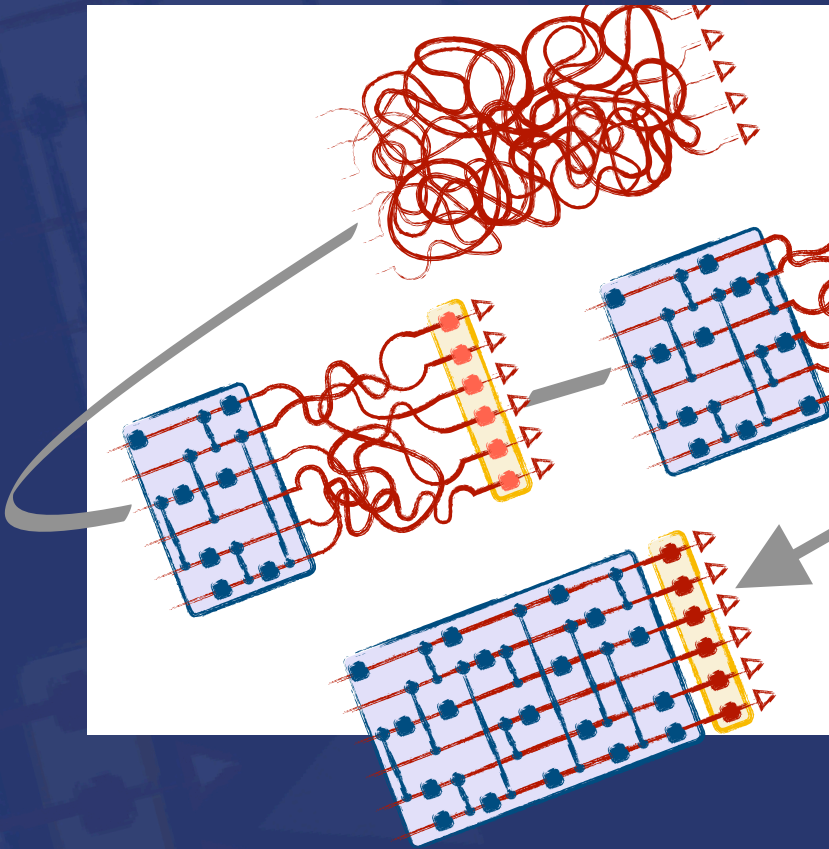


1

Tensor Network Techniques for Quantum Computation

*Mario Collura, Guglielmo Lami,
Nishan Ranabhat, Alessandro Santini*





This page was intentionally left blank.



This page was intentionally left blank.

Tensor Network Techniques for Quantum Computation

Mario Collura · Guglielmo Lami

Nishan Ranabhat · Alessandro Santini

Published by SISSA Medialab S.r.l.

Via Bonomea 265

34136 Trieste, Italy

<https://medialab.sissa.it/>

Cover: Giacomo Sanna — Dotik

Typesetting: Elia A. Calderan and Giorgia del Bianco — SISSA Medialab S.r.l.



This book is distributed under the terms of the Creative Commons Attribution License ([CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)), which permits any use, distribution and reproduction in any medium, provided the original author(s) and source are credited.

Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

The third party material in this book are not included in the Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the book's Creative Commons license, you will need to obtain permission directly from the copyright holder.

© 2024 Mario Collura, Guglielmo Lami, Nishan Ranabhat, Alessandro Santini



This eBook was published Open Access with funding support from the Sponsoring Consortium for Open Access Publishing in Particle Physics (SCOAP3).

Title: Tensor Network Techniques for Quantum Computation

Authors: Mario Collura, Guglielmo Lami, Nishan Ranabhat, Alessandro Santini

Keywords: 1. Tensor Networks 2. Matrix Product States (MPS) 3. Matrix Product Operators (MPO) 4. Quantum Computing 5. Hamiltonian Dynamics 6. Time-Dependent Variational Principle (TDVP) 7. Entanglement 8. Quantum Circuits 9. Quantum Annealing 10. Open Quantum Systems 11. Stabilizer Formalism 12. Clifford Group and Clifford-Dressed TDVP 13. Nonstabilizerness and Quantum Resource Theory

First published 2024

ISBN: 9788898587049 (electronic edition)

DOI: [10.22323/9788898587049](https://doi.org/10.22323/9788898587049)

I am withdrawing to write a book.
And another time: I am withdrawing
to construct a labyrinth.

*“Ts’ui Pên” in The Garden of Forking
Paths by Jorge Luis Borges*

Preface

The last decade has seen a revolutionary shift in the overlap between quantum information science and the study of quantum many-body systems, two fields that traditionally developed along separate tracks. This book, designed for undergraduate students and early researchers, builds a bridge between these disciplines by exploiting tensor networks—a powerful computational framework with deep relevance for both many-body physics and quantum computation.

Tensor networks emerged initially as a way to represent complex quantum states compactly in many-body physics. However, as quantum computing has advanced, tensor networks have also become indispensable tools for understanding and simulating quantum systems. They allow us to visualize and capture intricate entanglement structures and to represent states in a way that is both computationally manageable and mathematically rich. As a result, tensor networks offer a critical foundation for tackling challenges in quantum computation, where state complexity, entanglement, and even more elusive aspects of quantum mechanics—like quantum magic—play key roles.

The book begins with an introduction to fundamental concepts, presenting tensor networks as efficient tools for representing and manipulating quantum states. We cover the basics of matrix product states (MPS), the foundational building blocks of tensor network theory quantum circuits, which allow us to approximate low-entanglement quantum states with minimal computational resources. We examine how they allow us to quantify and manipulate entanglement and delve into topics that go beyond entanglement alone, including recent advances in quantifying non-stabilizer resources—referred to as quantum magic—which are vital for achieving quantum speedup. This book discusses the implications of these non-stabilizer resources in the context of stabilizer circuits and Clifford transformations, offering insight into how tensor networks can represent both stabilizer and non-stabilizer states within the same framework. Beside that, we revisit well known techniques like the Time-Dependent Variational Principle (TDVP) and Matrix Product Operator

(MPO) based Lindblad dynamics, in order to give the proper background to advance into new methods which combine tensor networks with Clifford stabilizer formalism.

Furthermore, the book explores the connection between tensor networks and classical simulations of quantum circuits, highlighting the power of tensor networks to simulate complex quantum algorithms. We delve into the current limitations and challenges associated with scaling these simulations and discuss how advances in tensor network methods might overcome these hurdles in the future.

Our goal with this book is not to provide an exhaustive review of tensor networks or quantum computing, but rather to present these ideas in a pedagogical, hands-on manner, using accessible language and representative examples to illuminate complex topics. By emphasizing recent developments in tensor network methods for studying quantum complexity, quantum magic, and non-stabilizerness, we aim to provide readers with a toolkit for exploring the cutting-edge intersections of quantum computing and quantum many-body physics.

It is our hope that this book will inspire readers to further explore the interface between quantum information science and many-body physics, equipping them with the theoretical insights and practical tools needed to contribute to this rapidly evolving field.

The Authors

Acknowledgments

We are beholden to numerous colleagues and students for their invaluable insights and suggestions. In particular, we gratefully acknowledge Pasquale Calabrese and Marcello Dalmonte for their critical reading of parts of the manuscript. Obviously, we take full responsibility for any remaining errors or oversights, and no blame should fall on the individuals mentioned above.

Contents

List of Definitions	ix
List of Examples	xi
I Preliminaries	xiii
1 Tensor Network Basics	5
1.1 Tensors	5
1.1.1 Special Tensors	9
1.2 Tensor Networks	14
1.3 Tensor Network Decomposition	18
1.3.1 Matrix product tensor network	20
1.3.2 Tree tensor network (TTN)	22
2 Quantum Physics with Tensors	25
2.1 Introduction to quantum mechanics	25
2.1.1 States and observables	25
2.1.2 Evolution	27
2.1.3 Measurements	29
2.1.4 Composite systems	30
2.2 Quantum mechanics of a single qubit	32
2.2.1 Bloch sphere	33
2.3 Many-body quantum systems and Entanglement	35
2.3.1 Universal set of gates	36
2.3.2 Schmidt decomposition	37
2.3.3 Entanglement	38
2.4 Tensor network representation of quantum states	45
2.4.1 Matrix Product States (MPS)	47
2.4.2 Matrix Product Operator (MPO)	54

3	Quantum computing with Tensor Networks	63
3.1	State preparation	63
3.1.1	Product state	65
3.1.2	MPS to quantum circuit	65
3.2	Quantum circuits	69
3.3	Measurements	73
3.4	Quantum Platforms	76
3.4.1	Superconducting Qubits	77
3.4.2	Neutral-Atoms Qubits	78
3.4.3	Trapped-Ions Qubits	81
II	Applications	83
4	Unitary Evolution of Hamiltonian Dynamics	89
4.1	Time-Dependent Variational Principle (TDVP)	90
4.2	Quadratic unconstrained binary optimization	99
4.2.1	Mapping to Ising model	103
4.3	Quantum Annealing	104
4.3.1	Digitized Quantum Annealing (dQA)	105
4.3.2	dQA with Tensor Network	106
4.3.3	Quantum approximate optimization algorithm	112
5	Open tensor network	115
5.1	Open system recap	116
5.1.1	Tensor Network and Density Operators	123
5.2	Minimally Entangled Typical Thermal State	126
5.2.1	Classical vs Quantum Typical Sample	126
5.2.2	METTS with Tensor Network	129
5.3	Lindblad dynamics	133
5.3.1	MPO formalism	135
5.3.2	Locally Purified Tensor Network	140
5.4	Unravelling and monitored dynamics	145
5.4.1	Measurement protocols	145
6	Tensor networks and quantum magic	151
6.1	Introduction to Stabilizer Formalism and Magic	152
6.1.1	Pauli group and stabilizer groups	152
6.1.2	The Clifford group	154
6.1.3	The tableau representation	156
6.1.4	Entanglement	159

6.1.5	Nonstabilizerness as a quantum resource	159
6.1.6	Stabilizer Rényi Entropies	161
6.2	Measuring the magic of Tensor Network states	163
6.2.1	Replica methods	163
6.2.2	Sampling approach	167
6.3	Clifford enhanced matrix product states (cMPS)	173
6.3.1	Hybrid stabilizer MPO	173
6.3.2	Clifford-Dressed TDVP for Hamiltonian Dynamics . .	178

Index **187**

List of Definitions

1.1	Tensor	6
1.2	Tensor order	6
1.3	Tensor dimension	7
2.1	Bipartite product state	38
2.2	von Neumann entropy	42
2.3	Entanglement entropy	43
2.4	Mutual information	43
2.5	Local Operations and Classical Communication	44
2.6	Negativity	45
2.7	Area law of bipartite entanglement entropy	46
2.8	Matrix Product State	48
2.9	MPS Transfer Matrix	49
5.1	Statistical Mixture	116
5.2	normalized Pauli Tensor	125
5.3	Positive Map	134
5.4	Completely Positive Map	134
6.1	Pauli group	152
6.2	Stabilizers	153
6.3	Stabilizer projector	153
6.4	Clifford group	154
6.5	Generator matrix	157
6.6	Projective Pauli group	161
6.7	Stabilizer Rényi Entropies	161

List of Examples

1.1	Glimpse on Quantum Computing	10
1.2	Non-local swapping	12
1.3	Contraction order do matter	16
1.4	Left canonical matrix product decomposition	21
1.5	Tree tensor network decomposition	22
2.1	Purification	31
2.2	Decomposition of n qubits unitaries	36
2.3	Bell states, Bell measurements and quantum teleportation	39
2.4	Multiply MPO to MPO or MPS	55
3.1	GHZ state as an MPS	68
3.2	W state as an MPS	69
3.3	Local unitary updates of MPS with TEBD	71
3.4	Implementation of the CZ and CNOT gates	80
4.1	Lanczos Exponential Solver	94
4.2	TDVP algorithm	95
4.3	Variational compression of a MPS	110
5.1	Exact MPO for classical thermal states	119
5.2	Vectorization of matrices	121
5.3	How METTS guarantees the correct distribution	129
5.4	One-Body particle loss and gain in hard-core bosons	138
6.1	Replica Pauli-MPS	165
6.2	Stabilizer group of an MPS	171
6.3	Implementation of the stabilizer MPO	174

Part I

Preliminaries

Tensor network methods encompass a set of strategies designed to comprehend and analyze multi-linear maps, proving particularly valuable in the realm of quantum computation and information processing, as well as in quantum many-body physics. These techniques serve as the foundation for tensor network contraction algorithms, essential for modeling physical systems. Employed within abstract graphical frameworks, these methods proficiently depict channels, maps, states, and algorithms applicable across various domains within the field of quantum physics.

While tensor networks span various subjects, existing literature tends to be highly specific, often catering to a narrow community. Here we take the typical approach of tensor network for many-body physics - by offering a instructive exploration of the foundational principles of tensor network theory - exploiting its broad applicability in the realm of quantum computation and information.

The chapters presented in this section serves as the essential foundation for understanding the core concepts and mathematical tools required to explore the applications of tensor networks in quantum physics and quantum computing. This part introduces the reader to the building blocks of tensor network theory, starting with the basic concept of tensors and expanding into more complex tensor network structures that form the backbone of modern computational methods in many-body quantum systems.

Tensors and Tensor Networks — At the heart of this discussion are *tensors*, which generalize matrices to higher dimensions. Tensors represent multi-indexed arrays that serve as the fundamental objects for encoding and manipulating data in various dimensions. In quantum physics, they play a critical role in simplifying and representing complex multi-particle quantum states. The section on “Special Tensors” explores key forms such as *identity tensors* and *Kronecker delta tensors*, which have specific properties essential for simplifying tensor operations.

Building on this, the book introduces *tensor networks*, a graphical formalism to represent complex tensors by breaking them down into smaller components. This decomposition of large tensors into interconnected networks enables computational efficiency, especially for simulating large quantum systems. A deep dive into specific tensor network architectures, such as *Matrix Product States (MPS)* and *Tree Tensor Networks (TTN)*, demonstrates their wide-ranging applicability to problems in quantum physics and quantum computation.

Quantum Physics with Tensors — To establish the relevance of tensor networks to quantum mechanics, the next section provides an overview of quantum physics, including its foundational concepts like *states*, *observables*, and *measurements*. The representation of quantum systems in terms of tensors naturally arises here. For instance, *quantum states*, especially in many-body systems, are often expressed using tensor networks, which significantly reduce the complexity of handling these states.

Matrix Product States (MPS) and *Matrix Product Operators (MPO)* are key methods discussed in this section. They are shown to efficiently represent quantum states and operations, even for systems with large numbers of qubits. Moreover, *entanglement* — a hallmark feature of quantum systems — naturally emerges in tensor network representations. In this section we exploit the *Schmidt decomposition* and other key tools for understanding and quantifying entanglement in many-body systems.

Quantum Computing with Tensor Networks — The preliminaries culminate with an exploration of how tensor networks can be leveraged for *quantum computing*. As quantum circuits grow more complex, traditional methods become computationally expensive. Tensor networks, particularly in their MPS form, offer a powerful way to simulate quantum circuits and manipulate quantum data. Topics like *state preparation*, *quantum gates*, and *measurements* are discussed in detail, demonstrating how tensor networks streamline these processes, particularly in noisy intermediate-scale quantum (NISQ) devices.

Additionally, the section highlights real-world implementations of these techniques across various *quantum platforms*, such as *superconducting qubits*, *neutral-atom qubits*, and *trapped-ion qubits*. Each of these platforms provides distinct advantages and challenges, and tensor network methods are presented as valuable tools for overcoming the computational challenges associated with simulating these systems.

In conclusion, the “Preliminaries” section lays the groundwork for the more advanced applications that follow, by equipping the reader with both the theoretical knowledge and practical tools needed to navigate the landscape of quantum physics and quantum computation using tensor networks.

Chapter 1

Tensor Network Basics

Everything should be made as simple as possible, but not simpler.

Albert Einstein

Tensor Network (TN) methods often refer to a comprehensive set of tools commonly utilized in contemporary quantum information science, condensed matter physics, mathematics, and computer science. Essentially, these methods constitute a set of techniques employed to systematically organize and manipulate vast numerical datasets arranged in multidimensional arrays, also known as tensors, interconnected to form a network [Bia20; Ran+20; Eve22]. These methods allow for a simple and appealing diagrammatic notation that facilitates the understanding of complex linear algebra operations in a succinct fashion, making them well suited for modern computational devices. In this section, we present a formal introduction to tensors, tensor networks, and essential operations associated with tensor networks. Let us start from the fundamental building block of any tensor network, namely, a tensor.

1.1 Tensors

In the realm of mathematics, tensors serve as a powerful algebraic entity that encodes a multi-linear relationship among sets of algebraic objects associated with a vector space. In practice, a generic tensor is a scalar-valued function of multiple parameters which are linear with respect to each other. Remarkably, tensors transcend specific bases, asserting their definition independent of any particular basis. Formally we have the following definition:

Definition 1.1: Tensor

Let V is a vectorial space with dimension d over the complex numbers \mathbb{C} , i.e. $V \simeq \mathbb{C}^d$. The dual space V^* is the vector space defined as the set of all linear map $\varphi : V \rightarrow \mathbb{C}$. It has dimension d as well. Its element are called *covectors*. A type (p, q) **tensor** is therefore a multi-linear map

$$T : \underbrace{V^* \times \dots \times V^*}_p \times \underbrace{V \times \dots \times V}_q \rightarrow \mathbb{C}$$

Therefore, a tensor T associates q vectors $\{v_1, \dots, v_q\}$ and p covectors $\{w_1, \dots, w_p\}$ a scalar $T(w_1, \dots, w_p, v_1, \dots, v_q)$.

However, in practical applications, tensors often find representation through their components within a basis tied to a specific coordinate system. These components assemble into an array, akin to a high-dimensional matrix, facilitating an easier understanding.

For example, following the Dirac convection of modern quantum physics, we may choose the Hilbert space $\mathcal{H} \simeq \mathbb{C}^d$ as the typical vector space wherein tensors act as multi-linear map,¹ with standard computational basis $\{|j\rangle : j = 0, \dots, d - 1\}$. In this context, a vector $|\psi\rangle \in \mathcal{H}$ is essentially an order-1 tensor, which we may express in terms of its tensor components $\psi_j = \langle j|\psi\rangle$ with respect to the computational basis, such that $|\psi\rangle = \sum_j \psi_j |j\rangle$. In strict analogy, we introduce the standard covectors of the dual space \mathcal{H}^* whose basis is $\{\langle j| : j = 0, \dots, d - 1\}$. Linear operators $\hat{O} \in \mathcal{H} \times \mathcal{H}^*$ can be represented as order-2 tensors with components $O_{ij} = \langle i|\hat{O}|j\rangle$, such that $\hat{O} = \sum_{i,j} O_{ij} |i\rangle\langle j|$. Here we implicitly introduced the concept of *order* of a tensor as

Definition 1.2: Tensor order

We define the **order of a tensor** as the number of indices it possesses, no matter whether the indices are spanning over the vectors or the covectors.

With this respect, and after having introduced a suitable basis, tensor serves as a generalization encompassing both vectors and matrices. Specifically, a generic order- k tensor is portrayed as a complex multidimensional array denoted by $T_{j_1, \dots, j_k} \in \mathbb{C}^{d_1} \times \dots \times \mathbb{C}^{d_k}$. Each index j_i for $i \in \{1, \dots, k\}$, colloquially referred to as a *leg* of the tensor, takes values from the corresponding set

¹We refer the reader to the next chapter to a tighter connection with quantum physics and quantum computing.

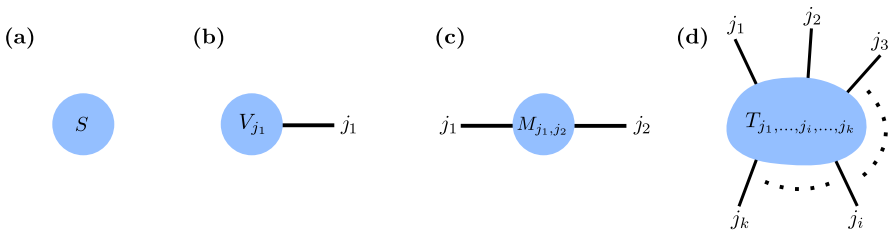
$\{1, 2, \dots, d_i\}$. In this context, d_i represents the dimension associated with the index j_i , and the overall dimension of the entire tensor is expressed as the product $\prod_{i=1}^k d_i$. Consequently, each individual element within the multi-dimensional array T_{j_1, \dots, j_k} can be unequivocally identified by a the k -tuple $\{j_1, \dots, j_k\}$. As passing by we introduce the following:

Definition 1.3: Tensor dimension

The number of values a tensor index can take is referred to as the **dimension of an index**, which is most often denoted by χ , but can also be denoted by d , especially when it spans a local physical vector space (or its dual). The product of all those dimensions defines the total dimension of a tensor.

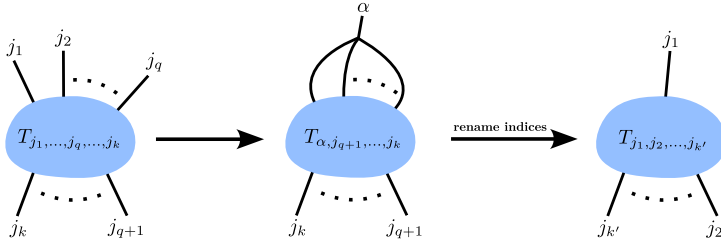
A scalar is an order-0 tensor, a vector is an order-1 tensor, and a matrix an order-2 tensor and so on. When dealing with tensors of higher order that feature multiple legs, tensor diagrams offer a powerful tool for representing them and eventually executing intricate operations. These diagrams serve as a visual and intuitive representation, simplifying complex computations and enhancing comprehension of tensor behavior.

In the following we illustrate the tensor network diagram for a scalar S , a vector V , a matrix M , and a generic order- k tensor T :

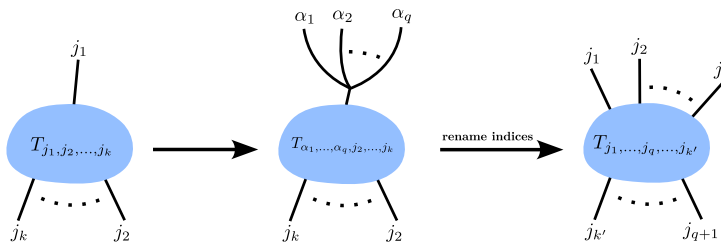


Tensor reshaping by index fusion — Given a generic order- k tensor, one can always reshape the tensor into an order- k' tensor with $k' < k$. This fundamental operation is called index fusion. We first identify the $q \leq k$ indices we want to fuse together, for example $\{j_1, \dots, j_q\}$. Each of those indices is labeling a local Hilbert space vector $\{|j_i\rangle : j_i = 1, \dots, d_i\}$. We then consider the tensor product of those Hilbert spaces $\mathcal{H}_1 \otimes \dots \otimes \mathcal{H}_q$ and its full computational basis $\{|j_1, \dots, j_q\rangle \equiv |\alpha\rangle : \alpha = 1, \dots, \prod_{i=1}^q d_i\}$. In the end we get a tensor with order $k' = k - q + 1$. Let us mention that, a tensor can be always reshaped back to its original form (or just partially reshaped) provided that we keep track of how the new index α is targeting the tensor product states $|\alpha\rangle \equiv |j_1, \dots, j_q\rangle$ as-

sociated with it, or in other words how it has been computed starting from the original indices $\{j_1, \dots, j_q\}$. The following tensor network diagram illustrates the reshaping of a generic tensor with index fusion,

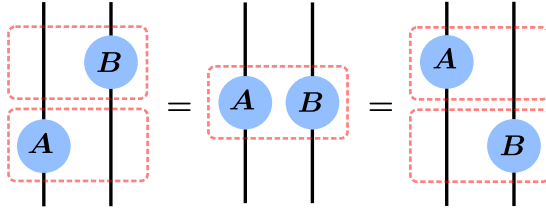


Tensor reshaping by index splitting — Alternatively, we can reshape a generic order- k tensor into an order- k' tensor with $k' > k$ by splitting one or several indices. This operation is the inverse of index fusion and involves splitting an index into several indices of smaller dimension, $\{|j_i\rangle : j_i = 1, \dots, d_i\} \rightarrow \{|j_i\rangle \equiv \{|\alpha_1, \dots, \alpha_q\rangle : \alpha_l = 1, \dots, d_l\}$ where, $d_i = \prod_{l=1}^q d_l'$. The result is a tensor of order $k' = k + q - 1$. Notably, reshaping a tensor by splitting an index is not a unique operation as an index of generic dimension d can be split into q indices in more than one ways. Therefore, index splitting is an inverse of index fusion only if we keep track of the way indices were fused. The following tensor network diagram illustrates the reshaping of a generic tensor with index splitting,



Tensor reshaping by outer product — When multiple disconnected tensors appear within the same diagram, they are combined through the tensor product operation. In quantum physics notation, this operation is denoted by the symbol \otimes . However, in abstract index notation, the tensor product sign is not explicitly indicated. Notice that, a tensor by itself is not carrying any information about the algebra of the vector spaces (and dual) connected to its legs. Therefore, when tensors are drawn near each other, they can freely

move through each others as



which, assuming applying tensors from bottom to up, basically means

$$(\mathbb{1} \otimes B)(A \otimes \mathbb{1}) = A \otimes B = (A \otimes \mathbb{1})(\mathbb{1} \otimes B), \tag{1.1}$$

where $\mathbb{1}$ is the identity matrix and it has a very special representation as a tensor diagram: just a simple wire.

Notice that, the tensor product of two or more tensors may results in a final tensor with higher order. Let us formalise this idea by considering the elementary outer product operation: A and B are two tensors, such that indices $\{\alpha_1, \dots, \alpha_p\}$ of tensor A need to be “tensorised” with indices $\{\beta_1, \dots, \beta_q\}$ of tensor B no matter the other indices of both tensors. We can do that in two steps: (i) we first fuse the indices $\{\alpha_1, \dots, \alpha_p\}$ and $\{\beta_1, \dots, \beta_q\}$ respectively into a single index α and β ; (ii) we then fuse the new bold indices into one single index $\gamma = (\alpha, \beta)$ such that

$$A_{(i_1, \dots, i_n), \alpha} B_{(j_1, \dots, j_m), \beta} \equiv C_{(i_1, \dots, i_n, j_1, \dots, j_m), \gamma}, \tag{1.2}$$

which defines the new tensor C .

1.1.1 Special Tensors

Here, we gather some special tensors that we will encounter frequently during the subsequent sections and chapters.

Identity tensor — In the graphical notation, there is one class of tensors which play a crucial role in many different shape manipulation at it always plays the role of the *Kronecker delta*: they are the so called *identity tensors*:

$$\left| \begin{array}{c} \text{---} \\ \text{---} \end{array} \right| = \delta_{ij}$$

These tensors enable (i) the contraction of tensor indices through diagrammatic connection (see next section), (ii) the raising and lowering of indices (when index position may have a meaning related to the vector space associated to the specific leg), and (iii) they establish a duality between maps, states, and linear maps in general (see next chapter).

However, as we have already stressed, a tensor by itself does not carry any information about the algebra of the vector spaces associated to the specific legs that are going to be contracted; the same consideration are valid for the identity tensor, which therefore act as simple connecting wire. Notwithstanding, as we will briefly anticipate in Example 1.1, and we will thoroughly use in the next sections, when identities are promoted to “operators”, then which states are contracted do matter.

Example 1.1: Glimpse on Quantum Computing

As previously mentioned, tensors serve as multilinear maps. They can be decomposed in any given basis and expressed in terms of their components. In the realm of quantum information science, it is common to introduce a computational basis $\{|0\rangle, |1\rangle\}$ for each Hilbert space and expand the tensors within it, utilizing kets $(| \rangle)$ for vectors and bras $(\langle |)$ for dual vectors. This expansion is represented as follows:

$$\hat{T} = \sum_{i,j,k} T_{j|k}^i |i\rangle\langle j,k|$$

Here, $T_{j|k}^i$ are the actual components of the tensor in the computational basis. Notice that, drawing the leg i pointing up, and the legs (j, k) pointing down is meaningless by itself (we could have rearranged the indices into any form without changing the meaning of the numeric tensor T); what is meaningful is the action of the **operator** \hat{T} once different Hilbert spaces have been linked to each leg of the tensor box T .

With this in mind, we can revisit the questions of how the Kronecker delta tensor may be used to construct different quantum operators. Let us consider two copies of a two-level quantum system, and let's introduce the computational basis $\{|0\rangle, |1\rangle\}$. We may thus construct the following tensor operators:

1.

$$\sum_{i,j} \left| \begin{array}{c} i \\ j \end{array} \right| |i\rangle\langle j| = \sum_{i,j} \underbrace{|i\rangle\langle j|} = |0\rangle\langle 0| + |1\rangle\langle 1| = \hat{1}$$

2.

$$\sum_{i,j} \left| \begin{array}{c} i \\ j \end{array} \right| |i,j\rangle = \sum_{i,j} \underbrace{|i,j\rangle} = |0,0\rangle + |1,1\rangle = \hat{2}$$

3.

$$\sum_{i,j} \left| \begin{array}{c} i \\ j \end{array} \right| \langle i,j| = \sum_{i,j} \underbrace{\langle i,j|} = \langle 0,0| + \langle 1,1| = \hat{3}$$

In this context, the map labeled as 1. represents the identity operator, whereas the tensor operator labeled as 2. (or 3.) operates on a two-bra (or ket) system, yielding a scalar as the outcome. The connection between these three equations is established through the bending of wires. In a given basis, the act of bending a wire signifies the conversion of a bra to a ket, and vice versa.

In fact, with an abuse of notation, we can easily combine operators 1., 2. and 3. as they were simple wires. For example, let us consider three local Hilbert spaces with each a two-level system, and let us label those systems as $\{a, b, c\}$. Operators \hat{C}_a^b and ${}_b^c \hat{C}$ can be combined together giving

$$\hat{C}_a^b \overset{c}{\hat{C}} = \sum_k \langle k|_a \langle k|_b \cdot \sum_k |k\rangle_b |k\rangle_c = \sum_k |k\rangle_c \langle k|_a = {}_c \hat{C}_a$$

Copy tensor — A very useful tensor is the multidimensional generalisation of the Kronecker delta, where many legs are forced to be equal. In graphic notation one has

$$\begin{array}{c} j_n \\ \curvearrowright \\ j_1 \\ \curvearrowleft \\ j_2 \\ \curvearrowleft \\ j_3 \end{array} = \delta_{j_1, j_2, \dots, j_n}$$

Interestingly, the natural decomposition of the multidimensional Kronecker delta $\delta_{j_1, j_2, \dots, j_n} = \delta_{j_1, j_2} \delta_{j_2, j_3} \cdots \delta_{j_{n-1}, j_n}$ (or for any other indices permutations)

induces the following graphical *tensor network decomposition*

$$\begin{array}{c} j_n \\ \swarrow \\ \bullet \\ \searrow \\ j_1 \quad j_2 \quad j_3 \end{array} = \begin{array}{c} \bullet \quad \bullet \quad \bullet \quad \dots \quad \bullet \\ | \quad | \quad | \quad \quad \quad | \\ j_1 \quad j_2 \quad j_3 \quad \quad \quad j_n \end{array}$$

Swap tensor — Let's examine the tensor product of n equivalent Hilbert spaces denoted as $\mathcal{H}^{\otimes n}$. We can use two *Identity tensors* to formally construct the *Swap tensor*, which switches the positions of two Hilbert spaces within a composite system. Specifically, it exchanges two consecutive local Hilbert spaces i th and $(i + 1)$ th. The diagrammatic representation is

$$\begin{array}{c} l \quad \quad j \\ \swarrow \quad \searrow \\ \quad \quad \times \\ \swarrow \quad \searrow \\ i \quad \quad k \end{array} = \delta_{ij} \delta_{lk}$$

which, as we have consistently emphasized, it lacks any algebraic significance unless it is viewed in conjunction with the Hilbert spaces upon which it operates. In fact, from the operatorial point of view, the *Swap Operator* reads

$$\hat{S} = \sum_{i,j,l,k} \begin{array}{c} l \quad \quad j \\ \swarrow \quad \searrow \\ \quad \quad \times \\ \swarrow \quad \searrow \\ i \quad \quad k \end{array} |i, l\rangle \langle k, j| = \sum_{i,j} |i, j\rangle \langle j, i|$$

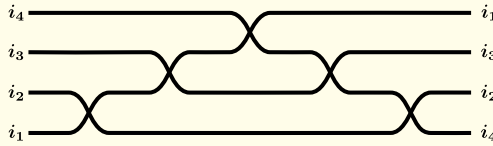
Notice that, this single operation it is sufficient to generate the all permutation group of the n local spaces.

Example 1.2: Non-local swapping

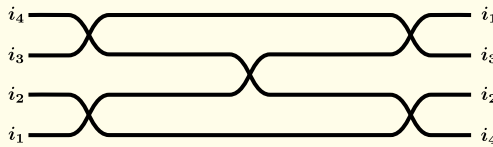
The Swap tensor serves as a generator for the permutation group. In this capacity, it enables the repositioning of two initially distant local systems, bringing them into proximity to facilitate interaction (cfr. Quantum Platform). This process can then be reversed to restore the entire system to its original configuration.

For clarity, let us consider a scenario involving four local Hilbert spaces denoted by $\{i_1, i_2, i_3, i_4\}$, where the objective is to swap the positions of the first and fourth systems. To achieve this, we can employ

five local swap operators in combination; for example in this way



where the swap operations need to be performed sequentially. Certainly, there exist implementation options where we can parallelize multiple permutations simultaneously within the specific platform. This strategy can significantly accelerate the procedure, nearly doubling its speed. For instance, one such approach involves the following:

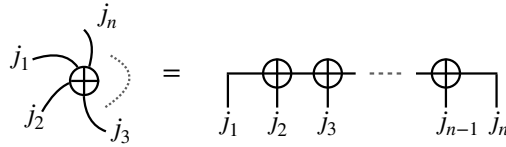


where, in this approach, the first swapping procedures involving $1 \leftrightarrow 2$ and $3 \leftrightarrow 4$ can be executed concurrently, as can the final $4 \leftrightarrow 2$ and $3 \leftrightarrow 1$ swaps.

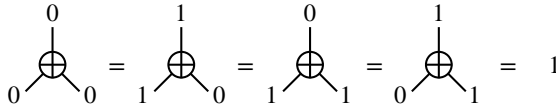
XOR tensor — The XOR (or Parity) tensor is a multi-dimensional array that operates on qubits or qubit-like systems in quantum computing. It represents a gate or operation that computes the parity (or XOR) of the qubits it acts upon. The XOR tensor outputs 1 if the number of qubits in the state with value 1 is even, and 0 if it is odd. Notice that, in this definition, there is an overall logical NOT operation if compared with the exclusive disjunction \oplus . In graphic notation one has

$$\begin{array}{c}
 j_n \\
 \curvearrowright \\
 j_1 \oplus \\
 \curvearrowleft \\
 j_2 \quad j_3
 \end{array}
 = \overline{j_1 \oplus j_2 \oplus j_3 \oplus \dots \oplus j_n}$$

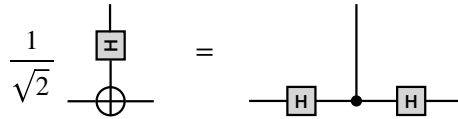
Also in this case, by using the associative property of the logical disjunction, we can easily decompose a n -legs XOR tensor into the following XOR chain



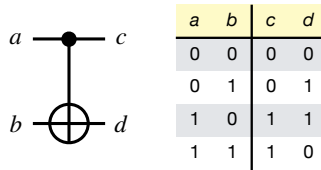
where we broke down the operation into 3-qubits XOR operations, such that



Let us mention the fact that, XOR and COPY tensors are related by local basis transformation induced by the Hadamard tensor H (cfr. Chapter 2), such that one has



CNOT tensor — The Control-NOT (CNOT) tensor, is a 4-order tensor which can be constructed by attaching a copy tensor with a xor tensor. It is of fundamental importance in quantum computation since it is used as entries of the homonym 2-qubit gate (cfr. Chapter 2). It is diagrammatic represented as



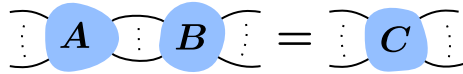
Basically, the copy tensor is attached to the control line, which let the input “a” to be copied to the output “c”; in the meanwhile this control line is controlling indeed the xor tensor, whose input “b” is then transformed to the output “d” according to the value of the control line (see logic table above).

1.2 Tensor Networks

A tensor network is a structured assemblage of tensors, where a specific subset (up to the entire set) of their total indices is systematically contracted.

The foundational operation within tensor networks involves the contraction of indices between two tensors, representing a fundamental and versatile generalization of traditional matrix multiplication. This operation lies at the core of tensor network methodologies, providing a framework for intricate manipulations and computations involving higher-order tensors.

Tensor index contraction — Diagrammatically, contracting two tensor indices corresponds to connect those indices with a single wire. Let's consider two tensors A and B of order respectively k and q , and dimensions $\prod_{i=1}^k d_i^{(A)}$ and $\prod_{i=1}^q d_i^{(B)}$. They can be *contracted* along shared indices: for example, the rightmost l indices of A with the leftmost l indices of B by just connecting the corresponding legs as below



which correspond in summing over the spaces of the shared indices, namely

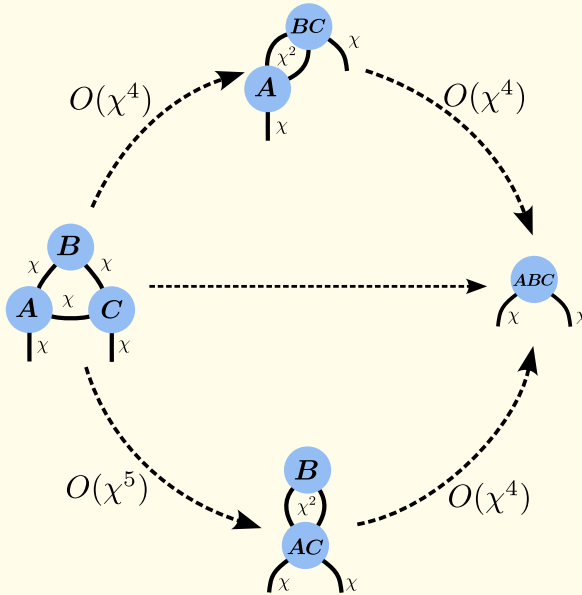
$$\sum_{j_1, \dots, j_l} A_{m_1, \dots, m_{k-l}, j_1, \dots, j_l} B_{j_1, \dots, j_l, n_1, \dots, n_{q-l}} = C_{m_1, \dots, m_{k-l}, n_1, \dots, n_{q-l}}. \quad (1.3)$$

The result is a order- $(k + q - 2l)$ tensor. Notice that the legs $\{j_1, \dots, j_l\}$ can be contracted provided that they are referring to the same vector (or dual) spaces in both tensor A and tensor B .

Alternatively, one can always achieve tensor contractions of arbitrary order and over arbitrary number of legs by resorting the operation to a vector-vector, matrix-vector or matrix-matrix multiplication. This involves initially *reshaping the tensors* into lower order tensors, by collecting via *index fusion* all non-contracted indices into a single index and all shared indices into another single index. In our previous example, we have considered the special case where the last l indices of A and the first l indices of B are the shared indices; this will correspond to reshape both tensors as $A_{m_1, \dots, m_{k-l}, j_1, \dots, j_l} \rightarrow A_{m, j}$ and $B_{j_1, \dots, j_l, n_1, \dots, n_{q-l}} \rightarrow B_{j, n}$. This is followed by regular a matrix-matrix multiplication along the shared index j , $\sum_j A_{m, j} B_{j, n} = C_{n, m}$. Finally, the free indices are reshaped back to the original indices $C_{n, m} \rightarrow C_{m_1, \dots, m_{k-l}, n_1, \dots, n_{q-l}}$. The modern scientific computing provides several options for performing efficiently algebraic manipulation of tensors.

Example 1.3: Contraction order do matter

At the core of all tensor network procedures lies the contraction of a network comprising multiple tensors, consolidating them into a single tensor. Here we illustrate an example of the problem where the goal is to contract the tensor network $\{A, B, C\}$ as depicted below



One straightforward approach to numerically evaluate this contraction would involve a direct summation over all the three shared indices, achievable through the implementation of a series of nested `for` loops. Although this approach yields the correct result, it is generally not the preferred method due to computational inefficiency. Specifically, for this example, assuming all indices have bond dimension χ , the direct summation over the internal legs requires $O(\chi^3)$ operations, for each entry of the tensor ABC , thus yielding to an overall computational cost $O(\chi^5)$.

We can opt for an alternative approach and break down the assessment of the tensor ABC into two distinct steps. But also in this case, we may follow different paths; for example:

1. $A, B, C \rightarrow AC, B \rightarrow ABC$
2. $A, B, C \rightarrow A, BC \rightarrow ABC$

Applying analogous reasoning as previously, it can be deduced that the computational cost for evaluating the intermediate tensor AC in the path 1 scales as $O(\chi^5)$, and only the subsequent contraction to get ABC needs for a smaller cost of $O(\chi^4)$. However, by following the path 2 we only need an overall computational cost of $O(\chi^4)$.

