined in (4.1.1) satisfying the assumptions 4.1.1. For any $1 \leq j \leq d$ and any $\ell \geq 0$, there are operators $O_L \in \mathscr{L}(\mathcal{H}_{1:j})$, $O_M \in \mathscr{L}(\mathcal{H}_{j-\ell:j+\ell})$ and $O_R \in \mathscr{L}(\mathcal{H}_{j+1:d})$ with $\|O_M\|, \|O_L\|, \|O_R\| \leq 1$ and there is $\beta > 0$ independent of ℓ and d and $C > 0$ depending polynomially on d such that*

$$\left\| (\mathrm{id}_{1:j-\ell-1} \otimes O_M \otimes \mathrm{id}_{j+\ell+1:d})(O_L \otimes \mathrm{id}_{j+1:d})(\mathrm{id}_{1:j} \otimes O_R) - |\boldsymbol{\Psi}_0^{(d)}\rangle\langle\boldsymbol{\Psi}_0^{(d)}| \right\| \leq C \exp(-\beta\ell). \tag{4.1.6}$$

From eq. (4.1.6), the area law and the TT approximation of the ground-state follows.

Corollary 4.1.7. *Let $\boldsymbol{\Psi}_0^{(d)}$ be the ground-state wave function of $\boldsymbol{\mathcal{H}}^{(d)}$ given by (4.1.1). Then the following assertions are true:*

(i). there is a constant S independent of L such that $S_\alpha(|\boldsymbol{\Psi}_0^{(d)}\rangle\langle\boldsymbol{\Psi}_0^{(d)}|) \leq S$;

(ii). for any $\varepsilon > 0$, there is a TT approximation $\mathrm{TT}_r \boldsymbol{\Psi}_0^{(d)}$ with TT rank r independent of d of $\boldsymbol{\Psi}_0^{(d)}$ such that

$$\|\mathrm{TT}_r \boldsymbol{\Psi}_0^{(d)} - \boldsymbol{\Psi}_0^{(d)}\| \leq \varepsilon.$$

Remark 4.1.8. *It is possible to choose the operators O_L, O_M and O_R to be nonnegative. By construction, O_L and O_R are nonnegative and by a little trick, O_M can also be chosen nonnegative [Has07].*

Sketch of an almost-proof of Theorem 4.1.6 The proof of the theorem relies on the following approximation of the ground-state projection

$$\rho_q = \frac{1}{\sqrt{2\pi q}} \int_{\mathbb{R}} e^{\mathrm{i}\boldsymbol{\mathcal{H}}^{(d)}t} e^{-\frac{t^2}{2q}} \,\mathrm{d}t, \tag{4.1.7}$$

where $q > 0$ is fixed later on. Using the spectral gap assumption, we see that

$$\|\rho_q - |\boldsymbol{\Psi}_0^{(d)}\rangle\langle\boldsymbol{\Psi}_0^{(d)}|\| \leq e^{-\frac{1}{2}\gamma^2 q}, \tag{4.1.8}$$

where γ is the spectral gap.

Using the NNI structure of the Hamiltonian, we can write

$$H = H_{L+R} + H_M,$$

with $H_M = \sum_{k=j-\frac{\ell}{2}}^{j+\frac{\ell}{2}} h_k$ and $H_{L+R} = \sum_{k<j-\frac{\ell}{2}} h_k + \sum_{k>j+\frac{\ell}{2}} h_k$. The evolution $e^{\mathrm{i}\boldsymbol{\mathcal{H}}^{(d)}t}$ can be written

$$e^{\mathrm{i}\boldsymbol{\mathcal{H}}^{(d)}t} = e^{\mathrm{i}H_{L+R}t + \mathrm{i}H_M t} e^{-\mathrm{i}H_{L+R}t} e^{\mathrm{i}H_{L+R}t}.$$

The trick is to realise that $e^{\mathrm{i}H_{L+R}t+\mathrm{i}H_M t}e^{-\mathrm{i}H_{L+R}t}$ is the solution to

$$\begin{cases} \mathrm{i}U'(t) = U(t)e^{\mathrm{i}H_{L+R}t} H_M e^{-\mathrm{i}H_{L+R}t} \\ U(0) = \mathrm{id}\,. \end{cases}$$

Since $H_M = \mathrm{id}_{1:j-\frac{\ell}{2}} \otimes \tilde{H}_M \otimes \mathrm{id}_{j+\frac{\ell}{2}+1:d}$, using Corollary 4.1.5, then for all $t \in \mathbb{R}$, there is $H_M^{(\ell)}(t) \in \mathscr{L}(\mathcal{H}_{j-\ell:j+\ell})$ such that

$$\left\| e^{\mathrm{i}H_{L+R}t} H_M e^{-\mathrm{i}H_{L+R}t} - \mathrm{id}_{1:j-\ell-1} \otimes H_M^{(\ell)}(t) \otimes \mathrm{id}_{j+\ell+1:d} \right\| \leq 2d\ell \|H_M\| \exp\bigl(-a(\tfrac{\ell}{2} - v|t|)\bigr).$$

Thus the operator $e^{\mathrm{i}H_{L+R}t+\mathrm{i}H_M t}e^{-\mathrm{i}H_{L+R}t}$ can be approximated by

$$e^{\mathrm{i}H_{L+R}t+\mathrm{i}H_M t}e^{-\mathrm{i}H_{L+R}t} = \mathcal{T}\exp\Bigl(\int_0^t \mathrm{id}_{1:j-\ell-1} \otimes H_M^{(\ell)}(\tau) \otimes \mathrm{id}_{j+\ell+1:d} \,\mathrm{d}\tau\Bigr)^*,$$

where for an operator $A(t)$, $\mathcal{T}\exp\bigl(\int_0^t A(\tau)\,\mathrm{d}\tau\bigr)$ is the time-ordered exponential defined by [RS75, Chapter X.12]

$$\mathcal{T}\exp\Bigl(\int_0^t A(\tau)\,\mathrm{d}\tau\Bigr) = \lim_{N\to\infty} e^{A(t_N)\Delta t} e^{A(t_{N-1})\Delta t} \cdots e^{A(t_1)\Delta t}, \quad t_k = k\Delta t, \quad \Delta t = \frac{t}{N}.$$

Using a Duhamel formula, the approximation of the ground-state projector is

$$
\begin{aligned}
|\mathbf{\Psi}_0^{(d)}\rangle\langle\mathbf{\Psi}_0^{(d)}| &= \frac{1}{\sqrt{2\pi q}}\int_{\mathbb{R}} e^{\mathrm{i}\boldsymbol{\mathcal{H}}^{(d)}t}e^{-\frac{t^2}{2q}}\,\mathrm{d}t + \mathcal{O}(e^{-\frac{1}{2}\gamma^2 q}) \\
&= \frac{1}{\sqrt{2\pi q}}\int_{\mathbb{R}} \mathcal{T}\exp\Big(\int_0^t \mathrm{id}_{1:j-\ell-1}\otimes H_M^{(\ell)}(\tau)\otimes \mathrm{id}_{j+\ell+1:d}\,\mathrm{d}\tau\Big)^* e^{\mathrm{i}H_{L+R}t}e^{-\frac{t^2}{2q}}\,\mathrm{d}t \\
&\qquad + \mathcal{O}(e^{-\frac{1}{2}\gamma^2 q} + q^{3/2}e^{-a\ell}).
\end{aligned}
$$

We would be done if it were possible to write $e^{\mathrm{i}H_{L+R}t} \simeq O_L \otimes \mathrm{id}_{j+1:d}\,\mathrm{id}_{1:j}\otimes O_R$ for $O_L \in \mathscr{L}(\mathcal{H}_{1:j})$ and $O_R \in \mathscr{L}(\mathcal{H}_{j+1:d})$ that are independent of t. In order to do so, another transformation is applied to H_M and H_{L+R} to guarantee that such a step is justified.

Proof of Theorem 4.1.6

Lemma 4.1.9. *Let $q>0$ and ρ_q be defined by*

$$
\rho_q = \frac{1}{\sqrt{2\pi q}}\int_{\mathbb{R}} e^{\mathrm{i}\boldsymbol{\mathcal{H}}^{(d)}t}e^{-\frac{t^2}{2q}}\,\mathrm{d}t. \tag{4.1.9}
$$

Then we have

$$
\|\rho_q - |\mathbf{\Psi}_0^{(d)}\rangle\langle\mathbf{\Psi}_0^{(d)}|\| \leq e^{-\frac{1}{2}\gamma^2 q}, \tag{4.1.10}
$$

where γ is the spectral gap.

Proof. This follows from the spectral gap assumption 4.1.1 and the fact that the Fourier transform of $t \mapsto \frac{1}{\sqrt{2\pi q}}e^{-\frac{t^2}{2q}}$ is $\omega \mapsto e^{-\frac{1}{2}\omega^2}$. □

Lemma 4.1.10. *For $1 \leq j \leq d$ and $\ell > 0$, let*

$$
H_M = \sum_{k=j-\frac{\ell}{3}}^{j+\frac{\ell}{3}} h_k, \quad H_L = \sum_{k<j-\frac{\ell}{3}} h_k, \quad H_R \sum_{k>j+\frac{\ell}{3}} h_k.
$$

For $q>0$, let

$$
H_M(q) = \frac{1}{\sqrt{2\pi q}}\int_{\mathbb{R}} e^{-\mathrm{i}\boldsymbol{\mathcal{H}}^{(d)}t}H_M e^{\mathrm{i}\boldsymbol{\mathcal{H}}^{(d)}t}e^{-\frac{t^2}{2q}}\,\mathrm{d}t - \langle\mathbf{\Psi}_0^{(d)}, H_M\mathbf{\Psi}_0^{(d)}\rangle \tag{4.1.11}
$$

$$
H_L(q) = \frac{1}{\sqrt{2\pi q}}\int_{\mathbb{R}} e^{-\mathrm{i}\boldsymbol{\mathcal{H}}^{(d)}t}H_L e^{\mathrm{i}\boldsymbol{\mathcal{H}}^{(d)}t}e^{-\frac{t^2}{2q}}\,\mathrm{d}t - \langle\mathbf{\Psi}_0^{(d)}, H_L\mathbf{\Psi}_0^{(d)}\rangle \tag{4.1.12}
$$

$$
H_R(q) = \frac{1}{\sqrt{2\pi q}}\int_{\mathbb{R}} e^{-\mathrm{i}\boldsymbol{\mathcal{H}}^{(d)}t}H_R e^{\mathrm{i}\boldsymbol{\mathcal{H}}^{(d)}t}e^{-\frac{t^2}{2q}}\,\mathrm{d}t - \langle\mathbf{\Psi}_0^{(d)}, H_R\mathbf{\Psi}_0^{(d)}\rangle. \tag{4.1.13}
$$

Then for all $q > 0$, we have

$$H = H_L(q) + H_M(q) + H_R(q), \tag{4.1.14}$$

and

$$\|H_M(q)\mathbf{\Psi}_0^{(d)}\|, \|H_L(q)\mathbf{\Psi}_0^{(d)}\|, \|H_R(q)\mathbf{\Psi}_0^{(d)}\| \leq \gamma J e^{-\frac{1}{2}\gamma^2 q}. \tag{4.1.15}$$

Proof. Since $H = H_L + H_M + H_R$, eq. (4.1.14) is clear. For eq. (4.1.15), we have

$$\begin{aligned} H_M(q)\mathbf{\Psi}_0^{(d)} &= \frac{1}{\sqrt{2\pi q}} \int_{\mathbb{R}} e^{-\mathrm{i}\boldsymbol{\mathcal{H}}^{(d)}t} H_M e^{\mathrm{i}\boldsymbol{\mathcal{H}}^{(d)}t} \mathbf{\Psi}_0^{(d)} e^{-\frac{t^2}{2q}} \, \mathrm{d}t - \langle \mathbf{\Psi}_0^{(d)}, H_M \mathbf{\Psi}_0^{(d)} \rangle \mathbf{\Psi}_0^{(d)} \\ &= \frac{1}{\sqrt{2\pi q}} \int_{\mathbb{R}} e^{-\mathrm{i}\boldsymbol{\mathcal{H}}^{(d)}t} P_0^{\perp} H_M \mathbf{\Psi}_0^{(d)} e^{-\frac{t^2}{2q}} \, \mathrm{d}t, \end{aligned}$$

where $P_0^{\perp} = \mathrm{id} - |\mathbf{\Psi}_0^{(d)}\rangle\langle\mathbf{\Psi}_0^{(d)}|$. We have

$$\|P_0^{\perp} H_M \mathbf{\Psi}_0^{(d)}\| \leq \gamma \|H H_M \mathbf{\Psi}_0^{(d)}\| \leq \gamma \|[H, H_M]\mathbf{\Psi}_0^{(d)}\| \leq \gamma J.$$

Hence using again the spectral gap of $\boldsymbol{\mathcal{H}}^{(d)}$, we obtain

$$\|H_M(q)\mathbf{\Psi}_0^{(d)}\| \leq \gamma J e^{-\frac{1}{2}\gamma^2 q}. \tag{4.1.16}$$

The same proof applies to H_L and H_R. □

The operators $H_L(q)$, $H_M(q)$ and $H_R(q)$ do not have the same support as H_L, H_M and H_R. In fact, their support is now the full Hilbert space $\mathcal{H}_{1:d}$. However, this can be solved by truncating the operators using Corollary 4.1.5.