Therefore, decomposing the network contraction into a series of smaller contractions, each involving only a pair of tensors (referred to as pairwise tensor contraction), proves to be computationally economical or even more cost-effective for any non-trivial bond dimension, provided that the chosen sequence of pairwise contractions have been properly selected.

Tensor contraction cost — We have just considered the problem of efficiently contracting tensors, and we have realized from the Example 1.3, that the elementary contraction within a tensor network is a *pairwise contraction*. This contraction is denoted as $(A \times B)$, and involves tensors A and B connected by one or more common indices. A straightforward method for executing such contractions, as exemplified in Equation (1.3), entails employing nested for loops to sum over the shared indices. The computational cost associated with this evaluation, measured in terms of the required number of scalar multiplications, can be precisely expressed as:

$$\text{cost}(A \times B) = \frac{\dim(A) \times \dim(B)}{\dim(A \cap B)} \quad (1.4)$$

where $\dim(A)$ ($\dim(B)$) represents the total dimension of A (B) (i.e., the product of its index dimensions), and $\dim(A \cap B)$ denotes the total dimension of the shared indices.

Alternatively, one can reinterpret a pairwise contraction as a matrix multiplication by following these steps: firstly, rearrange the free and contracted indices on each tensor so that they appear sequentially. Next, merge the free indices and the contracted indices into single separate indices. Subsequently, the contraction is performed through a single matrix-matrix multiplication operation, although the resulting product may need to be reshaped back into tensor form. This approach of recasting as matrix multiplication does not alter the formal computational cost outlined in Equation (1.4). However, in modern computing, utilizing highly optimized BLAS routines, matrix multiplications are typically executed more efficiently than equivalent for loop summations. Consequently, when dealing with tensors of large total dimensions, recasting as matrix multiplication is often favored for evaluating pairwise tensor

contractions, despite requiring additional computational overhead due to the necessity of rearranging tensor elements in memory using “reshaping” (i.e. index fusion). However, for practical uses there are several packages available in majority of modern programming languages designed for scientific computing that allows for a very efficient and direct contraction of a complex tensor network.

1.3 Tensor Network Decomposition

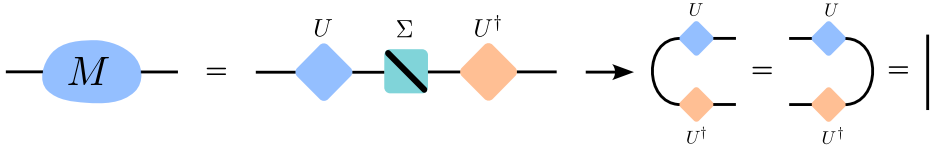
Tensor decomposition involves the process of taking a single tensor and breaking it down into two or more constituent tensors. The original tensor indices are distributed among these constituents, which are connected through shared internal indices. Conceptually, tensor decomposition can be seen as the inverse operation of tensor contraction. In this section, we explore two specific decompositions that play a pivotal role in essential tensor network algorithms for quantum physics, namely the *eigendecomposition*, *singular value decomposition* (SVD), and the *QR decomposition*. These decompositions are essentially matrix decomposition, for tensors of higher order decomposition is usually achieved by first reshaping the tensor into an appropriate matrix with index fusion followed by a matrix decomposition, and finally reshaping back to the higher order tensor by index splitting.

Eigendecomposition — Matrices that are diagonalizable can be factorized in terms of its eigen values and eigenvectors called eigendecomposition. Here we restrict to Hermitian matrix which is a special case of diagonalizable matrix relevant to quantum operators. In this case it is also called spectral decomposition. Consider a $n \times n$ Hermitian matrix M , the eigendecomposition of M is given by,

$$M_{\alpha,\beta} = U_{\alpha,\gamma} \Sigma_{\gamma,\gamma} U_{\gamma,\beta}^\dagger, \quad (1.5)$$

where U is a $n \times n$ unitary matrix ($UU^\dagger = U^\dagger U = \mathbb{1}$) whose columns are orthogonal eigenvectors of M and Σ is a $n \times n$ diagonal matrix whose diagonal entries are the corresponding eigenvalues. The computational cost for an eigendecomposition scales as $O(n^3)$. The following tensor network diagram shows the eigendecomposition of a generic Hermitian matrix M and the

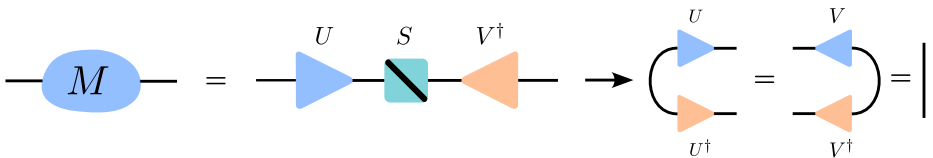
associated properties. The vertical line represents the identity matrix $\mathbb{1}$.



Singular Value Decomposition (SVD) — Unlike eigendecomposition singular Value decomposition exists for any generic matrix. SVD factorizes a $m \times n$ rectangular matrix M into three different components as follows,

$$M_{\alpha,\beta} = U_{\alpha,\gamma} S_{\gamma,\gamma} V_{\gamma,\beta}^\dagger, \tag{1.6}$$

where U is a $m \times m$ unitary matrix, S is a $m \times n$ rectangular diagonal matrix with positive entries, and V^\dagger is a $n \times n$ unitary matrix. The diagonal entries of S are known as the singular values and is usually written in descending order without loss of generality. The number of non-zero singular values $r = \min(m, n)$ is known as the rank of matrix M . This allows for a compact representation of SVD, $M_{\alpha,\beta} = \sum_{\gamma=1}^r U_{\alpha,\gamma} S_{\gamma,\gamma} V_{\gamma,\beta}^\dagger$. In compact representation U has dimension $m \times r$ and has orthonormal columns, $U^\dagger U = \mathbb{1}$ and are called left normalized, whereas V^\dagger has dimension $r \times n$ and has orthonormal rows, $V V^\dagger = \mathbb{1}$ and are called right normalized. The computational cost of compact SVD is $O(mn^2)$ (assuming $m > n$). The following tensor network diagram shows the SVD of a generic rectangular matrix M and the associated properties.

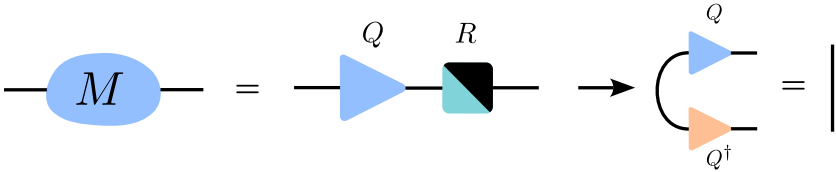


The eigendecomposition is related to SVD. The SVD of M implies the following two expressions, $MM^\dagger = U(SS^*)U^\dagger$ and $M^\dagger M = V(S^*S)V^\dagger$. Therefore the singular values are the square roots of the eigenvalues of MM^\dagger (and $M^\dagger M$). If M is Hermitian it has a unique eigendecomposition, $M = \bar{U}\Sigma\bar{U}^\dagger$ and the absolute eigenvalues of M are the singular values. If M is positive semi-definite the eigenvalues becomes equal to the singular values. In this scenario the SVD becomes equivalent to eigendecomposition.

QR decomposition — QR decomposition decomposes a matrix M of dimension $m \times n$ into two constituent components as,

$$M_{\alpha,\beta} = Q_{\alpha,\gamma} R_{\gamma,\beta} \quad (1.7)$$

where, Q is a $m \times m$ unitary matrix, R is a $m \times n$ upper triangular matrix. The lowest $(m - n)$ rows of matrix R are comprised of zeros, allowing for a more compact representation of the decomposition as follows: $M_{\alpha,\beta} = \sum_{\gamma} Q_{\alpha,\gamma}^1 R_{\gamma,\beta}^1$, where Q^1 denotes a rectangular matrix with dimensions $m \times n$ and features orthonormal columns ($Q^{\dagger} Q = \mathbb{1}$), while R^1 is an upper triangular square matrix sized at $n \times n$. Although Q^1 shares the characteristic of left orthonormality with matrix U in the SVD, it is important to note that these matrices are not identical. The following tensor network diagram shows the QR decomposition of a generic rectangular matrix M and the associated properties.



Analogously, we can define LQ and RQ decompositions, where Q is now right normalized, $QQ^\dagger = \mathbb{1}$ and L and R are lower and upper triangular matrices respectively.

1.3.1 Matrix product tensor network

The matrix decompositions outlined above can be used to decompose a generic tensor into constituent tensors of lower order [KB09; Ose11]. One such decomposition that has huge application in many body physics is matrix product decomposition where a generic tensor of order- k is decomposed as the product of k matrices (or more precisely order-3 tensors). This decomposition is obtained by a sequence of SVD or QR decompositions in a generic matrix. In the following example we will employ SVD to decompose a generic order- k tensor into right canonical matrix product network.

Example 1.4: Left canonical matrix product decomposition

Consider a generic order- k tensor T_{j_1, j_2, \dots, j_k} , we begin by reshaping the tensor into a matrix by index fusion, $T_{j_1, (j_2, \dots, j_k)}$ followed by a SVD,

$$T_{j_1, (j_2, \dots, j_k)} = \sum_{c_1} U_{j_1, c_1} S_{c_1, c_1} V_{c_1, (j_2, \dots, j_k)}^\dagger = \sum_{c_1} A_{j_1}^{c_1} T_{c_1, j_2, \dots, j_k}.$$

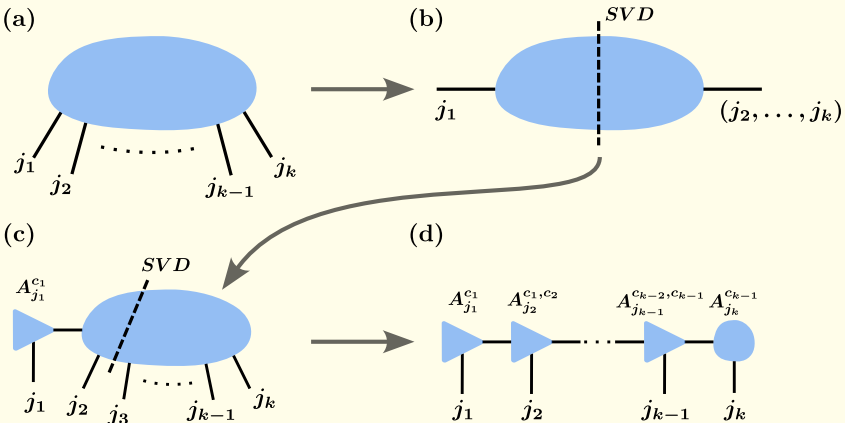
The left normalized matrix U_{j_1, c_1} is renamed as $A_{j_1}^{c_1}$ and S and V^\dagger is contracted and reshaped into T_{c_1, j_2, \dots, j_k} . In the second step we reshape T_{c_1, j_2, \dots, j_k} into a matrix $T_{(c_1, j_2), (j_3, \dots, j_k)}$ followed by a SVD,

$$\begin{aligned} \sum_{c_1} A_{j_1}^{c_1} T_{(c_1, j_2), (j_3, \dots, j_k)} &= \sum_{c_1, c_2} A_{j_1}^{c_1} U_{c_1, j_2, c_2} S_{c_2, c_2} V_{c_2, (j_3, \dots, j_k)}^\dagger \\ &= \sum_{c_1, c_2} A_{j_1}^{c_1} A_{j_2}^{c_1, c_2} S_{c_2, c_2} V_{c_2, (j_3, \dots, j_k)}^\dagger \\ &= \sum_{c_1, c_2} A_{j_1}^{c_1} A_{j_2}^{c_1, c_2} T_{c_2, j_3, \dots, j_k}. \end{aligned}$$

This procedure is sequentially repeated at each indices to obtain the full decomposition,

$$T_{j_1, j_2, \dots, j_k} = \sum_{c_1, c_2, \dots, c_{k-1}} A_{j_1}^{c_1} A_{j_2}^{c_1, c_2} \dots A_{j_{k-1}}^{c_{k-2}, c_{k-1}} A_{j_k}^{c_{k-1}}.$$

The full procedure is summarized in the following tensor network diagram,

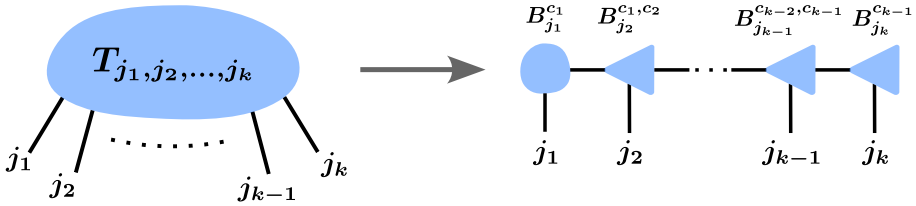


If we assume the dimension of the free indices to be χ then the dimension of the constituent tensors are $\chi \times \chi, \dots, \chi^{k/2-1} \times \chi \times \chi^{k/2}, \chi^{k/2} \times \chi \times \chi^{k/2-1}, \dots, \chi \times \chi$. This is called a left canonical decomposition since the constituent tensors (apart from the rightmost tensor $A_{j_k}^{c_{k-1}}$) are left normalized, $\sum_{c_{l-1}, j_l} A_{j_l}^{c_{l-1}, c_l} \left[A_{j_l}^{c_{l-1}, c'_l} \right]^\dagger = \delta^{c_l, c'_l}$.

The example presented above is not unique, alternatively we can start the SVD from the rightmost index j_k and proceed from right to left to obtain the right canonical matrix product decomposition,

$$T_{j_1, j_2, \dots, j_k} = \sum_{c_1, c_2, \dots, c_{k-1}} B_{j_1}^{c_1} B_{j_2}^{c_1, c_2} \dots B_{j_{k-1}}^{c_{k-2}, c_{k-1}} B_{j_k}^{c_{k-1}}.$$

This is summarized by the following tensor network diagram,



Each constituent tensors (apart from the leftmost tensor $B_{j_1}^{c_1}$) are right normalized, $\sum_{c_l, j_l} B_{j_l}^{c_{l-1}, c_l} \left[B_{j_l}^{c'_l, c_l} \right]^\dagger = \delta^{c_l, c'_l}$.

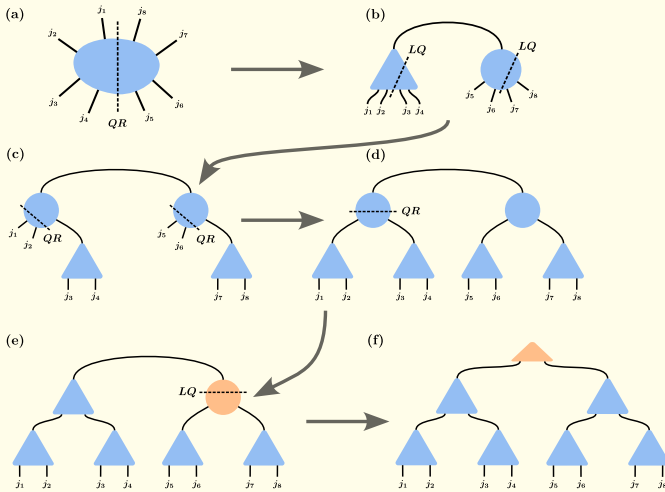
1.3.2 Tree tensor network (TTN)

Another relevant decomposition of a large tensor is the tree tensor network (TTN) where the constituent tensors of the network lie on the nodes of a connected acyclic tree graph[Mon18]. In the following examples we outline steps to decompose an order-8 tensor into a binary TTN (two tensors branching out of each tensor).

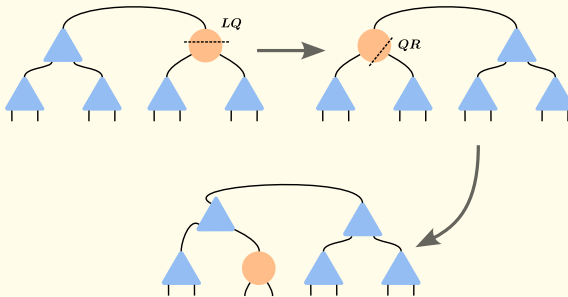
Example 1.5: Tree tensor network decomposition

We start from a generic tensor of order-8. The tensor is split into two order-5 tensors via QR decomposition as shown in (a). Similar steps of consecutive QR and LQ decompositions are applied to get six order-3 tensors as shown in (b),(c), and (d) respectively. The underlying struc-

ture is a binary tree with an input branch and two output branches stemming out of each nodes. In the final form five of the six constituent tensors are right or left orthogonal as shown in (e). The non-orthogonal tensor (in orange full circle) is known as the orthogonality center. A tensor in a tensor network is called an orthogonality center if all the indices stemming out of the tensor annihilate to identity when contracted with its conjugate. We can also separate the orthogonality center into an order-2 tensor (in orange full triangle) lying at the geometrical center of the tensor network.



The position of orthogonality center within a tree tensor network is flexible. We can shift the orthogonality center to any node by a sequence of QR and LQ decompositions. The following figure depicts the shifting of orthogonality center in the tree tensor network from one node to another.



References of Chapter 1

- [Bia20] J. Biamonte. *Lectures on Quantum Tensor Networks*. 2020. arXiv: [1912.10049](https://arxiv.org/abs/1912.10049) [quant-ph].
- [Eve22] G. Evenbly. “A Practical Guide to the Numerical Implementation of Tensor Networks I: Contractions, Decompositions, and Gauge Freedom.” In: *Frontiers in Applied Mathematics and Statistics* 8 (2022). ISSN: 2297-4687. DOI: [10.3389/fams.2022.806549](https://doi.org/10.3389/fams.2022.806549).
- [KB09] T. G. Kolda and B. W. Bader. “Tensor Decompositions and Applications.” In: *SIAM Review* 51.3 (2009), pp. 455–500. DOI: [10.1137/07070111X](https://doi.org/10.1137/07070111X).
- [Mon18] S. Montangero. *Introduction to Tensor Network Methods: Numerical Simulations of Low-Dimensional Many-Body Quantum Systems*. Vol. 922. Lecture Notes in Physics. Cham: Springer International Publishing, 2018. ISBN: 978-3-030-01408-7. DOI: [10.1007/978-3-030-01409-4](https://doi.org/10.1007/978-3-030-01409-4).
- [Ose11] I. V. Oseledets. “Tensor-Train Decomposition.” In: *SIAM Journal on Scientific Computing* 33.5 (2011), pp. 2295–2317. DOI: [10.1137/090752286](https://doi.org/10.1137/090752286).
- [Ran+20] S.-J. Ran et al. *Tensor Network Contractions: Methods and Applications to Quantum Many-Body Systems*. Springer International Publishing, 2020. ISBN: 9783030344894. DOI: [10.1007/978-3-030-34489-4](https://doi.org/10.1007/978-3-030-34489-4).

Chapter 2

Quantum Physics with Tensors

Reality is not always probable,
or likely.

Jorge Luis Borges

2.1 Introduction to quantum mechanics

Quantum mechanics is universally recognized as one of the main achievement in humanity's understanding of the natural world [Pen04]. Despite its incredible generality and astonishing predictive power, this theory is based on a few simple foundational elements. In the following, we will shortly summarize these postulates, without explaining how they were originally derived by the founding fathers of quantum mechanics, about a hundred years ago.

2.1.1 States and observables

First, any quantum system is associated with an Hilbert space \mathcal{H} , i.e. a linear vector space over the field of complex numbers \mathbb{C} with a metric induced by the inner product. Elements (vectors) of the Hilbert space represent possible states of the system, and are usually denoted with the ket notation $|\psi\rangle$. The inner scalar product between two states $|\psi\rangle, |\phi\rangle$ is denoted as $\langle\psi|\phi\rangle \in \mathbb{C}$. Physical states have norm 1, meaning that $\| |\psi\rangle \|^2 = \langle\psi|\psi\rangle = 1$. Since \mathcal{H} is a linear space, in principle any linear combination $\alpha |\psi\rangle + \beta |\phi\rangle$ ($\alpha, \beta \in \mathbb{C}$) of

two states $|\psi\rangle, |\phi\rangle$ is also a possible state (a part for an overall normalization constant).

Sometimes it is convenient to represent a state $|\psi\rangle$ in terms of its components with respect to a basis of the Hilbert space. Specifically, we consider an orthonormal basis, denoted as $\{|k\rangle\}_{k=1}^D$ where D is the Hilbert space dimension. The vectors of this basis satisfies: $\langle k|l\rangle = \delta_{kl}$ (orthogonality), and $\sum_{k=1}^D |k\rangle\langle k| = \hat{1}$ (completeness). The decomposition of $|\psi\rangle$ is the following

$$|\psi\rangle = \sum_k \psi_k |k\rangle, \quad (2.1)$$

and the components are given by

$$\psi_k = \langle\psi|k\rangle. \quad (2.2)$$

Essentially, by choosing a specific basis, we can express all the states in the Hilbert space in terms of their components (coordinates). Consequently, we effectively establish an equivalence between \mathcal{H} and \mathbb{C}^D . This is of great practical utility, as numbers can be easily handled for numerical calculations.

The second part of this first postulate asserts that any physical observable O (such as energy, momentum, magnetization, etc.) is associated with a Hermitian operator \hat{O} acting on \mathcal{H} (i.e. $\hat{O} : \mathcal{H} \rightarrow \mathcal{H}$ and $\hat{O}^\dagger = \hat{O}$, where \dagger indicates the Hermitian adjoint). Also operators can be expressed in coordinates as

$$\hat{O} = \sum_{k,l=1}^D O_{kl} |k\rangle\langle l|, \quad (2.3)$$

with

$$O_{kl} = \langle k|\hat{O}|l\rangle. \quad (2.4)$$

Notice that $|k\rangle\langle l|$ is a convenient notation for the linear operator acting as $|k\rangle\langle l|k'\rangle = |k\rangle\delta_{lk'}$ on a given vector of the basis $|k'\rangle$. The coordinates O_{kl} of the operator are D^2 complex numbers. The fact that the operator must be Hermitian implies that the matrix of coefficients must satisfy the condition $O_{kl}^* = O_{lk}$. Even for operators, there exists a vector space structure, as Hermitian matrices of size $D \times D$ form a vector space over the field of real numbers \mathbb{R} with dimension D^2 .

As a final point in this section, we observe that in general one can contemplate a broader scenario where a quantum system exists in a certain statistical mixture $\{\pi_i, |\psi_i\rangle\}$ of possible states (i.e. the system can be in the state $|\psi_i\rangle$ with probability $\pi_i \geq 0$, and $\sum_i \pi_i = 1$). In this context, the mathematical

object that is convenient to consider for describing the system is the following operator

$$\hat{\rho} = \sum_i \pi_i |\psi_i\rangle \langle \psi_i|, \quad (2.5)$$

which is known as density matrix. Here, $|\psi_i\rangle \langle \psi_i|$ is the projector operator on the state $|\psi_i\rangle$. The density matrix is:

- i. Hermitian $\hat{\rho}^\dagger = \hat{\rho}$;
- ii. positive semi-definite, meaning that $\langle \psi | \hat{\rho} | \psi \rangle \geq 0$, for any state $|\psi\rangle$;
- iii. with trace 1 since: $\text{Tr}[\hat{\rho}] = \sum_i \pi_i \text{Tr}[|\psi_i\rangle \langle \psi_i|] = \sum_i \pi_i = 1$.

In general, any operator that meets these conditions i), ii), iii) is as a possible density matrix. Notice that since $\hat{\rho}$ is Hermitian, it can be diagonalized as:

$$\hat{\rho} = \sum_k p_k |\varphi_k\rangle \langle \varphi_k|, \quad (2.6)$$

where $|\varphi_k\rangle$ are the eigenvectors of $\hat{\rho}$ and p_k the associated eigenvalues. Conditions ii) and iii) can also be rephrased saying that a density matrix should have non negative eigenvalues $p_k \geq 0$ which sum to 1, $\sum_k p_k = 1$. This means that eigenvalues of a density matrix can also be interpreted as a set of probabilities. As for states, also density matrices can be linearly combined. However, since the trace of the sum has to be 1, only convex superposition are allowed, as for instance $p\hat{\rho}_1 + (1-p)\hat{\rho}_2$, with $p \in [0, 1]$. Whenever a density matrix can be expressed as a projector into a single state $\hat{\rho} = |\psi\rangle \langle \psi|$ (i.e. the statistical mixture is trivial), it is said that the system is in a pure state.

2.1.2 Evolution

The second postulate defines how a quantum mechanical system evolves. It states that any closed quantum system undergoes its evolution through a unitary transformation. This implies that in the absence of interaction with the environment, any physical process is mathematically represented by a unitary operator \hat{U} acting on the Hilbert space and mapping a state $|\psi\rangle$ to a new state $U|\psi\rangle$. We recall that a linear operator is unitary when it preserves the inner product or equivalently when it satisfies $\hat{U}^\dagger \hat{U} = \hat{U} \hat{U}^\dagger = \hat{1}$. This condition allows consistency with the first postulate, since the evolved state $U|\psi\rangle$ will still have a norm of 1 (since $\| \hat{U} |\psi\rangle \|^2 = \langle \psi | \hat{U}^\dagger \hat{U} | \psi \rangle = \| |\psi\rangle \|^2 = 1$).

The most notable example of quantum evolution is given by the Schrödinger equation, which describes the evolution in time t of a closed quantum system, relating the derivative with respect to time of the state $|\psi(t)\rangle$

to the action of a suitable Hermitian operator \hat{H} called the Hamiltonian of the system [KP09]. The equation is:

$$\frac{\partial}{\partial t} |\psi(t)\rangle = -i\hat{H} |\psi(t)\rangle . \quad (2.7)$$

The Hermiticity property of the Hamiltonian, $\hat{H} = \hat{H}^\dagger$, implies that

$$\frac{d}{dt} \langle \phi(t) | \psi(t) \rangle = -i \langle \phi(t) | \hat{H} | \psi(t) \rangle + i \langle \phi(t) | \hat{H}^\dagger | \psi(t) \rangle = 0 \quad (2.8)$$

which implies at all time t

$$\langle \phi(t) | \psi(t) \rangle = \langle \phi(0) | \psi(0) \rangle . \quad (2.9)$$

Last equation signifies that the time evolution preserves the inner products between states. Thus, Schrödinger's time evolution is governed by a unitary operator \hat{U}_t such that

$$|\psi(t)\rangle = \hat{U}_t |\psi(0)\rangle . \quad (2.10)$$

It is not difficult to realize that

$$\hat{U}_t = \exp(-i\hat{H}t) \equiv \sum_{n=0}^{\infty} \frac{(-i\hat{H}t)^n}{n!} , \quad (2.11)$$

where we have defined the exponential of an operator through its Taylor expansion in powers [KP09], i.e.

$$\exp(\hat{A}) \equiv \sum_{n=0}^{\infty} \frac{\hat{A}^n}{n!} . \quad (2.12)$$

In general, the Hamiltonian operator can depend on time $\hat{H}(t)$. This often occurs when the quantum system is coupled with external fields that change over time. In this case, the correct definition of the unitary operator describing the evolution is [KP09]

$$\begin{aligned} \hat{U}_t &= \mathcal{T} \left[\exp \left(-i \int_0^t \hat{H}(s) ds \right) \right] \equiv \\ &\equiv \sum_{n=0}^{\infty} \frac{(-i)^n}{n!} \int_0^t ds_1 \int_0^{s_1} ds_2 \dots \int_0^{s_{n-1}} ds_n \mathcal{T} [\hat{H}(s_1) \hat{H}(s_2) \dots \hat{H}(s_n)] , \end{aligned} \quad (2.13)$$

where we introduced the time-ordering operator $\mathcal{T}[\dots]$ which arranges operators in chronological order [KP09]. For instance:

$$\mathcal{T}[\hat{H}(s_1)\hat{H}(s_2)] = \begin{cases} \hat{H}(s_1)\hat{H}(s_2) & \text{if } s_1 > s_2 \\ \hat{H}(s_2)\hat{H}(s_1) & \text{if } s_1 < s_2 \end{cases} \quad (2.14)$$

This type of evolution (unitary dynamics) is not the only possible one. Indeed, in the density matrix formalism, it is possible to identify a much more general class of physically allowed transformations. These also include cases where the system is coupled with an ancillary system, evolved unitarily, and then the ancillary system is traced out. The resulting dynamics for the system's density matrix $\hat{\rho}$ is thus non-unitary. This type of generalized evolution is described by a set of Kraus operators \hat{K}_i that must satisfy the completeness relation $\sum_i \hat{K}_i^\dagger \hat{K}_i = \hat{\mathbb{1}}$. The evolution is given by the mapping

$$\hat{\rho} \rightarrow \mathcal{E}(\hat{\rho}) = \sum_i \hat{K}_i^\dagger \hat{\rho} \hat{K}_i. \quad (2.15)$$

\mathcal{E} is a super-operator, meaning that it maps operators to operators. In fact, the form given in Eq. (2.15) represents the most general form for a class of super-operators known as completely positive trace-preserving super-operators. It can be rigorously demonstrated that if $\hat{\rho}$ is a well-defined density matrix, then $\mathcal{E}(\hat{\rho})$ will also be a well-defined density matrix. We refer the reader to Chapter 5 for a treatment of these topics.

2.1.3 Measurements

The third postulate of quantum mechanics concerns what happens when an external observer interacts with the quantum system to measure a physical quantity of interest. The possible outcomes of the measurement experiment are labeled by an index k , and for each of them we assume the existence of a certain operator \hat{M}_k describing the evolution of the system just after the measurement. The postulate asserts that, with probability $p_k = \langle \psi | \hat{M}_k^\dagger \hat{M}_k | \psi \rangle$, the measurement results in k and the system transitions (jumps) from its initial state $|\psi\rangle$ to

$$\frac{\hat{M}_k |\psi\rangle}{\sqrt{\langle \psi | \hat{M}_k^\dagger \hat{M}_k | \psi \rangle}}. \quad (2.16)$$

Notice that the post-measurement state is again properly normalized to 1. The following completeness equation must be fulfilled by the measurement operators

$$\sum_k \hat{M}_k^\dagger \hat{M}_k = \hat{\mathbb{1}}, \quad (2.17)$$

so that the probabilities p_k sum to 1. In case of mixed state $\hat{\rho}$, Eq. (2.16) is generalized to:

$$\frac{\hat{M}_k \hat{\rho} \hat{M}_k^\dagger}{\text{Tr}[\hat{M}_k^\dagger \hat{M}_k \hat{\rho}]}, \quad (2.18)$$

while the probability of outcome k is $p_k = \text{Tr}[\hat{M}_k^\dagger \hat{M}_k \hat{\rho}]$.

The simplest example of measurement process is given by projective measurements. These are a sub-class of the general scenario described above. In this case, the observer measures a Hermitian operator \hat{O} , and the measurements operators are the projectors on the eigenstates of \hat{O} . Since \hat{O} is Hermitian, we can use its spectral decomposition $\hat{O} = \sum_k O_k \hat{P}_k$, where \hat{P}_k is the projector onto the eigenspace of \hat{O} having eigenvalue O_k and $\hat{P}_k \hat{P}_l = \delta_{kl} \hat{P}_k$. Since \hat{P}_k are projectors, we have $p_k = \langle \psi | \hat{P}_k | \psi \rangle$. Furthermore, it is particularly simple to compute the expectation value of \hat{O} (namely, the value to which the empirical average of the outcomes converges as one repeats an increasing number of measurement experiments). Indeed,

$$\bar{O} = \sum_k p_k O_k = \sum_k \langle \psi | \hat{P}_k | \psi \rangle O_k = \langle \psi | \sum_k O_k \hat{P}_k | \psi \rangle = \langle \psi | \hat{O} | \psi \rangle, \quad (2.19)$$

where $\bar{\cdot}$ indicates average over possible outcomes. The fluctuations around this value are governed by the variance of the observed outcomes, namely

$$\overline{(O - \bar{O})^2} = \sum_k p_k (O_k - \bar{O})^2 = \langle \psi | \hat{O}^2 | \psi \rangle - \langle \psi | \hat{O} | \psi \rangle^2. \quad (2.20)$$

2.1.4 Composite systems

It is possible for a quantum system to consist of multiple quantum constituents. For instance, a molecule is composed of different atoms interacting with each other, and an atom consists of many electrons. Our last postulate specifies the relationship between the Hilbert space of the entire system \mathcal{H} and the Hilbert space of the individual constituents \mathcal{H}_i . The statement is that

$$\mathcal{H} \equiv \bigotimes_{i=1}^N \mathcal{H}_i, \quad (2.21)$$

i.e. the state space of a composite system is formed by the tensor product of the state spaces of the component physical systems. If we establish an orthonormal basis $|k_i\rangle_{k_i=1}^{D_i}$ for each individual Hilbert space \mathcal{H}_i , it follows that the basis for the composite Hilbert space \mathcal{H} is given by

$$|k_1, k_2 \dots k_N\rangle \equiv |k_1\rangle |k_2\rangle \dots |k_N\rangle. \quad (2.22)$$

The orthogonality condition becomes

$$\langle k_1, k_2 \dots k_N | k'_1, k'_2 \dots k'_N \rangle = \delta_{k'_1 k_1} \delta_{k'_2 k_2} \dots \delta_{k'_N k_N}. \quad (2.23)$$

Therefore the dimension of \mathcal{H} is the product of the dimension of individual Hilbert spaces, $D = \prod_{i=1}^N D_i$.

Now, suppose that our composite system is in a state $\hat{\rho}$. We divide the entire system into a specific subsystem A and its complement B , such that $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$. If our focus is solely on describing the behavior of subsystem A , we have somehow to marginalize over B , i.e. to sum over over all potential states of B . Indeed for any density matrix $\hat{\rho}$ of the full system and any observable $\hat{O}_A : \mathcal{H}_A \rightarrow \mathcal{H}_A$ acting only on A , we have

$$\text{Tr}[\hat{\rho}(\hat{O}_A \otimes \hat{\mathbb{1}}_B)] = \text{Tr}_A[\hat{\rho}_A \hat{O}_A] \quad (2.24)$$

where $\hat{\rho}_A$ is a density matrix for the subsystem only named reduced density matrix and defined as

$$\hat{\rho}_A \equiv \text{Tr}_B[\hat{\rho}]. \quad (2.25)$$

$\text{Tr}_B[\dots]$ is a mathematical operation called partial trace, defined from the following equality

$$\langle \alpha | \hat{\rho}_A | \alpha' \rangle = \sum_{\beta=1}^{D_B} \langle \alpha, \beta | \hat{\rho} | \alpha', \beta \rangle, \quad (2.26)$$

for any basis $\{|\alpha\rangle\}_{\alpha=1}^{D_A}$ and $\{|\beta\rangle\}_{\beta=1}^{D_B}$ of A, B . It is not difficult to show that if $\hat{\rho}$ is a well-defined density matrix, then so is $\hat{\rho}_A$, in the sense that it satisfies the three fundamental properties: i) $\hat{\rho}_A^\dagger = \hat{\rho}_A$; ii) $\hat{\rho}_A > 0$; iii) $\text{Tr}[\hat{\rho}_A^\dagger] = 1$. Besides, for any evolution operator \hat{U}_A acting on A only, we have: $\hat{U}_A^\dagger \hat{\rho}_A \hat{U}_A \equiv \text{Tr}_B[(\hat{U}_A \otimes \hat{\mathbb{1}}_B)^\dagger \hat{\rho} (\hat{U}_A \otimes \hat{\mathbb{1}}_B)]$, meaning that the result is the same regardless if we first apply $\hat{U}_A \otimes \hat{\mathbb{1}}_B$ and then trace away B , or we first then trace away B and then apply \hat{U}_A . Similar conclusions can be drawn in case of measurements performed on system A only.

Example 2.1: Purification

Purification refers to considering a mixed state (such as the infinite temperature state in the previous example) as derived from a pure state of a system and its environment by taking the partial trace over the environment. Specifically, consider a mixed state $\hat{\rho}_A$ for a system A with Hilbert space \mathcal{H}_A . It is always possible to find a complementary

system B with Hilbert space \mathcal{H}_B and a pure state $|\psi\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$ such that the reduced density matrix of $|\psi\rangle$ over A looks like $\hat{\rho}_A$, i.e.

$$\hat{\rho}_A = \text{Tr}_B[\hat{\rho}] \quad (2.27)$$

with $\hat{\rho} = |\psi\rangle\langle\psi|$. Indeed, to do this it is enough to write down $\hat{\rho}_A$ in its eigenbasis as: $\hat{\rho}_A = \sum_k p_k |\varphi_k^a\rangle\langle\varphi_k^a|$, where $\{|\varphi_k^a\rangle\}_k$ are the eigenvectors of $\hat{\rho}_A$ and p_k the associated eigenvalues. Afterwards, one can define $|\psi\rangle = \sum_k p_k^{1/2} |\varphi_k^a\rangle|\varphi_k^b\rangle$, where $\{|\varphi_k^b\rangle\}_k$ is any other set of mutually orthogonal and normalized states defined in a suitable system B .

2.2 Quantum mechanics of a single qubit

The simplest non-trivial quantum system is one in which the Hilbert space is spanned by only two states, which can be denoted as $|0\rangle$ and $|1\rangle$. Such a system is called a qubit (or quantum bit). A qubit is the fundamental unit of quantum information, analogous to the bit in classical computing. Unlike a classical bit, which can exist in only two distinct states 0 or 1, a qubit can be in a linear superposition of the two basis states, in accordance with the postulates of quantum mechanics. Any qubit state is therefore parameterized by two complex numbers α, β such that $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ (for physical states $|\alpha|^2 + |\beta|^2 = 1$). Thus, the Hilbert space has dimension $D = 2$ ($\mathcal{H} \sim \mathbb{C}^2$). Any operator acting on \mathcal{H} can be constructed using the identity matrix, plus the three Pauli matrices, whose graphical tensor representation is given by

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{array}{c} \text{---} \\ \text{X} \\ \text{---} \end{array} \quad \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = \begin{array}{c} \text{---} \\ \text{Y} \\ \text{---} \end{array} \quad \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{array}{c} \text{---} \\ \text{Z} \\ \text{---} \end{array}$$

Formally, the operators associated to these tensors are constructed by exploiting the effect of these matrices in the tensor product space $\mathcal{H} \otimes \mathcal{H}^*$. This results in a similar construction to what was obtained for the identity in Example 1.1,

$$\hat{X} = \sum_{i,j} \begin{array}{c} \text{---} \\ \text{X} \\ \text{---} \end{array} |i\rangle\langle j| = |0\rangle\langle 1| + |1\rangle\langle 0|$$

$$\hat{Y} = \sum_{i,j} \begin{array}{c} \text{---} \\ \text{Y} \\ \text{---} \end{array} |i\rangle\langle j| = -i|0\rangle\langle 1| + i|1\rangle\langle 0|$$

$$\hat{Z} = \sum_{i,j} \begin{array}{c} \text{---} \\ \text{Z} \\ \text{---} \end{array} |i\rangle\langle j| = |0\rangle\langle 0| - |1\rangle\langle 1|$$

Observe that the representation of Pauli operators underscores the particular basis selected for their expression. Indeed, typically in quantum computation, $|0\rangle$ is designated as the $+1$ eigenvector of \hat{Z} , and $|1\rangle$ as the -1 eigenvector of \hat{Z} .

Other relevant single qubit unitary transformations are the Hadamard gate \hat{H} , the phase gate \hat{S} and the \hat{T} gate, constructed in the similar way by exploiting the following tensors

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \boxed{\text{H}} \quad \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{pmatrix} = \boxed{\text{S}} \quad \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} = \boxed{\text{T}}$$

Note that we used the word ‘gate’ as a synonym for unitary transformation, borrowing it from the language of classical computation, where operations are performed by logical gates acting on bits of information (0 or 1).

2.2.1 Bloch sphere

An useful reparametrization of the qubit state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ can be utilized to elucidate the effects of unitary transformations. Indeed, since $|\alpha|^2 + |\beta|^2 = 1$, we can always find angles θ and ϕ such that¹

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle. \quad (2.28)$$

The fact that $\theta/2$ is used in this expression instead of θ can be clarified by calculating the expectation value of the Pauli matrices over the state. Indeed, we have:

$$\begin{aligned} \langle \psi | \hat{X} | \psi \rangle &= \cos \frac{\theta}{2} \sin \frac{\theta}{2} e^{i\phi} \langle 0 | \hat{X} | 1 \rangle + \cos \frac{\theta}{2} \sin \frac{\theta}{2} e^{-i\phi} \langle 1 | \hat{X} | 0 \rangle = \\ &= 2 \cos \frac{\theta}{2} \sin \frac{\theta}{2} \cos \phi = \sin \theta \cos \phi \\ \langle \psi | \hat{Y} | \psi \rangle &= \cos \frac{\theta}{2} \sin \frac{\theta}{2} e^{i\phi} \langle 0 | \hat{Y} | 1 \rangle + \cos \frac{\theta}{2} \sin \frac{\theta}{2} e^{-i\phi} \langle 1 | \hat{Y} | 0 \rangle = \\ &= 2 \cos \frac{\theta}{2} \sin \frac{\theta}{2} \sin \phi = \sin \theta \sin \phi \\ \langle \psi | \hat{Z} | \psi \rangle &= \cos^2 \frac{\theta}{2} \langle 0 | \hat{Z} | 0 \rangle + \sin^2 \frac{\theta}{2} \langle 1 | \hat{Z} | 1 \rangle = \cos^2 \frac{\theta}{2} - \sin^2 \frac{\theta}{2} = \cos \theta. \end{aligned} \quad (2.29)$$

Therefore, if we imagine placing the states $|0\rangle$ and $|1\rangle$ at the ‘‘North Pole’’ and ‘‘South Pole’’ of a sphere, respectively, we will find that the state $|\psi\rangle$ has polar

¹This expression holds true except for an overall global phase, which is entirely irrelevant as it cannot be detected by measurement experiments..

coordinates (θ, ϕ) , with $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi]$. This sphere is known as Bloch sphere. We will use the convention $|\theta, \phi\rangle$ to denote the state specified by the two angles θ, ϕ .

It should be noted that this is only an idealized framework and one should not think of the system as just a classic spin oriented in a certain direction. In fact, the Pauli operators exhibit non-zero (quantum) fluctuations from the expected value, since for instance $\langle \psi | \hat{Z}^2 | \psi \rangle - \langle \psi | \hat{Z} | \psi \rangle^2 = 1 - \cos^2 \theta \geq 0$. This means that the state is effectively in a kind of probabilistic superposition around the point in the Bloch sphere. In fact, every time a (projective) measurement of \hat{Z} is performed, the system will always be found in one of the two eigenstates of \hat{Z} , namely $|0\rangle$ or $|1\rangle$, never in rotated directions!

Now that we have an intuitive framework for single-qubit states, we can inquire about the unitary transformations that can implement rotations on the Bloch sphere. To answer, let us consider the following operator

$$\hat{R}_Z(\alpha) = \exp\left(-i\frac{\alpha}{2}\hat{Z}\right). \quad (2.30)$$

It is easy to realize that

$$\hat{R}_Z(\alpha) = \begin{pmatrix} e^{-i\frac{\alpha}{2}} & 0 \\ 0 & e^{+i\frac{\alpha}{2}} \end{pmatrix}. \quad (2.31)$$

Now we can apply this $\hat{R}_Z(\alpha)$ to the state $|\theta, \phi\rangle$, and result is

$$\hat{R}_Z(\alpha) |\theta, \phi\rangle = \cos\frac{\theta}{2} e^{-i\frac{\alpha}{2}} |0\rangle + e^{i\phi} \sin\frac{\theta}{2} e^{+i\frac{\alpha}{2}} |1\rangle = e^{-i\frac{\alpha}{2}} |\theta, \phi + \alpha\rangle. \quad (2.32)$$

So, up to a global phase, the operator $\hat{R}_Z(\alpha)$ has rotated the state on the Bloch sphere by an angle α around the Z-axis. In general, it is easy to realize that the operator

$$\hat{R}_{\mathbf{n}}(\alpha) = \exp\left(-i\frac{\alpha}{2}\mathbf{n} \cdot \hat{\boldsymbol{\sigma}}\right), \quad (2.33)$$

where

$$\hat{\boldsymbol{\sigma}} \equiv (\hat{X}, \hat{Y}, \hat{Z}) \quad (2.34)$$

implements a rotation of an angle α around the axis specified by the unit vector \mathbf{n} ($|\mathbf{n}| = 1$). The rotation is in the counter-clockwise direction. It is also straightforward to demonstrate that

$$\hat{R}_{\mathbf{n}}(\alpha) = \cos\left(\frac{\alpha}{2}\right) \hat{\mathbb{1}} - i \mathbf{n} \cdot \hat{\boldsymbol{\sigma}} \sin\left(\frac{\alpha}{2}\right). \quad (2.35)$$

It is now easy to realize that the Hadamard gate corresponds to a rotation of π around the axis $\frac{1}{\sqrt{2}}(1, 0, 1)$, while \hat{S} and \hat{T} are (apart for a global phase) rotations around the Z -axis of $\pi/2$ and $\pi/4$, respectively. In general, it is a well-known fact that every rotation can be decomposed into a composition of rotations around the axes X, Y, Z . It follows that every single qubit gate can be decomposed as:

$$\hat{U} = e^{i\delta} \hat{R}_{\mathbf{x}}(\alpha) \hat{R}_{\mathbf{y}}(\beta) \hat{R}_{\mathbf{z}}(\gamma), \quad (2.36)$$

where α, β, γ are known as Euler angles. In general, however, it is sufficient to implement rotations around two non-parallel generic axes \mathbf{n}, \mathbf{m} in order to decompose a generic unitary as:

$$\hat{U} = e^{i\delta} \hat{R}_{\mathbf{n}}(\alpha) \hat{R}_{\mathbf{m}}(\beta) \hat{R}_{\mathbf{n}}(\gamma), \quad (2.37)$$

2.3 Many-body quantum systems and Entanglement

Let us consider a system of n qubits. According to the general postulates of quantum mechanics, a basis for the global Hilbert space \mathcal{H} is given by the states

$$|\mathbf{x}\rangle \equiv |x_1\rangle |x_2\rangle \dots |x_n\rangle, \quad (2.38)$$

where the indices $x_i \in \{0, 1\}$ label the local computational basis (as before $|0\rangle$ is the $+1$ eigenvector of \hat{Z} , and $|1\rangle$ is the -1 eigenvector of \hat{Z}). Any state $|\psi\rangle$ can be therefore decomposed as

$$|\psi\rangle = \sum_{\mathbf{x}} \psi_{\mathbf{x}} |\mathbf{x}\rangle = \sum_{x_1} \sum_{x_2} \dots \sum_{x_n} \psi_{x_1 x_2 \dots x_n} |x_1 x_2 \dots x_n\rangle. \quad (2.39)$$

Notice that $\psi_{x_1 x_2 \dots x_n}$ is a multi-dimensional tensor with n indices. Using the tensor graphical representation introduced in Chapter 1 we can therefore represented this many-body wave function as:

$$|\psi\rangle = \sum_{x_1} \dots \sum_{x_n} \begin{array}{c} \psi \\ | \quad \dots \quad | \\ x_1 \quad \dots \quad x_n \end{array} |x_1\rangle \dots |x_n\rangle \quad (2.40)$$

The tensor $\psi_{x_1 x_2 \dots x_n}$ contains 2^n complex numbers encoding all the information about the state of the system. It is usually an extraordinarily complex object. A primary source of this complexity is that typical many-body states $|\psi\rangle$ possess significant quantum correlations among their constituents (qubits). These correlations, known as entanglement, are crucial for understanding quantum behavior and will be further discussed in Section 2.3.3.

2.3.1 Universal set of gates

In addition to single-qubit gates discussed in Section 2.2, exploring the full complexity of the quantum many-body Hilbert space necessitates the use of unitary gates that can make the qubits interact. Obviously this requires the implementation of gates that act on at least two qubits. The fundamental two qubits gate is often considered to be the **CNOT gate**. In fact, we will see in Section 2.3 that CNOT can be used to create prototypical entangled states, i.e. the Bell pairs. The CNOT gate is basically the operator associated to the CNOT tensor (cfr. Chapter 1)

$$\hat{C}_{NOT} = \sum_{i_1, i_2, j_1, j_2} \begin{array}{c} i_1 \text{---} \text{---} j_1 \\ | \\ \oplus \\ | \\ i_2 \text{---} \text{---} j_2 \end{array} |i_1, i_2\rangle\langle j_1, j_2| = |0\rangle\langle 0| \otimes \hat{I} + |1\rangle\langle 1| \otimes \hat{X}$$

This means that the control bit remains unchanged, while the target bit is flipped if the control bit is 1.

At this point, it is appropriate to ask what is the minimal set of gates necessary to have available in order to implement every unitary transformation \hat{U} . In particular, given \hat{U} , one seeks a quantum circuit \hat{U} composed of elementary gates that approximate it arbitrarily well, i.e. with error $\|\hat{U} - \hat{U}\| < \epsilon$ (where $\|\cdot\|$ represents some appropriate matrix norm). It turns out that a set of gates satisfying this universality property is for instance given by:

$$\{\text{CNOT}, \hat{H}, \hat{T}\} \quad (2.41)$$

Indeed:

Example 2.2: Decomposition of n qubits unitaries

Any n qubits unitary \hat{U} can be decomposed into single qubit unitaries and CNOT gates, with a number $O(n^2 4^n)$ of elementary gates.

We emphasize that the decomposition described above is generally inefficient, requiring a number of basic operations that scale exponentially with the number of qubits n . Indeed, implementing a generic unitary transformation \hat{U} involving n qubits necessitates exponentially many elementary gates, as the matrix \hat{U} is determined by $O(4^n)$ real parameters. A fundamental and still unresolved problem in quantum computation is identifying which special classes of unitary transformations can be computed in the quantum circuit model using a polynomial number of elementary gates.

Basically, in order to be universal a set of gates should be able to produce: complex wave function amplitudes, quantum superposition and quantum correlations (entanglement, see next sections). It is less obvious to observe that in addition to these properties, a set of gates must have an additional property, namely, not containing only unitaries from the Clifford group. These are defined as those unitaries that map Pauli matrices to Pauli matrices, i.e., they do not create superposition of Pauli operators. However, we refer the reader to Chapter ... for a discussion of this aspect.

2.3.2 Schmidt decomposition

We introduce a crucial tool in quantum mechanics known as the Schmidt decomposition [NC10], which is fundamentally an application of the singular value decomposition (SVD) introduced in Chapter 1. Let us consider a state $|\psi\rangle$ in a bipartite system, i.e. quantum mechanical system described by the tensor product of the two Hilbert spaces $\mathcal{H}_A \otimes \mathcal{H}_B$. Let us set a basis for the systems A and B , i.e $\mathcal{H}_A = \text{span}\{|a_j\rangle\}$ and $\mathcal{H}_B = \text{span}\{|b_j\rangle\}$. The state $|\psi\rangle$ can now be written as

$$|\psi\rangle = \sum_{i,j} \Phi_{ij} |a_i\rangle |b_j\rangle, \quad (2.42)$$

or graphically as

$$|\psi\rangle = \sum_{i,j} i \text{---} \Phi \text{---} j |a_i\rangle |b_j\rangle = \sum_k k \text{---} \Lambda \text{---} k |\varphi_k^a\rangle |\varphi_k^b\rangle \quad (2.43)$$

In the last graphical equation we carried out a SVD of the matrix $\Phi = U\Lambda V^\dagger$ and we defined new vectors $|\varphi_k^a\rangle = \sum_i U_{ik} |a_i\rangle$ and $|\varphi_k^b\rangle = \sum_i V_{ki}^* |b_i\rangle$. These are usually dubbed *Schmidt vectors*. They are orthonormal, in fact

$$\begin{aligned} \langle \varphi_l^a | \varphi_k^a \rangle &= \sum_{i,j} U_{jl}^* U_{ik} \langle a_j | a_i \rangle = \sum_i (U^\dagger)_{li} U_{ik} = \delta_{lk} \\ \langle \varphi_l^b | \varphi_k^b \rangle &= \sum_{i,j} V_{lj} V_{ki}^* \langle b_j | b_i \rangle = \sum_i V_{li} (V^\dagger)_{ik} = \delta_{lk} \end{aligned} \quad (2.44)$$

The real non-negative entries of the diagonal matrix Λ_{kk} are the so called Schmidt values. The number of non vanishing Schmidt values is usually dubbed Schmidt rank. Schmidt values (and vectors) are related to the eigenvalues (and eigenvectors) of the reduced density matrices of the subsystems A

and B. In fact

$$\begin{aligned}
 \hat{\rho}_A &= \text{Tr}_B |\Psi\rangle\langle\Psi| = \sum_{i,i'} \begin{array}{c} \textcircled{\Phi} \\ | \\ \textcircled{\Phi^*} \end{array} \begin{array}{c} i \\ | \\ i' \end{array} |a_i\rangle\langle a_{i'}| = \sum_k \begin{array}{c} \textcircled{\Lambda} \\ | \\ \textcircled{\Lambda} \end{array} \begin{array}{c} k \\ | \\ k \end{array} |\varphi_k^a\rangle\langle\varphi_k^a| \\
 \hat{\rho}_B &= \text{Tr}_A |\Psi\rangle\langle\Psi| = \sum_{j,j'} \begin{array}{c} \textcircled{\Phi} \\ | \\ \textcircled{\Phi^*} \end{array} \begin{array}{c} j \\ | \\ j' \end{array} |b_j\rangle\langle b_{j'}| = \sum_k \begin{array}{c} \textcircled{\Lambda} \\ | \\ \textcircled{\Lambda} \end{array} \begin{array}{c} k \\ | \\ k \end{array} |\varphi_k^b\rangle\langle\varphi_k^b|
 \end{aligned} \tag{2.45}$$

where last equality means

$$\begin{aligned}
 \hat{\rho}_A &= \sum_k (\Lambda_{kk})^2 |\varphi_k^a\rangle\langle\varphi_k^a| \\
 \hat{\rho}_B &= \sum_k (\Lambda_{kk})^2 |\varphi_k^b\rangle\langle\varphi_k^b|
 \end{aligned} \tag{2.46}$$

Notice that if the initial state $|\psi\rangle$ is normalized, the reduced density matrices $\hat{\rho}_A, \hat{\rho}_B$ are also normalized, in the sense that $\text{Tr}[\hat{\rho}_A] = \text{Tr}[\hat{\rho}_B] = 1$. From this fact it follows that

$$\sum_k \Lambda_{kk}^2 = 1. \tag{2.47}$$

Therefore the square of Schmidt coefficients can be also thought as a discrete probability distribution $p_k = \Lambda_{kk}^2$.

2.3.3 Entanglement

Entanglement is the fundamental property of composite quantum systems, and accounts for the intrinsic quantum correlations among the constituencies (for instance qubits). Consider for instance a composite quantum system $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$ and a pure state $|\psi\rangle \in \mathcal{H}$. A natural question arises: can $|\psi\rangle$ be factorized as a product of states on A and B , i.e. as $|\psi_A\rangle \otimes |\psi_B\rangle$?

Definition 2.1: Bipartite product state

Given a pure state $|\psi\rangle \in \mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$ we define it to be a bipartite product state if $|\psi\rangle = |\psi_A\rangle \otimes |\psi_B\rangle$, otherwise we say the two partitions A and B of $|\psi\rangle$ are entangled.

Example 2.3: Bell states, Bell measurements and quantum teleportation

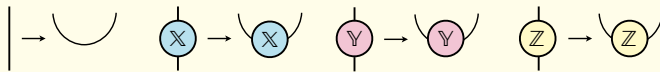
A prototypical example of entangled states is given by Bell pairs. Consider $N = 2$ qubits and the following four states

$$\begin{aligned}
 |1\rangle &= \frac{|00\rangle + |11\rangle}{\sqrt{2}} \\
 |X\rangle &= \frac{|01\rangle + |10\rangle}{\sqrt{2}} \\
 |Y\rangle &= i \frac{|01\rangle - |10\rangle}{\sqrt{2}} \\
 |Z\rangle &= \frac{|00\rangle - |11\rangle}{\sqrt{2}}
 \end{aligned}
 \tag{2.48}$$

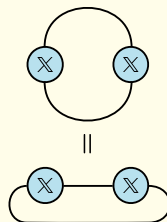
The reason we adopted the names $1, X, Y, Z$ becomes clear when one notices that these states are essentially the single qubit Pauli operators reshaped as vectors. Indeed we have for instance

$$\frac{1}{\sqrt{2}}(\hat{X})_{x_0, x_1} = \langle x_0, x_1 | X \rangle,
 \tag{2.49}$$

meaning that components of state $|X\rangle$ match with matrix elements of \hat{X} , a part for a normalization factor. The latter is chosen in order to make Bell states orthonormal. The reshaping can be graphically represent as follows

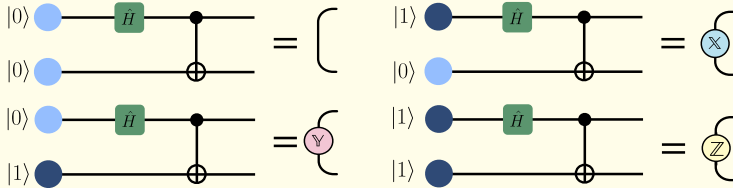


In this way, it is very easy to realize that inner scalar product between states is equal to Frobenius inner product of the corresponding Pauli matrices. For instance:

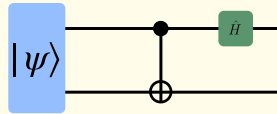


and therefore $\langle X | X \rangle = \frac{1}{2} \text{Tr}[X^\dagger X] = \frac{1}{2} \text{Tr}[\hat{1}] = 1.$

The set of Bell states form an orthonormal basis for the Hilbert space of a two qubits system. Consequently, they can be generated by applying a suitable unitary transformation \hat{U} to the states of standard computational basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. In particular, it is easy to verify that one can realize \hat{U} with a circuit consisting of one Hadamard gate, followed by a CNOT, i.e.

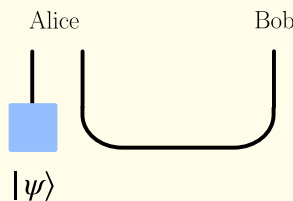


Thus, if an unknown Bell state $|\psi\rangle$ is prepared one can determine which of the four Bell states the two qubits are in by first applying \hat{U}^\dagger as follows

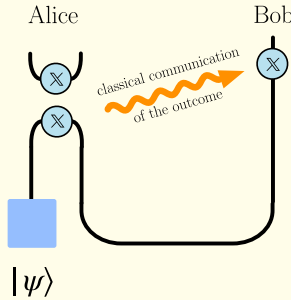


and then perform a projective measurement of \hat{Z} on both qubits. If, for instance we find $|0\rangle$ and $|0\rangle$, that means the unknown state was $|1\rangle$.

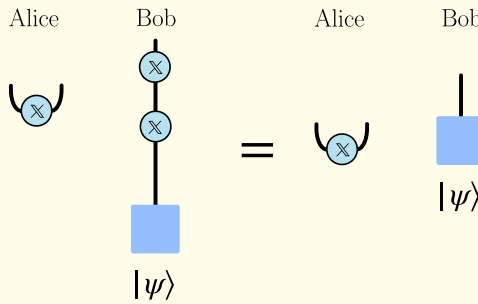
To conclude this paragraph, we illustrate an important use of entangled Bair pairs: the quantum teleportation. Suppose Alice possesses a qubit, yet she is unaware of its state $|\psi\rangle$. Bob is in urgent need of this qubit. However, Alice is limited to sending only classical bits of information to Bob. A protocol to teleport the qubit state $|\psi\rangle$ can be realized by exploiting a shared Bell pair. Indeed, suppose Alice and Bob share each one qubit of the state $|1\rangle$. Our initial set up can be therefore described by the following sketch



Then suppose Alice perform a Bell measurement of the Bell state qubit and the qubit to be teleported. This yields one of four possible measurement outcomes and project the two qubits in $|1\rangle$, $|X\rangle$, $|Y\rangle$ or $|Z\rangle$. Using a classical channel Alice can communicate the outcome to Bob, which can act with on its Bell qubit with the corresponding Pauli operator. The process can be summarized as follows:



in the case in which the outcome is X . Now notice the following graphical simplification



where we used the fact that $\hat{X}^2 = \hat{1}$. Thus, Bob has successfully received the state $|\psi\rangle$, effectively teleporting it to his qubit!

Now given a pure state $|\psi\rangle$, what quantity can we use to discriminate whether it is a product state or an entangled state? To answer this question, we first provide a definition.

Definition 2.2: von Neumann entropy

Given any density matrix $\hat{\rho}$, we define the von Neumann entropy as

$$S(\hat{\rho}) = -\text{Tr}[\hat{\rho} \log(\hat{\rho})], \quad (2.50)$$

with \log being matrix logarithm. This can be defined by using eigenstates $\{|\phi_k\rangle\}$ of $\hat{\rho}$ as a basis. We have $\hat{\rho} = \sum_k p_k |\phi_k\rangle\langle\phi_k|$, where p_k are the corresponding eigenvalues (notice that $p_k \geq 0$ and $\sum_k p_k = 1$). In this basis, the von Neumann entropy is defined as

$$S(\hat{\rho}) = -\sum_k p_k \log p_k. \quad (2.51)$$

The von Neumann entropy is a crucial quantity in quantum information, quantum computing and quantum many-body theory. The following properties holds:

1. invariance under unitary transformations: $S(\hat{U}^\dagger \hat{\rho} \hat{U}) = S(\hat{\rho})$.
2. non negativity: $S(\hat{\rho}) \geq 0$ and $S(\hat{\rho}) = 0$ if and only if the state is pure, i.e. $\hat{\rho} = |\psi\rangle\langle\psi|$. Indeed in this case, $\hat{\rho}$ is a projector and has a single eigenvalue 1, while all the others p_k are 0.
3. upper bound: $S(\hat{\rho}) \leq \log d$, where d is the dimension of the Hilbert space. The equality holds only for $\hat{\rho} = \frac{1}{d}$, which is named maximally mixed state. Indeed, in this case all eigenvalues p_k of $\hat{\rho}$ are $1/d$ and the sum Eq. (2.51) is maximized.
4. concavity: given a set of real positive numbers π_i such that $\sum_i \pi_i = 1$, then: $S(\sum_i \pi_i \hat{\rho}_i) \geq \sum_i \pi_i S(\hat{\rho}_i)$.
5. sub-additivity: if we have a composite system $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$ then $S(\hat{\rho}_{AB}) \leq S(\hat{\rho}_A) + S(\hat{\rho}_B)$, where $\hat{\rho}_A = \text{Tr}_B[\hat{\rho}_{AB}]$, $\hat{\rho}_B = \text{Tr}_A[\hat{\rho}_{AB}]$ and $\hat{\rho}_{AB}$ is the global density operator of the two systems. Furthermore, equality holds when the systems are uncorrelated, i.e. $S(\hat{\rho}_A \otimes \hat{\rho}_B) = S(\hat{\rho}_A) + S(\hat{\rho}_B)$.

Because of properties 2 and 3, we observe that the von Neumann entropy measures the intrinsic level of uncertainty encoded in the probability distribution of eigenvalues p_k of $\hat{\rho}$. When there is no uncertainty, for instance, when $p_1 = 1$ and all other $p_k = 0$, we have $S(\hat{\rho}) = 0$. In contrast, when there are d

equally distributed possibilities, meaning $p_k = \frac{1}{d}$ for each k , the uncertainty is maximized and $S(\hat{\rho}) = \log d$.

Definition 2.3: Entanglement entropy

The von Neumann entropy $S(\hat{\rho}_A)$ of the reduced density matrix $\hat{\rho}_A$ of a composite system is commonly referred to as the **entanglement entropy**.

Notice that if we assume the system to be in a pure state $|\psi\rangle$, by using the Schmidt decomposition (Eq. (2.46)) we find for $S(\hat{\rho}_A)$ an expression as Eq. (2.51) in terms of the eigenvalues $p_k = \Lambda_{kk}^2$ of the reduced density matrix. We have therefore

$$\begin{aligned} S_{A/B} &= -\text{Tr}_A \hat{\rho}_A \log \hat{\rho}_A = -\text{Tr}_B \hat{\rho}_B \log \hat{\rho}_B \\ &= -\sum_k p_k \log p_k = -\sum_k \Lambda_{kk}^2 \log \Lambda_{kk}^2, \end{aligned} \quad (2.52)$$

Notice that a bound χ in the Schmidt rank of the state $|\psi\rangle$ reflects in a maximum value of the entanglement entropy. Indeed if $p_k > 0$ only for $k \geq \chi$ it is easy to show that the entanglement is upper bounded as $S_{A/B} \leq \log \chi$. In the case of product states $|\psi\rangle = |\psi_A\rangle \otimes |\psi_B\rangle$, one has a single positive Schmidt value, and therefore $\chi = 1$. From this we find that $S_{A/B} = 0$. Thus, entanglement entropy is zero for product states and positive for entangled states.

Reflecting on property 5 of the von Neumann, we observe that we found a measure that is non-negative and reaches zero precisely when the two systems are uncorrelated ($\rho_{AB} = \rho_A \otimes \rho_B$). Hence, we can employ this as a metric for the overall level of correlations.

Definition 2.4: Mutual information

Given a composite quantum systems we define the mutual information as

$$I(\hat{\rho}_{AB}) = S(\hat{\rho}_A) + S(\hat{\rho}_B) - S(\hat{\rho}_{AB}). \quad (2.53)$$

It is not difficult to show that if $\hat{\rho}_{AB}$ is a pure state then $S(\hat{\rho}_A) = S(\hat{\rho}_B)$. Consequently, since $S(\hat{\rho}_{AB}) = 0$, the mutual information is $I(\hat{\rho}_{AB}) = 2S(\hat{\rho}_A) = 2S(\hat{\rho}_B)$.

We have so far provided definitions of entanglement based on rather simple arguments. However, it is possible to give a much deeper characterization of entanglement within the framework of the so-called quantum resource theories. In this context, entanglement is defined through measures known

as entanglement monotones. An entanglement monotone is a non-negative function whose value does not increase under a certain class of quantum transformations that are axiomatically assumed not to create entanglement. These transformations are known as Local Operations and Classical Communication.

Definition 2.5: Local Operations and Classical Communication

Local Operations and Classical Communication (LOCC) is a central concept in quantum information theory, especially in the study of quantum entanglement and quantum state manipulation [NC00; Chi+14]. The operations allowed under LOCC are:

1. **Local Operations:** each party can perform any quantum operation on their own subsystem. These operations include local unitary transformations, measurements, and the addition or removal of ancillary qubits.
2. **Classical Communication:** the parties can communicate with each other using classical channels. This communication can be used for instance to coordinate their local operations based on the measurement outcomes.

Importantly, LOCC do not allow for global unitary transformations and therefore the creation of genuine quantum correlations between parties, i.e. the generation of entanglement. As a result, LOCC cannot increase the amount of entanglement present in the system. However, they enable the manipulation and utilization of existing entanglement. For instance, LOCC can be used to concentrate entanglement, convert one form of entanglement into another, and distribute entanglement across different parties [WZ82; Ben+96a; Ben+96b; HHH99].

Essentially, entanglement can be defined as a kind of quantum correlation that cannot be created by LOCC alone. For pure states $|\psi\rangle$, the von Neumann entropy $S(\hat{\rho}_A)$ is recognized as a good measure of entanglement in this sense. In contrast, various functions have been developed to quantitatively measure entanglement in mixed states. Although a discussion of these aspects goes beyond the scope of this book, we can mention another highly relevant measure.

Definition 2.6: Negativity

The negativity of a subsystem A of a composite system in a mixed state $\hat{\rho}$ is defined as

$$N(\hat{\rho}) = \frac{\|\hat{\rho}^{T_B}\|_1 - 1}{2} \quad (2.54)$$

where T_B represent the partial transpose on system B . The trace norm $\|\cdot\|_1$ is defined as the sum of the absolute values of the eigenvalues. The logarithmic negativity is instead:

$$E(\hat{\rho}) = \log \|\hat{\rho}^{T_B}\|_1 \quad (2.55)$$

2.4 Tensor network representation of quantum states

Consider a many body wave function. Simply specifying its coefficients in a given local basis (for instance, the canonical computational basis in a quantum computing setup) does not provide any insight into the structure of entanglement among its constituents. In fact:

- (i) We anticipate that this structure will vary depending on the dimensionality of the system, resulting in differences between 1D, 2D, or higher-dimensional scenarios.
- (ii) Indeed, the structure of entanglement is influenced by more subtle factors such as the criticality of the state and its correlation length.

To be more precise, let us consider a generic state describing n qubits, namely $|\psi\rangle \in \mathcal{H}^{\otimes n}$, where \mathcal{H} is the local Hilbert space for a single qubit. When not differently specified, we always consider $\mathcal{H} = \text{span}\{|0\rangle, |1\rangle\}$, in terms of the eigenstates of the Pauli gate \hat{Z} . In particular, in quantum computation it is customary to set $|0\rangle$ as the $+1$ eigenvector of \hat{Z} , and $|1\rangle$ as the -1 eigenvector of \hat{Z} .

The many body vector decomposes as

$$|\psi\rangle = \sum_{x_1} \cdots \sum_{x_n} \psi_{x_1 \dots x_n} |x_1\rangle \dots |x_n\rangle \quad (2.56)$$

where, unfortunately, the order- n tensor ψ_{x_1, \dots, x_n} lacks explicit information about the entanglement properties. Therefore, it is desirable to find a rep-

resentation of quantum states where this information is explicit and readily accessible.

Tensor Networks provide a framework where this information is readily available; in a sense, Tensor Network states are inherently represented in a specific “entanglement representation”. The primary reason Tensor Networks serve as a key description of quantum many-body states is that the Hilbert space becomes exceedingly large, scaling exponentially with the number of local constituents (as 2^n in this case). Additionally, representing the state using coefficients in the computational basis is highly inefficient. The classical paradox example arises when we directly compare a generic random state with a random product state. In the case of a random product state, the associated tensor would simply be the product:

$$|\phi\rangle = \sum_{x_1} \cdots \sum_{x_n} \begin{array}{c} \phi_1 \\ | \\ x_1 \end{array} \begin{array}{c} \phi_2 \\ | \\ x_2 \end{array} \cdots \cdots \begin{array}{c} \phi_n \\ | \\ x_n \end{array} |x_1\rangle \cdots |x_n\rangle \quad (2.57)$$

When fused together, the local tensors ϕ_j collectively generate an order- n tensor that closely resembles the generic tensor ψ_{x_1, \dots, x_n} , being its entries basically indistinguishable from those of a completely random order- n tensor. Despite this, the complexity of a product state scales linearly with the number of qubits, with only $2n$ independent complex coefficients.

Actually, not all states within the Hilbert space of a many-body quantum system are equally important; some are more relevant than others. For instance, physical Hamiltonians often feature local interactions among constituents. The locality of these interactions leads to significant consequences, such as the emergence of area laws for entanglement in low-energy states of local gapped Hamiltonians. In other words, these states exhibit short-range correlations.

Definition 2.7: Area law of bipartite entanglement entropy

In quantum mechanics, a state adheres to an area law when the dominant contribution to its **entanglement entropy** increases at most proportionally to the boundary between two partitions. In one-dimensional systems, this implies, for instance, that the bipartite entanglement entropy would be strictly bounded.

Moreover, locally manipulating (via unitary gates) shortly correlated one-dimensional states yields a class of highly valuable states for harnessing quantum resources and implementing quantum computing algorithms. In fact, the

manifold generated after a series of local operations, whose number remains of the order of the system size, is exponentially small compared to the total Hilbert space. Matrix Product States (MPS) precisely provide an efficient ansatz for this portion of the Hilbert space. Numerous foundational articles and comprehensive reviews have been published on these topics over the past two decades. Here, we mention just a few of them, [Pér+07; McC07; Sch11; Orú14; BC17; Sil+19; Ran+20; Eve22], and we highly recommend that readers explore these works and seek out additional references cited in their respective bibliographies.

2.4.1 Matrix Product States (MPS)

The ingredient which stays at the basis of a MPS representation of a many-body quantum state is strictly related to the way the many-body coefficient ψ_{x_1, \dots, x_n} entering in Eq. (2.56) can be manipulated as a tensor, basically as has been already shown in Chapter 1.

Indeed, an efficient way of reshuffling the parameters of a quantum wave function, highlighting its entanglement content, is given by its *Schmidt decomposition* (see Section 2.3.2), which is a compact way of rewriting a quantum state of a system living in a bipartite universe.

Let us take the state in Eq. (2.56) and rewrite it as it is leaving in a the tensor product of two Hilbert spaces $\mathcal{H}_A \otimes \mathcal{H}_B$, such that $\mathcal{H}_A = \text{span}\{|a_j\rangle\}$ and $\mathcal{H}_B = \text{span}\{|b_j\rangle\}$. This can be done by arbitrarily fusing the indices of the tensor ψ_{x_1, \dots, x_n} and reshape it as a matrix. Then following what have been done in Section 2.3.2 the state can be rewritten as

$$|\psi\rangle = \sum_{ij} i \text{---} \Phi \text{---} j |a_i\rangle |b_j\rangle = \sum_k k \text{---} \Lambda \text{---} k |\varphi_k^a\rangle |\varphi_k^b\rangle \quad (2.58)$$

with *Schmidt vectors* $|\varphi_k^a\rangle$ and $|\varphi_k^b\rangle$, and real non-negative *Schmidt values* Λ_{kk} .

Notice that, in the original representation of the state we were using $\dim(\mathcal{H}_A) \times \dim(\mathcal{H}_B)$ complex numbers to describe the full wave function. However, if χ is the number of non-vanishing Schmidt (or singular) values, than, in the Schmidt basis, in addition to these real numbers, we are only using $\chi \times [\dim(\mathcal{H}_A) + \dim(\mathcal{H}_B)]$ complex numbers to encode the wave function. Therefore, if χ is relatively small, or only few of the entries of Λ_{kk} are relevant, we basically end up with a systematic way of reducing the complexity of the many-body state still retaining the leading correlations across the entire system. In fact, from the relation with the reduced density matrices, we can easily compute the bipartite **entanglement entropy** between subsystem A

and B

$$S_{A/B} = -\text{Tr}_A \hat{\rho}_A \log \hat{\rho}_A = -\text{Tr}_B \hat{\rho}_B \log \hat{\rho}_B = -\sum_k \Lambda_{kk}^2 \log \Lambda_{kk}^2, \quad (2.59)$$

which gives an indication of how much the original wave function is entangled. As already mentioned in section 2.3.2, a bound in the number χ of Schmidt values reflects in a maximum value of the entanglement entropy $S_{A/B} \leq \log \chi$, where we used the fact that $\sum_k \Lambda_{kk}^2 = 1$ due to the state normalisation.

However, if χ remains too large for the wave function to be efficiently handled by a classical computer, then we might question whether faithful compression is feasible. The Schmidt decomposition provides a solution to this fundamental query; specifically, we obtain an approximate state by retaining only the largest $\tilde{\chi} < \chi$ Schmidt vectors. As a matter of fact, the following state

$$|\tilde{\psi}\rangle = \sum_{k=1}^{\tilde{\chi}} \tilde{\Lambda}_{kk} |\varphi_k^a\rangle |\varphi_k^b\rangle \quad (2.60)$$

is exactly the one that minimise the norm-2 distance from the original state, namely $\| |\psi\rangle - |\tilde{\psi}\rangle \|^2$. Where in Eq. (2.60), in order to keep the state normalized, we re-scaled the original singular values as $\tilde{\Lambda}_{kk} \equiv \Lambda_{kk} / (\sum_{k=1}^{\tilde{\chi}} \Lambda_{kk}^2)^{1/2}$.

Now, since the bipartition we were considering was arbitrary, we can extend this reasoning to each qubit (or generic local Hilbert space) constituting the local degrees of freedom of the entire wave function. In practice, we can apply a process similar to what was described in Chapter 1 for decomposing a generic order- n tensor, but this time to decompose a quantum state into a one-dimensional ansatz where locality is preserved and entanglement compression is naturally achieved. This ansatz is commonly referred to as the **Matrix Product State**.

Definition 2.8: Matrix Product State

We say that a quantum many-body state is a **Matrix Product State** when its wave function is written as a matrix product tensor network (cfr. Chapter 1), thus having

$$\begin{aligned} |\psi\rangle &= \sum_{x_1} \cdots \sum_{x_n} \begin{array}{c} \textcircled{A_1} - \textcircled{A_2} - \cdots - \textcircled{A_n} \\ | \quad | \quad \quad \quad | \\ x_1 \quad x_2 \quad \quad \quad x_n \end{array} |x_1\rangle \cdots |x_n\rangle \\ &= \begin{array}{c} \textcircled{A_1} - \textcircled{A_2} - \cdots - \textcircled{A_n} \end{array} \end{aligned} \quad (2.61)$$

where in the last passage we have defined the following **vector-valued tensors**

$$\text{---} |A_j\rangle \text{---} = \sum_{x_j} \text{---} A_j \text{---} \downarrow_{x_j} |x_j\rangle$$

whose legs span auxiliary spaces whose dimension is typically termed the **bond or auxiliary dimension**, and denoted by χ . These legs guarantee the interconnection between the different local tensors, thereby enabling the encoding of correlations throughout the system. This capability extends beyond a non-correlated product state, which effectively corresponds to an MPS with bond dimension $\chi = 1$.

Given this representation, the norm of an MPS can be easily calculated as

$$\langle \psi | \psi \rangle = \begin{array}{ccccccc} \text{---} A_1 \text{---} & \text{---} A_2 \text{---} & \dots & \dots & \text{---} A_n \text{---} \\ | & | & & & | \\ \text{---} A_1^* \text{---} & \text{---} A_2^* \text{---} & \dots & \dots & \text{---} A_n^* \text{---} \end{array} \quad (2.62)$$

with a computational cost $O(n\chi^3)$. Here A_j^* is indicating the conjugate tensor. Let us notice that since the bond dimension essentially denotes the dimension of the auxiliary space between two physical lattice sites, such as j and $j + 1$, it could potentially vary with position, i.e. $\chi \rightarrow \chi_j$.

Definition 2.9: MPS Transfer Matrix

One of the central object in any MPS based description of a physical state is the so called **Transfer Matrix**. It is defined as

$$\mathcal{T}_j = \sum_x A_j^x \otimes (A_j^x)^* = \begin{array}{c} \text{---} A_j \text{---} \\ | \\ \text{---} A_j^* \text{---} \end{array} \quad (2.63)$$

as “super” operator which acts on the space of $\chi_j \times \chi_j$ matrices on its right or $\chi_{j-1} \times \chi_{j-1}$ matrices on its left, is defined as a completely positive map [Cho75], with the MPS matrices serving as Kraus operators. Notably, the transfer matrix exhibits the property that its leading eigenvalue is a positive number η . In general, this correspond to a

leading left (and right) eigenvector such that

$$\begin{array}{c} \text{---} A_j \text{---} \\ | \\ l_j \end{array} \begin{array}{c} \text{---} \\ | \\ A_j^* \text{---} \\ | \\ l_j \end{array} = \eta_j \begin{array}{c} \text{---} \\ | \\ l_j \end{array} \quad \begin{array}{c} \text{---} A_j \text{---} \\ | \\ r_j \end{array} \begin{array}{c} \text{---} \\ | \\ A_j^* \text{---} \\ | \\ r_j \end{array} = \eta_j \begin{array}{c} \text{---} \\ | \\ r_j \end{array}$$

where both l and r are hermitian matrices with non-negative eigenvalues and therefore which can be square decomposed as $l = L^\dagger L$ and $r = RR^\dagger$.

MPS gauge freedom — Whenever a Matrix Product Tensor ansatz is used to describe the many-body wave function of a quantum state, while the state is uniquely determined by the tensors A_j , the reverse is not necessarily true, as distinct tensors can yield the same physical state. This becomes apparent when considering the following gauge transformation:

$$\begin{array}{c} \text{---} A_j \text{---} \\ | \end{array} \Rightarrow \begin{array}{c} \text{---} X_{j-1} \text{---} A_j \text{---} X_j^{-1} \text{---} \\ | \end{array} \quad (2.64)$$

where X_j is an invertible matrix. Essentially, the previous transformation introduces a resolution of the identity $X_j^{-1}X_j$ in each local auxiliary bond of the MPS wave function, effectively preserving the integrity of the entire state. Indeed, it has been demonstrated [Pér+07; Pér+08] that this is the sole freedom in the parametrization, and it can be constrained (partially) by imposing canonical forms on the MPS tensors A_j .