Lemma 4.1.11. *There are self-adjoint operators $\widetilde{H}_L(q)$, $\widetilde{H}_M(q)$ and $\widetilde{H}_R(q)$ with respective support in $\mathcal{H}_{1:j}$, $\mathcal{H}_{j-2\ell/3:j+2\ell/3}$ and $\mathcal{H}_{j+1:d}$ such that*

$$\begin{aligned} \|H_M(q) - \widetilde{H}_M(q)\| &\lesssim \|h\|\ell^2 d e^{-a\ell/3} e^{qa^2v^2/2}, \\ \|H_L(q) - \widetilde{H}_L(q)\| &\lesssim \|h\|\ell^2 d e^{-a\ell/3} e^{qa^2v^2/2}, \\ \|H_R(q) - \widetilde{H}_R(q)\| &\lesssim \|h\|\ell^2 d e^{-a\ell/3} e^{qa^2v^2/2}. \end{aligned}$$

Proof. We only give the proof for $\widetilde{H}_M(q)$ as it is identical for the other truncations. By Corollary 4.1.5, there is an operator $H_M^{(\ell)}(t)$ with support in $\mathcal{H}_{j-2\ell/3:j+2\ell/3}$ such that

$$\|e^{-\mathrm{i}\boldsymbol{\mathcal{H}}^{(d)}t} H_M e^{\mathrm{i}\boldsymbol{\mathcal{H}}^{(d)}t} - H_M^{(\ell)}(t)\| \leq \|h\|\ell^2 d \exp(-a(\ell/3 - v|t|)).$$

Using that for $p, q > 0$, $\int_0^\infty e^{pt} e^{-\frac{t^2}{2q}} \, \mathrm{d}t \lesssim q^{1/2} e^{p^2 q/2}$.We deduce that there is an operator $\widetilde{H}_M(q)$ such that

$$\|H_M(q) - \widetilde{H}_M(q)\| \lesssim \|h\|\ell^2 d e^{-a\ell/3} e^{qa^2v^2/2}.$$

□

Lemma 4.1.12. *Let* $q > 0$ *and* $\widetilde{\rho}_q$ *be given by*

$$\widetilde{\rho}_q = \frac{1}{\sqrt{2\pi q}} \int_{\mathbb{R}} e^{\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_M(q)+\widetilde{H}_R(q))t} e^{-\frac{t^2}{2q}} \,\mathrm{d}t,$$

where $\widetilde{H}_L(q)$, $\widetilde{H}_M(q)$ *and* $\widetilde{H}_R(q)$ *are defined in Lemma 4.1.11. Then we have*

$$\left\|\widetilde{\rho}_q - |\mathbf{\Psi}_0^{(d)}\rangle\langle\mathbf{\Psi}_0^{(d)}|\right\| \lesssim \|h\|\ell^2 d q^{1/2} e^{-a\ell/3} e^{qa^2v^2/2} + e^{-\frac{1}{2}\gamma^2 q}. \tag{4.1.17}$$

Proof. The proof relies on a Duhamel formula:

$$\begin{aligned}
\|\widetilde{\rho}_q - |\mathbf{\Psi}_0^{(d)}\rangle\langle\mathbf{\Psi}_0^{(d)}|\| &\leq \|\widetilde{\rho}_q - \rho_q\| + \|\rho_q - |\mathbf{\Psi}_0^{(d)}\rangle\langle\mathbf{\Psi}_0^{(d)}|\|, \\
&\leq \frac{1}{\sqrt{2\pi q}} \int_{\mathbb{R}} \|e^{\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_M(q)+\widetilde{H}_R(q))t} - e^{\mathrm{i}\boldsymbol{\mathcal{H}}^{(d)}t}\| e^{-\frac{t^2}{2q}} \,\mathrm{d}t + e^{-\frac{1}{2}\gamma^2 q}, \\
&\lesssim \|h\|\ell^2 d q^{1/2} e^{-a\ell/3} e^{qa^2v^2/2} + e^{-\frac{1}{2}\gamma^2 q},
\end{aligned}$$

where we have used Lemma 4.1.11. □

Lemma 4.1.13. *Let* $\widetilde{H}_L(q)$ *and* $\widetilde{H}_R(q)$ *be the operators defined in Lemma 4.1.11. Let* $\alpha > 0$ *and* $O_R(q)$ *and* $O_L(q)$ *be the following spectral projections*

$$O_L(q) = \sum_{|\lambda|\leq\alpha} |\Phi_\lambda^{(L)}\rangle\langle\Phi_\lambda^{(L)}|, \quad O_R(q) = \sum_{|\lambda|\leq\alpha} |\Phi_\lambda^{(R)}\rangle\langle\Phi_\lambda^{(R)}|, \tag{4.1.18}$$

where $(\Phi_\lambda^{(L)})$ *and* $(\Phi_\lambda^{(R)})$ *are the normalised eigenvectors of* $\widetilde{H}_L(q)$ *and* $\widetilde{H}_R(q)$*. Then we have*

$$\|O_R O_L \mathbf{\Psi}_0^{(d)} - \mathbf{\Psi}_0^{(d)}\| \leq \frac{1}{\alpha}\Big(\|\widetilde{H}_L(q) - H_L(q)\| + \|\widetilde{H}_R(q) - H_R(q)\| + \|H_L\mathbf{\Psi}_0^{(d)}\| + \|H_R\mathbf{\Psi}_0^{(d)}\|\Big), \tag{4.1.19}$$

and

$$\|(e^{\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_R(q))t} - \mathrm{id})O_L O_R\| \leq 2\alpha|t|. \tag{4.1.20}$$

Proof. We first prove the estimate (4.1.19). Since $O_L(q)$ and $O_R(q)$ commute and are bounded operators by 1, we have

$$\|O_L O_R \mathbf{\Psi}_0^{(d)} - \mathbf{\Psi}_0^{(d)}\| \leq \|O_L\mathbf{\Psi}_0^{(d)} - \mathbf{\Psi}_0^{(d)}\| + \|O_R\mathbf{\Psi}_0^{(d)} - \mathbf{\Psi}_0^{(d)}\|. \tag{4.1.21}$$

We have

$$\begin{aligned}
\|O_L\mathbf{\Psi}_0^{(d)} - \mathbf{\Psi}_0^{(d)}\| &\leq \Big\| \int_{|\lambda|\geq\alpha} \mathrm{d}P_\lambda^{\widetilde{H}_L(q)}(\mathbf{\Psi}_0^{(d)}) \Big\| \\
&\leq \frac{1}{\alpha}\Big\| \int_{|\lambda|\geq\alpha} \lambda \mathrm{d}P_\lambda^{\widetilde{H}_L(q)}(\mathbf{\Psi}_0^{(d)}) \Big\| \\
&\leq \frac{1}{\alpha}\|\widetilde{H}_L(q)\mathbf{\Psi}_0^{(d)}\| \\
&\leq \frac{1}{\alpha}\big(\|\widetilde{H}_L(q) - H_L(q)\| + \|H_L\mathbf{\Psi}_0^{(d)}\|\big).
\end{aligned}$$

Estimate (4.1.20) follows from the definition of O_L and O_R. □

A final lemma is needed before completing the proof of Theorem 4.1.6 about the splitting of the evolution $e^{\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_M(q)+\widetilde{H}_R(q))t}$.

Lemma 4.1.14. *With the notation in Lemma 4.1.11, there is a family of operators $\widetilde{H}_M^{(\ell)}(t) \in \mathscr{L}(\mathcal{H}_{j-\ell:j+\ell})$ such that*

$$\left\| e^{\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_M(q)+\widetilde{H}_R(q))t} - \mathcal{T}\exp\left(\int_0^t \mathrm{id}_{1:j-\ell-1} \otimes \widetilde{H}_M^{(\ell)}(\tau) \otimes \mathrm{id}_{j+\ell+1:d}\, \mathrm{d}\tau\right)^* e^{\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_R(q))t} \right\| \leq t\|h\|\ell^2 d \exp(-a(\ell/3 - v|t|)), \quad (4.1.22)$$

where for a family of operators $A(t)$, $\mathcal{T}\exp\left(\int_0^t A(\tau)\,\mathrm{d}\tau\right)$ is the time-ordered exponential.

Proof. We can write

$$e^{\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_M(q)+\widetilde{H}_R(q))t} = e^{\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_M(q)+\widetilde{H}_R(q))t} e^{-\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_R(q))t} e^{\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_R(q))t}.$$

By differentiating we notice that $e^{\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_M(q)+\widetilde{H}_R(q))t} e^{-\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_R(q))t}$ is the solution to

$$\begin{cases} \mathrm{i}U'(t) = U(t) e^{\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_R(q))t} H_M e^{-\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_R(q))t} \\ U(0) = \mathrm{id}\,. \end{cases}$$

Alternatively, the solution to the equation above can be written

$$e^{\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_M(q)+\widetilde{H}_R(q))t} e^{-\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_R(q))t} = \mathcal{T}\exp\left(\int_0^t e^{\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_R(q))\tau} H_M e^{-\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_R(q))\tau}\, \mathrm{d}\tau\right)^*.$$

Using a Lieb-Robinson bound and Corollary 4.1.5, there is a family of operators $\widetilde{H}_M^{(\ell)}(t)$ such that for all $t \in \mathbb{R}$, $\widetilde{H}_M^{(\ell)}(t) \in \mathscr{L}(\mathcal{H}_{j-\ell:j+\ell})$ and

$$\left\| e^{\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_R(q))t} H_M e^{-\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_R(q))t} - \mathrm{id}_{1:j-\ell-1} \otimes \widetilde{H}_M^{(\ell)}(t) \otimes \mathrm{id}_{j+\ell+1:d} \right\| \leq \|h\|\ell^2 d \exp(-a(\ell/3 - v|t|)).$$

It remains to bound the difference between $\mathcal{T}\exp\left(\int_0^t e^{\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_R(q))\tau} H_M e^{-\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_R(q))\tau}\, \mathrm{d}\tau\right)$ and $\mathcal{T}\exp\left(\int_0^t \mathrm{id}_{1:j-\ell-1} \otimes \widetilde{H}_M^{(\ell)}(\tau) \otimes \mathrm{id}_{j+\ell+1:d}\, \mathrm{d}\tau\right)$. Recall that for a family of operators $A(t)$, the time-ordered exponential is defined by

$$\mathcal{T}\exp\left(\int_0^t A(\tau)\,\mathrm{d}\tau\right) = \lim_{N\to\infty} e^{A(t_N)\Delta t} e^{A(t_{N-1})\Delta t} \cdots e^{A(t_1)\Delta t}, \quad t_k = k\Delta t, \quad \Delta t = \frac{t}{N}.$$

By a Duhamel formula, the difference of the time-ordered exponentials can be bounded by

$$\Big\| \mathcal{T} \exp\Big(\int_0^t e^{\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_R(q))\tau} H_M e^{-\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_R(q))\tau}\, \mathrm{d}\tau \Big) - \mathcal{T} \exp\Big(\int_0^t \mathrm{id}_{1:j-\ell-1} \otimes \widetilde{H}_M^{(\ell)}(\tau) \otimes \mathrm{id}_{j+\ell+1:d}\, \mathrm{d}\tau \Big) \Big\| \leq t \|h\| \ell^2 d \exp(-a(\ell/3 - v|t|)).$$

This finishes the proof of the lemma. $\square$

We have now all the ingredients to prove Hastings area law 4.1.6.