As a matter of fact, for a finite system MPS with tensors $\{A_1, \dots, A_n\}$, starting from the leftmost tensor, and proceeding iteratively on j , we can follow what have been done in Chapter 1, but now employing the **QR decomposition** of the tensors in the following recursive way

$$\begin{array}{c} \text{---} A_j \text{---} \\ | \end{array} = \begin{array}{c} \tilde{A}_1 \end{array} \begin{array}{c} \text{---} R_1 \text{---} \\ | \end{array} \\
 \begin{array}{c} \text{---} R_{j-1} \text{---} A_j \text{---} \\ | \end{array} = \begin{array}{c} \tilde{A}_j \end{array} \begin{array}{c} \text{---} R_j \text{---} \\ | \end{array} \quad (2.65)$$

where we can identify the required Gauge transformation exactly with the upper triangular matrices R_j , such that the **Transfer Matrix** constructed

with the new tensors \tilde{A}_j has as leading left eigenvector the identity matrix, diagrammatically

$$\left[\begin{array}{c} \tilde{A}_j \\ \tilde{A}_j^* \end{array} \right] = \left[\right]$$

A similar iterative approach could lead to a completely right normalized decomposition where the transfer matrix now satisfies the following property

$$\left[\right] = \left[\begin{array}{c} \tilde{B}_j \\ \tilde{B}_j^* \end{array} \right]$$

Mixing the two strategies, and thus orthogonalising in iterative way the left-subsystems tensors from the left boundary, and the right subsystems tensors for the right boundary, leads to the following, extremely useful **Mixed-Canonical decomposition** of an MPS

$$\begin{aligned} |\psi\rangle &= \sum_{x_1, \dots, x_n} \left[\begin{array}{c} A_1 \\ x_1 \end{array} \right] \cdots \left[\begin{array}{c} A_{j-1} \\ x_{j-1} \end{array} \right] \left[\begin{array}{c} M_j \\ x_j \end{array} \right] \left[\begin{array}{c} B_{j+1} \\ x_{j+1} \end{array} \right] \cdots \left[\begin{array}{c} B_n \\ x_n \end{array} \right] |x_1, \dots, x_n\rangle \\ &= \sum_{\alpha, x_j, \beta} \alpha \left[\begin{array}{c} M_j \\ x_j \end{array} \right] \beta |\alpha\rangle |x_j\rangle |\beta\rangle \end{aligned} \tag{2.66}$$

with left and right vectors

$$|\alpha\rangle = \sum_{x_1, \dots, x_{j-1}} \left[\begin{array}{c} A_1 \\ x_1 \end{array} \right] \cdots \left[\begin{array}{c} A_{j-1} \\ x_{j-1} \end{array} \right] \alpha |x_1, \dots, x_{j-1}\rangle \tag{2.67}$$

$$|\beta\rangle = \sum_{x_{j+1}, \dots, x_n} \beta \left[\begin{array}{c} B_{j+1} \\ x_{j+1} \end{array} \right] \cdots \left[\begin{array}{c} B_n \\ x_n \end{array} \right] |x_{j+1}, \dots, x_n\rangle$$

which satisfy $\langle \alpha | \alpha' \rangle = \delta_{\alpha\alpha'}$ and $\langle \beta | \beta' \rangle = \delta_{\beta\beta'}$ thanks to tensor properties.

MPS overlap & expectation values — Computing overlaps between quantum states or expectation values of a local observable is generally an exponentially hard problem as the system size increases. However, in the MPS formalism, this problem becomes polynomially complex because the complexity is essentially constrained by the bond dimension of the network representing the state. As we have seen with the norm of an MPS, the overlap between two MPS states

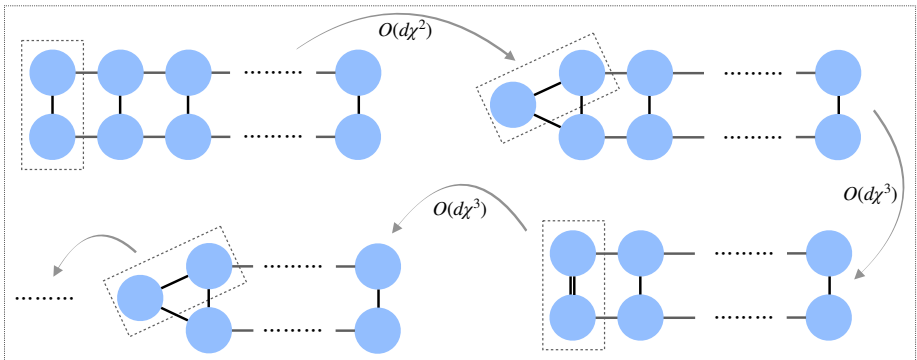
$$|\psi\rangle = \sum_{x_1} \cdots \sum_{x_n} \begin{array}{c} \text{---} A_1 \text{---} A_2 \text{---} \cdots \cdots \text{---} A_n \text{---} \\ | \quad | \quad \quad \quad | \\ x_1 \quad x_2 \quad \quad \quad x_n \end{array} |x_1\rangle \cdots |x_n\rangle$$

$$|\phi\rangle = \sum_{x_1} \cdots \sum_{x_n} \begin{array}{c} \text{---} B_1 \text{---} B_2 \text{---} \cdots \cdots \text{---} B_n \text{---} \\ | \quad | \quad \quad \quad | \\ x_1 \quad x_2 \quad \quad \quad x_n \end{array} |x_1\rangle \cdots |x_n\rangle$$

is easily computed as

$$\langle \phi | \psi \rangle = \begin{array}{c} A_1 \quad A_2 \quad \cdots \quad A_n \\ | \quad | \quad \quad \quad | \\ B_1^* \quad B_2^* \quad \cdots \quad B_n^* \end{array}$$

where we are not assuming any specific normalization of the MPS. Let us stress again that the compression scheme of the Tensor Network is very important to achieve an optimal procedure. In fact, for any open boundary tensor network, the contraction should start from one of the two boundaries (for example, from the left), as outlined in the following steps:



where we assume d is the local physical dimension and χ is the auxiliary dimension of both the MPS wavefunctions.

Evaluating expectation values is as straightforward as computing overlaps. To keep the discussion concrete, let's suppose we want to compute the average of a Pauli string.

$$\hat{\sigma}_1 \hat{\sigma}_2 \cdots \hat{\sigma}_n = \begin{array}{c} y_1 \\ | \\ \textcircled{\sigma_1} \\ | \\ x_1 \end{array} \begin{array}{c} y_2 \\ | \\ \textcircled{\sigma_2} \\ | \\ x_2 \end{array} \cdots \begin{array}{c} y_n \\ | \\ \textcircled{\sigma_n} \\ | \\ x_n \end{array} |x_1, x_2, \dots, x_n\rangle \langle y_1, y_2, \dots, y_n|$$

where $\hat{\sigma}_j \in \{\hat{I}, \hat{X}, \hat{Y}, \hat{Z}\}$. However, what we are going to discuss is valid for the tensor product of any local observable. As is typical with tensor networks, the expectation value of such an operator computed on an MPS state reduces to the following network contraction:

$$\langle \psi | \hat{\sigma}_1 \hat{\sigma}_2 \cdots \hat{\sigma}_n | \psi \rangle = \begin{array}{c} \textcircled{A_1} \textcircled{A_2} \cdots \textcircled{A_n} \\ | \quad | \quad \cdots \quad | \\ \textcircled{\sigma_1} \textcircled{\sigma_2} \cdots \textcircled{\sigma_n} \\ | \quad | \quad \cdots \quad | \\ \textcircled{A_1^*} \textcircled{A_2^*} \cdots \textcircled{A_n^*} \end{array}$$

Notice that, in this case as well, the contraction scheme is important. We must follow the same procedure outlined for the computation of the overlap, which involves starting from one of the two boundaries and sequentially updating the boundary matrix. Therefore, when adding the generic lattice site k , we compute the updating step as follows:

$$\begin{array}{c} \text{---} \textcircled{A_k} \text{---} \\ | \\ \boxed{\mathbb{L}_{k-1}} \\ | \\ \textcircled{\sigma_k} \\ | \\ \textcircled{A_k^*} \text{---} \end{array} = \boxed{\mathbb{L}_k}$$

which basically correspond to compute $\mathbb{L}_k = \sum_{i,j} (\sigma_k)_{ij} (A_k^i)^\dagger \cdot \mathbb{L}_{k-1} \cdot A_k^j$ thus keeping the computational cost of a single step $O(d^2 \chi^3)$. In such a recursive algorithm, $\mathbb{L}_0 = (1)$ is a 1×1 identity matrix (or a scalar if you wish); and finally, $\mathbb{L}_n = \langle \psi | \hat{\sigma}_1 \hat{\sigma}_2 \cdots \hat{\sigma}_n | \psi \rangle$ is the desired expectation value.

2.4.2 Matrix Product Operator (MPO)

Matrix Product Operators were originally introduced by Juan Ignacio Cirac [VGC04] and by Guifré Vidal [ZV04] as a Tensor Network Ansatz for density matrices, and they basically constitute the operator analogue of MPS.

Let us consider from the moment n lattice sites ordered in a one-dimensional chain. Each local Hilbert space \mathcal{H} having dimension d (with $d = 2$ for qubits) so that quantum many-body state is living in \mathcal{H}^n with total dimension d^n .

Any **linear operator** mapping a state $|\psi\rangle \in \mathcal{H}^n$ to another state $|\psi'\rangle \in \mathcal{H}^n$ is an object living in the tensor product $\mathcal{H}^n \otimes \mathcal{H}^{*n}$, so as in general we have

$$\hat{O} = \sum_{x_1, \dots, x_n} \sum_{y_1, \dots, y_n} \begin{array}{c} y_1 \quad \dots \quad y_n \\ | \quad \quad \quad | \\ \boxed{O} \\ | \quad \quad \quad | \\ x_1 \quad \dots \quad x_n \end{array} |x_1, \dots, x_n\rangle \langle y_1, \dots, y_n| \quad (2.68)$$

so it is natural to inquire whether the operator can be represented using the same tensor decomposition discussed in the previous section. Indeed, blindly applying the procedure outlined in Chapter 1 for the tensor in Eq. (2.68) leads to a similar decomposition of the operator's coefficient. We can fuse the indices corresponding to each local Hilbert space (x_j, y_j) and then perform the standard SVD procedure, resulting in:

$$\hat{O} = \sum_{x_1, \dots, x_n} \sum_{y_1, \dots, y_n} \begin{array}{c} y_1 \\ | \\ \boxed{o_1} \\ | \\ x_1 \end{array} - \begin{array}{c} y_2 \\ | \\ \boxed{o_2} \\ | \\ x_2 \end{array} - \dots - \begin{array}{c} y_n \\ | \\ \boxed{o_n} \\ | \\ x_n \end{array} |x_1, \dots, x_n\rangle \langle y_1, \dots, y_n| \quad (2.69)$$

where we reshuffled back each fused index, so that $O_j^{x_j y_j}$ are matrices like the $A_j^{x_j}$ entering in the MPS, with the key distinction staying in the fact that, as representations of operators, they require both outgoing and incoming physical indices. Notice that, in general, a systematic procedure based on iterative SVD from a many-body representation of the operator itself requires an exponentially large amount of resources and is unlikely to be feasible in practical scenarios. Instead, there are cases where an MPO representation of an operator is easy to construct in a systematic way.

The analogues of product state in the realm of the operators are *tensor product of local operators*, which have the straightforward representation

$$\hat{o}_1 \hat{o}_2 \dots \hat{o}_n = \sum_{x_1, \dots, x_n} \sum_{y_1, \dots, y_n} \begin{array}{c} y_1 \\ | \\ \boxed{o_1} \\ | \\ x_1 \end{array} \begin{array}{c} y_2 \\ | \\ \boxed{o_2} \\ | \\ x_2 \end{array} \dots \begin{array}{c} y_n \\ | \\ \boxed{o_n} \\ | \\ x_n \end{array} |x_1, \dots, x_n\rangle \langle y_1, \dots, y_n|$$

with local bond dimension equal to one. As a further example, an highly non-local operator whose MPO representation can be written straightforwardly is the **projector to a MPS state**

$$|\psi\rangle\langle\psi| = \sum_{x_1, \dots, x_n} \sum_{y_1, \dots, y_n} \begin{array}{c} y_1 \quad y_2 \quad \dots \quad y_n \\ | \quad | \quad \dots \quad | \\ \textcircled{A_1^*} - \textcircled{A_2^*} - \dots - \textcircled{A_n^*} \\ | \quad | \quad \dots \quad | \\ \textcircled{A_1} - \textcircled{A_2} - \dots - \textcircled{A_n} \\ | \quad | \quad \dots \quad | \\ x_1 \quad x_2 \quad \dots \quad x_n \end{array} |x_1, \dots, x_n\rangle\langle y_1, \dots, y_n|$$

where now the by fusing the two auxiliary spaces, we can identify the local MPO matrices $W_j^{x_j y_j} = A_j^{x_j} \otimes (A_j^*)^{y_j}$ with bond dimension given by the square of the MPS bond dimension.

Example 2.4: Multiply MPO to MPO or MPS

Let us for a moment forget about normalisation and suppose we have the following MPS and MPO:

$$|\psi\rangle = \begin{array}{c} \textcircled{A_1} - \textcircled{A_2} - \dots - \textcircled{A_n} \\ | \quad | \quad \dots \quad | \\ x_1 \quad x_2 \quad \dots \quad x_n \end{array} |x_1, \dots, x_n\rangle$$

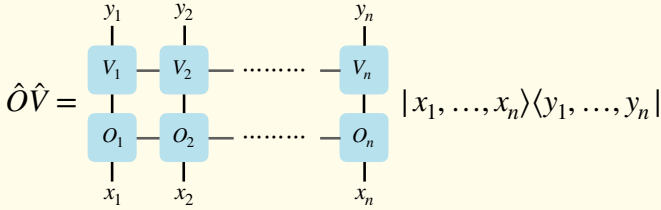
$$\hat{O} = \begin{array}{c} \textcircled{O_1} - \textcircled{O_2} - \dots - \textcircled{O_n} \\ | \quad | \quad \dots \quad | \\ y_1 \quad y_2 \quad \dots \quad y_n \\ | \quad | \quad \dots \quad | \\ x_1 \quad x_2 \quad \dots \quad x_n \end{array} |x_1, \dots, x_n\rangle\langle y_1, \dots, y_n|$$

with $\dim(A_j^{x_j}) = \chi_{j-1} \times \chi_j$ and $\dim(W_j^{x_j y_j}) = D_{j-1} \times D_j$, and where we use the implicit summation convention for repeated indices. Applying the operator \hat{O} to the state $|\psi\rangle$ actually results in the following contraction

$$\hat{O}|\psi\rangle = \begin{array}{c} \textcircled{B_1} = \textcircled{B_2} = \dots = \textcircled{B_n} \\ | \quad | \quad \dots \quad | \\ x_1 \quad x_2 \quad \dots \quad x_n \end{array} |x_1, \dots, x_n\rangle = \textcircled{B_j} = \begin{array}{c} \textcircled{A_j} \\ | \\ \textcircled{O_j} \\ | \\ x_j \end{array}$$

which is a new MPS state with matrices $B_j^{x_j}$ whose auxiliary dimensions are $\dim(A_j^{x_j}) = D_{j-1} \chi_{j-1} \times D_j \chi_j$. Notice that, a very similar procedure

applies for the product of two operators \hat{O} and \hat{V} both having an MPO representation:

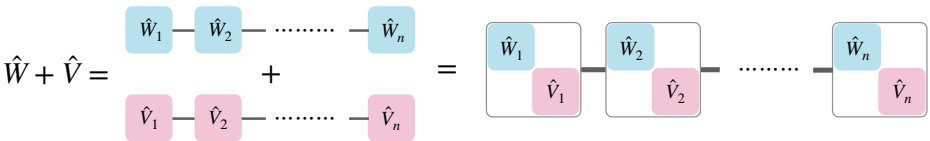


where the order of the physical indices contraction is relevant since in general $[\hat{O}, \hat{V}] \neq 0$.

Relevant finite-dimensional MPO — Constructing an exact compact MPO representation for certain physically relevant operators might initially appear daunting. However, analogous to the case of low-entangled states that admit an exact MPS representation, whenever “local” operators are involved (such as Hamiltonians with short-range interactions), an exact finite-dimensional MPO representation exists. Similarly to what we already did for MPS, we can rearrange a matrix product operator in terms of operator-value matrices, as follow

$$\hat{O} = \begin{array}{c} \boxed{\hat{o}_1} - \boxed{\hat{o}_2} - \dots - \boxed{\hat{o}_n} \end{array}, \quad \boxed{\hat{o}_j} = \sum_{x,y} \begin{array}{c} y \\ \downarrow \\ \boxed{o_j} \\ \downarrow \\ x \end{array} |x\rangle \langle y| \tag{2.70}$$

This representation is very useful to understand how to systematically construct exact MPO of relevant operators. In fact when considering the addition of two operators, \hat{W} and \hat{V} , with both MPO representations, the resulting MPO is formed by the direct sum of the local operator-value matrices for all sites $1 < j < n$, with the exception of the boundary sites where we add row and column vectors. Essentially, we get



where basically we are using block diagonal matrices to take into account the independent effects of both operators. However, for “local” operators, the

previous representation is sub-optimal, and we can systematically construct much better representations.

In the following we present a list of example of operators which admit an exact MPO representation.

1. Let us start from a simplest example of a local Hamiltonian acting on n qubits

$$\hat{H} = \sum_{j=1}^n h_j \hat{Z}_j \tag{2.71}$$

This compact form of Hamiltonian can be written in a more general tensor product notation,

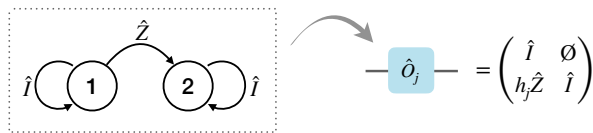
$$\hat{H} = \sum_{j=1}^n \hat{I}_1 \otimes \dots \otimes \hat{I}_{j-1} \otimes h_j \hat{Z}_j \otimes \hat{I}_{j+1} \otimes \dots \otimes \hat{I}_n \tag{2.72}$$

In this notation, it is evident that the size of the matrix corresponding to the operator \hat{H} is $d^n \times d^n$, where each term in the sum is an exact product operator, i.e., an MPO with auxiliary dimension equal to one. Clearly, a suitable MPO representation of the Hamiltonian (2.71) requires an auxiliary space with dimension $D > 1$. The question then becomes: what is the minimal extra resource needed to store all the information in a suitable operator-valued matrix product form $\hat{O}_1 \hat{O}_2 \dots \hat{O}_n$.

If we temporarily set aside the boundary vectors \hat{O}_1 and \hat{O}_n and focus on the bulk matrices, we can ask ourselves what these matrices need to accomplish and remember:

- Insert the identity operator regardless of the lattice site.
- Insert the operator \hat{Z} and remember having done so.
- After the previous step, insert only the identity operator.

Actually, we can accomplish, and visualise the effect of the operator-value matrices, via the following **auxiliary-state diagram**

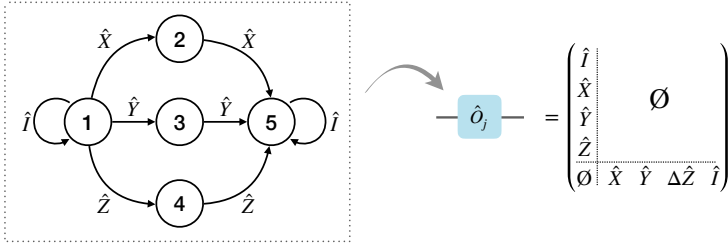


Finally, the boundary vectors can be easily obtained from the bulk matrix, by just piking up the last row and the first column, namely having $\hat{O}_1 = (h_1 \hat{Z} \ \hat{I})$ and $(\hat{O}_n)^t = (\hat{I} \ h_n \hat{Z})$.

- Let us consider now an interacting Hamiltonian like the Heisenberg one with anisotropy (i.e. the XXZ Hamiltonian)

$$\hat{H} = \sum_{j=1}^{n-1} \hat{X}_j \hat{X}_{j+1} + \hat{Y}_j \hat{Y}_{j+1} + \Delta \hat{Z}_j \hat{Z}_{j+1}, \quad (2.73)$$

where for simplicity we are considering the anisotropy Δ uniform along the chain. To find the MPO representation, let's focus on a generic neighboring interaction term $\sum_j \hat{S}_j \hat{S}_{j+1}$. Practically, as before, we need to insert the identity operator. Then, if the bulk matrix inserts the operator \hat{S} , immediately after we have to insert the same operator again. After this step, we complete the operator chain with the identity. This need to be done for each of the interaction terms. Using the auxiliary-state diagram we get



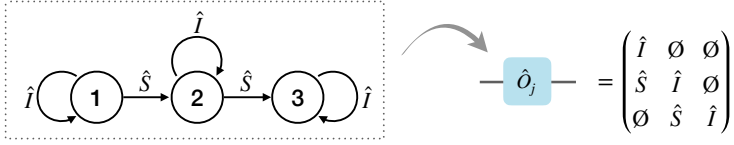
and similar consideration as before applies for the boundary vectors.

- Finally, we want to address arbitrary **long-range interacting Hamiltonian**. Let' us start with the preliminary example of a fully connected spin system, such that

$$\hat{H} = \sum_{j>i}^n \hat{S}_i \hat{S}_j, \quad (2.74)$$

and \hat{S} here stays for a generic local operator. A generic operator entering in that sum can be written explicitly as $\dots \hat{I} \otimes \hat{I} \otimes \hat{S} \otimes \hat{I}^{|j-i|} \otimes \hat{S} \otimes \hat{I} \otimes \hat{I} \dots$ for any possible distance $|j-i| \in \{1, \dots, n-1\}$. The only difference between a simple neighboring site interaction $\hat{S}_j \hat{S}_{j+1}$ is that, after inserting the operator \hat{S} at any lattice site j , our MPO must allow for the insertion of an arbitrary number of identity operators until the second insertion of the operator \hat{S} occurs. Thereafter, we complete our chain with the

remaining identities. In terms of auxiliary-state diagram this reads



Now it is clear that, if we have exponentially decaying interactions such that

$$\hat{H} = \sum_{j>i}^n e^{-|j-i|/\xi} \hat{S}_i \hat{S}_j = \sum_{j>i}^n \lambda^{|j-i|} \hat{S}_i \hat{S}_j \quad (2.75)$$

with $\lambda \equiv \exp(-1/\xi)$, we can slightly modify the state diagram of the fully connected Hamiltonian by introducing an additional numerical coefficient λ for each identity operator inserted by the auxiliary state 2, and another λ when transitioning from state 2 to state 3. This modification leads to the intriguing result that exponentially decaying interactions can be exactly encoded into an MPO with an auxiliary dimension of $D = 3$:

$$\hat{O}_j = \begin{pmatrix} \hat{I} & \emptyset & \emptyset \\ \hat{S} & \lambda \hat{I} & \emptyset \\ \emptyset & \lambda \hat{S} & \hat{I} \end{pmatrix}. \quad (2.76)$$

This result is extremely useful, because it allows to efficiently represent generic interacting Hamiltonians on a finite chain

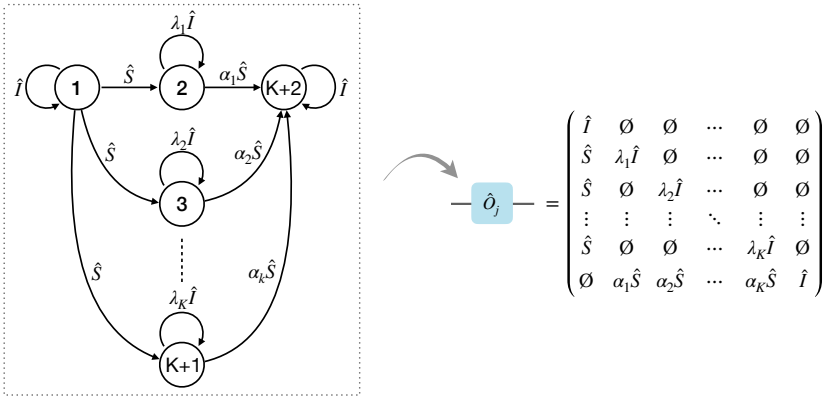
$$\hat{H} = \sum_{j>i}^n J(j-i-1) \hat{S}_i \hat{S}_j. \quad (2.77)$$

In fact, once we approximate the interactions as a sum of K different exponential $J(r) \simeq \sum_{k=1}^K \alpha_k \lambda_k^r$, we can rewrite the generic Hamiltonian as

$$\hat{H} = \sum_{k=1}^K \sum_{j>i}^n \alpha_k \lambda_k^{j-i-1} \hat{S}_i \hat{S}_j, \quad (2.78)$$

and using the previous argument we can easily draw the following

auxiliary-state diagram



which results in a MPO with auxiliary dimension $D = 2 + K$.

References of Chapter 2

- [BC17] J. C. Bridgeman and C. T. Chubb. “Hand-waving and interpretive dance: an introductory course on tensor networks.” In: *Journal of Physics A: Mathematical and Theoretical* 50.22 (2017), p. 223001. DOI: [10.1088/1751-8121/aa6dc3](https://doi.org/10.1088/1751-8121/aa6dc3).
- [Ben+96a] C. H. Bennett et al. “Concentrating partial entanglement by local operations.” In: *Physical Review A* 53.4 (1996), p. 2046.
- [Ben+96b] C. H. Bennett et al. “Mixed-state entanglement and quantum error correction.” In: *Physical Review A* 54.5 (1996), p. 3824.
- [Chi+14] E. Chitambar et al. “Everything You Always Wanted to Know About LOCC (But Were Afraid to Ask).” In: *Communications in Mathematical Physics* 328.1 (2014), pp. 303–326. DOI: [10.1007/s00220-014-1953-9](https://doi.org/10.1007/s00220-014-1953-9).
- [Cho75] M.-D. Choi. “Completely positive linear maps on complex matrices.” In: *Linear Algebra and its Applications* 10.3 (1975), pp. 285–290. ISSN: 0024-3795. DOI: [10.1016/0024-3795\(75\)90075-0](https://doi.org/10.1016/0024-3795(75)90075-0).
- [Eve22] G. Evenbly. “A Practical Guide to the Numerical Implementation of Tensor Networks I: Contractions, Decompositions, and Gauge Freedom.” In: *Frontiers in Applied Mathematics and Statistics* 8 (2022). ISSN: 2297-4687. DOI: [10.3389/fams.2022.806549](https://doi.org/10.3389/fams.2022.806549).

- [HHH99] M. Horodecki, P. Horodecki, and R. Horodecki. “General teleportation channel, singlet fraction, and quasidistillation.” In: *Physical Review A* 60.3 (1999), p. 1888.
- [KP09] K. Konishi and G. Paffuti. *Quantum Mechanics: A New Introduction*. OUP Oxford, 2009. ISBN: 9780199560264. URL: <https://books.google.it/books?id=YuQ-DgAAQBAJ>.
- [McC07] I. P. McCulloch. “From density-matrix renormalization group to matrix product states.” In: *Journal of Statistical Mechanics: Theory and Experiment* 2007.10 (2007), P10014. DOI: [10.1088/1742-5468/2007/10/P10014](https://doi.org/10.1088/1742-5468/2007/10/P10014).
- [NC00] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.
- [NC10] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. DOI: [10.1017/CBO9780511976667](https://doi.org/10.1017/CBO9780511976667).
- [Orú14] R. Orús. “A practical introduction to tensor networks: Matrix product states and projected entangled pair states.” In: *Annals of Physics* 349 (2014), pp. 117–158. ISSN: 0003-4916. DOI: [10.1016/j.aop.2014.06.013](https://doi.org/10.1016/j.aop.2014.06.013).
- [Pen04] R. Penrose. *The Road to Reality: A Complete Guide to the Laws of the Universe*. Science: Astrophysics. Jonathan Cape, 2004. ISBN: 9780224044479. URL: <https://books.google.it/books?id=c saaQgAACAAJ>.
- [Pér+07] D. Pérez-García et al. *Matrix Product State Representations*. 2007. arXiv: [quant-ph/0608197](https://arxiv.org/abs/quant-ph/0608197) [[quant-ph](https://arxiv.org/abs/quant-ph)].
- [Pér+08] D. Pérez-García et al. “String Order and Symmetries in Quantum Spin Lattices.” In: *Phys. Rev. Lett.* 100 (16 2008), p. 167202. DOI: [10.1103/PhysRevLett.100.167202](https://doi.org/10.1103/PhysRevLett.100.167202).
- [Ran+20] S.-J. Ran et al. *Tensor Network Contractions: Methods and Applications to Quantum Many-Body Systems*. Springer International Publishing, 2020. ISBN: 9783030344894. DOI: [10.1007/978-3-030-34489-4](https://doi.org/10.1007/978-3-030-34489-4).
- [Sch11] U. Schollwöck. “The density-matrix renormalization group in the age of matrix product states.” In: *Annals of Physics* 326.1 (2011), pp. 96–192. DOI: [10.1016/j.aop.2010.09.012](https://doi.org/10.1016/j.aop.2010.09.012).
- [Sil+19] P. Silvi et al. “The Tensor Networks Anthology: Simulation techniques for many-body quantum lattice systems.” In: *SciPost Phys. Lect. Notes* (2019), p. 8. DOI: [10.21468/SciPostPhysLectNotes.8](https://doi.org/10.21468/SciPostPhysLectNotes.8).
- [VGC04] F. Verstraete, J. J. García-Ripoll, and J. I. Cirac. “Matrix Product Density Operators: Simulation of Finite-Temperature and Dissipative Systems.” In: *Phys. Rev. Lett.* 93 (20 2004), p. 207204. DOI: [10.1103/PhysRevLett.93.207204](https://doi.org/10.1103/PhysRevLett.93.207204).

- [WZ82] W. K. Wootters and W. H. Zurek. “A single quantum cannot be cloned.” In: *Nature* 299.5886 (1982), pp. 802–803.
- [ZV04] M. Zwolak and G. Vidal. “Mixed-State Dynamics in One-Dimensional Quantum Lattice Systems: A Time-Dependent Superoperator Renormalization Algorithm.” In: *Phys. Rev. Lett.* 93 (20 2004), p. 207205. DOI: [10.1103/PhysRevLett.93.207205](https://doi.org/10.1103/PhysRevLett.93.207205).

Chapter 3

Quantum computing with Tensor Networks

I am made and remade continually.
Different people draw different words
from me.

Virginia Woolf

Quantum computing protocols essentially involve preparing a system in a specific state $|\psi\rangle$, evolving it through a circuit composed of logical quantum gates, and finally performing measurements on the resulting states. In a more general setting (adaptive quantum circuits) it is also possible to perform mid-circuit measurements, conduct classical computations on these outcomes, and subsequently apply unitary gates conditionally, depending on the results of these classical computations.

In this chapter, we will see how each of these three steps can be efficiently replicated using the MPS formalism, thereby enabling the creation of classical simulators of quantum computers. Of course, since quantum computers are generally believed to perform tasks considered NP-hard for classical computation, these classical simulators necessarily involve a certain level of approximation.

3.1 State preparation

In this section, we address the challenge of preparing a generic state with quantum unitary gates and/or Tensor Network representation. State preparation

is a fundamental task in quantum computation and quantum information. However, it is generally an inefficient process, often requiring an exponential number of gates relative to the number of qubits. To illustrate this, let us consider a quantum computer with n -qubit register initially in the computational basis state $|00 \dots 0\rangle$, and examine the preparation of an arbitrary quantum state. In general, we wish to prepare a generic many-body state as the following

$$|\psi\rangle = \sum_{\mathbf{x}} \psi_{\mathbf{x}} |\mathbf{x}\rangle \quad (3.1)$$

where $\psi_{\mathbf{x}}$ are complex coefficients that define our target state. This preparation typically involves a sequence of rotations and entangling operations. For example, the state preparation circuit might employ a series of $\hat{R}_n(\alpha)$ gates to initialize the qubits, followed by controlled-NOT (CNOT) gates to create the necessary entanglement.

Despite the apparent simplicity in specifying such a state, the actual implementation can become highly complex. The number of required operations grows exponentially with the number of qubits, posing significant practical challenges. In the following subsections, we will explore various techniques and algorithms that have been developed to mitigate this complexity, enabling more efficient state preparation for specific classes of states.

We note that, similar to classical computation, preparing a quantum state with a quantum computer generally requires an exponential number of operations. However, quantum computing has a significant advantage in terms of memory requirements. Specifically, a quantum computer can efficiently store a wave vector with n qubits, determined by 2^n complex numbers, which are the coefficients of its expansion over the computational basis. In contrast, a classical computer needs $O(2^n)$ bits to store these 2^n complex numbers.

This demonstrates the substantial memory efficiency of a quantum computer: it could perform the same task using only n qubits. This memory efficiency directly impacts the complexity of quantum state preparation. While classical systems require exponentially increasing resources, quantum computers leverage their qubit-based architecture to manage state vectors efficiently.

Furthermore, the connection between quantum state preparation and Matrix Product States (MPS) complexity offers additional insights. The auxiliary dimension in MPS reflects the entanglement present in the state, and states with lower entanglement can be represented with smaller auxiliary dimensions, reducing computational overhead. This connection suggests that quantum computers, which naturally handle entangled states, can efficiently prepare and manipulate states that would be complex to manage classically.

In essence, the exponential memory advantage of quantum computing, coupled with the understanding of MPS complexity, underscores the potential

of quantum systems to outperform classical counterparts in handling complex, high-dimensional quantum states.

3.1.1 Product state

In certain scenarios, a specific wave function can be prepared efficiently. An operation on a quantum computer is considered efficient if it requires a number of elementary gates that is polynomial in the number of qubits. For example, creating an equal superposition of all states in the computational basis,

$$\frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}} |\mathbf{x}\rangle,$$

can be achieved by applying n Hadamard gates, one to each qubit, starting from the initial state $|00 \dots 0\rangle$.

The key insight here is that the previous state is simply a product state along a different direction, which can be efficiently prepared by applying local single-qubit unitaries. In general, any product state, or Matrix Product State (MPS) with bond dimension $\chi = 1$, can be readily prepared using the following procedure, assuming the quantum platform allows for arbitrary single-qubit rotations:

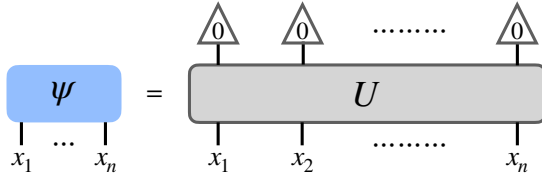
$$|\psi_1\rangle |\psi_2\rangle \dots |\psi_n\rangle = \begin{array}{c} \triangle 0 \\ | \\ \square u_1 \\ | \\ x_1 \end{array} \begin{array}{c} \triangle 0 \\ | \\ \square u_2 \\ | \\ x_2 \end{array} \dots \dots \begin{array}{c} \triangle 0 \\ | \\ \square u_n \\ | \\ x_n \end{array} |x_1, x_2, \dots, x_n\rangle \quad (3.2)$$

where u_j denotes a single-qubit unitary operation applied to the j -th qubit. This approach leverages the ability to independently manipulate each qubit to create an entanglement-free state representation suitable for various quantum computing tasks.

3.1.2 MPS to quantum circuit

In the realm of universal quantum computation, it is well-established that any quantum state living in an n -qubit register (which can be extended to quantum qudits) can fundamentally be expressed as a global unitary transformation applied to an elementary computational basis state. This foundational principle asserts that the many-body wave function can be succinctly repre-

sented as



However, for an n -qubit state, finding the global unitary transformation \hat{U} such that $|\Psi\rangle = \hat{U} |00 \dots 0\rangle$ is generally quite challenging. Additionally, having this matrix as a global transformation is practically useless for quantum computing purposes. Instead, we desire the unitary operator \hat{U} to be constructed from only one- and two-qubit gates, i.e., gates that can be feasibly implemented on a quantum device.

Here, we reduce the complexity of this problem by considering only **states that have an MPS representation with a local bond dimension equal to the dimension of the physical indices** [Sch+05]. In other words, for local qubits, we want $\chi = 2$ for all bonds except at the boundaries, of course. This approach can be easily generalized to qudits, where the physical dimension is now d [Ran20].

Therefore let us consider an MPS and assume to have it right-normalized (this condition can always been satisfied by means of a gauge transformation as we already have seen). The state wave function looks like



Now let's take a specific site j in the bulk ($1 < j < n$) and reshape the tensor B_j into a rectangular matrix of dimensions 2×4 by mapping $(B_j)_{\alpha,\beta}^x \rightarrow (B_j)_{\alpha,(x,\beta)}$. Under this transformation, the right-normalization condition becomes $B_j B_j^\dagger = I$. However, $B_j^\dagger B_j$ does not necessarily equal the identity matrix, indicating that B_j is an *isometric* matrix rather than a unitary one. To transform B_j into a unitary matrix u_j , we introduce an additional artificial index y (representing an auxiliary), as follows:

$$(B_j)_{\alpha,(x,\beta)} \rightarrow (u_j)_{(\alpha,y)(x,\beta)}$$

This can be achieved by augmenting the two 4-component orthonormal row vectors with two additional orthonormal vectors to complete the basis. This ensures the condition that $(u_j)_{(\alpha,0)(x,\beta)} = (B_j)_{\alpha,(x,\beta)}$. This condition can be

effectively represented using the following tensor diagram:

$$\alpha \text{---} B_j \text{---} \beta \quad \xrightarrow{\dots\dots\dots} \quad \alpha \text{---} u_j \text{---} \beta \quad \alpha \text{---} u_j \text{---} \beta \quad = \quad \alpha \text{---} B_j \text{---} \beta \quad (3.3)$$

Notice that, in principle, we can combine the column and row indices to construct the unitary matrices in any order. However, we have chosen to maintain a specific order, where the auxiliary qubit is tensorized with the row auxiliary-dimension leg in reverse order compared to the column tensorization order. This choice, represented by the curved dotted wires in the graphical representation, allows us to clearly identify each virtual qubit Hilbert space line.

Finally a different situation arises at the boundaries. When $j = 1$, the matrix B_1 is of dimension 1×4 (essentially, it is a single vector of length 4). To embed this into a unitary matrix of size 4×4 , we need to introduce two additional artificial qubits. Conversely, at the opposite boundary where $j = n$, the tensor B_n has dimensions 2×2 , thus being inherently a square matrix and thus already a single-qubit unitary. The full graphical transformation looks like

$$\text{[Chain of } u_j \text{ tensors]} \quad = \quad \text{[Staircase quantum circuit]} \quad (3.4)$$

which forms a valid staircase quantum circuit, consisting solely of single-qubit and two-qubit gates acting on neighboring sites.

In the following we present two examples of non-trivial quantum states that admit a simple MPS representation.

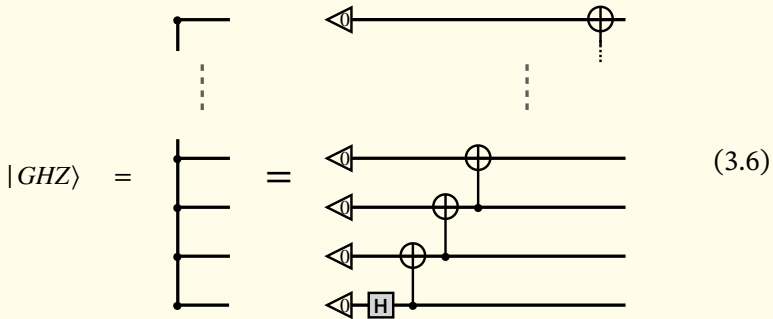
Example 3.1: GHZ state as an MPS

In the field of quantum information theory, a Greenberger–Horne–Zeilinger (GHZ) state represents a unique type of entangled quantum state encompassing at least three qubits. The study of the four-particle GHZ state was first conducted by Daniel Greenberger, Michael Horne, and Anton Zeilinger in 1989 [GHZ07]. These states may exhibit highly non-classical properties; however, it turns out that for an arbitrary number of qubits, they admit a very simple MPS (and thus, in a sense, “classical”) representation.

Indeed, the standard MPS of the GHZ state for n qubits is given as

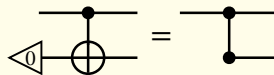
$$|GHZ\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \end{pmatrix} \begin{pmatrix} |0\rangle & 0 \\ 0 & |1\rangle \end{pmatrix}^n \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} (|00 \dots 0\rangle + |11 \dots 1\rangle) \tag{3.5}$$

which is an exact representation with bond dimension $\chi = 2$. Graphically it corresponds to a sequence of copy tensors



$$|GHZ\rangle = \text{[Diagrammatic representation of GHZ state as a sequence of copy tensors]} \tag{3.6}$$

where in the last passage we transformed to sequential CNOT operation exploiting the following identity



$$\text{[CNOT with copy tensor on control]} = \text{[CNOT with copy tensor on target]}$$

Let us conclude by noting that the “classical” nature of the GHZ state is further affirmed by the fact that, as we have seen, it has been constructed using only Clifford unitaries, making it a *stabilizer state*. We refer the reader to Chapter ... to learn more about *stabilizer states* and *magic*.

Example 3.2: W state as an MPS

The W state is an entangled quantum state of three qubits, represented in bra-ket notation as follows:

$$|W\rangle = \frac{1}{\sqrt{3}} (|001\rangle + |010\rangle + |100\rangle)$$

This state is remarkable for representing a specific type of multipartite entanglement and appears in several applications in quantum information theory. Particles prepared in this state exhibit the properties described by Bell's theorem, which asserts that no classical theory of local hidden variables can reproduce the predictions of quantum mechanics. The W state has been introduced for the first time by Wolfgang Dür, Guifré Vidal, and Ignacio Cirac in 2002 [DVC00].

The W state exemplifies one of the two classes of three-qubit states that cannot be separated into independent subsystems. The other class is exemplified by the 3-qubit GHZ state. These two states, $|W\rangle$ and $|GHZ\rangle$, cannot be transformed into each other, even probabilistically, via LOCC (local operations and classical communication). Therefore, they represent fundamentally different types of tripartite entanglement.

The concept of the W state has been extended to n qubits, referring to a quantum superposition where each term has equal coefficients, and exactly one qubit is in the state $|1\rangle$ while the rest are in the $|0\rangle$ state; it can be easily written as an MPS

$$\begin{aligned} |W_n\rangle &= \frac{1}{\sqrt{n}} \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} |0\rangle & 0 \\ |1\rangle & |0\rangle \end{pmatrix}^n \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= \frac{1}{\sqrt{n}} (|100 \dots 0\rangle + |010 \dots 0\rangle + |001 \dots 0\rangle + \dots + |000 \dots 1\rangle) \end{aligned} \quad (3.7)$$

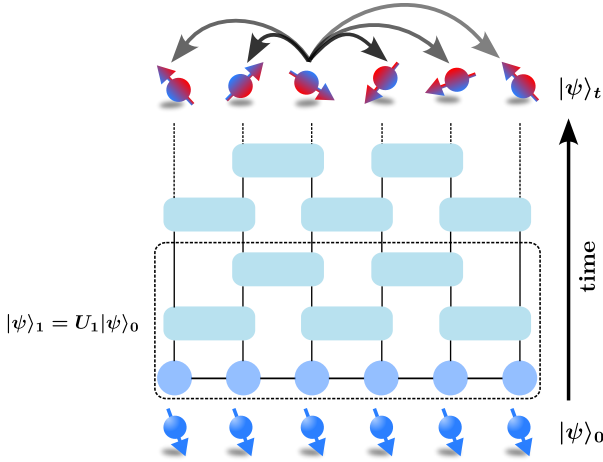
3.2 Quantum circuits

The state preparation is followed by the evolution of this initial state in time through a quantum circuit. In a traditional manybody setup the evolution is driven by the system Hamiltonian in continuous time. Quantum circuit on the other hand evolves the initial state in discrete time by application of sequence of unitary quantum operator, $|\psi\rangle_t = U_t |\psi\rangle_{t-1}$. The final evolution is given as $|\psi\rangle_t = U(0 \rightarrow t) |\psi\rangle_0$ where $U(0 \rightarrow t) = U_t U_{t-1} \dots U_2 U_1$. Here, U

can be a generic operator however we will consider a special case of brickwork circuit where each U is composed of alternating two site local unitary gates,

$$U_t = \begin{cases} \bigotimes_i u_t^{2i+1,2i+2} & i \in 0, 1, 2 \dots \\ \bigotimes_i u_t^{2i,2i+1} & i \in 1, 2, 3 \dots \end{cases} \quad (3.8)$$

Here each u is a 4×4 unitary matrix acting in the local Hilbert space of a pair of neighboring spins. The following figure illustrates the unitary evolution of a trivial initial state with a brickwork circuit where the discrete unitary evolution induces entanglement in the initial weakly entangled pure state over time.



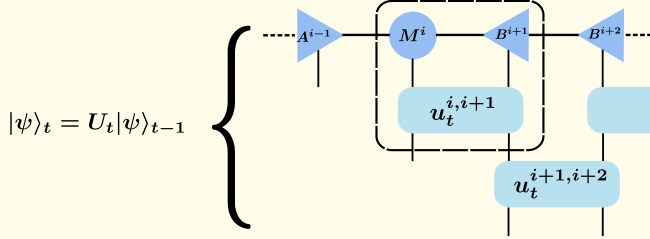
Here, the section inside the dotted rectangle represent the evolution of the initial state $|\psi\rangle_0$ over a unit time. Repeated evolution over similar units will generate the full unitary dynamics. With initial state as an MPS the brickwork Quantum circuit is a tensor network that can be evaluated by sequential contraction of the u matrices with the MPS at specific sites. The optimal algorithm for evaluating such tensor network architecture is the time evolving block decimation (TEBD) algorithm [Vid04]. The TEBD algorithm was introduced to solve the continuous Hamiltonian dynamics and involved the Suzuki-Trotter decomposition of the evolution operator $e^{-i\Delta t H}$ into a product of two site unitary operators. Applying TEBD algorithm in a brickwork circuit evolution circumvents the Trotter error as the operators are already in the form of product of two site unitaries. TEBD constitutes an efficient sequential application of two site unitaries onto the MPS state. The evolution through a unit time can be written as,

$$|\psi\rangle_t = U_t |\psi\rangle_{t-1} \rightarrow u_t^{1,2} \dots \otimes u_t^{j,j+1} \otimes \dots \otimes u_t^{N-1,N} |\psi\rangle_{t-1}. \quad (3.9)$$

Conventionally, the TEBD algorithm is performed iteratively starting from the left edge of the MPS and moving towards the right edge and back while applying the two site unitary operators on the corresponding sites. Furthermore, the MPS is kept in a mixed canonical form (see Section 2.4.1) throughout the evolution as the canonical form is preserved under the application of unitary operator. In the following example we will thoroughly outline the core of TEBD algorithm which is the application of a two site unitary gate onto the MPS.

Example 3.3: Local unitary updates of MPS with TEBD

In this example we will thoroughly explain the local unitary update during the operation $|\psi\rangle_t = U_t |\psi\rangle_{t-1}$. Specifically, we will consider the operation $u_t^{i,i+1} |\psi\rangle_{t-1}$ as highlighted inside the dotted rectangle in figure below.



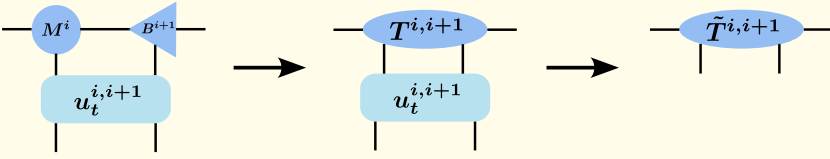
The state is in mixed canonical form (see equation (2.66)),

$$\begin{aligned} |\psi\rangle &= \sum_{\alpha, x_i, x_{i+1}, \beta} [M^{x_i} B^{x_{i+1}}]_{\alpha, \beta} |\alpha\rangle |x_i\rangle |x_{i+1}\rangle |\beta\rangle \\ &= \sum_{\alpha, x_i, x_{i+1}, \beta} T_{\alpha, \beta}^{x_i, x_{i+1}} |\alpha\rangle |x_i\rangle |x_{i+1}\rangle |\beta\rangle \end{aligned}$$

where $|\alpha\rangle$ and $|\beta\rangle$ are the left and right canonical vectors as defined in equation (2.67). In this representation applying the two site unitary operator onto the state is now straightforward,

$$\begin{aligned} u_{x_i, x_{i+1}}^{\tilde{x}_i, \tilde{x}_{i+1}} |\psi\rangle &= \sum_{\alpha, x_i, x_{i+1}, \beta} u_{x_i, x_{i+1}}^{\tilde{x}_i, \tilde{x}_{i+1}} T_{\alpha, \beta}^{x_i, x_{i+1}} |\alpha\rangle |x_i\rangle |x_{i+1}\rangle |\beta\rangle \\ &= \sum_{\alpha, x_i, x_{i+1}, \beta} \tilde{T}_{\alpha, \beta}^{x_i, x_{i+1}} |\alpha\rangle |x_i\rangle |x_{i+1}\rangle |\beta\rangle. \end{aligned}$$

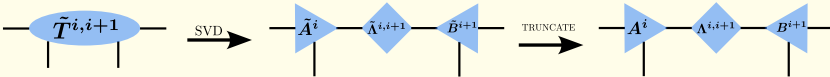
The operations outlined above are shown in the figure below,



Applying two site unitary operator on site i and $i + 1$ increases the entanglement at those bonds, in other words the bond dimension between sites i and $i + 1$ increases from χ to $d\chi$, where d is the local Hilbert space dimension ($d = 2$ for spin $1/2$ system). Successive application of local unitaries onto the bond during the evolution will exponentially increase the bond dimension rendering it computationally intractable. To this end the bond dimension at every bond is capped from above by χ_{\max} . This truncation process is carried out by doing a Singular Value Decomposition (SVD) of the tensor \tilde{T} as

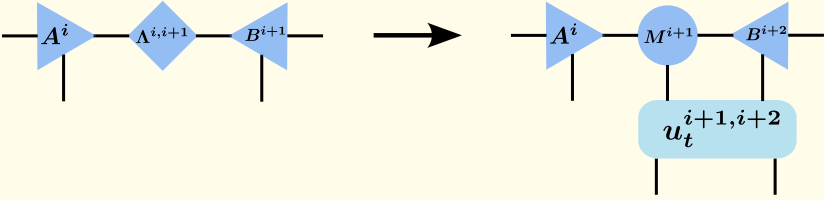
$$\tilde{T}_{\alpha,\beta}^{x_i,x_{i+1}} \xrightarrow{\text{reshape}} \tilde{T}_{[x_i,\alpha],[x_{i+1},\beta]} \xrightarrow{\text{SVD}} \tilde{A}_{\alpha,k}^{x_i} \tilde{\Lambda}_{k,k}^{i,i+1} \tilde{B}_{k,\beta}^{x_{i+1}},$$

where, \tilde{A}^{x_i} is right normalized tensor, $\tilde{\Lambda}^{i,i+1}$ is a diagonal matrix, and $\tilde{B}^{x_{i+1}}$ is right normalized tensor. Therefore by construction the MPS remains mixed canonical after the application of local unitaries. The matrix $\tilde{\Lambda}^{i,i+1}$ has descending diagonal entries whose squares sums to unity $\sum_k \tilde{\lambda}_{k,k}^2$. The truncation process happens at this stage where we keep only the χ_{\max} largest values and discard the rest. This incurs a truncation error, $\sum_{k > \chi_{\max}} \tilde{\lambda}_{k,k}^2$. Following the truncation the matrix is normalized. The following figure illustrates the decomposition and truncation process,



The state $|\psi\rangle$ remains in a mixed canonical form at the end of a local unitary update. The re-normalized diagonal matrix $\Lambda^{i,i+1}$ is now contracted with the tensor $B^{x_{i+1}}$ to build $M^{x_{i+1}}$. This brings us to the beginning of the outlined process just shifted by a site. Now following the same process we apply the unitary operator $u^{x_{i+1},x_{i+2}}$ on sites $i + 1$

and $i + 2$ respectively as shown in the figure below,



We proceed on a similar fashion by sequentially applying two site unitaries from left to right until we reach the right most site. This completes a unit left-to-right TEBD sweep and evolves the state through a unit time. This is followed by a similar sequential application of local unitaries starting from the rightmost site and moving toward left called right-to-left TEBD sweep. TEBD involves an alternative left-to-right and right-to-left sweeps.

3.3 Measurements

Measurements are the essential readout of any quantum computation, allowing to extract (classical) information from a quantum state $|\psi\rangle$.

Suppose $|\psi\rangle$ is represented as an MPS, and we aim to perform a measurement. We will specifically focus on the simplest type of measurements: projective measurements of \hat{Z}_i operators on qubits indexed by $i \in I$, where I is a subset of $1, 2, \dots, N$ representing the qubits to be measured.

Since the operators $\{\hat{Z}_i\}_{i \in I}$ commute with each other, we can perform the measurements in any order we prefer. First, we will consider the qubit with the smallest index i (i.e. the one closer to the left boundary of the MPS). Notice that

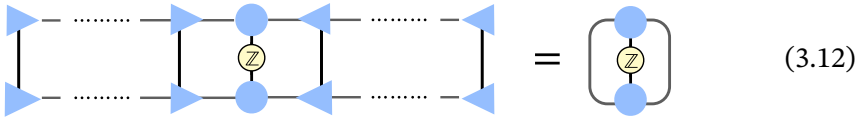
$$\hat{P}_0^{(i)} = \frac{\hat{1} + \hat{Z}_i}{2} \quad \hat{P}_1^{(i)} = \frac{\hat{1} - \hat{Z}_i}{2} \quad (3.10)$$

are projectors on subspaces where qubit i is respectively $|0\rangle$ or $|1\rangle$. We have therefore to compute outcomes probabilities

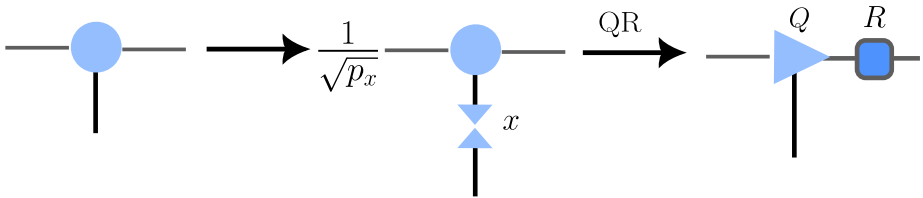
$$p_0^{(i)} = \frac{1 + \langle \psi | \hat{Z}_i | \psi \rangle}{2} \quad p_1^{(i)} = \frac{1 - \langle \psi | \hat{Z}_i | \psi \rangle}{2}. \quad (3.11)$$

Evaluating the expectation value $\langle \psi | \hat{Z}_i | \psi \rangle$ becomes particularly simple if we have the orthogonality center of the MPS on site i . Indeed, if sites $j < i$ are

left normalized and sites $j > i$ are right normalized, we get



This contraction can be evaluated efficiently at a cost $O(\chi^3)$. Having obtained p_0, p_1 , the task is to simulate the wave function collapse with these probabilities. This can be done numerically by drawing a uniform (pseudo-)random variable $\xi \in [0, 1]$ and setting the outcome x to 0 if $\xi < p_0$, and to 1 if $\xi > p_0$. Once the outcome x is determined, we must apply the measurement postulate (Eq. (2.16)) to project $|\psi\rangle$ into the eigenspace corresponding to x and then renormalize the state. This is achieved by setting the $\chi \times \chi$ matrix A_i^{1-x} to 0, effectively projecting away the component corresponding to $1 - x$ in the MPS tensor A_i . For the normalization, A_i is simply divided by $\sqrt{p_x^{(i)}}$ (which has already been evaluated). At this point, however, it is necessary to restore the correct MPS gauge by shifting the orthogonality center from i to $i + 1$. This can be done simply with a QR decomposition of the tensor A_i (after projecting it), and incorporating R into the tensor A_{i+1} . These last steps can be graphically represented as follows:



Starting from a right-normalized MPS $|\psi\rangle$, we can then perform an entire sweep of QR decomposition, iteratively bringing the tensors into a left-normalized form. Each time we reach a site $i \in I$, we need to measure p_0 and apply the outlined algorithm, extracting the outcome and projecting the MPS tensor, before moving on to site $i + 1$. The total cost of this algorithm is thus $O(N_{\text{meas}}N\chi^3)$, where N_{meas} is the number of measurements we want to extract.

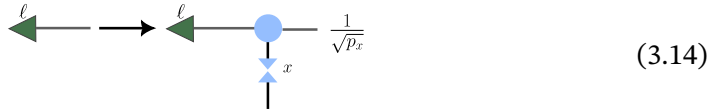
Simplifications occur, however, in the case where I coincides with the set of all qubits $1, 2, \dots, N$. Referring to Eq. (3.12), this is because if all sites $j < i$ have been measured, the bra and ket sides of the expectation value factorize. Consequently, the auxiliary line to the left of the right-hand side can be split, which simplifies the complexity of the evaluation. To be more

precise, Eq. (3.12) becomes



$$(3.13)$$

where we introduced a suitable left environment vector ℓ . Notice that evaluating Eq. (3.13) have cost $O(\chi^2)$, instead of $O(\chi^3)$. Once we get p_0, p_1 , we project A_i as before, but without the need of a QR decomposition. Indeed, it is enough to update the environment vector as



$$(3.14)$$

Again, we can start from a right-normalized MPS $|\psi\rangle$ and perform a sweep through all sites. The total cost is therefore $O(N_{\text{meas}}N\chi^2)$. This improved sampling algorithm has been firstly proposed in ref. [SW10].

Measuring all \hat{Z}_i operators is equivalent to sampling bit-strings \mathbf{x} from the computational basis, where each string is drawn with a probability given by $p(\mathbf{x}) = |\langle \mathbf{x} | \psi \rangle|^2$. This probability reflects the likelihood of obtaining the configuration \mathbf{x} in a measurement of the quantum state $|\psi\rangle$.¹

Collecting these samples, or shots, is the typical method experimentalists use to measure (diagonal) observables. For instance, if an observable \hat{O} is defined as $\hat{O} = \sum_{\mathbf{x}} O_{\mathbf{x}} |\mathbf{x}\rangle \langle \mathbf{x}|$, then $\langle \Psi | \hat{O} | \Psi \rangle = \sum_{\mathbf{x}} O_{\mathbf{x}} p(\mathbf{x})$, and therefore one can estimate $\langle \Psi | \hat{O} | \Psi \rangle$ as a sample average based on outcomes of \mathbf{x} . This situation is especially common when addressing classical problems embedded into quantum systems and proves useful numerically even when we lack an explicit MPO representation for \hat{O} .

Similar to what we have seen, every Pauli operator can be measured. For instance, to measure \hat{X}_i , we utilize the identity $\hat{X} = \hat{H}\hat{Z}\hat{H}^\dagger$. Thus, the measurement process can be expressed as:

$$\frac{\hat{1} \pm \hat{X}_i}{2} |\psi\rangle = \hat{H}_i \hat{P}_{0,1}^{(i)} |\tilde{\psi}\rangle \quad \text{and} \quad p_{0,1} = \langle \psi | \frac{\hat{1} \pm \hat{X}_i}{2} |\psi\rangle = \langle \tilde{\psi} | \hat{P}_{0,1}^{(i)} |\tilde{\psi}\rangle, \quad (3.15)$$

where $|\tilde{\psi}\rangle = \hat{H}_i^\dagger |\psi\rangle$. This method involves first applying the local gate \hat{H}_i^\dagger to the MPS tensor A_i and then calculating p_0 and p_1 as described. If the outcome is $x = 0$, for example, we apply $\hat{P}_0^{(i)}$ to $|\tilde{\psi}\rangle$ and then transform it back with \hat{H} .

¹For a more detailed examination of the sampling algorithm and its connection to conditional probabilities, we recommend referring to Chapter 5, where these concepts are thoroughly explored in the context of constructing the so-called minimal entangled typical thermal states (METs).

3.4 Quantum Platforms

In the year 2000, theoretical physicist David DiVincenzo proposed a set of conditions, known as the DiVincenzo Criteria, which are necessary for building a proper fault-tolerant quantum computer [DiV00]. The requirements are the following

1. A scalable system with well-characterized qubits.
2. The ability to initialize the state of the qubits to a simple fiducial state, such as $|00 \dots 0\rangle$.
3. Long relevant decoherence times, much longer than the gate operation time.
4. An *universal* set of quantum gates.
5. A high-fidelity readout method.

It is immediately clear that there is a certain level of contradiction in these requirements: on one hand, we need the quantum computer to be effectively protected from external influences to maintain its coherence, but on the other hand, we must interact with it intensively to set up the initial state, carry out the intended unitary evolution, and measure the final state. Nonetheless, in the last few years remarkable progress has been achieved thanks to the improvements in the control of quantum-mechanical systems, which allowed the transition of quantum hardware from laboratory curiosities to technical realities [Che+23].

We are going to focus on hardware platforms that show promise in realizing large-scale quantum computers, including superconducting qubits, neutral atoms and trapped-ions. In presenting these hardware implementations, our aim is not to favor any particular platform. Instead, we intend to demonstrate the potential application of tensor-networks methods in simulating the physics of these processors. It is important to notice that the field of quantum computing is still in a developmental phase, where multiple hardware platforms are being explored and tested for their efficacy in implementing quantum algorithms. Each of these platforms has its own set of advantages and challenges, making it unclear which one will ultimately prove to be the most effective for practical quantum computing. On the other hand, by effectively employing the physics and connectivity of quantum hardware, tensor-networks can serve as a crucial tool for advancing quantum technologies.

3.4.1 Superconducting Qubits

Superconducting qubits are advanced nonlinear circuits that utilize Josephson junctions to create quantized electromagnetic fields in the microwave frequency domain. These circuits are distinguished by their ability to be configured to exhibit customizable *atom-like* energy spectra, thanks to the tunable parameters of their circuit elements. Often referred to as *artificial atoms* due to their engineered energy levels, they are essential for quantum computing, offering high controllability and low noise in quantum logic operations. To function effectively, superconducting qubits require operation at cryogenic temperatures, typically around 10 millikelvin, to minimize thermal fluctuations. This is achieved using dilution refrigerators.

From a quantum circuit electrodynamics point of view, a Josephson junction consists of a nonlinear inductor $L_J = \phi_0 / (2\pi) I_C \cos(\phi)$ in parallel with a capacitor C_J , where I_C is the critical current, $\phi_0 = h / (2e)$ is the superconducting flux quantum, and ϕ is the phase difference across the junction. The two key parameters of the Josephson junction are the Josephson energy $E_J = \phi_0 I_C / (2\pi)$ and the charging energy $E_C = (2e)^2 / (2C_J)$.

Among all possible superconducting qubits, transmons are a widely used variant. In their design, a shunt capacitor C_S is added in parallel to the Josephson junction to decrease the charging energy, resulting in $E_C = (2e)^2 / (C_J + C_S)$, and ensuring $E_J \gg E_C$. The single-qubit Hamiltonian for transmons is given by:

$$H = E_C n^2 - E_J \cos(\phi) \quad (3.16)$$

where n is the number of Cooper pairs traversing the junction, and n and ϕ are conjugate variables with $[n, \phi] = i$. The shunt capacitor allows the low-energy eigenstates to be considered by expanding the potential term in a power series:

$$-E_J \cos(\phi) \approx -\frac{1}{2} E_J \phi^2 + \frac{1}{24} E_J \phi^4 \quad (3.17)$$

The first term represents a quantum harmonic oscillator with equidistant energy levels, while the quartic term introduces anharmonicity, particularly affecting the gap between the first and second excited states. This non-linearity allows the definition of a qubit in the subspace of the ground state $|0\rangle$ and the first excited state $|1\rangle$. The Hamiltonian projected in the two-state basis is:

$$H_Q = \frac{1}{2} (\epsilon \sigma_z - \Delta \sigma_x) \quad (3.18)$$

with $\epsilon = -E_C$ and $\Delta = E_J$. The energy splitting of the eigenvalues is $\Omega = \sqrt{\epsilon^2 + \Delta^2}$. By varying the voltage on the Josephson junction with pulses, the parameters ϵ and Δ can be manipulated in time to generate single-qubit gates.

Microwave pulses are integral to the control and measurement of transmon qubits. These pulses, typically shaped with specific envelopes (like Gaussian or sine-squared), drive transitions between qubit states by resonating at particular frequencies that correspond to the qubit's energy levels. A typical control pulse might have a duration of around 10 nanoseconds and is carefully calibrated in terms of amplitude and phase to ensure high-fidelity operations. In superconducting quantum processors, the CNOT gate is typically implemented using a method called the cross-resonance gate. In this method, the control qubit is driven at the resonance frequency of the target qubit. This drive induces a coupling between the qubits, creating an interaction described by a Hamiltonian of the form:

$$H = \frac{\omega}{2} \sigma_z \otimes \sigma_x \quad (3.19)$$

where ω is the drive amplitude. By carefully adjusting the amplitude and duration of the microwave drive, an entangling interaction is created that, combined with single-qubit rotations, realizes the CNOT gate.

For measurement, microwave pulses are used in conjunction with resonators coupled to the qubits. A readout pulse, often several microseconds long, is applied to the resonator. The qubit's state influences the resonator's response, altering the reflected or transmitted microwave signal. This signal is then analyzed to infer the qubit's state.

3.4.2 Neutral-Atoms Qubits

Neutral atom quantum computers have emerged as a promising platform for both near-term applications and long-term fault-tolerant universal quantum computation. These systems utilize atoms trapped in optical tweezers to create highly controllable qubits. The qubits are encoded in the electronic states or nuclear spin states of the atoms, which offer long coherence times and high-fidelity operations. Each atom is individually trapped by focused laser beams, forming a highly organized array that can be manipulated with precise laser pulses.

We start by considering the so-called *digital addressing* of the neutral-atom arrays. In this setting the qubit states are encoded in the two hyperfine ground-states of the system $\{|g\rangle \equiv |0\rangle, |h\rangle \equiv |1\rangle\}$, corresponding to $F = 1$ and $F = 2$ which for atoms such as rubidium have a very long lifetime. An additional quantum state dubbed Rydberg state $|r\rangle$ mediate the interaction between the atoms using the phenomenon of the Rydberg blockade.

Single-qubit gates are implemented by using tightly focused laser beams to drive transitions between the internal states of the atoms, creating rotations

around the Bloch sphere. Indeed, a transition, driven by an external pulse, between two energy levels $\{|g\rangle, |h\rangle\}$ or $\{|g\rangle, |r\rangle\}$ of an atom in the array can be mapped to a time-dependent spin 1/2 Hamiltonian

$$H^D(t) = \frac{1}{2}\mathbf{\Omega}(t) \cdot \hat{\boldsymbol{\sigma}}. \quad (3.20)$$

The rotation vector is given by $\mathbf{\Omega}(t) = (\Omega(t) \cos \phi, -\Omega(t) \sin \phi, -\delta(t))^T$ where Ω is the Rabi frequency of the transition while δ represent the detuning. The driving hamiltonian therefore implements a rotation around the axis $\mathbf{n} = \mathbf{\Omega}/|\mathbf{\Omega}|$ with angular velocity $|\mathbf{\Omega}| = \sqrt{\Omega^2 + \delta^2}$. In the case of a resonant pulse $\delta = 0$ of duration τ the Hamiltonian evolution generates the following unitary

$$U = \exp\left\{-i\frac{\theta}{2}(\cos \phi \hat{X} - \sin \phi \hat{Y})\right\} = R_z(-\phi)R_x(\theta)R_z(\phi), \quad (3.21)$$

where we defined

$$\theta = \int_0^\tau dt \Omega(t) dt. \quad (3.22)$$

Finally, applying a further rotation around the z-axis of an angle $\gamma + \phi$ we obtain an arbitrary single-qubit gate

$$U(\gamma, \theta, \phi) = R_z(\gamma)R_x(\theta)R_z(\phi). \quad (3.23)$$

Multi-qubit entangling gates are realized through the Rydberg blockade mechanism. In this method, atoms are excited to high-lying Rydberg states where their interactions are significantly enhanced, allowing for the creation of entangled states across multiple qubits. The Hamiltonian describing the ground-rydberg transitions is given by

$$\mathcal{H}^{gr}(t) = \sum_i \left(H_i^D(t) + \sum_{i < j} \frac{C_6}{R_{ij}^6} \hat{n}_i \hat{n}_j \right). \quad (3.24)$$

where \hat{n}_i denotes the projector $|r\rangle\langle r|_i$ on the i -th atom. Where the second term, which is the long-range dipole-dipole interactions via Rydberg states with strength C_6 , implements the so-called Rydberg blockade effect which prevents the simultaneous excitation of the two nearby atoms in the state $|rr\rangle$.

Example 3.4: Implementation of the CZ and CNOT gates

In neutral-atom devices the natural native two-qubits gate is given by the controlled-Z (CZ) gate. There are several implementations of the CZ gate

$$\text{CZ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}. \quad (3.25)$$

We illustrate one realized with a sequence of three pulses on the control-target two-qubit basis, namely $\{|0_c 0_t\rangle, |0_c 1_t\rangle, |1_c 0_t\rangle, |1_c 1_t\rangle\}$, which reads

$$\text{CZ} = e^{i\pi} R_x^{(c)}(\pi) \otimes \mathbb{1}^{(t)} \cdot \mathbb{1}^{(c)} \otimes R_x^{(t)}(2\pi) \cdot R_x^{(c)}(\pi) \otimes \mathbb{1}^{(t)}. \quad (3.26)$$

We note the following

- Initial state $|0_c 0_t\rangle$. The control qubit is excited to the Rydberg state $|r\rangle$ with a pulse π . The second pulse, because of the Rydberg blockade, leaves the target state unchanged. The final pulse brings the system back to the initial state with a phase $e^{i\pi} = -1$.
- Initial state $|0_c 1_t\rangle$. The second pulse is off-resonant because of the energy difference between $\Delta E = E_h - E_g$, the target state is unchanged. The first and third pulses bring the system back to the initial state with a phase $e^{i\pi} = -1$.
- Initial state $|1_c 0_t\rangle$. The first and third pulses are off-resonant, the control state is unchanged. The second pulses brings the system back to the initial state with a phase $e^{i\pi} = -1$.
- Initial state $|1_c 1_t\rangle$. all the pulses are off-resonant, the system state remains unchanged with no phase factor.

We finally notice that the CZ gate, composed with two additional Hadamard gates, can be used to implement a CNOT by applying the following sequence of gates $\mathbb{1}^{(c)} \otimes H^{(t)} \cdot \text{CZ} \cdot \mathbb{1}^{(c)} \otimes H^{(t)}$

3.4.3 Trapped-Ions Qubits

We finally address the case of trapped-ion qubits. Individual ions, such as ytterbium or calcium, are confined in electromagnetic traps using a combination of static and radio-frequency electric fields. These ions have long-lived electronic states that serve as qubits, which are manipulated using laser or microwave fields. The ions are held in traps where electrodes create a potential well that confines them in a linear chain, with mutual Coulomb repulsion maintaining their separation and allowing for individual addressing.

Single-qubit gates, as for neutral atoms, are implemented by applying laser pulses that induce transitions between the qubit states, with the pulses typically shaped and timed precisely to drive rotations around the Bloch sphere. For instance, resonant laser fields at the qubit transition frequency can perform rotations $R_x(\theta)$ and $R_y(\theta)$, where θ is the rotation angle determined by the pulse duration and amplitude.

Entangling gates, such as the CNOT gate, are often realized using the Mølmer-Sørensen interaction [SM99] or the Cirac-Zoller gate [CZ95]. The Mølmer-Sørensen gate uses bichromatic laser fields to couple the internal states of two ions via their shared motional modes, creating an effective spin-spin interaction that entangles the qubit states. The Mølmer-Sørensen gate applies a state-dependent force to the ions, displacing their motional states conditioned on their internal states, and carefully timing these interactions to entangle the ions' internal states.

Measurement of trapped-ion qubits is performed using state-dependent fluorescence. A laser tuned to a transition that only one of the qubit states can absorb causes the ion to scatter photons if it is in that state. The presence or absence of fluorescence, detected by a photomultiplier or a CCD camera, indicates the qubit state with high fidelity.

References of Chapter 3

- [Che+23] B. Cheng et al. “Noisy intermediate-scale quantum computers.” In: *Frontiers of Physics* 18.2 (2023), p. 21308. ISSN: 2095-0470. DOI: [10.1007/s11467-022-1249-z](https://doi.org/10.1007/s11467-022-1249-z).
- [CZ95] J. I. Cirac and P. Zoller. “Quantum Computations with Cold Trapped Ions.” In: *Phys. Rev. Lett.* 74 (20 1995), pp. 4091–4094. DOI: [10.1103/PhysRevLett.74.4091](https://doi.org/10.1103/PhysRevLett.74.4091).
- [DiV00] D. P. DiVincenzo. “The Physical Implementation of Quantum Computation.” In: *Fortschritte der Physik* 48.9-11 (2000), pp. 771–783. DOI: [10.1002/1521-3978\(200009\)48:9/11<771::AID-PROP771>3.0.CO;2-E](https://doi.org/10.1002/1521-3978(200009)48:9/11<771::AID-PROP771>3.0.CO;2-E).

- [DVC00] W. Dür, G. Vidal, and J. I. Cirac. “Three qubits can be entangled in two inequivalent ways.” In: *Phys. Rev. A* 62 (6 2000), p. 062314. DOI: [10.1103/PhysRevA.62.062314](https://doi.org/10.1103/PhysRevA.62.062314).
- [GHZ07] D. M. Greenberger, M. A. Horne, and A. Zeilinger. *Going Beyond Bell’s Theorem*. 2007. arXiv: [0712.0921](https://arxiv.org/abs/0712.0921).
- [Ran20] S.-J. Ran. “Encoding of matrix product states into quantum circuits of one- and two-qubit gates.” In: *Phys. Rev. A* 101 (3 2020), p. 032310. DOI: [10.1103/PhysRevA.101.032310](https://doi.org/10.1103/PhysRevA.101.032310).
- [Sch+05] C. Schön et al. “Sequential Generation of Entangled Multiqubit States.” In: *Phys. Rev. Lett.* 95 (11 2005), p. 110503. DOI: [10.1103/PhysRevLett.95.110503](https://doi.org/10.1103/PhysRevLett.95.110503).
- [SM99] A. Sørensen and K. Mølmer. “Quantum Computation with Ions in Thermal Motion.” In: *Phys. Rev. Lett.* 82 (9 1999), pp. 1971–1974. DOI: [10.1103/PhysRevLett.82.1971](https://doi.org/10.1103/PhysRevLett.82.1971).
- [SW10] E. M. Stoudenmire and S. R. White. “Minimally entangled typical thermal state algorithms.” In: *New Journal of Physics* 12.5 (2010), p. 055026. DOI: [10.1088/1367-2630/12/5/055026](https://doi.org/10.1088/1367-2630/12/5/055026).
- [Vid04] G. Vidal. “Efficient Simulation of One-Dimensional Quantum Many-Body Systems.” In: *Phys. Rev. Lett.* 93 (4 2004), p. 040502. DOI: [10.1103/PhysRevLett.93.040502](https://doi.org/10.1103/PhysRevLett.93.040502).

Part II

Applications

Tensor networks (TNs) have emerged as one of the most powerful tools for simulating complex quantum systems, particularly when dealing with the exponential growth of Hilbert space in many-body quantum mechanics. This Part of the book looks into the practical applications of tensor networks in modeling quantum dynamics from optimization problems to open system dynamics and resource theory and non-stabilizerness. We specifically focus into their application for addressing a subset of key challenges in quantum computation, particularly in the context of many-body systems. These problems, while not exhaustive due to the authors' preferences, expertise, and space constraints, are closely related by their shared reliance on tensor network techniques and the fundamental principles of quantum mechanics. Through these carefully selected topics, we aim to provide a representative exploration of how tensor networks can be employed to efficiently simulate quantum dynamics and optimize complex quantum systems.

Here, we will highlight the key frameworks, methods, and challenges tackled through the lens of tensor networks, setting the stage for a deeper exploration of these concepts in subsequent chapters.

Unitary Evolution of Hamiltonian Dynamics — The accurate simulation of quantum systems evolving under unitary transformations governed by time-dependent Hamiltonians remains a core challenge in quantum physics. Tensor networks, particularly the Matrix Product States (MPS) and Matrix Product Operators (MPO) formalisms, offer a scalable approach to representing quantum states and operators, allowing for simulations that would otherwise be computationally intractable. One of the fundamental methods employed is the *Time-Dependent Variational Principle (TDVP)*, which provides an efficient means to approximate the evolution of quantum states with a fixed bond dimension. TDVP has been extensively studied and applied in the context of one-dimensional systems, where its ability to control truncation errors is particularly valuable for long-time simulations.

In addition, tensor networks have found widespread application in *Quadratic Unconstrained Binary Optimization (QUBO)* problems, which are integral to classical and quantum optimization algorithms. These problems can be mapped to Ising-like models, enabling the use of TN methods to solve real-world optimization challenges that are NP-hard.

Quantum Annealing and Digitized Approaches — One of the most intriguing applications of TNs is in *Quantum Annealing (QA)*, a method designed to find the ground state of a Hamiltonian by slowly varying a control parameter. QA can be effectively simulated on both quantum and classical

devices. The *Digitized Quantum Annealing (dQA)* approach, a discretized version of QA, allows for a gate-based implementation on quantum simulators. Tensor networks provide a natural framework for dQA simulations, where they allow for the compression of the exponentially growing Hilbert space into more manageable forms. Notably, TN methods have been employed to simulate dQA with a high degree of precision, as demonstrated in studies where MPS and MPO techniques are used to efficiently capture the state evolution under quantum annealing protocols.

Furthermore, *Quantum Approximate Optimization Algorithm (QAOA)*, a variational quantum algorithm, leverages principles of both QA and gate-based quantum computation. Tensor networks can be used to analyze the performance of QAOA, optimizing the classical parameters that drive the quantum circuit. This provides an avenue for classical simulations of quantum algorithms aimed at solving combinatorial optimization problems.

Open Tensor Networks and Open System Dynamics — In real-world quantum systems, interactions with the environment cannot be ignored, giving rise to *open quantum systems*. TNs have been successfully adapted to simulate such systems by extending traditional MPS techniques to handle mixed quantum states and non-unitary evolutions. *Tensor Network and Density Operators* techniques allow for an efficient representation of mixed states, where MPO formalism plays a crucial role. A prominent approach within this context is the *Minimally Entangled Typical Thermal States (METTS)* algorithm, which uses tensor networks to sample typical thermal states, providing insight into quantum thermodynamics.

Incorporating environmental effects often requires the use of the *Lindblad Master Equation*, a formalism that describes the evolution of open quantum systems. TN-based algorithms such as the *Locally Purified Tensor Network* method offer a pathway to simulate open quantum dynamics in an efficient manner, enabling the study of decoherence, dissipation, and other non-unitary processes.

Finally, TNs are crucial for simulating quantum measurement protocols in open systems, particularly when dealing with *monitored dynamics*, where the evolution of the system depends on continuous measurement outcomes. These *unravelling schemes*, in combination with TN methods, provide an effective means to model complex quantum behavior in open systems.

Tensor Networks and Quantum Magic — The last part of this section addresses the interplay between tensor networks and quantum “magic” — a term that refers to the non-stabilizer nature of quantum states, which plays

a crucial role in the power of quantum computation beyond Clifford operations. Understanding how tensor networks can capture and quantify the non-stabilizerness of a quantum system is essential for leveraging their full potential in quantum information processing.

The chapter begins by introducing the stabilizer formalism, which underpins much of the classical simulability of certain quantum systems. Stabilizer states, those that can be expressed using Pauli and Clifford operators, can be simulated efficiently on classical computers, thanks to the Gottesman-Knill theorem. This efficiency, however, highlights the limitation of such states from the perspective of quantum computational advantage. Truly leveraging the power of quantum computation requires moving beyond stabilizer circuits — into the domain of nonstabilizer, or “magic” states.

Quantum magic is identified as a critical resource for achieving quantum advantage. This concept refers to non-classicality beyond stabilizer operations and is essential for quantum speedup. Measuring the degree of nonstabilizerness is crucial for evaluating how far a quantum system departs from stabilizer form, thus revealing its true computational power.

Following this foundational introduction, the chapter explores measuring magic in tensor network states, where tensor network representations provide efficient descriptions of quantum many-body systems. The sampling techniques and replica methods outlined in this chapter serve as fundamental tools for quantifying the magic present in tensor network states, allowing for a deeper understanding of their complexity and power.

The latter part of the chapter introduces advanced techniques for handling tensor networks in the presence of magic. In particular, the development of Clifford-enhanced Matrix Product States (CMPS) offers a novel way to represent complex quantum states by combining the structure of stabilizer states with the flexibility of tensor networks. This hybrid approach is extended to incorporate stabilizer MPOs (Matrix Product Operators) and Clifford-dressed TDVP (Time-Dependent Variational Principle), providing a framework that balances computational efficiency with the need to capture nonstabilizer dynamics.

Chapter 4

Unitary Evolution of Hamiltonian Dynamics

Life is a dynamic rather than a static process, and when we don't change it kills us. It's not running away, it's moving on.

Irvine Welsh

Much of Quantum mechanics involves solving the Schrödinger's equation,

$$\frac{\partial}{\partial t} |\psi\rangle = -i\hat{H} |\psi\rangle. \quad (4.1)$$

This fundamental equation dictates the unitary evolution of a pure quantum state, akin to how Newton's laws describe the evolution of classical systems. This seemingly harmless first-order time differential equation is analytically tractable only for a select few problems. Particularly for many-body systems, the curse of dimensionality strikes, and its unitary dynamics become an extremely challenging problem in quantum mechanics. Several popular numerical methods have been developed over time, each with certain benefits and drawbacks compared to others. Tensor network methods are a recent addition to this list. In the first two introductory chapters, we introduced tensor networks and their efficiency in representing one-dimensional quantum many-body systems, with states written as MPS and the Hamiltonian as MPO. Furthermore in the second chapter we also gave a comprehensive introduction to the TEBD algorithm (see Section 3.2) which is the first tensor network based algorithm to calculate the unitary dynamics of pure quantum

state. The TEBD algorithm decomposes the full Hamiltonian into a sequence of two-site local unitary gates. In this regard it is also a suitable algorithm for the discrete evolution of the quantum state via a sequence of random local unitaries. In this chapter we will focus in another approach for unitary evolution of a pure quantum state, specifically the one that preserves the structure of the Hamiltonian and is continuous in time, namely the Time-Dependent Variational Principle (TDVP) algorithm. The TDVP [Hae+11; Hae+16] is a variational approach based on the idea of MPS manifold and tangent space.