Proof of Theorem 4.1.6. Let O_L and O_R be the operators defined in Lemma 4.1.13. Then we have

$$|\mathbf{\Psi}_0^{(d)}\rangle\langle\mathbf{\Psi}_0^{(d)}| = |\mathbf{\Psi}_0^{(d)}\rangle\langle\mathbf{\Psi}_0^{(d)}| O_L O_R + \frac{1}{\alpha}\mathcal{O}(\|\widetilde{H}_L(q) - H_L(q)\| + \|\widetilde{H}_R(q) - H_R(q)\| + \|H_L \mathbf{\Psi}_0^{(d)}\| + \|H_R \mathbf{\Psi}_0^{(d)}\|).$$

Thus with Lemma 4.1.10 and Lemma 4.1.11, we obtain

$$|\mathbf{\Psi}_0^{(d)}\rangle\langle\mathbf{\Psi}_0^{(d)}| = |\mathbf{\Psi}_0^{(d)}\rangle\langle\mathbf{\Psi}_0^{(d)}| O_L O_R + \frac{1}{\alpha}\mathcal{O}\Big(\gamma J e^{-\frac{1}{2}\gamma^2 q} + \|h\| \ell^2 d q^{1/2} e^{-a\ell/3} e^{q a^2 v^2}\Big).$$

Using that O_L and O_R are bounded operators by 1, in combination with Lemma 4.1.12, we get

$$\begin{aligned}
|\mathbf{\Psi}_0^{(d)}\rangle\langle\mathbf{\Psi}_0^{(d)}| &= \frac{1}{\sqrt{2\pi q}} \int_{\mathbb{R}} e^{\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_M(q)+\widetilde{H}_R(q))t} e^{-\frac{t^2}{2q}} O_L O_R\, \mathrm{d}t + \mathcal{O}\Big(\frac{\|h\|\ell^2 d}{\alpha} q^{1/2} e^{-a\ell/3} e^{q a^2 v^2} + \frac{\gamma J}{\alpha} e^{-\frac{1}{2}\gamma^2 q}\Big) \\
&= \frac{1}{\sqrt{2\pi q}} \int_{\mathbb{R}} \mathcal{T} \exp\Big(\int_0^t e^{\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_R(q))\tau} H_M e^{-\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_R(q))\tau}\, \mathrm{d}\tau \Big)^* e^{-\frac{t^2}{2q}} e^{\mathrm{i}(\widetilde{H}_L(q)+\widetilde{H}_R(q))t} O_L O_R\, \mathrm{d}t \\
&\quad + \mathcal{O}\Big(\frac{\|h\|\ell^2 d}{\alpha} q^{1/2} e^{-a\ell/3} e^{q a^2 v^2} + \frac{\gamma J}{\alpha} e^{-\frac{1}{2}\gamma^2 q}\Big),
\end{aligned}$$

where we have used Lemma 4.1.14. By Lemma 4.1.13, we thus have

$$\begin{aligned}
|\mathbf{\Psi}_0^{(d)}\rangle\langle\mathbf{\Psi}_0^{(d)}| = \frac{1}{\sqrt{2\pi q}} \int_{\mathbb{R}} \mathcal{T} \exp\Big(\int_0^t \mathrm{id}_{1:j-\ell-1} \otimes \widetilde{H}_M^{(\ell)}(\tau) \otimes \mathrm{id}_{j+\ell+1:d}\, \mathrm{d}\tau \Big)^* e^{-\frac{t^2}{2q}} O_L O_R\, \mathrm{d}t \\
+ \mathcal{O}\Big(\alpha q^{1/2} + \frac{\|h\|\ell^2 d}{\alpha} q^{1/2} e^{-a\ell/3} e^{q a^2 v^2} + \frac{\gamma J}{\alpha} e^{-\frac{1}{2}\gamma^2 q}\Big).
\end{aligned}$$

All it remains to do is to set the parameters α and q to prove Theorem 4.1.6. Taking $q = \tilde{q}\ell$ such that $\big(\frac{\gamma^2}{2} + a v^2\big)\tilde{q} < \frac{a}{3}$ and $\alpha < e^{-\frac{1}{2}\gamma^2 \tilde{q}\ell}$ give (4.1.6). $\square$

Chapter 5

DMRG for the electronic Schrödinger equation

Density matrix renormalisation group [Whi92] (DMRG) is an alternating scheme to solve linear problems or eigenvalue problems in the tensor train format. In the mathematical community, it is also referred to the *alternating linear scheme* (ALS) in its simplest version or to the *modified ALS (MALS)* [HRS12a], which is the equivalent to the two-site DMRG. In DMRG, given a symmetric matrix $\boldsymbol{\mathcal{H}} \in \mathbb{R}^{n_1 \cdots n_d \times n_1 \cdots n_d}$, we want to solve for $\boldsymbol{x}_* \in \mathbb{R}^{n_1 \cdots n_d}$ the linear problem

$$\boldsymbol{\mathcal{H}} \boldsymbol{x}_* = \boldsymbol{b}, \tag{5.0.1}$$

for a given $\boldsymbol{b} \in \mathbb{R}^{n_1 \cdots n_d}$, or for $(\lambda, \boldsymbol{x}_*) \in \mathbb{R} \times \left(\mathbb{R}^{n_1 \cdots n_d} \setminus \{0\}\right)$ the lowest eigenvalue problem

$$\boldsymbol{\mathcal{H}} \boldsymbol{x}_* = \lambda \boldsymbol{x}_*, \tag{5.0.2}$$

For both problems, a tensor train representation of the operator $\boldsymbol{\mathcal{H}}$ is needed in order to efficiently implement the DMRG algorithm.

5.1 Tensor train operators

5.1.1 Graphical representation of tensors

As we are going to manipulate formulas involving more and more tensors, it can be helpful to have graphical representations of the summation over shared indices between tensors. This operation is called *tensor contraction*.

Let $\boldsymbol{u} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ be a tensor. The graphical representation of $\boldsymbol{u}$ is given by Figure 5.1. Elementary operations between vectors and matrices are explained in Figure 5.2.

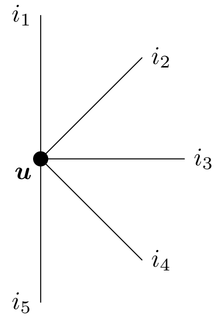
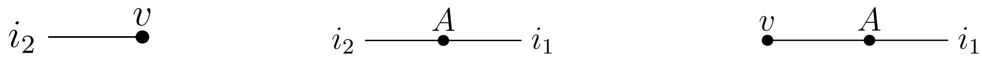


Figure 5.1: Graphical representation of an order 5 tensor $\boldsymbol{u}$. The tensor $\boldsymbol{u}$ is represented by its vertex and its indices by the five free edges.



(a) Vector v_{i_2}. (b) Matrix $A_{i_1}^{i_2}$. (c) Matrix-vector product $(Av)_{i_1} = \sum_{i_2} A_{i_1}^{i_2} v_{i_2}$.

Figure 5.2: Contraction of tensors. Every pair of connected edges is a summation over the shared index.

5.1.2 Definition of tensor train operators

Definition 5.1.1 (Tensor train operator). *Let $\boldsymbol{\mathcal{H}} \in \mathbb{R}^{n_1 \cdots n_d \times n_1 \cdots n_d}$ be a matrix. A tensor train operator (TTO) representation of the matrix is any tuple of order 4 tensors $(H_1, \dots, H_d)$, $H_k \in \mathbb{R}^{R_{k-1} \times n_k \times n_k \times R_k}$ $(R_0 = R_d = 1)$ such that*

$$\forall \mathbf{i}, \mathbf{j} \in [\![\mathbf{n}]\!], \boldsymbol{\mathcal{H}}_{i_1 \dots i_d}^{j_1 \dots j_d} = H_1[i_1, j_1] \cdots H_d[i_d, j_d],$$

or written with the strong Kronecker product

$$\boldsymbol{\mathcal{H}} = H_1 \bowtie \cdots \bowtie H_d.$$

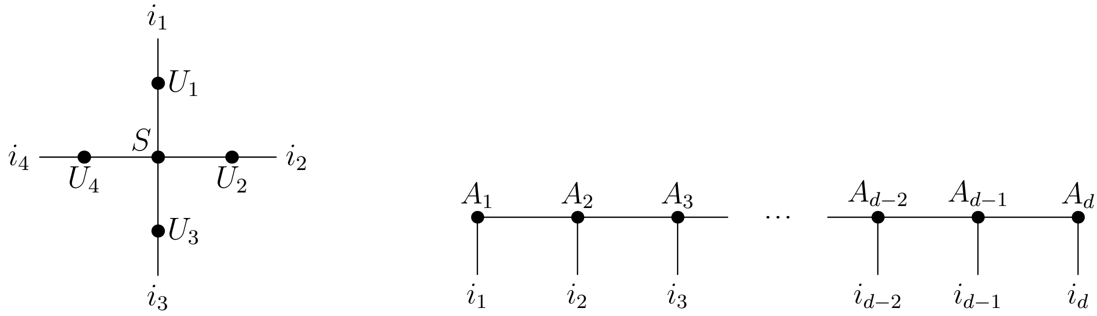
$(R_0, \dots, R_d)$ are the TTO ranks of the TTO representation $(H_1, \dots, H_d)$. $(H_1, \dots, H_d)$ are the TTO cores of the TTO representation.

In the context of tensor trains, this is the natural generalisation of the tensor product of operators. Indeed let $h_k \in \mathbb{R}^{n_k \times n_k}$ for $k \in [\![d]\!]$, then the operator $\boldsymbol{\mathcal{H}} = h_1 \otimes \cdots \otimes h_d$ has a TTO representation of TTO rank 1 with TTO cores given by $H_k[i_k, j_k] = (h_k)_{i_k, j_k}$ for $k \in [\![d]\!]$.

The diagrammatic representation of a TTO is similar to the diagrammatic of a TT as illustrated in Figure 5.4.

A TTO representation of a matrix can be obtained by reordering the indices of the matrix $\boldsymbol{\mathcal{H}}$ and performing a TT-SVD on the resulting tensor. More precisely, by defining the tensor $\widetilde{\boldsymbol{\mathcal{H}}} \in \mathbb{R}^{n_1^2 \times \cdots \times n_d^2}$

$$\widetilde{\boldsymbol{\mathcal{H}}}_{i_1 j_1; \dots; i_d j_d} = \boldsymbol{\mathcal{H}}_{i_1 \dots i_d}^{j_1 \dots j_d},$$



(a) Tucker decomposition (b) Tensor train decomposition

Figure 5.3: Tucker and tensor train decompositions. From the graphical representation, at first sight we see that the Tucker format still has an exponential dependence in the order of the tensor, whereas this exponential dependence has disappeared in the TT format.

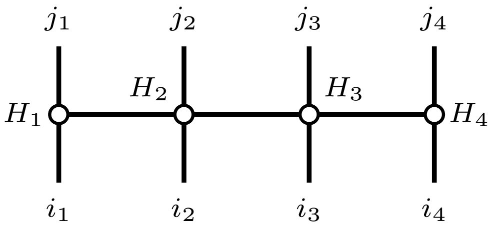


Figure 5.4: Diagrammatic representation of a TTO

we realise that a TTO representation is simply a TT representation of $\widetilde{\boldsymbol{\mathcal{H}}}$.

5.1.3 Algebraic properties

Like the TT representation of vectors, the TTO format has some algebraic stability property.

Proposition 5.1.2. *Let* $\boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{H}} \in \mathbb{R}^{n_1 \cdots n_d \times n_1 \cdots n_d}$ *be matrices and* $(G_1, \dots, G_d)$, $G_k \in \mathbb{R}^{R^G_{k-1} \times n_k \times n_k \times R^G_k}$ *and* $(H_1, \dots, H_d)$, $H_k \in \mathbb{R}^{R^H_{k-1} \times n_k \times n_k \times R^H_k}$ *be respectively TTO representations of* $\boldsymbol{\mathcal{G}}$ *and* $\boldsymbol{\mathcal{H}}$*. Let* $\boldsymbol{x} \in \mathbb{R}^{n_1 \cdots n_d}$ *be vectors with respective TT representations* $(X_1, \dots, X_d)$, $X_k \in \mathbb{R}^{n_k \times r^A_{k-1} \times r^A_k}$*. Then*

(i). the sum $\boldsymbol{\mathcal{G}} + \boldsymbol{\mathcal{H}}$ *has a TTO representation* $(S_1, \dots, S_d)$ *given by*

$$\begin{gathered} S_1[i_1, j_1] = \begin{bmatrix} G_1[i_1, j_1] & H_1[i_1, j_1] \end{bmatrix}, \quad S_d[i_d, j_d] = \begin{bmatrix} G_d[i_d, j_d] \\ H_d[i_d, j_d] \end{bmatrix} \\ S_k[i_k, j_k] = \begin{bmatrix} G_k[i_k, j_k] & 0 \\ 0 & H_k[i_k, j_k] \end{bmatrix}, k = 2, \dots, d-1 \end{gathered} \tag{5.1.1}$$

(ii). the matrix-vector product $\boldsymbol{y} = \boldsymbol{\mathcal{H}}\boldsymbol{x}$ *has a TT representation* $(Y_1, \dots, Y_d)$ *with* $Y_k[j_k] \in \mathbb{R}^{R^H_{k-1} r^X_{k-1} \times R^H_k r^X_k}$

$$Y_k[i_k] = \sum_{j_k=1}^{n_k} H_k[i_k, j_k] \otimes X_k[j_k], \quad k \in [\![d]\!]. \tag{5.1.2}$$

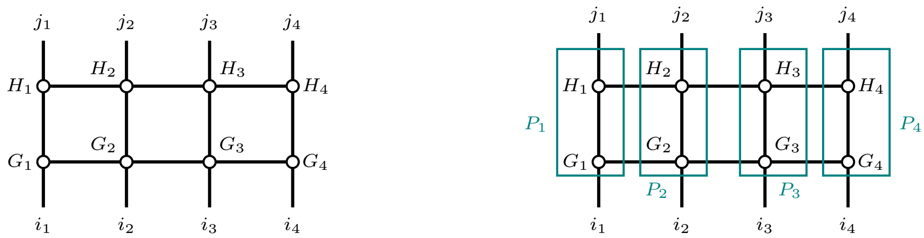
(iii). the product $\boldsymbol{\mathcal{G}}\boldsymbol{\mathcal{H}}$ *has a TTO representation* $(P_1, \dots, P_d)$ *given by*

$$P_k[i_k, j_k] = \sum_{\ell_k=1}^{n_k} G_k[i_k, \ell_k] \otimes H_k[\ell_k, j_k], \quad k \in [\![d]\!]. \tag{5.1.3}$$

Proof. This is a direct computation. □

Remark 5.1.3. *The TTO representations of the sum and the product of the operators are not optimal. This is clear in the case of the sum* $\boldsymbol{\mathcal{G}} + \boldsymbol{\mathcal{H}}$ *when we consider* $\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{H}}$*. A TT rounding step is required in order to reduce the TTO ranks of the representation. This is not innocuous as essential properties of the matrix can be lost in the rounding procedure (symmetry for instance).*

A diagrammatic proof of the formula for the product of two TTO representations is given in Figure 5.5, avoiding cumbersome computations.



(a) Diagrammatic representation of the product of two TTO

(b) Diagrammatic representation of the product of two TTO

Figure 5.5: Diagrammatic proof of the product of two TTO. The left panel is the diagrammatic representation of the product of two TTO. On the right panel, the boxed tensors P_k are the TTO cores of a TTO representation of the product GH, provided that the double edges shared between neighbouring P_k are gathered into one edge.

Example 5.1.4. *Let us consider the following matrix* $\boldsymbol{\mathcal{H}} \in \mathbb{R}^{n^d \times n^d}$

$$\boldsymbol{\mathcal{H}} = h \otimes \mathrm{id} \otimes \cdots \otimes \mathrm{id} + \cdots + \mathrm{id} \otimes \mathrm{id} \otimes \cdots \otimes h,$$

where $h \in \mathbb{R}^{n \times n}$ *is a symmetric matrix and* id *is the identity in* $\mathbb{R}^{n \times n}$*. Since the matrix* $h \otimes \mathrm{id} \otimes \cdots \otimes \mathrm{id}$ *is a TTO of rank 1, a naïve application of Proposition 5.1.2 yields a TTO* *representation of $\boldsymbol{\mathcal{H}}$ of rank d. However it is possible to achieve a rank 2 representation with the following construction*

$$\begin{aligned} H_1[i_1, j_1] &= \begin{bmatrix} h_{i_1 j_1} & \delta_{i_1 j_1} \end{bmatrix}, \quad H_d[i_d, j_d] = \begin{bmatrix} \delta_{i_d j_d} \\ h_{i_d j_d} \end{bmatrix} \\ H_k[i_k, j_k] &= \begin{bmatrix} \delta_{i_k j_k} & 0 \\ h_{i_k j_k} & \delta_{i_k j_k} \end{bmatrix}, \quad k = 2, \dots, d-1. \end{aligned} \tag{5.1.4}$$

Note that this representation also satisfies the property stated in Proposition 5.1.5.

Proposition 5.1.5. *Let $\boldsymbol{\mathcal{H}} \in \mathbb{R}^{n_1 \cdots n_d \times n_1 \cdots n_d}$ be a symmetric matrix. Then there is a TTO representation of $\boldsymbol{\mathcal{H}}$ such that*

$$\forall\, 1 \leq i_k, j_k \leq n_k, \ H_k[i_k, j_k] = H_k[j_k, i_k], \quad k = 1, \dots, L. \tag{5.1.5}$$

Proof. □

5.1.4 The electronic Hamiltonian as a TTO

The electronic Hamiltonian operator in second quantisation is given by

$$\boldsymbol{\mathcal{H}} = \sum_{i,j=1}^{d} h_{ij} c_i^\dagger c_j + \frac{1}{2} \sum_{i,j,k,\ell=1}^{d} V_{ijk\ell} c_i^\dagger c_j^\dagger c_\ell c_k, \tag{5.1.6}$$

where h_{ij} (resp. $V_{ijk\ell}$) correspond to the one-electron integrals and two-electron integrals with Mulliken's convention [HJO14]. The tensor representation of the creation $c_i^\dagger$ and annihilation c_i operators can be written as a tensor product of 2×2 matrices

$$c_i = Z \otimes \cdots \otimes Z \otimes C \otimes \mathrm{id}_2 \otimes \cdots \otimes \mathrm{id}_2 \in \mathbb{R}^{2^d \times 2^d}, \tag{5.1.7}$$

$$c_i^\dagger = Z \otimes \cdots \otimes Z \otimes C^\top \otimes \mathrm{id}_2 \otimes \cdots \otimes \mathrm{id}_2 \in \mathbb{R}^{2^d \times 2^d}, \tag{5.1.8}$$

where C (resp. $C^\top$) appears in the i-th position and

$$C = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \text{and} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Since the creation and annihilation operators are written as Kronecker products, their TTO rank is 1. Using the algebraic properties of TTOs in Proposition 5.1.2, a naïve implementation of the TTO of an electronic Hamiltonian has TTO rank scaling as d^4.

Noticing that the reshape of the two-body interaction at any cut is at most of rank d^2, we deduce that the TTO rank of the electronic Hamiltonian can be reduced to $\mathcal{O}(d^2)$ [CKN+16, BGP22]. The TT-SVD is useful to compress these ranks to the optimal ones. To preserve the particle conservation and the symmetry of the Hamiltonian, this procedure has to done with great care.

Remark 5.1.6. *In popular implementations of QC-DMRG, it is usual to work in the* space orbital picture. *Namely instead of having sites that can be either occupied or unoccupied, sites can be unoccupied, occupied with spin up or down, or doubly occupied. The expression of the electronic Hamiltonian is similar to the spin orbital case. The main reason of using this representation is that it is more suited for an implementation that preserves the $SU(2)$ symmetry.*

5.2 The DMRG algorithm

The DMRG algorithm [Whi92] is an algorithm to solve linear systems $\boldsymbol{\mathcal{H}}\boldsymbol{x}_* = \boldsymbol{b}$ or the lowest eigenvalue problem $\boldsymbol{\mathcal{H}}\boldsymbol{x}_* = \lambda \boldsymbol{x}_*$ using the variational characterisation of the solution to both problems. As such it is limited in the resolution of linear problems with *symmetric* and positive-definite matrices. In the following, we assume that $\boldsymbol{\mathcal{H}}$ is a symmetric, positive-definite matrix.

Assumption 5.2.1. *The matrix $\boldsymbol{\mathcal{H}} \in \mathbb{R}^{n_1 \cdots n_d \times n_1 \cdots n_d}$ is symmetric and positive-definite.*

The solution to the linear system $\boldsymbol{\mathcal{H}}\boldsymbol{x}_* = \boldsymbol{b}$ is also the minimiser of the functional

$$\boldsymbol{x}_* = \operatorname*{arg\,min}_{\boldsymbol{x} \in \mathbb{R}^{n_1 \cdots n_d}} \frac{1}{2} \langle \boldsymbol{x}, \boldsymbol{\mathcal{H}}\boldsymbol{x} \rangle - \langle \boldsymbol{b}, \boldsymbol{x} \rangle. \tag{5.2.1}$$

Using the Rayleigh-Ritz principle, the lowest eigenvalue of $\boldsymbol{\mathcal{H}}$ is given by

$$\boldsymbol{x}_* = \operatorname*{arg\,min}_{x \in \mathbb{R}^{n_1 \cdots n_d}} \frac{\langle \boldsymbol{x}, \boldsymbol{\mathcal{H}}\boldsymbol{x} \rangle}{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}. \tag{5.2.2}$$

The first idea in DMRG is to reduce the minimisation set to the set of TT tensors with prescribed TT ranks $\mathbf{r}$

$$\mathscr{M}_{\mathrm{TT}_{\leq \mathbf{r}}} = \Big\{ \boldsymbol{u} \mid \exists (A_k)_{k \in [\![d]\!]} \in \bigtimes_{k \in [\![d]\!]} \mathbb{R}^{n_k \times r_{k-1} \times r_k},\ \forall \mathbf{i} \in [\![\mathbf{n}]\!],\ \boldsymbol{u}_{i_1 \dots i_d} = A_1[i_1] \cdots A_d[i_d] \Big\},$$

and thus solve, for example in the linear solve case, the following problem

$$\boldsymbol{x}_* = \operatorname*{arg\,min}_{\boldsymbol{x} \in \mathscr{M}_{\mathrm{TT}_{\leq \mathbf{r}}}} \frac{1}{2} \langle \boldsymbol{x}, \boldsymbol{\mathcal{H}}\boldsymbol{x} \rangle - \langle \boldsymbol{b}, \boldsymbol{x} \rangle. \tag{5.2.3}$$

The practical trump of the DMRG algorithm now relies on the fact that the approximate minimisation problem above is solved by a sequence of much smaller symmetric positive-definite linear systems of size $\mathcal{O}(r_{\mathrm{TT}}^2 R_{\mathrm{TTO}})$. These problems are tractable and moreover it is possible to import all the technology developed in numerical linear algebra to solve these problems efficiently.

5.2.1 General algorithm

The DMRG algorithm, also known as *alternating linear scheme* (ALS) [HRS12a], is an alternating optimisation over the set $\mathscr{M}_{\mathrm{TT}_{\leq \mathbf{r}}}$. The general idea is to perform a descent step for each TT core separately. Let TT be the map

$$\mathrm{TT}\ :\begin{cases}\mathbb{R}^{r_0\times n_1\times r_1}\times\cdots\times\mathbb{R}^{r_{d-1}\times n_d\times r_d}\to\mathbb{R}^{n_1\cdots n_d}\\ \qquad (X_1,\dots,X_d)\mapsto X_1\bowtie\cdots\bowtie X_d=(X_1[i_1]\cdots X_d[i_d]),\end{cases}$$

Introducing the functional $J(\boldsymbol{x})=\frac{1}{2}\langle\boldsymbol{x},\boldsymbol{\mathcal{H}}\boldsymbol{x}\rangle-\langle\boldsymbol{b},\boldsymbol{x}\rangle$ and j the map

$$j(X_1,\dots,X_d)=J\circ\mathrm{TT}(X_1,\dots,X_d), \tag{5.2.4}$$

then minimising J over the manifold $\mathscr{M}_{\mathrm{TT}_{\leq \mathbf{r}}}$ is the same as minimising the functional j over the product space $\mathbb{R}^{n_1\times r_1}\times\mathbb{R}^{r_1\times n_2\times r_2}\times\cdots\times\mathbb{R}^{r_{d-1}\times n_d}$:

$$\min_{\boldsymbol{x}\in\mathscr{M}_{\mathrm{TT}_{\leq \mathbf{r}}}}\frac{1}{2}\langle\boldsymbol{x},\boldsymbol{\mathcal{H}}\boldsymbol{x}\rangle-\langle\boldsymbol{b},\boldsymbol{x}\rangle=\min_{X_1,\dots,X_d}j(X_1,\dots,X_d).$$

In the one-site DMRG procedure, the minimisation of j is carried out sequentially over (X_k) by freezing all the TT cores but one and by solving the minimisation problem for the remaining core. More precisely, for $k\in[\![d]\!]$, let P_k be defined by

$$P_k:\begin{cases}\mathbb{R}^{r_{k-1}\times n_k\times r_k}\to\mathbb{R}^{n_1\times\cdots\times n_d}\\ \qquad V\mapsto X_1\bowtie\cdots\bowtie X_{k-1}\bowtie V\bowtie X_{k+1}\bowtie\cdots X_d.\end{cases} \tag{5.2.5}$$

The minimisation problem to solve is the following

$$\min_{V\in\mathbb{R}^{r_{k-1}\times n_k\times r_k}}J(P_kV)=\min_{V\in\mathbb{R}^{r_{k-1}\times n_k\times r_k}}\frac{1}{2}\langle P_kV,\boldsymbol{\mathcal{H}}P_kV\rangle-\langle\boldsymbol{b},P_kV\rangle.$$

If the minimiser is denoted by Y_k, it thus solves

$$P_k^{\mathsf{T}}\boldsymbol{\mathcal{H}}P_kY_k=P_k^{\mathsf{T}}\boldsymbol{b}. \tag{5.2.6}$$

A natural condition to impose on P_k is that it is a partial isometry, for the following reason.