4.1 Time-Dependent Variational Principle (TDVP)

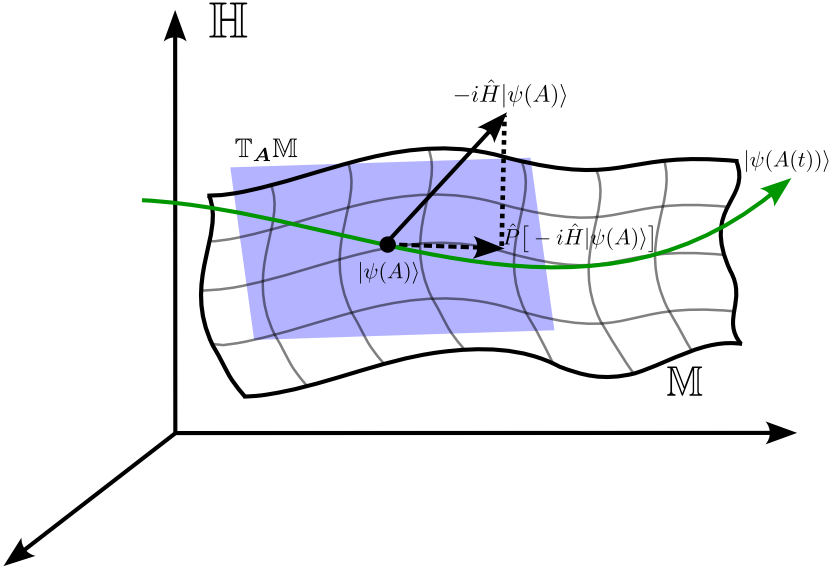
With the quantum state written as an MPS the Schrödinger's equation can be written as,

$$\frac{\partial}{\partial t} |\psi(A(t))\rangle = -i\hat{H} |\psi(A(t))\rangle, \quad (4.2)$$

where,

$$|\psi(A)\rangle = \sum_{x_1} \cdots \sum_{x_n} \begin{array}{c} \text{A}_1 \\ | \\ x_1 \end{array} \text{---} \begin{array}{c} \text{A}_2 \\ | \\ x_2 \end{array} \text{---} \cdots \cdots \text{---} \begin{array}{c} \text{A}_n \\ | \\ x_n \end{array} |x_1\rangle \cdots |x_n\rangle \quad (4.3)$$

The MPS $|\psi(A)\rangle$ resides in a smooth manifold \mathbb{M}_χ , which is a subset of the Hilbert space, $\mathbb{M}_\chi \subset \mathbb{H}$. This manifold \mathbb{M}_χ is characterized by the constraint that it exclusively contains MPS with a fixed bond dimension χ . It should be noted that although \mathbb{M}_χ is a subset of the Hilbert space it is not a vector space because adding two MPS with bond dimension χ results in a MPS of a larger bond dimension and will therefore not lie within \mathbb{M}_χ . Furthermore, applying a Hamiltonian onto the MPS also increases the bond dimension, therefore the right hand side of the Schrödinger's equation (4.2) leads the MPS out of the manifold. The central idea of TDVP is to constrain the unitary evolution within the manifold \mathbb{M}_χ or in terms of MPS language we restrict the growth of the bond dimension to $\chi_{\max} = \chi$. This is achieved by projecting the right hand side of the Schrödinger's equation, $-i\hat{H} |\psi(A(t))\rangle$ back into the tangent space $\mathbb{T}_A\mathbb{M}$ defined at point $|\psi(A)\rangle$. The figure below demonstrates this process, the smooth curved space is the manifold \mathbb{M}_χ embedded inside the Hilbert space \mathbb{H} , the blue plane is the tangent space $\mathbb{T}_A\mathbb{M}$ defined at point $|\psi(A)\rangle$, the bold arrow pointing out of the manifold is the right hand side of the Schrödinger's equation which is projected back to the tangent space as shown by the dashed arrow, finally, the green curve that lies entirely within the manifold represents the optimal evolution of the quantum state.



The effective Schrödinger’s equation such that the quantum state always remains in the manifold \mathbb{M}_χ is,

$$\frac{\partial}{\partial t} |\psi(A(t))\rangle = -i\hat{P}_{\mathbb{T}_A^{[2]}\mathbb{M}_\chi} \hat{H} |\psi(A(t))\rangle, \tag{4.4}$$

Here, the superscript [2] signifies two site TDVP which is more accurate and adaptive compared to the one-site TDVP [Hae+16]. $\hat{P}_{\mathbb{T}_A^{[2]}\mathbb{M}_\chi}$ projects the MPS $-i\hat{H} |\psi(A(t))\rangle$ onto the tangent space $\mathbb{T}_A \mathbb{M}_\chi$ and is defined as,

$$\hat{P}_{\mathbb{T}_A^{[2]}\mathbb{M}_\chi} = \sum_{l=1}^{n-1} \hat{P}_L^{l-1} \otimes \mathbb{I}^l \otimes \mathbb{I}^{l+1} \otimes \hat{P}_R^{l+2} - \sum_{l=1}^{n-2} \hat{P}_L^l \otimes \mathbb{I}^{l+1} \otimes \hat{P}_R^{l+2}. \tag{4.5}$$

The full derivation of the projector can be found in the following articles: [Hae+11; Hae+16]. The parts of the projector are defined as

$$\hat{P}_L^{l-1} \otimes \mathbb{I}^l \otimes \mathbb{I}^{l+1} \otimes \hat{P}_R^{l+2} = \begin{array}{c} \begin{array}{ccccccc} 1 & l-2 & l-1 & l & l+1 & l+2 & n \\ \begin{array}{c} \text{---} \blacktriangleright \text{---} \text{---} \blacktriangleright \text{---} \text{---} \blacktriangleright \text{---} \\ \text{---} \blacktriangleright \text{---} \text{---} \blacktriangleright \text{---} \text{---} \blacktriangleright \text{---} \end{array} \\ \hat{P}_L^{l-1} & \mathbb{I}^l & \mathbb{I}^{l+1} & \hat{P}_R^{l+2} \end{array} \end{array} \tag{4.6}$$

$$\hat{P}_L^l \otimes \mathbb{I}^{l+1} \otimes \hat{P}_R^{l+2} =$$

$$(4.7)$$

Replacing equation (4.5) into equation (4.4) we get,

$$\frac{\partial}{\partial t} |\psi\rangle = -i \sum_{l=1}^{N-1} \hat{P}_L^{l-1} \otimes \mathbb{I}^l \otimes \mathbb{I}^{l+1} \otimes \hat{P}_R^{l+2} \hat{H} |\psi\rangle + i \sum_{l=1}^{N-2} \hat{P}_L^l \otimes \mathbb{I}^{l+1} \otimes \hat{P}_R^{l+2} \hat{H} |\psi\rangle. \quad (4.8)$$

The exact solution of equation (4.8) is not tractable, however, the individual terms are integrable exactly. This kind of problem can be solved by Lie-Trotter splitting of the operators [Hol+10] and sequentially evolving a pair of differential equations for a time step of Δt ,

$$\frac{\partial}{\partial t} |\psi\rangle = -i \hat{P}_L^{l-1} \otimes \mathbb{I}^l \otimes \mathbb{I}^{l+1} \otimes \hat{P}_R^{l+2} \hat{H} |\psi\rangle, \quad (4.9)$$

$$\frac{\partial}{\partial t} |\psi\rangle = i \hat{P}_L^l \otimes \mathbb{I}^{l+1} \otimes \hat{P}_R^{l+2} \hat{H} |\psi\rangle. \quad (4.10)$$

This approach incurs a local time step error of $O(\Delta t^2)$. These two differential equations have opposite signs, this can be interpreted as a set of forward and backward evolution in time. Let us introduce a two-site map $\bar{\psi}_{l-1}^L \otimes \mathbb{I}^l \otimes \mathbb{I}^{l+1} \otimes \bar{\psi}_{l+2}^R$ and single-site map $\bar{\psi}_l^L \otimes \mathbb{I}^{l+1} \otimes \bar{\psi}_{l+2}^R$ with the following definitions,

$$\bar{\psi}_{l-1}^L \otimes \mathbb{I}^l \otimes \mathbb{I}^{l+1} \otimes \bar{\psi}_{l+2}^R =$$

$$(4.11)$$

$$\bar{\psi}_L^l \otimes \mathbb{I}^{l+1} \otimes \bar{\psi}_R^{l+2} =$$

$$(4.12)$$

We apply the two-site map (4.11) onto equation (4.9) and the single-site map (4.12) onto equation (4.10). The resulting left hand side of these equations

are two-site and single-site tensors given as,

$$\bar{\psi}_L^{l-1} \otimes \mathbb{I}^l \otimes \mathbb{I}^{l+1} \otimes \bar{\psi}_R^{l+2} |\psi\rangle = \text{Diagram} = T^{l,l+1}, \quad (4.13)$$

$$\bar{\psi}_L^l \otimes \mathbb{I}^{l+1} \otimes \bar{\psi}_R^{l+2} |\psi\rangle = \text{Diagram} = A^{l+1}. \quad (4.14)$$

Here we assumed that $|\psi\rangle$ is in a mixed canonical form with orthogonality center at site $l + 1$ and exploited the left and right unitary properties of the canonical tensors. Similarly, the right hand side of these equations are also two-site and single-site tensors albeit more complex given by the following two diagrammatic equations,

$$\begin{aligned} \hat{H}|\psi\rangle & \left\{ \begin{array}{l} \text{Diagram 1} \\ \text{Diagram 2} \\ \text{Diagram 3} \end{array} \right\} = \text{Diagram 4} \\ \hat{P}_L^{l-1} \otimes \mathbb{I}^l \otimes \mathbb{I}^{l+1} \otimes \hat{P}_R^{l+2} & \left\{ \begin{array}{l} \text{Diagram 1} \\ \text{Diagram 2} \\ \text{Diagram 3} \end{array} \right\} \\ \bar{\psi}_L^{l-1} \otimes \mathbb{I}^l \otimes \mathbb{I}^{l+1} \otimes \bar{\psi}_R^{l+2} & \left\{ \begin{array}{l} \text{Diagram 1} \\ \text{Diagram 2} \\ \text{Diagram 3} \end{array} \right\} \end{aligned} = \hat{H}_{\text{eff}}^{l,l+1} T^{l,l+1}, \quad (4.15)$$

$$\begin{aligned} \hat{H}|\psi\rangle & \left\{ \begin{array}{l} \text{Diagram 1} \\ \text{Diagram 2} \\ \text{Diagram 3} \end{array} \right\} = \text{Diagram 4} \\ \hat{P}_L^l \otimes \mathbb{I}^{l+1} \otimes \hat{P}_R^{l+2} & \left\{ \begin{array}{l} \text{Diagram 1} \\ \text{Diagram 2} \\ \text{Diagram 3} \end{array} \right\} \\ \bar{\psi}_L^l \otimes \mathbb{I}^{l+1} \otimes \bar{\psi}_R^{l+2} & \left\{ \begin{array}{l} \text{Diagram 1} \\ \text{Diagram 2} \\ \text{Diagram 3} \end{array} \right\} \end{aligned} = \hat{H}_{\text{eff}}^{l+1} A^{l+1}. \quad (4.16)$$

Finally, the equations (4.9) and (4.10) becomes a pair of two-site and single-site effective equations given by,

$$\frac{\partial}{\partial t} T^{l,l+1}(t) = -i \hat{H}_{\text{eff}}^{l,l+1} T^{l,l+1}(t) \rightarrow T^{l,l+1}(t) = e^{-it \hat{H}_{\text{eff}}^{l,l+1}} T^{l,l+1}(0), \quad (4.17)$$

$$\frac{\partial}{\partial t} A^{l+1}(t) = i \hat{H}_{\text{eff}}^{l+1} A^{l+1}(t) \rightarrow A^{l+1}(t) = e^{it \hat{H}_{\text{eff}}^{l+1}} A^{l+1}(t). \quad (4.18)$$

The two-site and single-site effective Hamiltonians $\hat{H}_{\text{eff}}^{l,l+1}$, $\hat{H}_{\text{eff}}^{l+1}$ are both rank-8 tensors as defined in the equations (4.15) and (4.16) respectively. The

set of equations (4.17) and (4.18) can be integrated using iterative Lanczos exponential solver which has a computational cost of $O(\chi^3)$ [Lan50; Sim84; ML03]. Furthermore, the error incurred during Lanczos solver is well controlled and can be made arbitrarily small by sufficiently increasing the number of Krylov vectors. Solving these local equations forms the bottleneck of TDVP algorithm which comprises of solving these local equations iteratively over the system sites.

Example 4.1: Lanczos Exponential Solver

The Lanczos algorithm was initially developed for the purpose of tri-diagonalizing a Hermitian matrix. Essentially, the Lanczos process converts an $N \times N$ Hermitian matrix M into an $N \times N$ tridiagonal matrix over the course of n steps. The importance of this process lies in its ability to approximate the extreme eigenvalues and eigenvectors of the Hermitian matrix M by using the extreme eigenvalues and eigenvectors of a $k \times k$ tridiagonal matrix after just $k \ll N$ iterations. The Lanczos algorithm starts with an $N \times N$ Hermitian matrix M and an initial guess vector v_{input} . After k iterations, it produces a $k \times k$ tridiagonal matrix $T_{k \times k}$ along with a set of orthogonal column vectors $V_{N \times k}$. The algorithm can be described concisely as follows:

```

1:  $v_1 = v_{\text{input}} / |v_{\text{input}}|$  ▷ input vector
2: for  $i \in [2, k + 1]$  do
3:    $v_i = M \times v_{i-1}$ 
4:   for  $j \in [i - 2, i - 1]$  do ▷ Building tridiagonal matrix
5:     if  $j \geq 1$  then
6:        $T_{i-1,j} = \text{dot}(v_i, v_j)$ 
7:        $T_{j,i-1} = T_{i-1,j}^*$ 
8:     end if
9:   end for
10:  for  $f \in [1, i - 1]$  do ▷ Full reorthonormalization
11:     $v_i = v_f - \text{dot}(v_i, v_f) \times v_f$ 
12:     $v_i = v_i / |v_i|$ 
13:  end for
14: end for
15: return  $T, [v_1, v_2, \dots, v_k]$ 

```

The set of orthonormal vectors $V_{n \times k} = [v_1, v_2, \dots, v_k]$ are known as

Krylov vectors that forms the basis of Krylov subspace [Lan50; Pai71]. The number k is known as the Krylov dimension.

Finally, the matrix exponential applied to a vector can be approximated by the tridiagonal matrix as,

$$e^M \times v_{\text{input}} \approx V_{n \times k} \times e^{T_{k \times k}} \times \mathbb{1}_{k \times k}[:, 1], \quad (4.19)$$

where, $\mathbb{1}_{k \times k}[:, 1]$ is the first column of the $k \times k$ identity matrix [ML03].

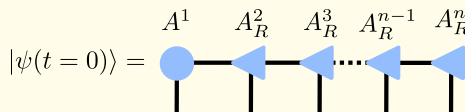
Note: in equations (4.17) and (4.18) $\hat{H}_{\text{eff}}^{l,l+1}$, $\hat{H}_{\text{eff}}^{l+1}$, $T^{l,l+1}$, and A^{l+1} are not matrices and vectors but rather high rank tensors. In this case the Lanczos algorithm can be adapted for exponential of a higher rank tensor applied to another higher rank tensor.

Example 4.2: TDVP algorithm

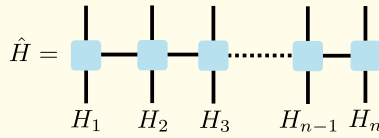
The TDVP algorithm is fairly similar to the Density Matrix Renormalization Group (DMRG) algorithm [Sch11] in the sense that it involves sequential sweeping from left-to-right and right-to-left through the system sites. The right-to-left sweep is adjoint to the left-to-right sweep, giving a second-order symmetric integrator scheme so that the error time step error becomes of order $O(\Delta t^3)$. In the algorithm we then set $\Delta t \rightarrow \Delta t/2$ and consider the left-to-right and right-to-left sweep as a single time step [Hae+16]. The TDVP algorithm can be decomposed into three main parts;

1. **Initialization:** This part involves setting up the system ready before the sweeps. The requirements are as follows:

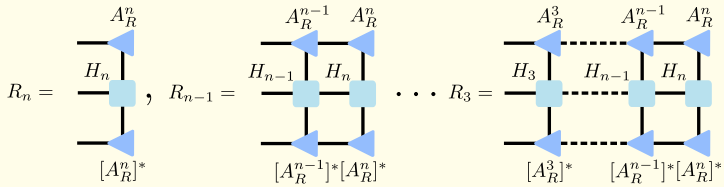
- (a) The initial state $|\psi(t=0)\rangle$ in right canonical form. This can be achieved by a sequence of SVD/QR decompositions starting from the n^{th} site to the first site.



(b) The Hamiltonian \hat{H} as an MPO.

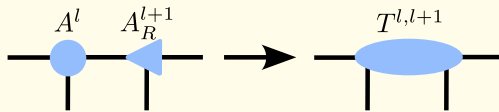


(c) Right environment tensors $\{R_i\}_{i=3}^{i=n}$.

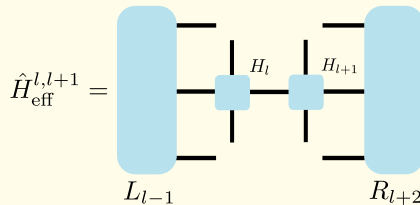


2. Left-to-right sweep

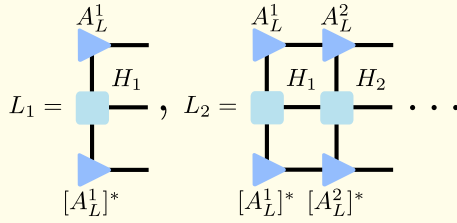
- 1: **for** $l \in [1, n - 1]$ **do**
- 2: $T^{l,l+1} \leftarrow A^l A_R^{l+1}$



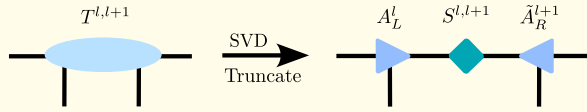
- 3: Solve the forward local evolution: $T^{l,l+1}(t + \Delta t/2) \leftarrow e^{-i \frac{\Delta t}{2} \hat{H}_{\text{eff}}^{l,l+1}} T^{l,l+1}(t)$, where $\hat{H}_{\text{eff}}^{l,l+1}$ is defined as,



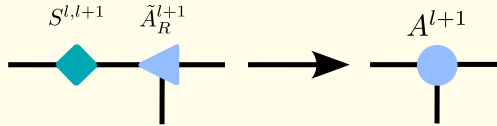
Here, L_{l-1} is the $l - 1^{th}$ left environment tensor akin to the right environment tensors. Note that L_0 and R_{n+1} corresponds to a scalar 1.0. They are build successively as,



4: $A_L^l S^{l,l+1} \tilde{A}_R^{l+1} \leftarrow T^{l,l+1} \quad \triangleright \text{SVD and truncation}$

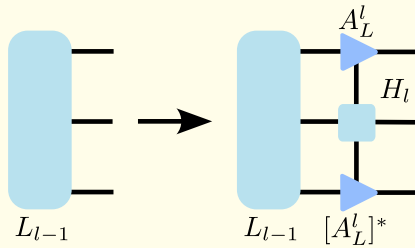


5: $S^{l,l+1} \tilde{A}_R^{l+1} \leftarrow A^{l+1}$

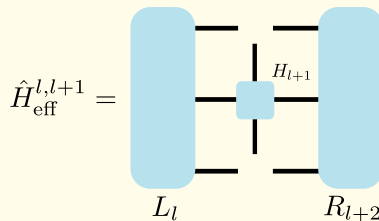


6: **if $l \neq n - 1$ then**

7: $L_l \leftarrow L_{l-1} [A_L^l]^* H_l A_L^l$



8: Solve the backward local evolution: $A^{l+1}(t - \Delta t/2) \leftarrow e^{i \frac{\Delta t}{2} \hat{H}_{\text{eff}}^{l+1}} A^{l+1}(t)$, where $\hat{H}_{\text{eff}}^{l+1}$ is defined as,



- ```

9: Delete R_{l+2}
10: end if
11: end for

```
3. **Right-to-left sweep:** At the end of left-to-right sweep the MPS is in left canonical form and we have a set of left environment matrices  $\{L_i\}_{i=1}^{n-1}$ . All the steps involved in the right-to-left sweep are equivalent to left-to-right sweeps except for the direction of iteration. So we will outline only the algorithm without the diagrams.
- ```

1: for  $l \in [n, 2]$  do
2:    $T^{l-1,l} \leftarrow A_L^{l-1} A^l$ 
3:    $T^{l-1,l}(t + \Delta t/2) \leftarrow e^{-i \frac{\Delta t}{2} \hat{H}_{\text{eff}}^{l-1,l}} T^{l-1,l}(t)$ 
4:    $\tilde{A}_L^{l-1} S^{l-1,l} A_R^l \leftarrow T^{l-1,l}(t + \Delta t)$   $\triangleright$  SVD and truncation
5:    $\tilde{A}_L^{l-1} S^{l-1,l} \leftarrow A^{l-1}$ 
6:   if  $l \neq 2$  then
7:      $R_l \leftarrow R_{l+1} [A_R^l]^* H_l A_R^l$ 
8:      $A^{l-1}(t - \Delta t/2) \leftarrow e^{i \frac{\Delta t}{2} \hat{H}_{\text{eff}}^{l-1}} A^{l-1}(t)$ 
9:     Delete  $L_{l-2}$ 
10:  end if
11: end for

```
4. Steps 2 and 3 evolves the state by a time step Δt : $|\psi(t + \Delta t)\rangle \leftarrow |\psi(t)\rangle$. The full evolution is achieved by successive iteration of these steps.

The TDVP algorithm encounters four main types of errors. The first error emerges during the projection of the exact Schrödinger equation onto (4.4), stemming from the limited bond dimensions of the auxiliary indices in MPS tensors. To gauge this error, one can observe how TDVP data converges as the bond dimensions increase. The second error is related to the finite time step when approximating the differential equation (4.8) with a series of solvable local differential equations (4.9) and (4.10). For the second-order integrator described earlier, this error scales as $O(\delta t^3)$ per time-step. The third type of error is due to an insufficient number of Krylov vectors in solving the local differential equations (4.17) and (4.18). This error can be made arbitrarily small by using a sufficiently large number of Krylov vectors in the Lanczos exponential solver. The final source of error arises from the truncation of the singular values in the local MPS tensors. Reducing the time step Δt

will decrease the time step error and the error incurred during solving local differential equations. The other two errors, projection error and truncation error are however independent of the discrete time step and therefore taking smaller Δt will wind up more overall projection and truncation error since it requires more discrete steps to reach the final state. In a real simulation the optimal Δt is often determined by several rounds of hit and trials.

4.2 Quadratic unconstrained binary optimization

The Quadratic Unconstrained Binary Optimization (QUBO) problem is a form of optimization where the goal is to find the values of binary variables that minimize (or maximize) a quadratic objective function. This is pivotal in various scientific and engineering fields, such as finance, logistics, machine learning, and especially quantum computing, because of its compatibility with quantum annealers and certain quantum algorithms.

The QUBO problem involves the optimization of the following function

$$f(\mathbf{x}) = \sum_{ij} x_i Q_{ij} x_j + \sum_i c_i x_i \quad (4.20)$$

where c_i represent the weight associated to each component and Q_{ij} the weights associated with each pair of indices, defining the structure of the problem, and $x_j \in \{0, 1\}$ are binary variables.

The optimal solution, \mathbf{x}^* , minimizes the objective function and satisfies the condition

$$\forall \mathbf{x} \in \{0, 1\}^n \quad f(\mathbf{x}^*) \leq f(\mathbf{x}). \quad (4.21)$$

In its most general case, the QUBO problem is NP-hard, meaning it cannot be efficiently solved by any algorithm that runs in polynomial time. Before dwelling on how to map QUBO problems in a quantum mechanical problem, let us give some examples of such problems.

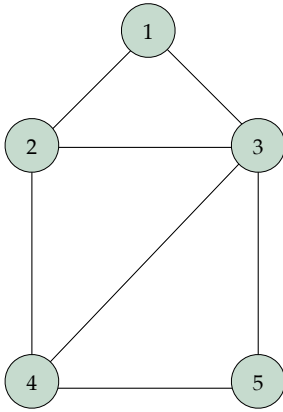
Unweighted graphs — A graph is a mathematical structure that is defined as a pair $G = (V, E)$, where V represents the set of vertices (or nodes) and E represents the set of edges (or links) between these vertices. The vertices can represent entities such as points, objects, or locations, while the edges represent the relationships or connections between pairs of vertices.

One common way to represent a graph is through its adjacency matrix A . The adjacency matrix A is a square matrix of size $n \times n$, where n is the

number of vertices in the graph. The entries of A are defined as follows

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge between vertex } i \text{ and vertex } j, \\ 0 & \text{otherwise.} \end{cases} \quad (4.22)$$

For undirected graphs, the adjacency matrix A is symmetric, meaning $A_{ij} = A_{ji}$. In contrast, for directed graphs, the adjacency matrix may not be symmetric, as the presence of an edge from vertex i to vertex j does not necessarily imply the presence of an edge from vertex j to vertex i .



$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \quad (4.23)$$

Weighted graphs — In addition to simple graphs, there are also weighted graphs, where each edge is assigned a numerical value or weight. A weighted graph can be represented by a triplet $G = (V, E, w)$, where V is the set of vertices, E is the set of edges, and $w : E \rightarrow \mathbb{R}$ is a weight function that assigns a real number to each edge.

In the context of a weighted graph, the adjacency matrix A is often extended to include these weights. Specifically, the weighted adjacency matrix of a weighted graph is defined as follows

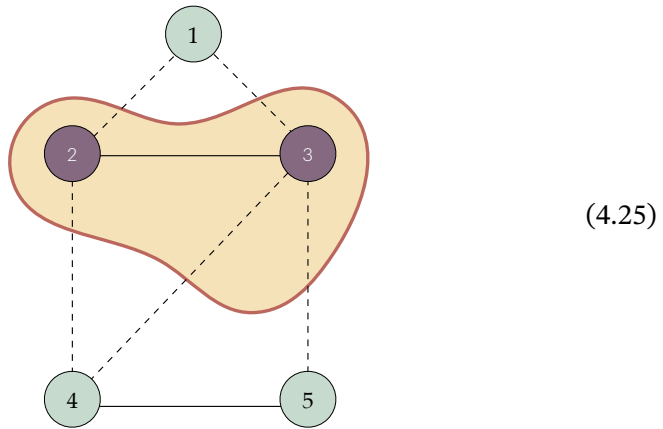
$$W_{ij} = \begin{cases} w_{ij} & \text{if there is an edge between vertex } i \text{ and vertex } j, \\ 0 & \text{otherwise.} \end{cases} \quad (4.24)$$

Here, w_{ij} represents the weight of the edge between the vertex i and the vertex j . For undirected weighted graphs, the adjacency matrix remains symmetric, that is, $W_{ij} = W_{ji}$, while for directed weighted graphs, the matrix may still be asymmetric depending on the direction and weights of the edges.

Weighted graphs are particularly useful in modeling real-world problems where relationships between entities have different strengths, costs, or capacities. Examples include road networks where edges represent distances or

travel times, communication networks where edges represent bandwidth or latency, and financial networks where edges represent transaction amounts.

Max-Cut problem — The Max-Cut problem is a fundamental problem in graph theory and combinatorial optimization. Given an undirected graph $G = (V, E)$, the objective of the Max-Cut problem is to partition the set of vertices into two disjoint subsets such that the number of edges between the two subsets is maximized. Equivalently, the problem seeks to maximize the weight of the cut, where the weight is defined as the sum of the weights of the edges that have endpoints in different subsets. For instance, the Max-Cut of (4.23) is the following



It is straightforward to express the Max-Cut problem as a QUBO optimization. We begin by assigning a binary variable x_j to each vertex j , where $x_j = 0$ if the vertex is in the first subset and $x_j = 1$ if it is in the second subset. Then, it is easy to figure out that the matrix Q in the MaxCut problem is related to the adjacency matrix A of the undirected graph. Indeed, we have to minimize the function

$$f(\mathbf{x}) = -\sum_{i,j} A_{ij}(2x_i - 1)(2x_j - 1). \quad (4.26)$$

which is such that for every connected $x_i \neq x_j$ contributes -1 to the value of $f(\mathbf{x})$, while every $x_i = x_j$ contribute with $+1$. Equivalently, discarding an unnecessary additive constant the problem is equivalent to find the \mathbf{x}^* which minimizes

$$f(\mathbf{x}) = \sum_{ij} A_{ij} (2x_i x_j - x_i - x_j) \quad (4.27)$$

Traveling salesman problem — The Traveling Salesman Problem (TSP) is a classic optimization problem in the field of combinatorial mathematics and computer science. It involves determining the most efficient route for a seller who must visit a given set of cities, stopping at each city exactly once, and then returning to the starting point. The challenge lies in finding the shortest possible route that accomplishes this task, given the distances or costs between every pair of cities.

The problem is known for its computational complexity; as the number of cities increases, the number of possible routes grows factorially, making an exhaustive search impractical for large instances. Again, as for the Max-Cut problem, despite its simplicity in formulation, TSP is NP-hard, meaning that no efficient algorithm is known to solve all instances of the problem in polynomial time.

The TSP can be formulated as a QUBO optimization problem. For N cities at distances d_{ij} , we define a set of $n = N^2$ binary variables $x_{i,\mu}$, where i represents the index of a city, and μ denotes the position of that city in the sequence of the route. Specifically, if $x_{2,3} = 1$, it indicates that the second city is the third one visited in the route.

To ensure that the solution represents a valid tour, where each city is visited exactly once and every position in the sequence corresponds to exactly one city, we impose the following penalties or constraints

$$\sum_{i=1}^N \left(1 - \sum_{\mu=1}^N x_{i,\mu} \right)^2, \quad \sum_{\mu=1}^N \left(1 - \sum_{i=1}^N x_{i,\mu} \right)^2. \quad (4.28)$$

These equations guarantee that each city is assigned a unique position in the route and that each position in the route is occupied by only one city. The first equation ensures that every city appears exactly once in the sequence, while the second equation ensures that each position in the route is filled by one and only one city. If some cities are not reachable directly from other cities, also include the equation

$$\sum_{ij} (1 - A_{ij}) x_{i,\mu} x_{j,\mu+1} \quad (4.29)$$

where A is the adjacency matrix of the unweighted graph. Finally, we have to include the cost of the route in the objective function, which is given by

$$h \sum_{ij} d_{ij} \sum_{\mu} x_{i,\mu} x_{j,\mu+1}. \quad (4.30)$$

The scaling factor h has to be small enough that it is never favorable to violate that the solution represents a valid tour, that is, $0 < h \max d < 1$. The QUBO problem that solves the TSP is therefore the following

$$f(\mathbf{x}) = \sum_{i=1}^N \left(1 - \sum_{\mu=1}^N x_{i,\mu} \right)^2 + \sum_{\mu=1}^N \left(1 - \sum_{i=1}^N x_{i,\mu} \right)^2 + \sum_{ij} (1 - A_{ij}) x_{i,\mu} x_{j,\mu+1} + h \sum_{ij} d_{ij} \sum_{\mu} x_{i,\mu} x_{j,\mu+1} \quad (4.31)$$

4.2.1 Mapping to Ising model

The QUBO framework is relevant in the field of quantum computing since it is computationally equivalent to ground state search of a classical Ising model, which is described by the Hamiltonian function

$$H(\sigma) = - \sum_{\langle i,j \rangle} J_{ij} \sigma_i \sigma_j - \sum_j h_j \sigma_j \quad (4.32)$$

where the parameters h_j, J_{ij} are real values applicable to all pairs (i, j) . As usual, in the Ising model, the spin variables σ_j are binary, taking values from $\{-1, +1\}$.

These spin variables are usually arranged in a lattice, allowing only adjacent pairs to interact significantly. Applying the transformation $\sigma \mapsto 2x - 1$, we can equivalently express the Ising model as a QUBO problem. The equivalent QUBO function is then given by

$$\begin{aligned} f(x) &= \sum_{\langle i,j \rangle} -J_{ij} (2x_i - 1)(2x_j - 1) + \sum_j h_j (2x_j - 1) \\ &= \sum_{\langle i,j \rangle} (-4J_{ij} x_i x_j + 2J_{ij} x_i + 2J_{ij} x_j - J_{ij}) + \sum_j (2h_j x_j - h_j) \\ &= \sum_{\langle i,j \rangle} (-4J_{ij} x_i x_j) + \sum_j \left(\sum_{\langle i,k=j \rangle} (2J_{ki} + 2J_{ik}) + 2h_j \right) x_j - \sum_{\langle i,j \rangle} J_{ij} - \sum_j h_j, \end{aligned} \quad (4.33)$$

where we define the QUBO matrix Q as

$$Q_{ij} = \begin{cases} -4J_{ij} & \text{if } i \neq j \\ \sum_{\langle i,k=j \rangle} (2J_{ki} + 2J_{ik}) + 2h_j & \text{if } i = j \end{cases} \quad (4.34)$$

and the constant term C as

$$C = - \sum_{\langle i,j \rangle} J_{ij} - \sum_j h_j. \quad (4.35)$$

Since the constant C does not influence the optimization of \mathbf{x}^* , it can generally be ignored during the optimization process, although it is critical for calculating the original Hamiltonian function's value.

4.3 Quantum Annealing

Recent explorations into adiabatic quantum optimization have sparked significant interest due to its potential to tackle QUBO problems. This quantum computational approach takes advantage of a dynamical adiabatic change in Hamiltonian parameters from a simple initial Hamiltonian H_0 , whose ground state is easily realizable, to a problem-specific Hamiltonian H_p , whose ground state encodes the solution to the desired problem.

The method relies on the principle of keeping the quantum system in its ground state while slowly changing from \hat{H}_x to \hat{H}_z (with $[\hat{H}_x, \hat{H}_z] \neq 0$)

$$\hat{H}(t) = (1 - s)\hat{H}_x + s\hat{H}_z, \quad (4.36)$$

with an interpolation parameter $s = s(t) \in [0, 1]$ such that $s(0) = 0$ and $s(\tau) = 1$, τ being the total annealing time. According to the adiabatic theorem, if T is sufficiently large, compared to the inverse square of the minimal spectral gap of $\hat{H}(s)$, the system remains in its lowest energy state throughout the evolution, resulting in a solution at $t = \tau$.

The efficiency of adiabatic quantum optimization has been under scrutiny, particularly concerning its potential to surpass classical computational speeds. The relationship between the system size N and the time τ , given by $\tau = O(\exp(\alpha N^\beta))$, suggests that maintaining the ground state, and thus avoiding transitions to excited states due to minute energy gaps (Landau-Zener transitions), demands exponentially increasing time as N approaches infinity. Although solving NP-complete problems in polynomial time using adiabatic optimization remains implausible, the actual coefficients α and β might be competitively lower than those in classical approaches, suggesting a possible advantages for specific problem classes.

In the case of a QUBO problem \hat{H}_z is the quantum counterpart of the classical Ising model, which, as already pointed out, typically involve a system of N spins $\sigma_i = \pm 1$ described by the Hamiltonian

$$H = - \sum_{i < j}^N J_{ij} \sigma_i \sigma_j - \sum_{i=1}^N h_i \sigma_i, \quad (4.37)$$

so that its quantum counterpart reads

$$\hat{H}_z = \hat{H}(\hat{Z}_1, \dots, \hat{Z}_N), \quad (4.38)$$

where \hat{Z}_i are Pauli matrices acting on the i -th qubit. To initialize the system, the initial state is the ground state of

$$\hat{H}_x = - \sum_{i=1}^N \hat{X}_i, \quad (4.39)$$

creating a superposition of all possible states in the computational basis. This setup creates an opportunity for employing tensor network methods. In particular, $\hat{H}(s)$ has a clear MPO description and the time evolution can be implemented for example with a TDVP algorithm. This allows to simulate and benchmark the performance of quantum annealing processes. This approach not only aids in verifying the efficacy of quantum annealing solutions but also enhances our understanding of the scalability and limitations of current quantum optimization technologies.

4.3.1 Digitized Quantum Annealing (dQA)

In its digitized form, Quantum Annealing (QA) can be implemented on digital quantum simulators or classically simulated through a process known as digitized Quantum Annealing (dQA) [Bar+16; BAS19; BFS19]. This procedure discretizes the continuous time evolution of QA into P time steps, each of duration $\delta t = \tau/P$, followed by a Trotter decomposition to separate the non-commuting Hamiltonian terms. The lowest-order Trotter splitting is given by

$$e^{-i\hat{H}(s_p)\delta t} \simeq e^{-i(1-s_p)\hat{H}_x\delta t} e^{-is_p\hat{H}_z\delta t} + \mathcal{O}(\delta t)^2, \quad (4.40)$$

where $p = 1, 2, \dots, P$ and $s_p = t_p/\tau = p/P$. The parameters $\beta_p = (1 - s_p)\delta t$ and $\gamma_p = s_p\delta t$ allow the previous expression to be compactly rewritten as

$$e^{-i\hat{H}(s_p)\delta t} \simeq \hat{U}_x(\beta_p) \hat{U}_z(\gamma_p) + \mathcal{O}(\delta t)^2 \quad (4.41)$$

$$\hat{U}_x(\beta_p) = e^{-i\beta_p\hat{H}_x}, \quad \hat{U}_z(\gamma_p) = e^{-i\gamma_p\hat{H}_z},$$

yielding a quantum state at the end of the digitized annealing process:

$$|\psi_p\rangle = e^{-i\beta_p\hat{H}_x} e^{-i\gamma_p\hat{H}_z} \dots e^{-i\beta_p\hat{H}_x} e^{-i\gamma_p\hat{H}_z} \dots e^{-i\beta_1\hat{H}_x} e^{-i\gamma_1\hat{H}_z} |\psi_0\rangle, \quad (4.42)$$

where $|\psi_0\rangle = |+\rangle^{\otimes N}$ is the ground state of the transverse field term \hat{H}_x , and $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.

This framework allows the digitized QA to closely approximate real-time Quantum Annealing dynamics as $P \rightarrow \infty$ and $\delta t \rightarrow 0$, while maintaining $\tau = P\delta t$ finite. However, for finite P , the trade-off between Trotter errors

and the annealing time τ becomes crucial [BAS19]. Small δt results in non-adiabatic evolution, while large δt makes the Trotter approximation too rough, introducing spurious quantum correlations.

Despite the rough approximation introduced by Trotterization, numerical studies have shown that even for values of δt on the order of $\mathcal{O}(1)$, the discrete-time evolution often yields surprisingly accurate results without needing a higher-order split-up. This makes it a practical approach for both simulation and implementation of QA dynamics on near-term quantum devices.

4.3.2 dQA with Tensor Network

A novel tensor network approach was introduced to efficiently simulate digitized Quantum Annealing (dQA) for a broad class of classical Hamiltonians or cost functions [Lam+23]. These Hamiltonians take the form:

$$H(\boldsymbol{\sigma}) = \sum_{\mu=1}^{N_{\xi}} \mathfrak{h}(\boldsymbol{\xi}^{\mu} \cdot \boldsymbol{\sigma}), \quad (4.43)$$

where the variables $\xi_i^{\mu} \in \{-1, +1\}$ (with $\mu = 1, 2, \dots, N_{\xi}$ and $i = 1, 2, \dots, N$) represent spin or qubit configurations, often referred to as “patterns”, and \mathfrak{h} is any sufficiently regular function.

Notice that the QUBO Hamiltonian can be easily rewritten in this form; specifically the quadratic term only — the so called Hopfield model [Meh21; Her+91; Hop82; AGS85] — looks like

$$\sum_{i,j} J_{ij} \sigma_i \sigma_j = \sum_{\mu=1}^{N_{\xi}} (\boldsymbol{\xi}^{\mu} \cdot \boldsymbol{\sigma})^2, \quad J_{ij} = \sum_{\mu=1}^{N_{\xi}} \xi_i^{\mu} \xi_j^{\mu}, \quad (4.44)$$

where a sufficiently large number of patterns N_{ξ} have been chose to reconstruct the random couplings J_{ij} .

The methods discussed in this work are notably versatile, applying to any Hamiltonian that can be reformulated in this general structure, reminiscent of the cost functions used in simple discrete neural networks. This framework opens up new possibilities for efficiently simulating complex quantum systems within the scope of dQA. In the following sections, we will explore this methodology in more detail.

Let us begin by noting that the initial state $|\psi_0\rangle = |+\rangle^N$ is a simple product state, i.e. a Matrix Product State (MPS) with a bond dimension of $\chi = 1$.

$$|+\rangle^N = \sum_{\sigma} \begin{array}{c} \bigcirc \\ | \\ \sigma_1 \end{array} \begin{array}{c} \bigcirc \\ | \\ \sigma_2 \end{array} \cdots \begin{array}{c} \bigcirc \\ | \\ \sigma_N \end{array} |\sigma\rangle \quad (4.45)$$

The next objective is to express the unitaries $\hat{U}_z(\gamma_p)$ and $\hat{U}_x(\beta_p)$ as Matrix Product Operators (MPOs).

The \hat{U}_x gate — Let's begin with \hat{U}_x which admits an elementary decomposition into an MPO of bond dimension $\chi = 1$, namely in the computational basis one has

$$\langle \sigma' | \hat{U}_x(\beta_p) | \sigma \rangle = \begin{array}{c} \sigma_1 \\ | \\ e^{-i\beta_p \otimes} \\ | \\ \sigma'_1 \end{array} \begin{array}{c} \sigma_2 \\ | \\ e^{-i\beta_p \otimes} \\ | \\ \sigma'_2 \end{array} \cdots \cdots \begin{array}{c} \sigma_N \\ | \\ e^{-i\beta_p \otimes} \\ | \\ \sigma'_N \end{array} \quad (4.46)$$

and as we know now very well, applying those operators to an MPS is trivial and it is not increasing the auxiliary dimension of its representation.