Proposition 5.2.2. *If P_k is a partial isometry, then the linear system* (5.2.6) *has a unique solution.*

Moreover the condition number of the linear system (5.2.6) *is bounded by the condition number of $\boldsymbol{\mathcal{H}}$,* i.e.

$$\mathrm{cond}_2\,P_k^{\mathsf{T}}\boldsymbol{\mathcal{H}}P_k\leq\mathrm{cond}_2\,\boldsymbol{\mathcal{H}}.$$

Proof. Since P_k is a partial isometry, the matrix $P_k^\mathsf{T} \boldsymbol{\mathcal{H}} P_k$ is symmetric positive-definite, thus the linear system has a unique solution.

The bound on the condition number follows from the fact that P_k is a partial isometry, as we have the inequalities $\lambda_{\min}(P_k^\mathsf{T} \boldsymbol{\mathcal{H}} P_k) \geq \lambda_{\min}(\boldsymbol{\mathcal{H}})$ and $\lambda_{\max}(P_k^\mathsf{T} \boldsymbol{\mathcal{H}} P_k) \leq \lambda_{\max}(\boldsymbol{\mathcal{H}})$. □

It is rather simple to impose that P_k defines a partial isometry, by imposing that the left TT cores are left-orthogonal while the right TT cores are right-orthogonal.

Lemma 5.2.3. *Let $(A_j)_{j \in [\![d]\!]}$ be a TT representation of some tensor $\boldsymbol{x} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$. For $k \in [\![d]\!]$, if $(A_1, \dots, A_{k-1})$ is left-orthogonal and $(A_{k+1}, \dots, A_d)$ is right-orthogonal, then P_k is a partial isometry.*

Proof. For $V \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$, we have

$$\begin{aligned}
\|P_k V\|^2 &= \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} \mathrm{Tr}\left(X_d[i_d]^\mathsf{T} \cdots X_{k+1}[i_{k+1}]^\mathsf{T} V[i_k]^\mathsf{T} X_{k-1}[i_{k-1}]^\mathsf{T} \cdots X_1[i_1]^\mathsf{T} \right. \\
&\qquad\qquad \left. X_1[i_1] \cdots X_{k-1}[i_{k-1}] V[i_k] X_{k+1}[i_{k+1}] \cdots X_d[i_d]\right) \\
&= \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} \mathrm{Tr}\left(V[i_k]^\mathsf{T} X_{k-1}[i_{k-1}]^\mathsf{T} \cdots X_1[i_1]^\mathsf{T} X_1[i_1] \cdots X_{k-1}[i_{k-1}] V[i_k] \right. \\
&\qquad\qquad \left. X_{k+1}[i_{k+1}] \cdots X_d[i_d] X_d[i_d]^\mathsf{T} \cdots X_{k+1}[i_{k+1}]^\mathsf{T}\right) \\
&= \sum_{i_k=1}^{n_k} \mathrm{Tr}\left(V[i_k]^\mathsf{T} V[i_k]\right) = \|V\|^2,
\end{aligned}$$

where we have used the cyclicity of the trace and the orthogonality of the TT cores. Thus P_k is indeed a partial isometry. □

The final algorithm is described in Algorithm 5.1, where at each step of the algorithm, we perform a linear solve for a reduced matrix $P_k^\mathsf{T} \boldsymbol{\mathcal{H}} P_k$ and a root shifting of the orthogonality center of the TT.

The optimisation steps (5.2.7) and (5.2.8) are called *microsteps*. An iteration over the loop s is called a sweep. Notice that at each microstep (5.2.7) or (5.2.8) the left TT cores are left-orthogonal and the right-TT cores are right-orthogonal, thanks to the root shifting step in the ALS algorithm.

5.2.2 Implementation details

In this part, we give some details about the implementation of the DMRG algorithm described in Algorithm 5.1, as well as the total computational cost of a sweep. Each microstep corresponds to solving a linear system of size $\mathcal{O}(nr^2)$ (where $n = \max n_k$ and $r = \max r_k$), hence at first glance, the storage cost would scale as $\mathcal{O}(n^2 r^4)$ and the computational cost of solving each linear system would scale as $\mathcal{O}(n^3 r^6)$ with a direct solver and $\mathcal{O}(n^2 r^4)$ for an iterative solver. Using the structure of the matrix $P_k^\mathsf{T} \boldsymbol{\mathcal{H}} P_k$, better scalings can be achieved.

Algorithm 5.1 One-site DMRG with sweeps

Input: $(X_1^{(0)}, \dots, X_d^{(0)})$ in right-orthogonal TT representation
Output: $(X_1^{(s)}, \dots, X_d^{(s)}) \in \mathscr{M}_{\mathrm{TT}_{\leq \mathbf{r}}}$ approximation of the minimiser in $\mathscr{M}_{\mathrm{TT}_{\leq \mathbf{r}}}$ of J

function ONE-SITE-DMRG($(X_1^{(0)}, \dots, X_d^{(0)})$)
  $s = 0$
  while `not converged` **do**
    for $k = 1, 2, \dots, d-1$ **do** ▷ Forward half-sweep

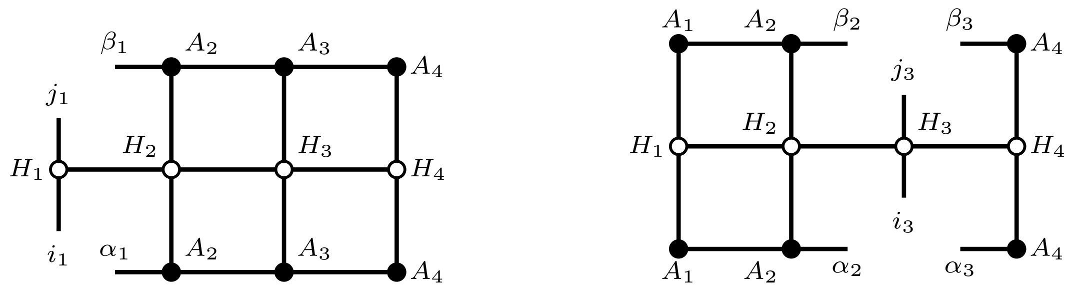
$$Y_k^{(s+\frac{1}{2})} = \operatorname*{arg\,min}_{V_k \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}} j(X_1^{(s+\frac{1}{2})}, \dots, X_{k-1}^{(s+\frac{1}{2})}, V_k, X_{k+1}^{(s)}, \dots, X_d^{(s)}) \tag{5.2.7}$$

      $Q, R = \mathtt{qr}\big(\big(Y_k^{(s+\frac{1}{2})}\big)_{\alpha_{k-1} i_k}^{\beta_k}\big)$ ▷ QR decomposition
      $\big(X_k^{(s+\frac{1}{2})}[i_k]\big)_{\alpha_{k-1}}^{\alpha_k} = Q_{\alpha_{k-1} i_k}^{\alpha_k}$ ▷ Keep Q
      $\big(X_{k+1}^{(s)}[i_{k+1}]\big)_{\alpha_k}^{\alpha_{k+1}} \leftarrow \big(R X_{k+1}^{(s)}[i_{k+1}]\big)_{\alpha_k}^{\alpha_{k+1}}.$ ▷ Shift R to the right
    end for
    for $k = d, d-1, \dots, 2$ **do** ▷ Backward half-sweep

$$Y_k^{(s+1)} = \operatorname*{arg\,min}_{V_k \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}} j(X_1^{(s+\frac{1}{2})}, \dots, X_{k-1}^{(s+\frac{1}{2})}, V_k, X_{k+1}^{(s+1)}, \dots, X_d^{(s+1)}) \tag{5.2.8}$$

      $L, Q = \mathtt{lq}\Big(\big(Y_k^{(s+1)}\big)_{\alpha_{k-1}}^{\beta_k i_k}\Big)$ ▷ LQ decomposition
      $\big(X_k^{(s+1)}[i_k]\big)_{\alpha_{k-1}}^{\alpha_k} = \big(Q\big)_{\alpha_{k-1}}^{\alpha_k i_k}$ ▷ Keep Q
      $\big(X_{k-1}^{(s+\frac{1}{2})}[i_{k-1}]\big)_{\alpha_{k-2}}^{\alpha_{k-1}} \leftarrow \big(X_{k-1}^{(s+\frac{1}{2})}[i_{k-1}] L\big)_{\alpha_{k-2}}^{\alpha_{k-1}}$ ▷ Shift L to the left
    end for
    $s = s + 1$
  end while
  return $(X_1^{(s)}, \dots, X_d^{(s)})$
end function

The matrix $P_k^\mathsf{T}\boldsymbol{\mathcal{H}}P_k$ A critical step in DMRG is to efficiently implement the effective matrix $P_k^\mathsf{T}\boldsymbol{\mathcal{H}}P_k$ (see Figure 5.6).



(a) Effective matrix $(P_1^\mathsf{T}\boldsymbol{\mathcal{H}}P_1)_{\alpha_1 i_1}^{\beta_1 j_1} \in \mathbb{R}^{r_1 n_1 \times r_1 n_1}$

(b) Effective matrix $(P_3^\mathsf{T}\boldsymbol{\mathcal{H}}P_3)_{\alpha_2 i_2 \alpha_3}^{\beta_2 j_3 \beta_3} \in \mathbb{R}^{r_2 n_3 r_3 \times r_2 n_3 r_3}$

Figure 5.6: Examples of $P_k^\mathsf{T}\boldsymbol{\mathcal{H}}P_k$

As the TT ranks can be large (of the order of $10^3 - 10^4$), it is inefficient and useless to build the effective matrix $P_k^\mathsf{T}\boldsymbol{\mathcal{H}}P_k$. Instead, what is needed is the matrix-vector product $P_k^\mathsf{T}\boldsymbol{\mathcal{H}}P_k X_k$ where $X_k \in \mathbb{R}^{r_{k-1}n_k r_k}$. For this, a splitting of the effective Hamiltonian is used and it is written

$$\left(P_k^\mathsf{T}\boldsymbol{\mathcal{H}}P_k\right)_{\alpha_{k-1}i_k\alpha_k}^{\beta_{k-1}j_k\beta_k} = \sum_{\nu_k=1}^{R_k} \left(\boldsymbol{\mathcal{L}_k}\right)_{\alpha_{k-1}i_k}^{\beta_{k-1}j_k\nu_k} \left(\boldsymbol{\mathcal{R}_k}\right)_{\alpha_k\nu_k}^{\beta_k}. \tag{5.2.9}$$

This splitting is illustrated in Figure 5.7.

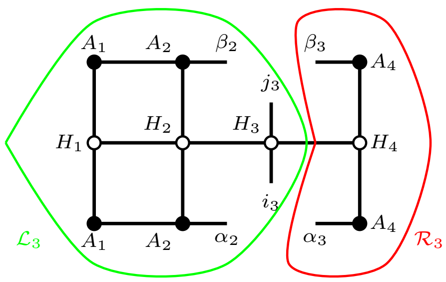


Figure 5.7: Splitting of the effective Hamiltonian

For iterative solvers, it is more relevant to focus on the computation of the matrix-vector multiplication. It goes as follows (see also Figure 5.8)

$$\left(P_k^\mathsf{T}\boldsymbol{\mathcal{H}}P_k\right)_{\alpha_{k-1}i_k\alpha_k}^{\beta_{k-1}j_k\beta_k}\left(X_k\right)_{\beta_{k-1}j_k\beta_k} = \left(\left(\boldsymbol{\mathcal{L}_k}\right)_{\alpha_{k-1}i_k}^{\beta_{k-1}j_k\nu_k}\left(X_k\right)_{\beta_{k-1}j_k\beta_k}\right)\left(\boldsymbol{\mathcal{R}_k}\right)_{\alpha_k\nu_k}^{\beta_k}, \tag{5.2.10}$$

i.e.

(i). first, we compute for $i_k \in [\![n_k]\!]$, $\nu_k \in [\![R_k]\!]$, $\alpha_{k-1} \in [\![r_{k-1}]\!]$, $\beta_k \in [\![r_k]\!]$ the sum

$$\sum_{\beta_{k-1}=1}^{r_{k-1}} \sum_{j_k=1}^{n_k} (\boldsymbol{\mathcal{L}_k})_{\alpha_{k-1} i_k}^{\beta_{k-1} j_k \nu_k} (X_k)_{\beta_{k-1} j_k \beta_k}.$$

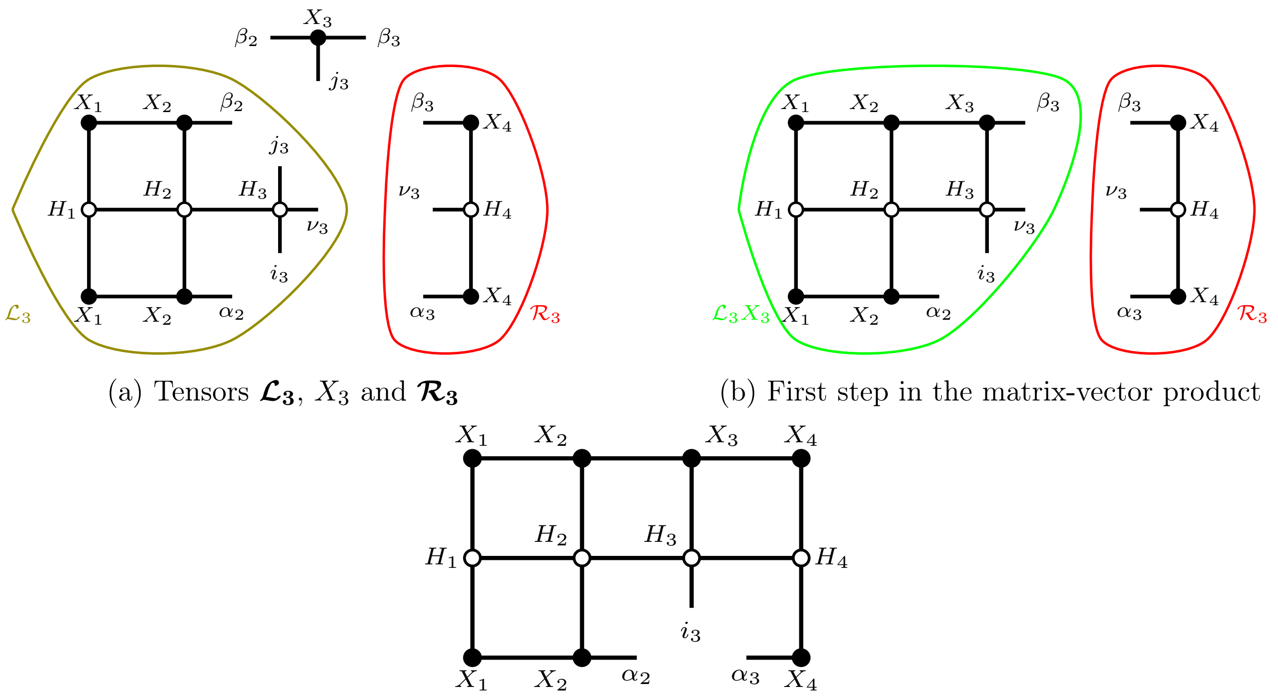
This scales as $\mathcal{O}(n^2 r^2 R)$.

(ii). in the second step, the previous tensor is contracted with $\boldsymbol{\mathcal{R}_k}$: for $\alpha_{k-1} \in [\![r_{k-1}]\!]$, $\alpha_k \in [\![r_k]\!]$, $i_k \in [\![n_k]\!]$, we sum

$$\sum_{\nu_k=1}^{R_k} \sum_{\beta_k=1}^{r_k} (\boldsymbol{\mathcal{L}_k} X_k)_{\alpha_{k-1} i_k \beta_k}^{\nu_k} (\boldsymbol{\mathcal{R}_k})_{\alpha_k \nu_k}^{\beta_k}$$

This scales as $\mathcal{O}(n r^3 R)$.

So overall the matrix-vector multiplication costs $\mathcal{O}(n^2 r^2 R + n r^3 R)$.

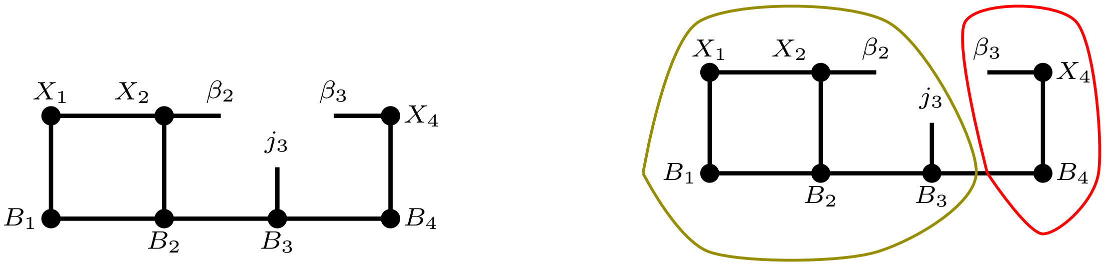


(a) Tensors $\boldsymbol{\mathcal{L}_3}$, X_3 and $\boldsymbol{\mathcal{R}_3}$

(b) First step in the matrix-vector product

(c) Result of the matrix-vector product

Figure 5.8: Matrix-vector product (5.2.10)



(a) $P_3^\mathsf{T}\boldsymbol{b}$ (b) Possible splitting of $P_3^\mathsf{T}\boldsymbol{b}$

Figure 5.9: Graphical representation of $P_3^\mathsf{T}\boldsymbol{b}$

The RHS $P_k^\mathsf{T}\boldsymbol{b}$ The assembly of the RHS is simpler than of the effective Hamiltonian. Similarly, it is possible to precompute the left and the right parts of the RHS as depicted in Figure 5.9b.

Operator updates The final main contribution to the total DMRG cost is the update of the effective Hamiltonians from one microstep to the next one. We would like to compute $\boldsymbol{\mathcal{L}_{k+1}}, \boldsymbol{\mathcal{R}_{k+1}}$ from $\boldsymbol{\mathcal{L}_k}, \boldsymbol{\mathcal{R}_k}$. In the case of a forward half-sweep, let us focus on the computation of $\boldsymbol{\mathcal{L}_{k+1}}$ from $\boldsymbol{\mathcal{L}_k}$. This can be done in 3 steps:

(i). as in the matrix-vector product we first compute for $i_k \in [\![n_k]\!], \nu_k \in [\![R_k]\!]$, $\alpha_{k-1} \in [\![r_{k-1}]\!], \beta_k \in [\![r_k]\!]$ the sum

$$\sum_{\beta_{k-1}=1}^{r_{k-1}} \sum_{j_k=1}^{n_k} (\boldsymbol{\mathcal{L}_k})_{\alpha_{k-1} i_k}^{\beta_{k-1} j_k \nu_k} (X_k)_{\beta_{k-1} j_k \beta_k}.$$

This scales as $\mathcal{O}(n^2 r^2 R)$.

(ii). we then contract the result of the operation above with X_k, so for each $\alpha_k \in [\![r_k]\!]$, $\nu_k \in [\![R_k]\!]$, we need to compute the following sum

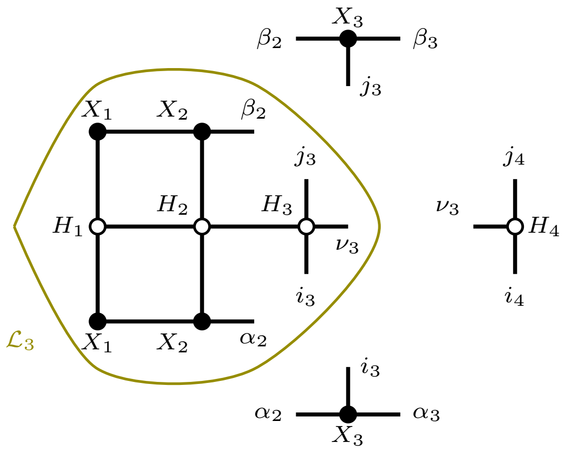
$$\sum_{\beta_{k-1}=1}^{r_{k-1}} \sum_{j_k=1}^{n_k} (\boldsymbol{\mathcal{L}_k} X_k)_{\alpha_{k-1} i_k \beta_k}^{\nu_k} (X_k)_{\alpha_{k-1} i_k \alpha_k}$$

This scales as $\mathcal{O}(n r^3 R)$.

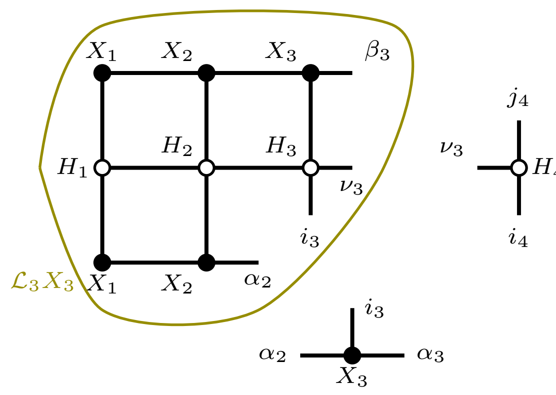
(iii). finally, once the previous step is performed, one needs to contract with TTO core H_{k+1}, so for each $i_{k+1}, j_{k+1} \in [\![n_{k+1}]\!]$, $\nu_{k+1} \in [\![R_{k+1}]\!]$, $\alpha_k, \beta_k \in [\![r_k]\!]$, the following sum is computed

$$\sum_{\beta_{k-1}=1}^{r_{k-1}} \sum_{j_k=1}^{n_k} (X_k^\mathsf{T} \boldsymbol{\mathcal{L}_k} X_k)_{\alpha_k \beta_k}^{\nu_k} (H_{k+1})_{\nu_k i_{k+1} j_{k+1} \nu_{k+1}}$$

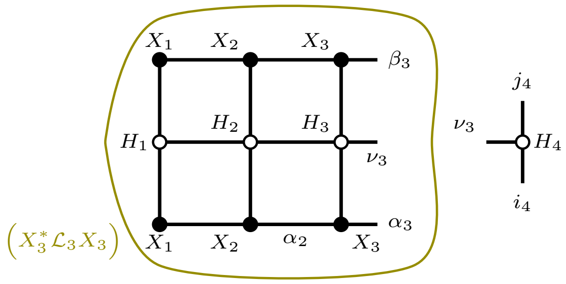
This scales as $\mathcal{O}(n^2 r^2 R^2)$.



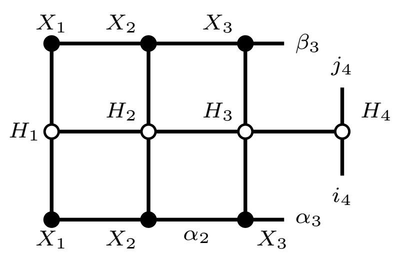
(a) Tensors $\mathcal{L}_3$, X_3 and $\mathcal{H}_4$



(b) First step in the update



(c) Second step in the update



(d) Third step in the update

Figure 5.10: Microstep operators updates

The cost of assembling $\boldsymbol{\mathcal{R}_k}$ from $\boldsymbol{\mathcal{R}_{k+1}}$ has the same scaling. The total cost of DMRG is summarised in the following Proposition.

Proposition 5.2.4 (Total cost of DMRG)**.** *The computational cost of DMRG scales as* $N_{\mathrm{sweep}} d((n^2 r^2 R + n r^3 R) N_{\mathrm{matvec}} + n^2 r^2 R^2)$*, where* N_{sweep} *is the number of total DMRG sweeps and* N_{matvec} *is the maximal number of matrix-vector products in all the microsteps.*

5.3 Convergence of DMRG

The global convergence of DMRG is a difficult problem, as the TT manifold is not a convex set. The convergence results on DMRG are local and assume that the Hessian of the functional j is of full-rank.