The \hat{U}_z gate — The real novelty comes in decomposing \hat{U}_z as an MPO. Since the target Hamiltonian encodes different pattern, we can first decompose the total unitary in N_ξ layers

$$\hat{U}_z(\gamma_p) = \prod_{\mu=1}^{N_\xi} \hat{U}_z^\mu(\gamma_p) = \prod_{\mu=1}^{N_\xi} e^{-i\gamma_p \mathfrak{h}(\xi^\mu \cdot \hat{\mathbf{Z}})}, \quad (4.47)$$

with $\hat{\mathbf{Z}} = \{\hat{Z}_1, \dots, \hat{Z}_N\}$ a vector of Pauli matrices. These operators are diagonal in the computational basis with entries

$$\langle \sigma' | \hat{U}_z^\mu(\gamma_p) | \sigma \rangle = \delta_{\sigma' \sigma} e^{-i\gamma_p \mathfrak{h}(\xi^\mu \cdot \sigma)} \quad (4.48)$$

Let us now take advantage of the particular structure of the Hamiltonian \hat{H}_z . By its very nature, this Hamiltonian depends on the spin configuration σ only through the following variables:

$$x^\mu(\sigma) = \frac{N - \xi^\mu \cdot \sigma}{2} = \sum_{j=1}^N \left(\frac{1 - \xi_j^\mu \sigma_j}{2} \right), \quad (4.49)$$

which quantify the number of bits in σ that differ from those in ξ^μ . This expression represents the well-known Hamming distance between the configurations σ and ξ^μ , where $x^\mu \in \{0, 1, \dots, N\}$.

This observation is critical, as it constitutes the only assumption underpinning the approach. Since x (for any pattern μ) is a discrete integer variable

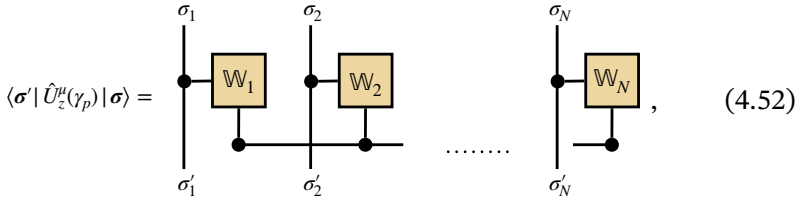
taking values in $\{0, 1, \dots, N\}$, we can represent any function by applying the Discrete Fourier Transform (DFT). In particular for what we concern we get

$$\langle \sigma' | \hat{U}_z^\mu(\gamma_p) | \sigma \rangle = \delta_{\sigma' \sigma} \frac{1}{\sqrt{N+1}} \sum_{k=0}^N \tilde{U}_{kp} e^{i \frac{2\pi}{N+1} kx^\mu(\sigma)} \quad (4.50)$$

$$\tilde{U}_{kp} = \frac{1}{\sqrt{N+1}} \sum_{x=0}^N e^{-i \frac{2\pi}{N+1} kx} e^{-i \gamma_p \mathfrak{h}(N-2x)}, \quad (4.51)$$

where Fourier components \tilde{U}_{kp} implicitly depend on the angle γ_p , and can be treated as a matrix with dimensions $(N+1) \times P$.

Through the use of Fourier decomposition, an “efficient” representation of $\hat{U}_z^\mu(\gamma_p)$ as a MPO becomes possible. In fact, the wave numbers $k = 0, 1, \dots, N$ naturally emerge as “auxiliary indices” within the MPO formalism. This enables the transformation of the operator into the form



$$\langle \sigma' | \hat{U}_z^\mu(\gamma_p) | \sigma \rangle = \begin{array}{c} \sigma_1 \\ | \\ \bullet \\ | \\ \sigma'_1 \end{array} \begin{array}{c} \boxed{W_1} \\ | \\ \bullet \end{array} \begin{array}{c} \sigma_2 \\ | \\ \bullet \\ | \\ \sigma'_2 \end{array} \begin{array}{c} \boxed{W_2} \\ | \\ \bullet \end{array} \dots \begin{array}{c} \sigma_N \\ | \\ \bullet \\ | \\ \sigma'_N \end{array} \begin{array}{c} \boxed{W_N} \\ | \\ \bullet \end{array}, \quad (4.52)$$

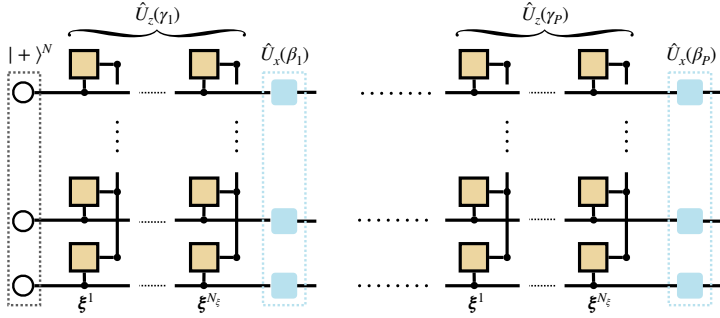
where the tensors \mathbb{W}_j , have the following entries

$$\sigma \begin{array}{c} \boxed{W_j} \\ | \\ k \end{array} = \left(\frac{\tilde{U}_{kp}}{\sqrt{N+1}} \right)^{\frac{1}{N}} e^{i \frac{2\pi}{N+1} (1 - \xi_j^\mu) \sigma}, \quad (4.53)$$

The tensors \mathbb{W}_j are notably diagonal in both the auxiliary indices, meaning that they essentially depend on only a single auxiliary index k and one physical spin σ_j , since they only depends on the Pauli matrix \hat{Z}_j . As usual, at the left (right) boundary, the tensor \mathbb{W}_1 (\mathbb{W}_N) reduces to a row (column) vector in the auxiliary space. In addition, it's important to recognize that they also implicitly rely on the angle γ_p and the pattern index μ , even though these dependencies aren't explicitly shown in the notation. This formulation captures the critical factors needed for efficient tensor manipulations while abstracting the underlying dependencies.

Given these findings, the complete digitized quantum annealing (dQA) time evolution can be faithfully depicted as a 2D Tensor Network, as illustrated

in the following figure:



This network effectively models the application of successive Matrix Product Operators (MPOs) to the initial Matrix Product State (MPS) $|\psi_0\rangle$, which initially possesses a trivial structure.

This observation forms the foundation of an MPS-based algorithm for simulating dQA classically, outlined in the following pseudo-code:

Input: the dQA parameters τ , P , $\delta t = \tau/P$, the MPS maximum bond dimension χ

- 1: Compute the Fourier matrix $\tilde{U}_{k,p}$ (Eq. (4.51)) of dimension $(N+1) \times P$
- 2: Define the initial state $|\psi_0\rangle$ as an MPS of bond dimension $\chi = 1$ and set $|\psi\rangle = |\psi_0\rangle$
- 3: **for** ($p = 1, p \leq P, p++$) **do**
- 4: **for** ($\mu = 1, \mu \leq N_\xi, \mu++$) **do**
- 5: apply $\hat{U}_z^\mu(\gamma_p)$ to the current MPS $|\psi\rangle$ and if needed compress the bond dimension to χ
- 6: **end for**
- 7: apply $\hat{U}_x(\beta_p)$ to the current MPS $|\psi\rangle$
- 8: **end for**

Output: the final optimized MPS $|\psi\rangle$

Algorithm 1. dQA with MPS.

A significant challenge in this approach is the rapid growth of the bond dimension of the MPS as the number of Trotter steps P increases. To mitigate this, a suitable **compression strategy** needs to be implemented (see Example 4.3). This compression is crucial for maintaining computational feasibility, as a critical input for the algorithm is the predefined maximum bond dimension.

Let' finally mention that the MPO structure based on the DFT strategy offers an additional advantage: it enables the Hamiltonian itself to be ex-

pressed as an MPO with a bond dimension of $\chi = N + 1$. This is a noteworthy feature, as it facilitates the *precise computation* of the classical cost function — specifically, the expectation value of \hat{H}_z — for *any* MPS. Consequently, while running the algorithm, one can accurately monitor the evolution of the cost function over time.

Example 4.3: Variational compression of a MPS

As already explained in Chapter 1 and Chapter 2, the SVD is a fundamental tool for performing a compression sweep over an MPS. Here we outline another procedure which is more accurate and is based on a variational optimisation. In practice, the compression technique restricts the evolving system to the manifold of MPS with a predefined maximum bond dimension χ . This can be accomplished using an iterative approach, as outlined in works like Refs. [Sch11; Sab+09], which efficiently solves the optimization problem step-by-step at each lattice site.

In practice, lets suppose we want to compress an MPS state with bond dimension D , such that $|\psi\rangle = \sum_{\sigma_1, \dots, \sigma_N} A^{\sigma_1} \dots A^{\sigma_N} |\sigma_1, \dots, \sigma_N\rangle$. We basically want to find a new set of optimal tensors $\{\tilde{A}^{\sigma_j}\}$ with bond dimension $\chi \ll D$, such that the compressed state will be $|\tilde{\psi}\rangle = \sum_{\sigma_1, \dots, \sigma_N} \tilde{A}^{\sigma_1} \dots \tilde{A}^{\sigma_N} |\sigma_1, \dots, \sigma_N\rangle$. This is achieved by solving the following local variational equations

$$\frac{\partial}{\partial \tilde{A}^{\sigma_j}} \|\tilde{\psi} - \psi\|^2 = \frac{\partial}{\partial \tilde{A}^{\sigma_j}} (\langle \tilde{\psi} | \tilde{\psi} \rangle - \langle \tilde{\psi} | \psi \rangle) = 0. \quad (4.54)$$

In practice, this process involves carrying out a series of localized optimizations by sweeping across all the sites in the system multiple times, denoted by the number of sweeps N_{sweeps} . Each sweep progressively refines the approximation, ensuring convergence to a more accurate solution.

To optimize the tensor at position j , the best method involves having the MPS for $|\tilde{\psi}\rangle$ in its mixed canonical form relative to that specific site. This ensures that the optimal update is simply:

$$\alpha - \tilde{A}_j - \beta = \begin{array}{c} \text{--- } A_1 \text{ --- } \dots \text{ --- } A_{j-1} \text{ --- } A_j \text{ --- } A_{j+1} \text{ --- } \dots \text{ --- } A_N \text{ --- } |\psi\rangle \\ \downarrow \quad \quad \quad \downarrow \quad \quad \quad \downarrow \quad \quad \quad \downarrow \\ \tilde{A}_1^* \text{ --- } \dots \text{ --- } \tilde{A}_{j-1}^* \text{ --- } \tilde{A}_j^* \text{ --- } \tilde{A}_{j+1}^* \text{ --- } \dots \text{ --- } \tilde{A}_N^* \text{ --- } \langle \tilde{\psi}| \\ \uparrow \quad \quad \quad \uparrow \quad \quad \quad \uparrow \quad \quad \quad \uparrow \\ \alpha \quad \quad \quad \beta \end{array} \quad (4.55)$$

Here, the contraction on the right-hand side has a computational cost which scales as $ND\chi^2 + ND^2\chi \sim ND^2\chi$, when $D \gg \chi$. Notably, unlike the compressed state $|\tilde{\psi}\rangle$, the known state $|\psi\rangle$ doesn't necessarily need to be in mixed canonical form. After computing \tilde{A}^{σ_j} , the next step depends on the direction of the sweep. If sweeping from right-to-left, one performs a QR decomposition, while for a left-to-right sweep, an LQ decomposition is used to move efficiently to the next lattice site at position $j + 1$ or $j - 1$ respectively. This ensures that the MPS remains in a canonical form appropriate for the chosen direction of the sweep.

As with most iterative procedures, this method benefits from a good initial representation of $|\tilde{\psi}\rangle$, which, in this case, is provided through the canonical SVD compression.

Optimised compression for dQA — Whilst this strategy is generic, when applied within the dQA algorithm the state $|\psi\rangle$ is obtained by applying a unitary evolution operator to the MPS from the previous step, expressed as $|\psi\rangle = \hat{U}_z^\mu(\gamma_p)|\phi\rangle$. We learned that, when an MPO is applied to an MPS, the bond dimensions increase as their product, meaning $D = (N + 1)\chi$ at regime. As a result, applying blindly the previous scheme, the cost of computing the local optimised tensor becomes $\sim N^3\chi^3$, where χ is the maximum bond dimension fixed in the dQA algorithm. However, the compression efficiency can be greatly improved by exploiting the particular diagonal structure of the MPO. By explicitly writing the action of the MPOs \hat{W}_j in Eq. (4.53) we easily get

$$\alpha \text{---} \tilde{A}_j \text{---} \beta \text{---} \sigma = \text{[Diagrammatic representation of the tensor network contraction]} \quad (4.56)$$

reducing the computational cost to $\sim N^2\chi^3$, since we have basically to compress $N + 1$ times a tensor network which has a computational cost $\sim N\chi^3$.

4.3.3 Quantum approximate optimization algorithm

The Quantum Approximate Optimization Algorithm (QAOA) is a variational quantum algorithm that builds upon the digitized quantum annealing (dQA) process by treating the parameters $\boldsymbol{\beta} = \beta_1, \dots, \beta_P$ and $\boldsymbol{\gamma} = \gamma_1, \dots, \gamma_P$ as variational variables. This approach transforms dQA into a Variational Quantum Algorithm (VQA), with QAOA playing a crucial role in modern quantum computing research [FGG14; Cer+21].

In the context of QAOA, as in the previous sections, the goal is to minimize a problem-specific cost function, typically encoded in a Hamiltonian \hat{H}_P , by evolving an initial state $|+\rangle^{\otimes N}$ through a sequence of alternating unitary operators. These unitaries correspond to the time-evolved versions of the mixing Hamiltonian \hat{H}_x and the problem Hamiltonian \hat{H}_z , with the evolution parameters being the variational angles $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$. The alternating application of these Hamiltonians is mathematically described as:

$$|\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle = \prod_{i=1}^P e^{-i\beta_i \hat{H}_x} e^{-i\gamma_i \hat{H}_P} |+\rangle^{\otimes N}. \quad (4.57)$$

The variational nature of QAOA comes from tuning the parameters $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ so that the expectation value of the problem Hamiltonian \hat{H}_P in the evolved state $|\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle$ is minimized

$$\langle \psi(\boldsymbol{\beta}, \boldsymbol{\gamma}) | \hat{H}_P | \psi(\boldsymbol{\beta}, \boldsymbol{\gamma}) \rangle. \quad (4.58)$$

This expectation value serves as a cost function to be minimized using classical optimization techniques. The success of the algorithm depends on finding optimal parameters and determining their quantity, balancing exploration and exploitation. As the number of layers P increases, QAOA approaches the performance of adiabatic quantum computing, but even with small P , it provides useful approximations, making it suitable for near-term quantum devices.

In QAOA the entanglement between qubits grows with circuit depth and problem complexity, often requiring large bond dimensions in MPS to accurately capture this entanglement [Dup+22]. This challenge highlights the classical hardness of simulating QAOA for deeper circuits, as the entanglement barrier increases. As a result, simulating QAOA on classical devices often involves a trade-off between accuracy and computational cost.

Nevertheless, TN methods serve as an excellent classical benchmark to evaluate the performance of quantum algorithms such as QAOA. By adjusting the bond dimension, TN simulations can provide upper bounds on the entanglement that classical methods can handle. Comparing the performance of

quantum hardware with TN benchmarks allows us to quantify the quantum advantage. Thus, TN methods offer a practical way to calibrate the difficulty of QAOA problems and assess quantum hardware performance in realistic scenarios.

References of Chapter 4

- [AGS85] D. J. Amit, H. Gutfreund, and H. Sompolinsky. “Spin-glass models of neural networks.” In: *Phys. Rev. A* 32 (2 1985), pp. 1007–1018. DOI: [10.1103/PhysRevA.32.1007](https://doi.org/10.1103/PhysRevA.32.1007).
- [Bar+16] R. Barends et al. “Digitized adiabatic quantum computing with a superconducting circuit.” In: *Nature* 534 (2016), p. 222.
- [BAS19] G. Bigan Mbeng, L. Arceci, and G. E. Santoro. “Optimal working point in digitized quantum annealing.” In: *Phys. Rev. B* 100 (22 2019), p. 224201. DOI: [10.1103/PhysRevB.100.224201](https://doi.org/10.1103/PhysRevB.100.224201).
- [BFS19] G. Bigan Mbeng, R. Fazio, and G. Santoro. “Quantum Annealing: a journey through Digitalization, Control, and hybrid Quantum Variational schemes.” In: *arXiv e-prints* (2019). arXiv: [1906.08948](https://arxiv.org/abs/1906.08948) [quant-ph].
- [Cer+21] M. Cerezo et al. “Variational quantum algorithms.” In: *Nature Reviews Physics* 3 (2021).
- [Dup+22] M. Dupont et al. “Entanglement perspective on the quantum approximate optimization algorithm.” In: *Phys. Rev. A* 106 (2 2022), p. 022423. DOI: [10.1103/PhysRevA.106.022423](https://doi.org/10.1103/PhysRevA.106.022423).
- [FGG14] E. Farhi, J. Goldstone, and S. Gutmann. “A Quantum Approximate Optimization Algorithm.” In: *arXiv e-prints* (2014). arXiv: [1411.4028](https://arxiv.org/abs/1411.4028) [quant-ph].
- [Hae+11] J. Haegeman et al. “Time-Dependent Variational Principle for Quantum Lattices.” In: *Physical Review Letters* 107.7 (2011). ISSN: 1079-7114. DOI: [10.1103/physrevlett.107.070601](https://doi.org/10.1103/physrevlett.107.070601).
- [Hae+16] J. Haegeman et al. “Unifying time evolution and optimization with matrix product states.” In: *Physical Review B* 94.16 (2016). ISSN: 2469-9969. DOI: [10.1103/physrevb.94.165116](https://doi.org/10.1103/physrevb.94.165116).
- [Her+91] J. A. Hertz et al. *Introduction To The Theory Of Neural Computation, Volume I*. Addison-Wesley Computation and Neural Systems Series. Avalon Publishing, 1991. ISBN: 9780201503951. URL: <https://books.google.it/books?id=Z4BqAAAAMAAJ>.
- [Hol+10] H. Holden et al. *Splitting Methods for Partial Differential Equations with Rough Solutions: Analysis and MATLAB programs*. 2010. ISBN: 978-3-03719-078-4. DOI: [10.4171/078](https://doi.org/10.4171/078).

- [Hop82] J. J. Hopfield. “Neural networks and physical systems with emergent collective computational abilities.” In: *Proceedings of the National Academy of Sciences* 79.8 (1982), pp. 2554–2558. DOI: [10.1073/pnas.79.8.2554](https://doi.org/10.1073/pnas.79.8.2554).
- [Lam+23] G. Lami et al. “Quantum annealing for neural network optimization problems: A new approach via tensor network simulations.” In: *SciPost Phys.* 14 (2023), p. 117. DOI: [10.21468/SciPostPhys.14.5.117](https://doi.org/10.21468/SciPostPhys.14.5.117).
- [Lan50] C. Lanczos. “An Iteration Method for the Solution of the Eigenvalue I Problem. of Linear Differential and Integral Operators.” In: *Journal of Research of the National Bureau of Standards* 45 (4 1950), pp. 255–282. DOI: [10.6028/jres.045.026](https://doi.org/10.6028/jres.045.026).
- [Meh21] B. Mehlig. *Machine Learning with Neural Networks: An Introduction for Scientists and Engineers*. Cambridge University Press, 2021. ISBN: 9781108849562. URL: <https://books.google.it/books?id=DQ1IEAAAQBAJ>.
- [ML03] C. Moler and C. Loan. “Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later.” In: *Society for Industrial and Applied Mathematics* 45 (2003), pp. 3–49. DOI: [10.1137/S00361445024180](https://doi.org/10.1137/S00361445024180).
- [Pai71] C. C. Paige. “The computation of eigenvalues and eigenvectors of very large sparse matrices.” In: 1971. URL: <https://api.semanticscholar.org/CorpusID:19735646>.
- [Sab+09] H. Saberi et al. “Constrained optimization of sequentially generated entangled multiqubit states.” In: *Physical Review A* 80.2 (2009). DOI: [10.1103/physreva.80.022334](https://doi.org/10.1103/physreva.80.022334).
- [Sch11] U. Schollwöck. “The density-matrix renormalization group in the age of matrix product states.” In: *Annals of Physics* 326.1 (2011), pp. 96–192. DOI: [10.1016/j.aop.2010.09.012](https://doi.org/10.1016/j.aop.2010.09.012).
- [Sim84] H. D. Simon. “Analysis of the symmetric Lanczos algorithm with reorthogonalization methods.” In: *Linear Algebra and its Applications* 61 (1984), pp. 101–131. ISSN: 0024-3795. DOI: [10.1016/0024-3795\(84\)90025-9](https://doi.org/10.1016/0024-3795(84)90025-9).

Chapter 5

Open tensor network: from thermal states to open system dynamics

No man ever steps in the same river twice, for it is not the same river and he is not the same man.

Heraclitus

Until now, aside from a brief introduction to density matrices, we have primarily discussed scenarios in which the state of a quantum system is fully described by a single unit vector $|\psi\rangle$ in its Hilbert space. However, in many real-world situations, the exact state vector $|\psi\rangle$ is not known. Consider, for instance, a beam of thermal neutrons emitted from a radioactive source. In this scenario, we do not have specific information about the energy of each neutron. Instead, we only know the overall distribution of their energies, which might follow a canonical distribution at a given temperature. This exemplifies a situation where our knowledge of the system is incomplete.

In this chapter, we will consider scenarios where a single unit vector in Hilbert space is insufficient to fully describe the system under investigation. This situation often arises when the system is in thermal equilibrium at a non-zero temperature or interacts with an environment about which we lack detailed knowledge. Additionally, continuous monitoring of the system by external probes can influence its dynamics in a non-unitary manner, necessitating a more comprehensive description. There are numerous situations

where a pure state description falls short; here, we will focus on a select few that are particularly pertinent to tensor network methods and quantum computing. These include thermal equilibrium states, open quantum systems dynamics, and measurement-induced dynamics, highlighting their relevance and application in contemporary quantum research.

5.1 Open system recap

Continuing from the introduction of this chapter, our lack of information about the state of a quantum system is mathematically represented by the fact that we cannot assign a unique state in the Hilbert space to our system. Instead, we must describe it as an ensemble of states, each associated with a certain probability:

$$\{|\psi_1\rangle, p_1\}, \{|\psi_2\rangle, p_2\}, \dots, \{|\psi_D\rangle, p_D\} \quad (5.1)$$

such that $p_k \geq 0 \forall k \in \{1, \dots, D\}$, and $\sum_{k=1}^D p_k = 1$ to ensure a proper probability distribution. It is important to note that the states $|\psi_k\rangle$ are not necessarily orthogonal to each other.

It is fundamentally impossible to describe a statistical mixture using a single “typical” vector; instead, it can be described using either a distribution of “typical” vectors or just a “typical” operator, commonly known as the **density operator** or **density matrix**.

Definition 5.1: Statistical Mixture

We describe a statistical mixture, also known as a mixed state, when instead of having a single pure state representing the system, we have an ensemble of states with corresponding probabilities $\{|\psi_k\rangle, p_k\}$, such that $\sum_k p_k = 1$. Practically, this ensemble is represented in quantum mechanics through the **density operator**:

$$\hat{\rho} = \sum_k p_k |\psi_k\rangle\langle\psi_k|, \quad (5.2)$$

where $\text{Tr}(\hat{\rho}) = 1$.

According to the rules of quantum mechanics, the density operator provides the probabilities of the associated statistical mixture. Specifically, for a pure state $|\psi_k\rangle$ with probability p_k , this probability can be computed using the density operator as $p_k = \text{Tr}(\hat{\rho} |\psi_k\rangle\langle\psi_k|)$.

In addition, consider any observable \hat{O} , which can be decomposed in terms of its eigenvectors $|o_j\rangle$ and eigenvalues o_j :

$$\hat{O} = \sum_j o_j |o_j\rangle\langle o_j|. \quad (5.3)$$

The probability of obtaining the outcome o_j when measuring the observable \hat{O} on the state described by the density operator $\hat{\rho}$ is given by:

$$p(o_j) = \text{Tr}(\hat{\rho} |o_j\rangle\langle o_j|) = \sum_k p_k |\langle o_j|\psi_k\rangle|^2. \quad (5.4)$$

This formula encapsulates how the density operator effectively captures the statistical distribution of measurement outcomes over the ensemble of states. The expectation value of the observable \hat{O} , which represents the average measurement outcome, is then given by:

$$\langle \hat{O} \rangle = \sum_j o_j p(o_j) = \text{Tr}(\hat{O}\hat{\rho}). \quad (5.5)$$

To summarize, the density operator $\hat{\rho}$ not only encodes the probabilities of different pure states in the mixture but also allows us to calculate the probabilities of measurement outcomes and the expectation values of observables in a straightforward manner. This is crucial for understanding the behavior of quantum systems where the state is not precisely known and must be described as a mixture of several possible states [Ben+19].

Mixed system unitary evolution — When a mixed system evolve in time according to the quantum mechanics postulates (see Chapter 2), assuming the evolution is induced then by a unitary transformation, aka the Schrodinger equation, the time dependent statistical mixture reads

$$\hat{\rho}(t) = \sum_k p_k |\psi_k(t)\rangle\langle\psi_k(t)|, \quad (5.6)$$

where $\{|\psi_k(t)\rangle\}$ are the pure states evolving according to the Schrodinger equation and $\{p_k\}$ are the corresponding time-independent probabilities.

To derive the *von Neumann equation*, which govern the time evolution of the density operator, we start from taking the time derivative of $\hat{\rho}(t)$ as written above:

$$\partial_t \hat{\rho}(t) = \sum_k p_k [(\partial_t |\psi_k(t)\rangle)\langle\psi_k(t)| + |\psi_k(t)\rangle(\partial_t \langle\psi_k(t)|)], \quad (5.7)$$

then using the Schrödinger equation for the pure states we easily obtain:

$$\partial_t \hat{\rho}(t) = -i [\hat{H}, \hat{\rho}(t)], \quad (5.8)$$

where in the entire formulation we fixed $\hbar = 1$. This equation shows that the time evolution of the density matrix $\hat{\rho}(t)$ is governed by the commutator of the Hamiltonian \hat{H} with $\hat{\rho}(t)$. This formalism is crucial for describing the dynamics of mixed states in quantum mechanics, allowing for a complete representation of the statistical properties of quantum systems.

Thermal (or Canonical) ensemble — What does a typical wavefunction of a quantum system look like at a finite temperature? In statistical mechanics, a fundamental concept is indeed the density matrix which describes the state of a system in thermal equilibrium. For a system with Hamiltonian \hat{H} at an inverse temperature $\beta = 1/kT$, the density matrix is given by:

$$\hat{\rho} = \frac{1}{Z} \exp(-\beta \hat{H}), \quad (5.9)$$

where $Z = \text{Tr}\{\exp(-\beta \hat{H})\}$ is the *partition function*. This density matrix can be understood in several ways. For example, it can arise from averaging over an ensemble of pure states, from taking the long-time average of one system, or from the quantum mechanical entanglement with a heat bath, which produces mixed states. Regardless of the interpretation, the predictions of statistical mechanics depend only on $\hat{\rho}$.

However, in general, even though, as we have seen in Chapter 2, local Hamiltonians often admit simple MPO representations, exponentiating an MPO is typically a challenging task. This difficulty is analogous to exponentiating a matrix, which, in general, if it lacks special properties, is nontrivial.

A very trivial example of thermal state which is Hamiltonian independent, is the **infinite temperature state**, i.e. $\beta = 0$, which in fact a part from a suitable normalisation, reduces to the **identity operator**

$$\hat{\rho}_{\beta=0} \propto \hat{I} = \sum_{x_1, \dots, x_n} \sum_{y_1, \dots, y_n} \begin{array}{c} y_1 \\ | \\ x_1 \end{array} \begin{array}{c} y_2 \\ | \\ x_2 \end{array} \cdots \begin{array}{c} y_n \\ | \\ x_n \end{array} |x_1, \dots, x_n\rangle \langle y_1, \dots, y_n|$$

where the norm of the identity operator, for n qubits state, is $Z = \text{Tr} \hat{I} = 2^n$.

Example 5.1: Exact MPO for classical thermal states

Let us consider the **classical ferromagnetic Ising model in one dimension**, in the presence of an external magnetic field h . Here, we assume homogeneous couplings between neighboring qubits. The Hamiltonian reads

$$\hat{H} = -J \sum_j \hat{Z}_j \hat{Z}_{j+1} - h \sum_j \hat{Z}_j \equiv \hat{H}_J + \hat{H}_h, \quad (5.10)$$

where we arbitrarily choose to work with Pauli operators in the \hat{z} direction, and we split the Hamiltonian into two commuting operators. Here we consider a system with n Spin-1/2 (or qubits) with open boundary conditions. We have already seen in Chapter 2 that this Hamiltonian admits an exact MPO representation. However, here we want to compute its **thermal density operator**

$$\exp(-\beta\hat{H}) = \exp(-\beta\hat{H}_J) \exp(-\beta\hat{H}_h), \quad (5.11)$$

where we used the fact that $[\hat{H}_J, \hat{H}_h] = 0$. The magnetic contribution to the density operator is trivial due to the fact that each term acts locally; in practice, we have a product operator, namely an MPO with bond dimension $\chi_h = 1$.

$$\exp(-\beta\hat{H}_h) = \prod_j e^{\beta h \hat{Z}_j}$$

More interesting is the contribution from the neighboring coupling term, which reads (using the properties of Pauli matrices)

$$\exp(-\beta\hat{H}_J) = \prod_j [\cosh(\beta J) + \sinh(\beta J) \hat{Z}_j \hat{Z}_{j+1}]. \quad (5.12)$$

In fact, we can recast this as an MPO by explicitly rewriting each term in the product as a scalar product between two local neighboring vectors, i.e.,

$$\cosh(\beta J) + \sinh(\beta J) \hat{Z}_j \hat{Z}_{j+1} = \begin{pmatrix} \hat{I}_j & \hat{Z}_j \end{pmatrix} \cdot \begin{pmatrix} \cosh(\beta J) \hat{I}_j \\ \sinh(\beta J) \hat{Z}_{j+1} \end{pmatrix}, \quad (5.13)$$

where we formally reintroduce the local identity operator. Using this last result into the Eq. (5.12), and collecting the terms acting on the

same qubit, we can rewrite the exponential factor as an MPO with bond dimension $\chi_J = 2$

$$\exp(-\beta\hat{H}_J) = \hat{w}_1 - \hat{w}_2 - \dots - \hat{w}_n$$

with uniform bulk tensors

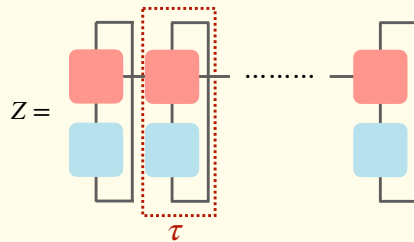
$$\hat{w}_j = \begin{pmatrix} \cosh(\beta J) \hat{I} & \cosh(\beta J) \hat{Z} \\ \sinh(\beta J) \hat{Z} & \sinh(\beta J) \hat{I} \end{pmatrix} \quad (5.14)$$

and boundary tensors

$$\hat{w}_1 = (\hat{I} \quad \hat{Z}), \quad \hat{w}_n = \begin{pmatrix} \cosh(\beta J) \hat{I} \\ \sinh(\beta J) \hat{Z} \end{pmatrix}. \quad (5.15)$$

At this point, it is straightforward to combine the two MPO representations of the exponential factors into a single MPO representing $\exp(-\beta\hat{H})$ with bond dimension $\chi = \chi_J\chi_h = 2$, by following the rules of the tensor network contractions.

Let us finally conclude this example by considering the fact that, with this representation in hand, we can easily compute the **Partition Function** $Z = \text{Tr}[\exp(-\beta\hat{H})]$ of this model. This computation corresponds to the normalization of the density operator. The connection with thermodynamics is established by identifying the **Free Energy** with the logarithm of the partition function, namely $\mathcal{F} = -\frac{\log Z}{\beta}$. Specifically, taking the trace of the full MPO representation of the thermal density operator involves wrapping the physical wires locally for each qubit, thus obtaining



in terms of the 2×2 classical transfer matrix (for the bulk)

$$\tau = 2 \begin{pmatrix} \cosh(\beta J) \cosh(\beta h) & \cosh(\beta J) \sinh(\beta h) \\ \sinh(\beta J) \sinh(\beta h) & \sinh(\beta J) \cosh(\beta h) \end{pmatrix}. \quad (5.16)$$

For instance, in the simplest scenario where the magnetic field vanishes ($h = 0$), τ simplifies to a diagonal matrix. By considering the boundary vectors, we obtain the well-known result from classical statistical mechanics:

$$Z_{h=0} = 2^n \cosh(\beta J)^{n-1}. \tag{5.17}$$

The general case with a finite magnetic field can be solved by diagonalizing the classical transfer matrix, which is left as an exercise for the reader.

Finite temperature states — Finite temperature states for systems on a lattice, such as registers of qubits, can be efficiently prepared by extending the canonical Time-Evolving Block Decimation (TEBD) scheme to handle mixed states and performing imaginary time evolution [ZV04]. This extension allows for the simulation of thermal states by evolving the system in imaginary time, effectively capturing the thermal properties at any desired temperature.

In fact, we can map operators, such as the density matrix $\hat{\rho}$, into “superkets” $|\rho\rangle\rangle$, and superoperators, acting on linear operators, into linear mappings acting on the superkets. For instance, the commutator in the Liouville equation of motion transforms as $[\hat{H}, \cdot] \rightarrow \hat{H} \otimes \hat{I} - \hat{I} \otimes \hat{H}^T$. Notice that, all those transformation are simply based on the **vectorization of matrices**.

Example 5.2: Vectorization of matrices

Vectorization of a matrix is a specific reshaping operation where an $m \times n$ matrix is transformed into a vector of length mn . In tensor network terminology, this corresponds to **bending the physical wires** of an MPO to get an MPS, resulting in:

$$|\rho\rangle\rangle = \sum_{x_1, \dots, x_n} \sum_{y_1, \dots, y_n} \left[\begin{array}{c} \rho_1 \\ x_1 \quad y_1 \end{array} \right] \left[\begin{array}{c} \rho_2 \\ x_2 \quad y_2 \end{array} \right] \dots \left[\begin{array}{c} \rho_n \\ x_n \quad y_n \end{array} \right] |x_1 y_1, \dots, x_n y_n\rangle$$

Similarly, the infinite temperature state, represented by the identity matrix, can be decomposed into a tensor product of non-normalized Bell states as follow:

$$|I\rangle\rangle = \sum_{x_1, \dots, x_n} \sum_{y_1, \dots, y_n} \left[\begin{array}{c} \\ x_1 \quad y_1 \end{array} \right] \left[\begin{array}{c} \\ x_2 \quad y_2 \end{array} \right] \dots \left[\begin{array}{c} \\ x_n \quad y_n \end{array} \right] |x_1 y_1, \dots, x_n y_n\rangle$$

Vectorization basically is going to construct a Hilbert space specifically for density matrices by defining an appropriate scalar product. This procedure effectively transforms matrices into vectors in this Hilbert space, where the density matrix $\hat{\rho}$ is represented as $|\rho\rangle\rangle$. This Hilbert space is known as the Fock-Liouville Space (FLS).

In this framework, the scalar product between two matrices, $\hat{\phi}$ and $\hat{\rho}$, is given by: $\langle\langle\phi|\rho\rangle\rangle \equiv \text{Tr}[\hat{\phi}^\dagger \hat{\rho}]$. With this setup, the Liouville superoperator becomes an operator acting within the FLS. The primary advantage of the Fock-Liouville Space is that it enables a convenient matrix representation for the evolution operator, facilitating the analysis and simulation of dynamic processes in quantum systems.

Before we delve into the procedure for using imaginary-time TEBD on superkets to generate finite temperature states, it is important to mention a key aspect of MPS-based algorithms. These algorithms typically preserve the standard ket-normalization, $\langle\langle\rho|\rho\rangle\rangle$, which results in a normalization condition at the operator level of $\text{Tr}(\hat{\rho}^2)$. However, this may not be the desired norm to maintain throughout the simulation.

Now let's consider a nearest neighbor Hamiltonian \hat{H} and an inverse temperature β . The thermal state of interest at this temperature can be constructed by using the superket formalism and simulating the imaginary time evolution from the completely mixed state,

$$e^{-\beta\hat{H}} = e^{-\beta\hat{H}/2} \hat{I} e^{-\beta\hat{H}/2} \rightarrow |e^{-\beta\hat{H}}\rangle\rangle = e^{-\beta\hat{\mathbb{H}}} |\hat{I}\rangle\rangle \quad (5.18)$$

with $\hat{\mathbb{H}} = (\hat{H} \otimes \hat{I} + \hat{I} \otimes \hat{H}^t)/2$.

To efficiently compute the thermal state $|\rho\rangle\rangle$, we can utilize the Trotter decomposition to approximate the exponential operator $e^{-\beta\hat{\mathbb{H}}}$ as a sequence of operations involving only two adjacent sites. This method allows us to update the MPS using the Time-Evolving Block Decimation (TEBD) algorithm (see section 2.5.2).

The Trotter decomposition breaks down the exponential of a sum of operators into a product of exponentials of the individual operators. Specifically, for a Hamiltonian $\hat{\mathbb{H}}$ that can be written as a sum of terms $\hat{\mathbb{H}}_1$ and $\hat{\mathbb{H}}_2$ (where each term acts on different pairs of adjacent sites), we can approximate the exponential as:

$$e^{-\beta\hat{\mathbb{H}}} \approx \left(e^{-\Delta\beta\hat{\mathbb{H}}_1} e^{-\Delta\beta\hat{\mathbb{H}}_2} \right)^{\beta/\Delta\beta}, \quad (5.19)$$

where $\Delta\beta$ is a small imaginary time step. By iteratively applying this process, the thermal state $|\rho\rangle\rangle$ is constructed through a series of local updates involving only pairs of adjacent sites.

An important advantage of this approach is that a single run of the simulation generates the thermal state for any intermediate value of $\beta' \in [0, \beta]$. This is because the TEBD algorithm incrementally builds up the thermal state from the infinite temperature state (completely mixed state) to the desired temperature. To implement the Trotter decomposition in practice, follow these steps (for a qubits system):

1. *Initialize the System* — Start with the infinite temperature state, which is represented as an MPS with bond dimension $\chi = 1$, each tensor describing the normalized Bell state $(|00\rangle + |11\rangle)/\sqrt{2}$, so that to have a proper normalized superket MPS.
2. *Apply Local Updates* — Sequentially apply the strings of local two-site operators $e^{-\Delta\beta\hat{H}_1}$ and $e^{-\Delta\beta\hat{H}_2}$ to the MPS. This can be done using the TEBD algorithm, which efficiently updates the MPS by considering one pair of adjacent sites at a time.
3. *Repeat* — Iterate the above step $\beta/\Delta\beta$ times.

Let us finally remember that, due to the anomalous normalisation of a thermal state in the superket MPS representation, namely $\langle\langle \rho | \rho \rangle\rangle = \text{Tr}\{\hat{\rho}^2\} = 1$, we do have to compute the partition function

$$Z = \langle\langle I | \rho \rangle\rangle = \text{Diagram} \quad (5.20)$$

in order to properly measure observables

$$\langle \hat{O} \rangle = \frac{1}{Z} \text{Tr}\{\hat{O}\hat{\rho}\} = \frac{\langle\langle I | \hat{O} \otimes \hat{I} | \rho \rangle\rangle}{Z}, \quad (5.21)$$

where in the MPO formalism, the expectation value reads

$$\langle\langle I | \hat{O} \otimes \hat{I} | \rho \rangle\rangle = \text{Diagram} \quad (5.22)$$

5.1.1 Tensor Network and Density Operators

In the language of Tensor Networks, any operator can, in principle, be represented as a Matrix Product Operator (MPO), provided we have sufficient

computational resources. As discussed in Chapter 2, this applies to any operator, including the density matrix. Therefore, the density matrix can be expressed in the computational basis as an MPO:

$$\hat{\rho} = \sum_{x_1, \dots, x_n} \sum_{y_1, \dots, y_n} \begin{array}{c} y_1 \\ | \rho_1 \\ x_1 \end{array} - \begin{array}{c} y_2 \\ | \rho_2 \\ x_2 \end{array} - \dots - \begin{array}{c} y_n \\ | \rho_n \\ x_n \end{array} | x_1, \dots, x_n \rangle \langle y_1, \dots, y_n | \quad (5.23)$$

which in particular, by using the normalized Pauli tensors (see Definition 5.2), can be even rewritten in the normalized Pauli basis as

$$\hat{\rho} = \sum_{\mu_1, \dots, \mu_n} \begin{array}{c} e_1 \\ | \mu_1 \end{array} - \begin{array}{c} e_2 \\ | \mu_2 \end{array} - \dots - \begin{array}{c} e_n \\ | \mu_n \end{array} \hat{\zeta}^{\mu_1} \otimes \hat{\zeta}^{\mu_2} \otimes \dots \otimes \hat{\zeta}^{\mu_n} \quad (5.24)$$

where using the properties of the Pauli tensor we can pass from one representation to the other via the following identity:

$$\begin{array}{c} \alpha \\ | e \\ \mu \end{array} - \beta = \mu - \begin{array}{c} \alpha \\ | \zeta^* \\ \mu \end{array} - \begin{array}{c} \alpha \\ | \rho \\ \mu \end{array} - \beta$$

Notice that, this change of basis induced a trivial tensor transformation for the infinite temperature state, where indeed the local coefficient in the Pauli basis reduce to $e_\mu(\beta = 0) = \text{Tr} \zeta^\mu = \sqrt{2} \delta_{\mu 0}$.