Assumption 5.3.1. *At the local minimiser* $\boldsymbol{x}_*$*, the Hessian* j'' *is of full rank*

$$\operatorname{rank} j''(\boldsymbol{x}_*) = \sum_{i=1}^{d} r_{i-1} n_i r_i - \sum_{i=1}^{d-1} r_i^2, \quad \textit{i.e.} \ \ker j''(\boldsymbol{x}_*) = T_{\boldsymbol{x}_*} \mathcal{M}_{\mathrm{TT}_{\leq r}}. \tag{5.3.1}$$

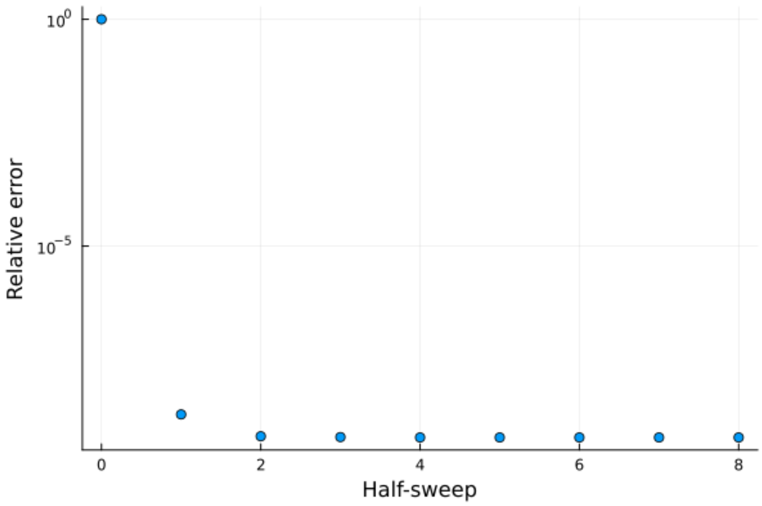


Figure 5.11: Convergence to the solution of $\boldsymbol{\mathcal{H}}\boldsymbol{x}_* = \boldsymbol{b}$ with $\boldsymbol{\mathcal{H}}$ the discrete Laplacian in $\mathbb{R}^{4^8 \times 4^8}$, $\boldsymbol{b}$ a random tensor of TT rank 2. The reference solution has TT rank 10.

5.3.1 Local convergence of DMRG

Assumption 5.3.1 ensures that the Hessian is invertible at the solution to the DMRG equations.

Theorem 5.3.2 ([RU13, Theorem 2.7]). *Let $\boldsymbol{x}_*$ be a local minimiser of the problem (5.2.3). There exists a neighbourhood $\mathrm{W} \subset \mathscr{M}_{\mathrm{TT}_{\leq \mathbf{r}}}$ in $\mathscr{M}_{\mathrm{TT}_{\leq \mathbf{r}}}$ of $\boldsymbol{x}_*$ such that Algorithm 5.1 initiated with $\boldsymbol{x}^{(0)} \in \mathrm{W}$ converges to the minimiser $\boldsymbol{x}_*$.*

5.3.2 Half-sweep convergence

A more surprising result states that if the TT ranks in the DMRG algorithm are exactly the TT ranks of the sought solution, then DMRG returns the *exact* solution in a half-sweep (see Figure 5.11).

This result is shown in the case of $\boldsymbol{\mathcal{H}} = \mathrm{id}$ in [HRS12a].

Proposition 5.3.3 ([HRS12a, Lemma 4.2]). *Let $\boldsymbol{b} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ with TT ranks $(r_0, \dots, r_d)$. Let $(B_1, \dots, B_d)$ be a left-orthogonal TT representation of $\boldsymbol{b}$. Let $(X_1, \dots, X_d)$ be a right-orthogonal TT with TT ranks $(r_0, \dots, r_d)$. Suppose that $(X_1, \dots, X_d)$ is such that for all $k \in [\![2; d]\!]$, the matrix $G_k \in \mathbb{R}^{r_{k-1} \times r_{k-1}}$ defined by*

$$\big(G_k\big)_{\beta_{k-1}\alpha_{k-1}} = \sum_{i_k,\dots,i_d} \sum_{\substack{\alpha_k \cdots \alpha_{d-1} \\ \beta_k \cdots \beta_{d-1}}} \big(X_k[i_k]\big)_{\alpha_{k-1}}^{\alpha_k} \cdots \big(X_d[i_d]\big)_{\alpha_{d-1}} \big(B_k[i_k]\big)_{\beta_{k-1}}^{\beta_k} \cdots \big(B_d[i_d]\big)_{\beta_{d-1}}.$$

is invertible. The DMRG algorithm applied with $H = \mathrm{id}$ converges in a half-sweep.

The condition on the initial guess is related to a nondeficiency of the initialisation of the DMRG algorithm.

Proof. We are going to prove by recurrence that there are $Q_k \in \mathbb{R}^{r_k \times r_k}$ for $k \in [\![d-1]\!]$ such that the solution of the DMRG microstep k can be written $X_k^{(\frac{1}{2})}[i_k] = Q_{k-1} B_k[i_k] Q_k^\mathsf{T}$.

Initialisation: since $(X_1, \dots, X_d)$ is right-orthogonal, we have that $P_1^\mathsf{T} P_1 = \mathrm{id}$. The solution to the first microstep is simply given by

$$Y_1[i_1]_{\alpha_1} = \sum_{\beta_1} B_1[i_1]_{\beta_1} (G_2)_{\alpha_1}^{\beta_1}.$$

Let Q_1^T, R_1 be the QR factorisation of G_2. Then

$$X_1^{(\frac{1}{2})}[i_1]_{\alpha_1} = \sum_{\beta_1} B_1[i_1]_{\beta_1} (Q_1)_{\beta_1}^{\alpha_1}.$$

Iteration: suppose that for all $j \in [\![k-1]\!]$, we have

$$X_j^{(\frac{1}{2})}[i_j]_{\alpha_j}^{\alpha_{j-1}} = \sum_{\beta_{j-1},\beta_j} (Q_{j-1})_{\beta_{j-1}}^{\alpha_{j-1}} (B_j[i_j])_{\beta_j}^{\beta_{j-1}} (Q_j)_{\beta_j}^{\alpha_j}.$$

At microstep k, by left-orthogonality of $(X_j^{(\frac{1}{2})})_{1 \le j \le k-1}$ and right-orthogonality of $(X_j)_{k+1 \le j \le d}$, again the solution to the microstep k is given by

$$Y_k[i_k]_{\alpha_k}^{\alpha_{k-1}} = \sum_{\substack{\alpha_1 \dots \alpha_{k-1} \\ \beta_1 \dots \beta_k}} B_1[i_1]_{\beta_1} \cdots B_k[i_k]_{\beta_k}^{\beta_{k-1}} X_1^{(\frac{1}{2})}[i_1]_{\alpha_1} \cdots X_{k-1}^{(\frac{1}{2})}[i_{k-1}]_{\alpha_{k-1}}^{\alpha_{k-2}} (G_{k+1})_{\alpha_k}^{\beta_k}.$$

By the recurrence hypothesis and the orthogonality of the TT cores $(B_j)_{1 \le j \le k-1}$, the above expression simplifies to

$$Y_k[i_k]_{\alpha_k}^{\alpha_{k-1}} = \sum_{\beta_{k-1},\beta_k} B_k[i_k]_{\beta_k}^{\beta_{k-1}} (Q_{k-1})_{\beta_{k-1}}^{\alpha_{k-1}} (G_{k+1})_{\alpha_k}^{\beta_k}.$$

Now let Q_k^T, R_k be the QR factorisation of G_{k+1}, then

$$X_k[i_k]_{\alpha_k}^{\alpha_{k-1}} = \sum_{\beta_{k-1},\beta_k} (Q_{k-1})_{\beta_{k-1}}^{\alpha_{k-1}} B_k[i_k]_{\beta_k}^{\beta_{k-1}} (Q_k)_{\beta_k}^{\alpha_k},$$

which is exactly $X_k[i_k] = Q_{k-1} B_k[i_k] Q_k^\mathsf{T}$. This finishes the proof of the proposition. □

Remark 5.3.4. *A similar result holds for tensor rings, see [CLL20].*

5.4 Two-site DMRG: how to dynamically adapt the TT ranks

The main limitation of the one-site DMRG algorithm is the inability to dynamically adapt the TT ranks of the approximate solution during the course of the iterations. A small modification of the one-site DMRG makes it possible to have more flexibility in the TT ranks. The main idea is to solve the microstep in DMRG not only on one-site but on two neighbouring sites.

In that case, at each microstep k, the functional that is minimised is

$$j_2^{(k)} : \begin{cases} \mathbb{R}^{n_1 \times r_0 \times r_1} \times \cdots \times \mathbb{R}^{r_{k-1} \times n_k \times n_k \times r_{k+1}} \times \cdots \times \mathbb{R}^{n_d \times r_{d-1} \times r_d} \to \mathbb{R} \\ (X_1, \dots, X_{k-1}, X, X_k, \dots, X_d) \mapsto J \circ \widetilde{\mathrm{TT}}_k(X_1, \dots, X_{k-1}, X, X_k, \dots, X_d) \end{cases} \tag{5.4.1}$$

where

$$\widetilde{\mathrm{TT}}_k : \begin{cases} \mathbb{R}^{n_1 \times r_0 \times r_1} \times \cdots \times \mathbb{R}^{r_{k-1} \times n_k \times n_k \times r_{k+1}} \times \cdots \times \mathbb{R}^{n_d \times r_{d-1} \times r_d} \to \mathbb{R}^{n_1 \cdots n_d} \\ (X_1, \dots, X_{k-1}, X, X_{k+1}, \dots, X_d) \mapsto (X_1[i_1] \cdots X_{k-1}[i_{k-1}] X[i_k, i_{k+1}] X_{k+1}[i_{k+1}] \cdots X_d[i_d]). \end{cases}$$

The TT rank adaptivity comes in the transformation of the microstep solution back to a suitable TT form by a truncated SVD

$$(X_{\alpha_{k-1} i_k}^{\alpha_{k+1} i_{k+1}}) = U_\varepsilon S_\varepsilon V_\varepsilon^\mathsf{T} + \mathcal{O}(\varepsilon),$$

where $U_\varepsilon \in \mathbb{R}^{r_{k-1} n_k \times r}$, $S_\varepsilon \in \mathbb{R}^{r \times r}$ and $V_\varepsilon \in \mathbb{R}^{r_{k+1} n_{k+1} \times r}$ and r is chosen such that the truncated SVD has error ε. U_ε is (up to a reshape) the new TT core X_k and r is the corresponding TT rank. The full algorithm is given in Algorithm 5.2.

In practice, the truncation level ε is used to monitor the error in DMRG. It can also be used to extrapolate some quantities as the lowest eigenvalue as depicted in Figure 5.12 [WPAV14]. Theoretically, unlike the one-site algorithm, there is no convergence result on the two-site algorithm (except in the case where no truncation is made).

5.5 DMRG on eigenvalue problems

DMRG is primarily used to solve eigenvalue problems. In that case, the functional to minimise is $J(x) = \frac{\langle x, Hx \rangle}{\langle x, x \rangle}$. At each microstep, instead of solving a linear system, the following generalised eigenvalue problem has to be solved for the lowest eigenvalue

$$P_k^\mathsf{T} \boldsymbol{\mathcal{H}} P_k V = \lambda P_k^\mathsf{T} P_k V.$$

In that case, it is numerically beneficial to ensure the good orthogonality conditions for the approximate solution in the TT form, so that $P_k^\mathsf{T} P_k = \mathrm{id}$.

Apart from this change, the algorithms 5.1 and 5.2 can be modified in a straightforward way to solve eigenvalue problems instead.

For multiple lowest eigenvalues, there are two main options

Algorithm 5.2 Two-site DMRG with sweeps

Input: $(X_1^{(0)}, \dots, X_d^{(0)})$ in right-orthogonal TT representation with initial TT ranks $(r_0^{(0)}, \dots, r_d^{(0)})$, $\varepsilon_{\mathrm{TT}}$

Output: $(X_1^{(s)}, \dots, X_d^{(s)}) \in \mathscr{M}_{\mathrm{TT}_{\leq \mathbf{r}}}$ approximation of the minimiser of J

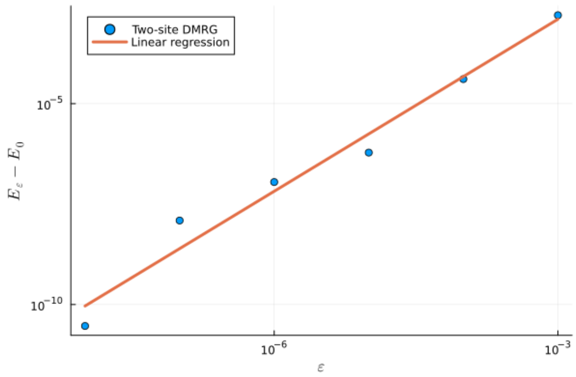
function TWO-SITE-DMRG$((X_1^{(0)}, \dots, X_d^{(0)}), \varepsilon_{\mathrm{TT}})$
  $s = 0$
  while `not converged` **do**
    for $k = 1, 2, \dots, d-2$ **do** ▷ Forward half-sweep

$$Y_k^{(s+\frac{1}{2})} = \underset{V_k \in \mathbb{R}^{r_{k-1}^{(s+\frac{1}{2})} \times n_k \times n_k \times r_{k+1}^{(s)}}}{\arg\min} j_2^{(k)}(X_1^{(s+\frac{1}{2})}, \dots, X_{k-1}^{(s+\frac{1}{2})}, V_k, X_{k+2}^{(s)}, \dots, X_d^{(s)}) \tag{5.4.2}$$

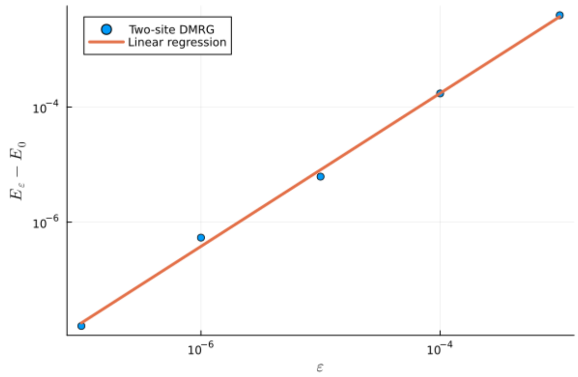
      $U, S, V^{\mathsf{T}} = \mathtt{svd}_{\varepsilon_{\mathrm{TT}}}\big(\big(Y_k^{(s+\frac{1}{2})}\big)_{\alpha_{k-1} i_k}^{\beta_{k+1} i_{k+1}}\big)$ ▷ Truncated SVD of Y_k
      $r_k^{(s+\frac{1}{2})}$ = rank of the SVD truncation to level $\varepsilon_{\mathrm{TT}}$ ▷ Update TT rank
      $\big(X_k^{(s+\frac{1}{2})}[i_k]\big)_{\alpha_{k-1}}^{\alpha_k} = U_{\alpha_{k-1} i_k}^{\alpha_k}$ ▷ Update X_k
      $\big(X_{k+1}^{(s)}[i_{k+1}]\big)_{\alpha_k}^{\alpha_{k+1}} = \big(SV^{\mathsf{T}}\big)_{\alpha_k}^{\alpha_{k+1} i_{k+1}}$ ▷ Update X_{k+1}
    end for
    for $k = d-1, d-2, \dots, 2$ **do** ▷ Backward half-sweep

$$Y_k^{(s+1)} = \underset{V_k \in \mathbb{R}^{r_{k-2}^{(s+\frac{1}{2})} \times n_k \times n_k \times r_k^{(s+1)}}}{\arg\min} j_2^{(k)}(X_1^{(s+\frac{1}{2})}, \dots, X_{k-1}^{(s+\frac{1}{2})}, V_k, X_{k+2}^{(s+1)}, \dots, X_d^{(s+1)}) \tag{5.4.3}$$

      $U, S, V^{\mathsf{T}} = \mathtt{svd}_{\varepsilon_{\mathrm{TT}}}\big(\big(Y_k^{(s+\frac{1}{2})}\big)_{\alpha_{k-1} i_k}^{\beta_{k+1} i_{k+1}}\big)$ ▷ Truncated SVD of Y_k
      $r_k^{(s+1)}$ = rank of the SVD truncation to level $\varepsilon_{\mathrm{TT}}$ ▷ Update TT rank
      $\big(X_{k+1}^{(s+1)}[i_{k+1}]\big)_{\alpha_k}^{\alpha_{k+1}} = V_{\alpha_{k+1} i_{k+1}}^{\alpha_k}$ ▷ Update X_{k+1}
      $\big(X_k^{(s)}[i_k]\big)_{\alpha_{k-1}}^{\alpha_k} = \big(US\big)_{\alpha_{k-1} i_k}^{\alpha_k}$ ▷ Update X_k
    end for
    $s = s + 1$
  end while
  return $(X_1^{(s)}, \dots, X_d^{(s)})$
end function



(a) Hubbard model with 10 sites



(b) Hubbard model with 20 sites

Figure 5.12: Extrapolation of the ground-state energy E_ε for the Hubbard model where E_ε is computed with the two-site DMRG algorithm with truncation ε

(i). deflate the computed eigenvalues

(ii). use the following characterisation of the k smallest eigenvalues $(\lambda_1, \dots, \lambda_k)$ of $\boldsymbol{\mathcal{H}}$

$$\sum_{i=1}^{k} \lambda_i = \min_{\boldsymbol{X} \in \mathbb{R}^{n_1 \cdots n_d \times k}} \frac{\operatorname{Tr}(\boldsymbol{X}^\mathsf{T} \boldsymbol{\mathcal{H}} \boldsymbol{X})}{\operatorname{Tr}(\boldsymbol{X}^\mathsf{T} \boldsymbol{X})}.$$

This approach is described in [DKOS14]. Essentially, the TT representing $\boldsymbol{X} \in \mathbb{R}^{n_1 \cdots n_d \times k}$ has an extra index accounting for the number of eigenvalues sought. At each microstep, this index is "moved" to the next microstep during the QR/SVD step.

Bibliography

[AKLV13] Itai Arad, Alexei Kitaev, Zeph Landau, and Umesh Vazirani. An area law and sub-exponential algorithm for 1d systems. *arXiv preprint arXiv:1301.1162*, 2013.

[AL70] Huzihiro Araki and Elliott H. Lieb. Entropy inequalities. *Comm. Math. Phys.*, 18:160–170, 1970.

[Ali21] Mazen Ali. On the ordering of sites in the density matrix renormalization group using quantum mutual information, 2021.

[BGP22] Markus Bachmayr, Michael Götte, and Max Pfeffer. Particle number conservation and block structures in matrix product states. *Calcolo*, 59(2):24, 2022.

[BLMR11] G. Barcza, Ö. Legeza, K. H. Marti, and M. Reiher. Quantum-information analysis of electronic states of different molecular structures. *Phys. Rev. A*, 83:012508, Jan 2011.

[BSU16] Markus Bachmayr, Reinhold Schneider, and André Uschmajew. Tensor networks and hierarchical tensors for the solution of high-dimensional partial differential equations. *Found. Comput. Math.*, 16(6):1423–1472, 2016.

[CHG02] Garnet Kin-Lic Chan and Martin Head-Gordon. Highly correlated calculations with a polynomial cost algorithm: A study of the density matrix renormalization group. *The Journal of Chemical Physics*, 116(11):4462–4476, 2002.

[CKN+16] Garnet Kin Chan, Anna Keselman, Naoki Nakatani, Zhendong Li, and Steven R White. Matrix product operators, matrix product states, and ab initio density matrix renormalization group algorithms. *The Journal of chemical physics*, 145(1), 2016.

[CL14] Eric A. Carlen and Elliott H. Lieb. Remainder terms for some quantum entropy inequalities. *J. Math. Phys.*, 55(4):042201, 5, 2014.

[CLL20] Ziang Chen, Yingzhou Li, and Jianfeng Lu. Tensor ring decomposition: optimization landscape and one-loop convergence of alternating least squares. *SIAM Journal on Matrix Analysis and Applications*, 41(3):1416–1442, 2020.

[DKOS14] Sergey V. Dolgov, Boris N. Khoromskij, Ivan V. Oseledets, and Dmitry V. Savostyanov. Computation of extreme eigenvalues in higher dimensions using block tensor train format. *Computer Physics Communications*, 185(4):1207–1216, April 2014. arXiv:1306.2269 [cond-mat].

[DMNS98] Dukelsky, J., Martín-Delgado, M. A., Nishino, T., and Sierra, G. Equivalence of the variational matrix product method and the density matrix renormalization group applied to spin chains. *Europhys. Lett.*, 43(4):457–462, 1998.

[DSL08] Vin De Silva and Lek-Heng Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1084–1127, 2008.

[FKK+18] Leon Freitag, Sebastian Keller, Stefan Knecht, Yingjin Ma, Christopher Stein, and Markus Reiher. A quick user guide to the qcmaquis software suite for openmolcas. 2018.

[Gra10] Lars Grasedyck. Hierarchical singular value decomposition of tensors. *SIAM J. Matrix Anal. Appl.*, 31(4):2029–2054, 2009/10.

[Hac12] Wolfgang Hackbusch. *Tensor spaces and numerical tensor calculus*, volume 42 of *Springer Series in Computational Mathematics*. Springer, Heidelberg, 2012.

[Hac14] Wolfgang Hackbusch. Numerical tensor calculus. *Acta Numer.*, 23:651–742, 2014.

[Has07] Matthew B Hastings. An area law for one-dimensional quantum systems. *Journal of statistical mechanics: theory and experiment*, 2007(08):P08024, 2007.

[HJO14] Trygve Helgaker, Poul Jorgensen, and Jeppe Olsen. *Molecular electronic-structure theory*. John Wiley & Sons, 2014.

[HRS12a] Sebastian Holtz, Thorsten Rohwedder, and Reinhold Schneider. The Alternating Linear Scheme for Tensor Optimization in the Tensor Train Format. *SIAM Journal on Scientific Computing*, 34(2):A683–A713, January 2012.

[HRS12b] Sebastian Holtz, Thorsten Rohwedder, and Reinhold Schneider. On manifolds of tensors of fixed TT-rank. *Numer. Math.*, 120(4):701–731, 2012.

[KSZ91] A Klumper, A Schadschneider, and J Zittartz. Equivalence and solution of anisotropic spin-1 models and generalized t-j fermion models in one dimension. *Journal of Physics A: Mathematical and General*, 24(16):L955–L959, aug 1991.

[LMW13] Noah Linden, Milán Mosonyi, and Andreas Winter. The structure of rényi entropic inequalities. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 469(2158):20120737, 2013.

[LOV15] Christian Lubich, Ivan V. Oseledets, and Bart Vandereycken. Time integration of tensor trains. *SIAM Journal on Numerical Analysis*, 53(2):917–941, January 2015.

[LRH03] Ö. Legeza, J. Röder, and B. A. Hess. Controlling the accuracy of the density-matrix renormalization-group method: The dynamical block state selection approach. *Phys. Rev. B*, 67:125114, Mar 2003.

[NS06] Bruno Nachtergaele and Robert Sims. Lieb-robinson bounds and the exponential clustering theorem. *Communications in mathematical physics*, 265:119–130, 2006.

[OR95] Stellan Östlund and Stefan Rommer. Thermodynamic limit of density matrix renormalization. *Phys. Rev. Lett.*, 75:3537–3540, Nov 1995.

[Ose11] Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[OT09] I. V. Oseledets and E. E. Tyrtyshnikov. Breaking the curse of dimensionality, or how to use SVD in many dimensions. *SIAM J. Sci. Comput.*, 31(5):3744–3759, 2009.

[RNW06] Jörg Rissler, Reinhard M Noack, and Steven R White. Measuring orbital interaction using quantum information theory. *Chemical Physics*, 323(2-3):519–531, 2006.

[RS75] Michael Reed and Barry Simon. Methods of modern mathematical physics. II: Fourier analysis, self- adjointness. New York - San Francisco - London: Academic Press, a subsidiary of Harcourt Brace Jovanovich, Publishers. XV, 361 p. $ 24.50; £ 11.75 (1975)., 1975.

[RU13] Thorsten Rohwedder and André Uschmajew. On Local Convergence of Alternating Schemes for Optimization of Convex Problems in the Tensor Train Format. *SIAM Journal on Numerical Analysis*, 51(2):1134–1162, January 2013.

[Sch08] Erhard Schmidt. On the theory of linear and nonlinear integral equations. III: On the solution of nonlinear integral equations and their bifurcations. *Math. Ann.*, 65:370–399, 1908.

[Sch11] Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Ann. Physics*, 326(1):96–192, 2011.

[SWVC08] Norbert Schuch, Michael M Wolf, Frank Verstraete, and J Ignacio Cirac. Entropy scaling and simulability by matrix product states. *Physical review letters*, 100(3):030504, 2008.

[T$^+$15] Joel A Tropp et al. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

[UV20] André Uschmajew and Bart Vandereycken. Geometric methods on low-rank matrix and tensor manifolds. In *Handbook of variational methods for nonlinear geometric data*, pages 261–313. Cham: Springer, 2020.

[VC04] F. Verstraete and J. I. Cirac. Renormalization algorithms for quantum-many body systems in two and higher dimensions. (arXiv:cond-mat/0407066), 2004. arXiv:cond-mat/0407066.

[VC06] F. Verstraete and J. I. Cirac. Matrix product states represent ground states faithfully. *Phys. Rev. B*, 73:094423, Mar 2006.

[Vid03] Guifré Vidal. Efficient classical simulation of slightly entangled quantum computations. *Phys. Rev. Lett.*, 91:147902, Oct 2003.

[Whi92] Steven R White. Density matrix formulation for quantum renormalization groups. *Physical review letters*, 69(19):2863, 1992.

[WPAV14] Sebastian Wouters, Ward Poelmans, Paul W. Ayers, and Dimitri Van Neck. CheMPS2: a free open-source spin-adapted implementation of the density matrix renormalization group for ab initio quantum chemistry. *Computer Physics Communications*, 185(6):1501–1514, 2014.