In addition, both representations have their advantages and disadvantages. For example, the density matrix is a Hermitian operator, which implies that we must enforce the condition $\hat{\rho}^\dagger = \hat{\rho}$; this condition results in nontrivial constraints in the computational basis representation of the MPO. However, when using the Pauli basis representation, due to the Hermiticity of the basis itself, the condition simplifies to requiring only real numbers for the entries of the tensors representing the operator.

On the contrary, when the density matrix represents a global pure state, it reduces to a projector. Its representation in the computational basis splits into a very convenient tensor product of two MPS (see section 2.4.2), each representing the same state (apart from conjugation). However, when transitioning to the Pauli basis representation, this tensor structure is lost due to the local mixing of each tensor and its conjugate induced by the change of basis described above.

Moreover, from the very definition of a statistical mixture in Eq. (5.2), it is clear that a density operator is *non-negative*, which implies that it can be rewritten as $\hat{\rho} = \hat{\Phi}\hat{\Phi}^\dagger$. It turns out that the MPO representation in the computational basis is more suitable for enforcing this condition, as we will see in the positive tensor network approach in Section 5.3.2.

Definition 5.2: normalized Pauli Tensor

For TN computations is very useful to introduce the so called **normalized Pauli tensor** ζ_{ij}^μ which basically incorporate in one single order-3 tensor all 3 Pauli matrices plus the identity matrix, which form a complete basis for single qubit operators. Basically we have in graphical notation (for $\mu \in \{0, 1, 2, 3\}$)

$$\zeta_{ij}^\mu = \frac{1}{\sqrt{2}} \left\{ \begin{array}{c} j \\ | \\ i \end{array} \right\}, \begin{array}{c} j \\ | \\ \otimes \\ | \\ i \end{array}, \begin{array}{c} j \\ | \\ \otimes \\ | \\ i \end{array}, \begin{array}{c} j \\ | \\ \otimes \\ | \\ i \end{array} \right\}_\mu$$

which satisfies the following orthogonality and completeness properties

$$\text{Tr}\{\zeta^\mu \zeta^\nu\} = \mu \begin{array}{c} \mu \\ | \\ \zeta^* \\ | \\ \zeta \\ | \\ \nu \end{array} = \mu \text{---} \nu \quad \sum_\mu \zeta^\mu \otimes \zeta^{*\mu} = \begin{array}{c} j \\ | \\ \zeta \\ | \\ i \end{array} \begin{array}{c} j' \\ | \\ \zeta^* \\ | \\ i' \end{array} = \begin{array}{c} j \\ \text{---} \\ i \end{array} \begin{array}{c} j' \\ \text{---} \\ i' \end{array}$$

Notice that any operator can be decomposed in terms of the normalized Pauli tensor; for example in the single qubit case one gets

$$\hat{O} = \sum_{x,y} \begin{array}{c} y \\ | \\ o \\ | \\ x \end{array} |x\rangle\langle y| = \sum_\mu \begin{array}{c} c \\ | \\ \mu \end{array} \zeta^\mu$$

with the local tensors which are related by the following local transformation

$$\begin{array}{c} c \\ | \\ \mu \end{array} = \mu \begin{array}{c} \mu \\ | \\ \zeta^* \\ | \\ o \end{array}$$

5.2 Minimally Entangled Typical Thermal State

Here we present a class of states originally introduced by Steven R. White in ref. [Whi09], along with the related algorithm proposed by Steven R. White and Erwin M. Stoudenmire in ref. [SW10].

The approach is exactly been devised to overcome the difficulty we have riced at the end of the previous section, and again it is inspired by the generic and very important question to describe a system at thermal equilibrium.

To make this concept more tangible, imagine we have a collection of pure states that represent different possible configurations of our system, as is typically the case for a mixture. Each pure state is like a snapshot of the system at a specific moment. When we average over many such snapshots, we get a picture of the system’s behavior at thermal equilibrium.

For instance, consider a quantum system weakly coupled to a heat bath. As the system equilibrates with the heat bath, it reaches a state described by the density matrix $\hat{\rho}$. If we then remove the heat bath, the system retains this mixed state. From this perspective, we can think of the system’s state as an ensemble of pure states that collectively generate $\hat{\rho} = \exp(-\beta\hat{H})/Z$.

This ensemble-of-pure-states viewpoint is particularly useful because it aligns with how we often simulate quantum systems. For example, when using techniques like diagonalization, the Density Matrix Renormalization Group (DMRG), or tensor network approaches, we can generate these pure states and efficiently compute finite temperature properties.

By proposing a set of idealized states, we argue that these states can be considered “typical” for a system in thermal equilibrium. The algorithm we introduce to generate these states not only helps us understand the system better but also provides a more efficient way to determine finite temperature properties of lattice models.

5.2.1 Classical vs Quantum Typical Sample

In quantum statistical mechanics, “typical states” refer to a set of states $\{|\phi(x)\rangle\}$ with associated unnormalized probabilities $P(x)$, from which we can sample to represent the thermal properties of a quantum system. The requirement for these states is that their ensemble should replicate the thermal density matrix of the system:

$$\sum_x P(x) |\phi(x)\rangle \langle \phi(x)| = e^{-\beta\hat{H}} \quad (5.25)$$

This ensures that the expectation value of any observable \hat{A} can be approximated by averaging $\langle \phi(x) | \hat{A} | \phi(x) \rangle$ over these typical states, with each state

$|\phi(x)\rangle$ selected according to the probability $P(x)/Z$, with $Z = \sum_x P(x)$. This framework bridges classical and quantum statistical mechanics, ensuring that the sampled states are representative of the system's thermal properties. Notice that the states $|\phi(x)\rangle$ need neither be orthogonal to each other nor constitute a complete basis set for the Hilbert space.

For classical lattice systems, typical states are **classical product states (CPS)**, which are configurations like $|x\rangle = \prod_j |x_j\rangle$, where x_j denotes the state of the j -th site. For instance, in an Ising model, a CPS might be $|\uparrow\downarrow\uparrow\downarrow \dots\rangle$, while in a register of qubits the analogous would be $|0101 \dots\rangle$. These states are intuitive and straightforward to generate numerically and experimentally, providing insight into the system's properties that might be obscured by the full density matrix. However, in quantum lattice models (spin systems, qubits registers, etc.), CPS do not typically represent the system's thermal state, especially at zero temperature, where the ground state is generally not a CPS. This is because CPS lack the entanglement necessary to capture the correlations present in a quantum system's ground state or thermal state.

Energy Eigenstates — An alternative approach might consider the energy eigenstates of the system. The energy eigenstates $|k\rangle$ and eigenvalues E_k satisfy:

$$\hat{\rho} = \frac{1}{Z} \sum_k e^{-\beta E_k} |k\rangle\langle k| \quad (5.26)$$

Thus, the thermal density matrix can be expressed in terms of these eigenstates. However, these eigenstates should not be considered typical states for several reasons:

1. *Equilibration and Dynamics*: Schrödinger highlighted that for large systems, especially at non-zero temperatures, the system does not equilibrate to any single energy eigenstate. The time required for such equilibration would be exponentially long in the number of particles N , making it impractical and unrealistic [Sch89].
2. *Sensitivity to Hamiltonian Uncertainties*: The energy eigenstates are exponentially sensitive to small changes in the Hamiltonian. Any slight uncertainty or perturbation in the system's parameters can drastically alter these states, making them unreliable for practical use in representing thermal properties.
3. *Superposition States*: In phases with broken symmetry, eigenstates often form highly non-classical superpositions of states with different values

of the order parameter. Such superpositions are not representative of typical states that one would expect to find in nature or experiments.

4. *Entanglement and Decoherence*: Energy eigenstates tend to be highly entangled, which is contrary to the effects of decoherence in realistic physical systems. Decoherence tends to suppress such high entanglement, leading to states that are more classically correlated.
5. *Computational Intractability*: From a computational perspective, obtaining energy eigenstates, especially for large systems, is highly challenging. The small energy level spacings require a full diagonalization of the Hamiltonian, which is computationally prohibitive for all but the smallest systems.

Constructing Typical States — To address these issues, we can construct typical states that satisfy the thermal property requirement using a complete orthonormal basis $\{|x\rangle\}$. These states are defined as:

$$|\phi(x)\rangle = P(x)^{-1/2} e^{-\beta\hat{H}/2} |x\rangle \quad (5.27)$$

where $P(x) = \langle x | e^{-\beta\hat{H}} | x \rangle$ is the unnormalized probability distribution. This construction ensures that:

$$\langle \hat{A} \rangle = \frac{1}{Z} \text{Tr}(\hat{\rho}\hat{A}) = \frac{1}{Z} \sum_x P(x) \langle \phi(x) | \hat{A} | \phi(x) \rangle \quad (5.28)$$

This approach avoids the need for computing the full spectrum of energy eigenstates, making it computationally feasible for large systems.

Minimally Entangled Typical Thermal States (METTS) are a specific type of typical state designed to have minimal entanglement. METTS are generated by choosing the initial set $\{|x\rangle\}$ as CPS. This minimization of entanglement makes METTS particularly suitable for numerical simulations using tensor network methods such as the Density Matrix Renormalization Group (DMRG). In fact, they provide a series of advantages:

1. *Low Entanglement*: By design, METTS have low entanglement, simplifying their numerical representation and manipulation.
2. *Physical Realism*: METTS incorporate decoherence effects naturally, making them more realistic for physical systems where such effects are significant.
3. *Symmetry Breaking*: METTS can spontaneously break symmetries, which is important for studying systems with long-range order.

4. *Revealing Short-Range Order*: Even in systems without broken symmetries, METTS can reveal underlying short-range order.

By leveraging the properties of METTS, we can effectively study the thermal properties of quantum systems, gaining insights that are computationally accessible and physically meaningful.

5.2.2 METTS with Tensor Network

The pure state method in ref. [SW10] involves a straightforward algorithm for generating a series of Matrix Product States that are distributed correctly according to the probability $P(x)/Z$. The procedure is as follows: (1) at first select a random CPS $|x\rangle$; (2) compute the METTS $|\phi(x)\rangle = e^{-\beta\hat{H}/2} |x\rangle P(x)^{-1/2}$, store it, if not too much resource consuming, otherwise evaluate all observable of interest and store the results; (3) Collapse to a new classical pure state $|y\rangle$ from $|\phi(x)\rangle$ with probability $p(x \rightarrow y) = |\langle y|\phi(x)\rangle|^2$, and return to step (2) for the next imaginary evolution.

Example 5.3: How METTS guarantees the correct distribution

Let us explore why the method presented in this section for producing METTS guarantees the correct distribution.

Consider an initial ensemble of classical product states (CPS) $|x\rangle$ distributed according to the probability $P(x)/Z$. If we select a CPS $|x\rangle$ at random from this ensemble and then proceed with the algorithm described previously, the probability of observing a specific CPS $|y\rangle$ at the end of step (3) is given by

$$\sum_x \frac{P(x)}{Z} p(x \rightarrow y) = \sum_x \frac{P(x)}{Z} |\langle y|\phi(x)\rangle|^2. \quad (5.29)$$

Substituting the definition $|\phi(x)\rangle = P(x)^{-1/2} e^{-\beta\hat{H}/2} |x\rangle$, we get

$$\sum_x \frac{\langle y| e^{-\beta\hat{H}/2} |x\rangle \langle x| e^{-\beta\hat{H}/2} |y\rangle}{Z} = \frac{P(y)}{Z}. \quad (5.30)$$

Thus, the probability of observing $|y\rangle$ aligns with the desired distribution $P(y)/Z$. The consistency of the resulting ensemble of $|y\rangle$ with the original distribution $P(x)/Z$ is ensured indeed by the detailed balance condition: $P(x)p(x \rightarrow y) = P(y)p(y \rightarrow x)$.

Thus, the ensemble of CPS distributed according to $P(x)/Z$ remains invariant under this process (is basically a fixed point of the algorithm).

Consequently, since each METTS $|\phi(x)\rangle$ is deterministically generated from a CPS $|x\rangle$ in step (2), the METTS themselves are also distributed according to $P(x)/Z$.

Notice that the aforementioned procedure generates a Markov chain of states. After a typical relaxation time, which may depend on the specific choice of the CPS basis states, the states $\{|\phi(x)\rangle\}$ are distributed according to the target distribution $P(x)/Z$. Consequently, any expectation value can be estimated by averaging over those states. In other words,

$$\langle \hat{A} \rangle = \mathbb{E}_{x \sim P(x)/Z} [\langle \phi(x) | \hat{A} | \phi(x) \rangle], \quad (5.31)$$

where $\mathbb{E}_{x \sim P(x)/Z}$ is indicating the ensemble average.

We now discuss the practical implementation of these steps assuming we are working with n -qubits register. In this scenario, a CPS state is represented by a classical configuration of the register, namely $|x_1, x_2, \dots, x_n\rangle$ with $x_j \in \{0, 1\}$. Extension to any local base dimension is straightforward.

METTS generation — The second step of the algorithm involves generating the typical thermal state $|\phi(x)\rangle$ from a classical product state $|x\rangle$. In the tensor network framework, this can be efficiently achieved using either the Time-Evolving Block Decimation (TEBD) or the Time-Dependent Variational Principle (TDVP) for Matrix Product States (MPS) in imaginary time. The choice of algorithm depends on the specific form of the Hamiltonian's interactions. We refer the reader to Chapter 2 and Chapter 3 for a detailed explanation of the methods. Here, we will briefly review the TEBD approach for nearest-neighbor Hamiltonians.

For instance, for short-range interacting Hamiltonians with couplings only between neighboring sites, represented as $\hat{H} = \sum_i \hat{h}_{i,i+1}$, the imaginary time evolution for a small step $\Delta\beta$ can be approximated by

$$e^{-\Delta\beta\hat{H}} \simeq e^{-\Delta\beta\hat{h}_{1,2}/2} \dots e^{-\Delta\beta\hat{h}_{n-1,n}/2} e^{-\Delta\beta\hat{h}_{n-1,n}/2} \dots e^{-\Delta\beta\hat{h}_{1,2}/2}, \quad (5.32)$$

with a Trotter error of order $\sim (\Delta\beta)^3$. This operator can be applied to an MPS by performing a left-to-right sweep followed by a right-to-left sweep, sequentially updating the local tensors and performing local Singular Value Decomposition (SVD). This process normalizes the state locally, effectively moving the central site of the mixed canonical form of the MPS. These sweeps are repeated until the desired inverse temperature $\beta/2$ is reached.

CPS sampling of an MPS — Here we assume $|\phi\rangle$ is in the right canonical form, i.e. with the orthogonality center at the first site. We have omitted the variable indicating the CPS state from which the METTS was generated, as it is irrelevant for the sampling procedure.

The goal is to sample a CPS state $|x_1, \dots, x_n\rangle$ with a probability $P_\phi(x_1, \dots, x_n) \equiv |\langle x_1, \dots, x_n | \phi \rangle|^2$ which in general is an exponentially hard problem due to the Hilbert space size.

To overcome this difficulty, we rewrite the full probability in terms of conditional and marginal probabilities as

$$P_\phi(x_1, \dots, x_n) = p_\phi(x_1)p_\phi(x_2|x_1) \cdots p_\phi(x_n|x_1, \dots, x_{n-1}), \quad (5.33)$$

where $p_\phi(x_j|x_1, \dots, x_{j-1}) = \frac{p_\phi(x_1, \dots, x_j)}{p_\phi(x_1, \dots, x_{j-1})}$ is the probability that the state $|x_j\rangle$ occurs at position j given that the classical configuration $\{x_1, \dots, x_{j-1}\}$ has already been found at positions $\{1, \dots, j-1\}$, regardless of the outcomes in the rest of the system (i.e., marginalizing over all possible outcomes for the remaining qubits $\{j+1, \dots, n\}$). We defined the marginal probability as $p_\phi(x_1, \dots, x_j) = \sum_{x_{j+1}, \dots, x_n} |\langle x_1, \dots, x_n | \phi \rangle|^2$.

The key advantage here is that the chain of conditional probabilities can be efficiently evaluated in an MPS state. To illustrate, let's start with the orthogonality center at the first site. The probability of the first outcome is given by:

$p_\phi(x_1) =$

(5.34)

and we can sample x_1 based on this probability. Suppose the specific outcome x_1 is selected.¹ The state is then updated by projecting the first qubit into this classical configuration, resulting in a new MPS state for the remaining qubits

¹For a qubit system where $x_1 \in \{0, 1\}$, we can uniformly draw $r \in [0, 1]$. If $r \leq p_\phi(0)$, then $x_1 = 0$; otherwise, $x_1 = 1$.

$\{2, \dots, n\}$:

$$\begin{aligned}
 |\phi_{x_1}\rangle &= \frac{1}{\sqrt{p_{\phi}(x_1)}} \left[\text{MPS diagram with } M_1, B_2, \dots, B_n \text{ and } x_1 \text{ output} \right] \\
 &= \left[\text{MPS diagram with } \mathbb{L}_1, B_2, \dots, B_n \right]
 \end{aligned}
 \tag{5.35}$$

with an updated left-boundary vector \mathbb{L}_j (where $j = 1$ at the end of the first step). Next, we compute the conditional probability for the second qubit given the first:

$$p_{\phi}(x_2 | x_1) = p_{\phi_{x_1}}(x_2) = \left[\text{MPS diagram with } \mathbb{L}_1, B_2, B_2^*, \mathbb{L}_1^* \text{ and } x_2 \text{ output} \right]
 \tag{5.36}$$

Using this probability, we sample x_2 , following the same procedure as in the first step. Upon determining a classical configuration for the second qubit, we update the state by projecting the second qubit, yielding a new MPS for the remaining qubits $\{3, \dots, n\}$:

$$\begin{aligned}
 |\phi_{x_1, x_2}\rangle &= \frac{1}{\sqrt{p_{\phi}(x_2 | x_1)}} \left[\text{MPS diagram with } \mathbb{L}_1, B_2, B_3, \dots, B_n \text{ and } x_2 \text{ output} \right] \\
 &= \left[\text{MPS diagram with } \mathbb{L}_2, B_3, \dots, B_n \right]
 \end{aligned}
 \tag{5.37}$$

The outlined procedure can be reformulated as an **iterative algorithm** from the leftmost to the rightmost qubit of the MPS. After initializing the boundary vector $\mathbb{L}_0 = (1)$, at each j -th step:

1. Compute the conditional probability $p_{\phi}(x_j | x_1, \dots, x_{j-1}) = p_{\phi_{x_1 \dots x_{j-1}}}(x_j)$.
2. Generate a classical local configuration accordingly.
3. Update the boundary vector: $\mathbb{L}_j = p_{\phi}(x_j | x_1, \dots, x_{j-1})^{-1/2} \mathbb{L}_{j-1} \cdot B^{x_j}$.

4. Repeat steps (1) to (3) until the rightmost qubit is sampled.

At the end of the algorithm, we obtain a CPS state $|x_1, x_2, \dots, x_n\rangle$ with probability $|\langle x_1, \dots, x_n | \phi \rangle|^2$.

Note that the outlined algorithm requires only matrix-to-vector multiplications along the auxiliary space of the MPS, resulting in an overall computational complexity scaling as $\sim \chi^2$, where χ is the maximum bond dimension of the MPS.

5.3 Lindblad dynamics

In this section, we aim to explore the most general class of Markovian transformations that can be applied to density matrices. Up to this point, we have considered two primary modes of quantum evolution: coherent evolution and state collapse following measurement, as described by postulates of quantum mechanics (see Chapter 2). Significant effort has been devoted to reconciling these two forms of quantum evolution, but a definitive unification remains elusive [Man20].

The central question we address is: What is the most general transformation that can be performed on a quantum system, and how can we describe this transformation using a dynamical equation?

To answer this, we focus on maps that transform density matrices into density matrices. We denote by $\mathfrak{D}_{\mathcal{H}}$ the space of all density matrices associated to a Hilbert space \mathcal{H} . We seek a map $\mathcal{V} : \mathfrak{D}_{\mathcal{H}} \rightarrow \mathfrak{D}_{\mathcal{H}}$ that acts on this space and maps it onto itself. For the map \mathcal{V} to ensure that its output is still a valid density matrix, it must satisfy the following criteria:

- **Trace Preservation:** The map must preserve the trace, meaning for any operator \hat{A} in the space of linear operators acting on the Hilbert space \mathcal{H} , the trace of the transformed operator should equal the trace of the original operator, i.e. $\text{Tr}[\mathcal{V}(\hat{A})] = \text{Tr}[\hat{A}]$.
- **Complete Positivity:** The map must be completely positive. This condition is more nuanced and requires further discussion, as it ensures that the map remains positive even when extended to a larger Hilbert space.

Maps that satisfy both of these properties are referred to as *completely positive and trace-preserving maps* (CPT maps). While trace preservation is straightforward and self-evident, complete positivity involves a more detailed definition.

Definition 5.3: Positive Map

A *positive map* \mathcal{V} between two spaces of operators is a linear map that preserves the positivity of operators. Formally, let $\mathcal{B}(\mathcal{H})$ denote the space of bounded linear operators on a Hilbert space \mathcal{H} . A map $\mathcal{V} : \mathcal{B}(\mathcal{H}) \rightarrow \mathcal{B}(\mathcal{K})$, where \mathcal{H} and \mathcal{K} are Hilbert spaces, is called positive if:

$$\mathcal{V}(\hat{A}) \geq 0 \text{ whenever } \hat{A} \geq 0. \quad (5.38)$$

In other words, \mathcal{V} is positive if it maps positive semidefinite operators to positive semidefinite operators. Here, $\hat{A} \geq 0$ denotes that \hat{A} is a positive semidefinite operator, meaning that $\langle \psi | \hat{A} | \psi \rangle \geq 0$ for all $|\psi\rangle \in \mathcal{H}$.

Definition 5.4: Completely Positive Map

A map \mathcal{V} between spaces of operators is called *completely positive* if it is positive and, additionally, remains positive when extended to larger spaces. Formally, let $\mathcal{B}(\mathcal{H})$ and $\mathcal{B}(\mathcal{K})$ denote the spaces of bounded linear operators on Hilbert spaces \mathcal{H} and \mathcal{K} , respectively. A map $\mathcal{V} : \mathcal{B}(\mathcal{H}) \rightarrow \mathcal{B}(\mathcal{K})$ is completely positive if for every integer n and for every matrix $\{\hat{A}_{ij}\}_{i,j=1}^n$ of operators $\hat{A}_{ij} \in \mathcal{B}(\mathcal{H})$, the following condition holds:

$$\mathcal{V} \otimes \mathbb{1}_{\mathcal{B}(\mathbb{C}^n)} \left(\sum_{i,j=1}^n \hat{A}_{ij} \otimes |i\rangle\langle j| \right) \geq 0, \quad (5.39)$$

where $\mathbb{1}_{\mathcal{B}(\mathbb{C}^n)}$ is the identity map on $\mathcal{B}(\mathbb{C}^n)$, and $|i\rangle\langle j|$ are the standard matrix units in \mathbb{C}^n . In simpler terms, \mathcal{V} is completely positive if it remains positive when extended to act on a larger Hilbert space by tensoring with the identity on a finite-dimensional space.

Lindblad equation — A widely used approach for deriving the Lindblad master equation stems from the study of open quantum systems. In this context, the Lindblad equation describes the dynamics of a subsystem that is part of a larger, more complex system. For a detailed exposition of this derivation, one can refer to standard texts such as Breuer and Petruccione’s *The Theory of Open Quantum Systems* [BP02] and Gardiner and Zoller’s *Quantum Noise* [GZ00].

The scope of this book is not to give a detailed derivation of such equation whilst explain how to integrate that equation via TN methods.

Let's just mention that a Lindblad master equation provides a powerful framework for describing the dynamics of the system's density matrix $\hat{\rho}$. This equation accounts for both the unitary evolution governed by the system's Hamiltonian and the effects of dissipation and decoherence due to interaction with the environment. The Lindblad master equation can be expressed as:

$$\partial_t \hat{\rho}(t) = -i[\hat{H}, \hat{\rho}(t)] + \sum_i \gamma_i \left(\hat{L}_i \hat{\rho}(t) \hat{L}_i^\dagger - \frac{1}{2} \{ \hat{L}_i^\dagger \hat{L}_i, \hat{\rho}(t) \} \right), \quad (5.40)$$

where:

- $\hat{\rho}(t)$ is the density matrix of the system at time t .
- \hat{H} is the Hamiltonian, representing the coherent evolution of the system.
- \hat{L}_i are the Lindblad operators (known also as **jump operators**), which model the interaction of the system with its environment and the resulting dissipative effects.
- $\gamma_i \geq 0$ are the damping rates, which quantify the strength of the dissipative processes.

If all the damping rates γ_i are zero, the Lindblad equation reduces to the von Neumann equation (5.8). This approach simplifies the description of the dynamics, making it easier to analyze and simulate the behavior of open quantum systems.

5.3.1 MPO formalism

When simulating the Lindblad dynamics of a system of qubits using Matrix Product Operator (MPO) techniques, two distinct approaches can be employed, each leveraging the unique structure of the quantum system's density matrix.

The first approach parallels the method used for imaginary time evolution in preparing thermal states. In this method, the density matrix $\hat{\rho}$ is treated as a "superket" $|\rho\rangle\rangle$, effectively doubling the Hilbert space by associating an ancillary qubit with each physical qubit. The resulting state is then approximated as a Matrix Product State (MPS), where each local tensor encompasses the physical and ancillary qubits in a tensor product space. This approach is intuitive as it extends the well-established MPS formalism to mixed states by simply reinterpreting the density matrix as a state vector in an enlarged Hilbert space.

The second approach retains the operatorial nature of the density matrix. Instead of converting $\hat{\rho}$ into a state vector, it is expressed directly in the Pauli

matrix basis as represented in Eq. (5.24). This technique leverages the natural decomposition of quantum operators in terms of the Pauli matrices, allowing the density matrix to be represented as a linear combination of Pauli operators. The advantage of this approach lies in its direct connection to the operator algebra and the potential for a more compact and efficient representation, especially when dealing with qubits.

However, similarly to what happens in the computational basis representation, in this case as well, when treating the wave function associated with the density matrix as an MPS, the algorithm inherently preserves the norm square of the wave function. This preservation corresponds to maintaining $\text{Tr}(\hat{\rho}^2)$. To properly compute expectation values, we need to compute first the partition function

$$Z = \text{Tr}(\hat{\rho}) = \begin{array}{c} \textcircled{e_1} \text{---} \textcircled{e_2} \text{---} \dots \text{---} \textcircled{e_n} \\ \downarrow \quad \downarrow \quad \quad \quad \downarrow \\ \nabla \quad \nabla \quad \quad \quad \nabla \end{array} \quad (5.41)$$

which allows us to restore the correct normalization of the density matrix. In this framework, due to the anomalous normalization of the density operator, the purity of a state, which quantifies how mixed the state is, will be given by $\text{Purity} = Z^{-2}$. This expression shows that the purity is inversely proportional to the square of the partition function, and reflects the impact of the normalization on assessing how close the state is to being pure.

In the following, we will provide a detailed exposition of this second approach, elucidating how the Pauli basis expansion facilitates the integration of Lindblad dynamics in a computationally efficient manner using the MPO formalism. We will also compare and contrast this with the superket method, highlighting the contexts in which each approach is most effective.

First and foremost, the integration of the Lindblad equation involves exponentiating the right-hand side of Eq. (5.40). To accomplish this, and assuming that we are evolving a density matrix over a small time step dt , we apply a Trotter decomposition to the evolution super-operator. This decomposition separates the evolution into two distinct parts: the unitary part and the dissipative part, as follows: $\mathcal{U}[\cdot] = \mathcal{U}_u[\cdot]\mathcal{U}_d[\cdot]$ where the unitary part acts as usual

$$\mathcal{U}_u[\hat{\rho}] = e^{-i\hat{H}dt} \hat{\rho} e^{i\hat{H}dt} \quad (5.42)$$

while the dissipative part can be formally written as

$$\mathcal{U}_d[\hat{\rho}] = e^{\mathcal{D}[\hat{\rho}]dt} \quad (5.43)$$

with

$$\mathcal{D}[\hat{\rho}] = \sum_i \mathcal{D}_i[\hat{\rho}] \equiv \sum_i \gamma_i \left(\hat{L}_i \hat{\rho}(t) \hat{L}_i^\dagger - \frac{1}{2} \{ \hat{L}_i^\dagger \hat{L}_i, \hat{\rho}(t) \} \right), \quad (5.44)$$

where the sum runs over the local qubits.

Unitary part — The unitary part of the evolution, for short-range interacting Hamiltonians, can be integrated using the TEBD scheme, but now in the Pauli basis representation. Specifically, given a Hamiltonian of the form $\hat{H} = \sum_i \hat{h}_{i,i+1}$, the elementary brick-wall evolution operator is $\hat{u} \equiv e^{-i\hat{h}_{i,i+1} dt}$, which acts on the density matrix as $\hat{u}\hat{\rho}\hat{u}^\dagger$. This induces a transformation of the local tensors in the Pauli-basis MPO formalism of the density matrix as follows:

$$\text{Diagram illustrating the transformation of a local tensor } \mathbb{R} \text{ under a unitary evolution operator } \hat{u}. \quad (5.45)$$

where we basically defined the Pauli-basis unitary tensor

$$\mathbb{R}_{\nu\nu'}^{\mu\mu'} = \text{Tr} \left[\hat{\zeta}^\nu \otimes \hat{\zeta}^{\nu'} \hat{u} \hat{\zeta}^\mu \otimes \hat{\zeta}^{\mu'} \hat{u}^\dagger \right]. \quad (5.46)$$

As is customary in the TEBD algorithm, it is advantageous to apply the local tensor \mathbb{R} when the MPO representation of the density matrix is in mixed canonical form with respect to one of the two sites involved in the local update. Once in this form, the standard TEBD procedure is followed by performing a truncated singular value decomposition (SVD). This SVD step results in a compressed pair of new local tensors, thereby maintaining the efficiency and accuracy of the MPO representation.

Dissipative part — Frequently, the dissipative component of the Lindblad equation is comprised of a series of non-unitary operations applied to individual lattice sites. When this is the case, as in Eq. (5.44), we can apply each local exponential superoperator $e^{\mathcal{D}_i[\cdot]dt}$ sequentially on the local tensor \mathcal{g}_i of the Pauli-basis MPS representation of the density matrix. This process can be effectively implemented by considering the superket reshaping of the Pauli

basis, allowing for the computation of the resulting tensor.

$$\mathbb{L}_{\nu\mu} = \langle\langle \zeta^\nu | e^{\mathbb{D} dt} | \zeta^\mu \rangle\rangle = \begin{array}{c} \mu \\ | \\ \boxed{\mathbb{L}} \\ | \\ \nu \end{array} = \begin{array}{c} \mu \\ | \\ \textcircled{\zeta} \\ | \\ \textcircled{e^{\mathbb{D} dt}} \\ | \\ \textcircled{\zeta^*} \\ | \\ \nu \end{array} \quad (5.47)$$

where $\mathbb{D} \equiv \gamma_i \left[\hat{L}_i \otimes \hat{L}_i^* - \frac{1}{2} (\hat{L}_i^\dagger \hat{L}_i \otimes \hat{I} + \hat{I} \otimes \hat{L}_i^\dagger \hat{L}_i^*) \right]$, and we have omitted the subscript i indicating the lattice site. Subsequently, we can update each local MPS tensor by contracting its physical index with this new tensor via the operation $\sum_\mu \mathbb{L}_{\nu\mu} \rho_{\alpha\beta}^\mu$, without altering the auxiliary bond dimension.

Example 5.4: One-Body particle loss and gain in hard-core bosons

As an example here we consider hard-core Boson particle jumping in a 1D lattice. These are particles that basically behave as bosons, a part from the fact that each lattice site cannot be occupied by more than one single bosons. As a matter of fact, the unitary dynamics is governed by an Hamiltonian which can be described in terms of a fictitious spin-1/2

$$\hat{H} = - \sum_j \left(\hat{\sigma}_j^+ \hat{\sigma}_{j+1}^- + \hat{\sigma}_j^- \hat{\sigma}_{j+1}^+ \right), \quad (5.48)$$

where the local boson density is related to a fictitious \hat{z} -magnetisation via $\hat{n}_j = (1 - \hat{\sigma}_j^z)/2$.

The simple unitary part can be mapped to a non-interacting fermionic theory and the dynamics can be analytically solved. However, in a real scenario, each lattice sites may be imperfect, and incoherent fluctuations in the number of particles can occur due to a one-body loss and gain, resulting in a Markovian evolution governed by the following Lindblad equation for the density matrix

$$\begin{aligned} \partial_t \hat{\rho} = & -i[\hat{H}, \hat{\rho}] + \gamma_L \sum_j \left(\hat{L}_j \hat{\rho} \hat{L}_j^\dagger - \frac{1}{2} \{ \hat{L}_j^\dagger \hat{L}_j, \hat{\rho} \} \right) \\ & + \gamma_G \sum_j \left(\hat{L}_j^\dagger \hat{\rho} \hat{L}_j - \frac{1}{2} \{ \hat{L}_j \hat{L}_j^\dagger, \hat{\rho} \} \right), \end{aligned} \quad (5.49)$$

where $\gamma_L, \gamma_G > 0$ set the loss and gain rates, and $\hat{L}_j = \hat{\sigma}_j^+$ removes a particles from the site j .

Notice that in the previous equation one has two incoherent terms acting independently in each lattice site. Nonetheless, we can collect both single-site loss and gain and compute the total local exponential $\exp\{\mathbb{D} dt\}$ as in the main text

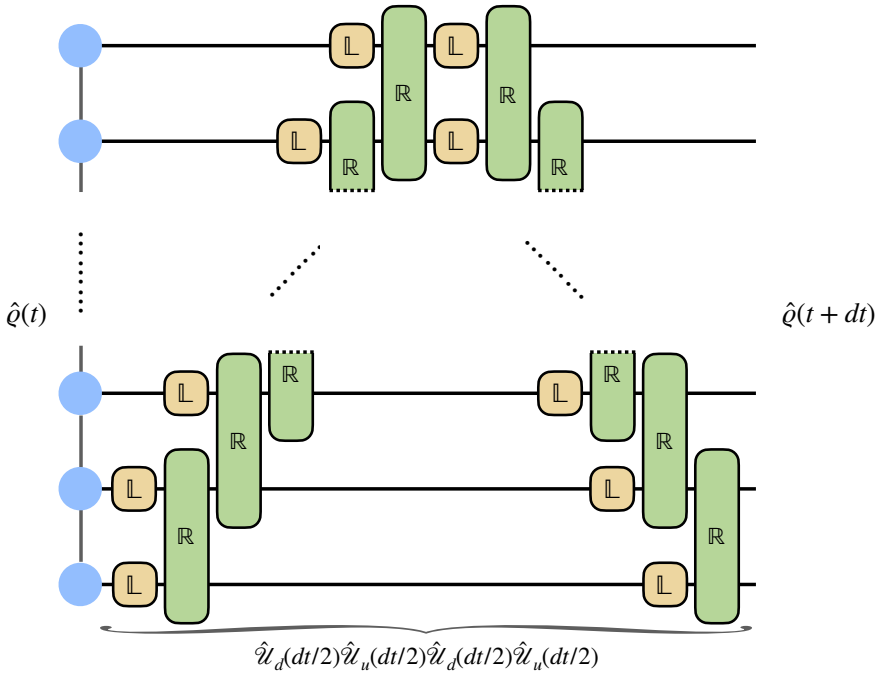
$$\mathbb{L}_{\nu\mu} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{-\gamma dt/2} & 0 & 0 \\ 0 & 0 & e^{-\gamma dt/2} & 0 \\ (1 - 2n_s)(1 - e^{-\gamma dt}) & 0 & 0 & e^{-\gamma dt} \end{pmatrix}, \quad (5.50)$$

where we defined the total decay rate $\gamma = \gamma_L + \gamma_G$ and the stationary density in the steady-state $n_s = \gamma_G/(\gamma_L + \gamma_G)$. In fact, since the Lindbladian is uniform, one may expect $\hat{\rho}$ to relax to a homogeneous state. Indeed, it is easy to show that the stationary state is an infinite temperature state in the sector with a fixed number of particles

$$\hat{\rho}_\infty = \frac{1}{Z} e^{\mu \sum_j \hat{n}_j} = \prod_j \frac{e^{\mu \hat{n}_j}}{1 + e^\mu}, \quad \mu = \log \frac{\gamma_G}{\gamma_L},$$

where the chemical potential is determined by the condition that the dissipator annihilates $\hat{\rho}_\infty$, as \hat{H} trivially commutes with it because of the particle number conservation.

Collecting both the dissipative and the unitary part, and using a time step of $dt/2$, we can easily implement the following second-order integration scheme whose graphical representation looks like (evolution is from left to right)



5.3.2 Locally Purified Tensor Network

As already mentioned at the beginning of this chapter, one of the conditions which is sacrificed by both the “superket” and the Pauli-MPS approach outlined in the previous section, is the requirement that a well defined density operator needs to be *non-negative*. And in fact, a more restrictive ansatz can be introduced which is non-negative by construction: the purification ansatz, known also as **Matrix Product Density Operator (MPDO)** ansatz [VGC04; Cue+13; Wer+16].

As a matter of fact, in quantum mechanics, it is well-established that for any mixed state $\hat{\rho}$, there exists a corresponding purification. This means that one can always find a pure state $|\Phi_{SA}\rangle$ in a larger Hilbert space, involving an additional *ancillary* system, such that tracing out the ancillary degrees of freedom yields the original mixed state of the system. Mathematically, if the enlarged system is described by the pure state $|\Phi_{SA}\rangle$ in the combined Hilbert space $\mathcal{H}_S \otimes \mathcal{H}_A$, then $\hat{\rho} = \text{Tr}_A (|\Phi_{SA}\rangle \langle \Phi_{SA}|)$ returns the original mixed state $\hat{\rho}$ on the system’s Hilbert space \mathcal{H}_S .

The concept of purification is particularly powerful in the context of tensor networks, where it underpins the construction of Matrix Product Density Operators. The purification ansatz leverages the idea of representing the purified state as a Matrix Product State on an extended system. Specifically, for every physical site in the system, an ancillary site of the same dimension is introduced, effectively doubling the physical dimension of each site.

This means that the Hilbert space of the combined system, originally of dimension d per site (two for qubits), is extended to $d \times d$ per site (four for qubits) in the purified MPS. The enlarged site thus accommodates both the physical and ancillary degrees of freedom, leading to an MPS representation that efficiently encodes the purification

$$|\Phi_{SA}\rangle = \sum_{x_1, \dots, x_n} \sum_{y_1, \dots, y_n} \left(\begin{array}{c} \text{---} A_1 \text{---} \\ \text{---} A_2 \text{---} \\ \vdots \\ \text{---} A_n \text{---} \end{array} \right)_{\substack{x_1, y_1 \\ x_2, y_2 \\ \vdots \\ x_n, y_n}} |\mathbf{x}\rangle |\mathbf{y}\rangle \quad (5.51)$$

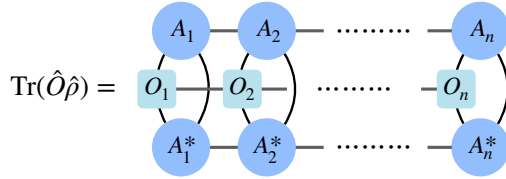
Here, the MPS describes the local purified state with indices \mathbf{x} corresponding to the system’s physical sites and indices \mathbf{y} corresponding to the ancillary sites. The crucial point is that by tracing out the ancillary indices, one recovers the original MPDO representation of the mixed state, i.e.

$$\hat{\rho} = \text{Tr} |\Phi_{SA}\rangle \langle \Phi_{SA}| = \sum_{x_1, \dots, x_n} \sum_{x'_1, \dots, x'_n} \left(\begin{array}{c} \text{---} A_1^* \text{---} \\ \text{---} A_2^* \text{---} \\ \vdots \\ \text{---} A_n^* \text{---} \end{array} \right)_{\substack{x'_1, x'_2 \\ \vdots \\ x'_n}} \left(\begin{array}{c} \text{---} A_1 \text{---} \\ \text{---} A_2 \text{---} \\ \vdots \\ \text{---} A_n \text{---} \end{array} \right)_{\substack{x_1, x_2 \\ \vdots \\ x_n}} |\mathbf{x}\rangle \langle \mathbf{x}'| \quad (5.52)$$

Since it is constructed from the partial trace of a normalized state, the resulting MPDO is a valid physical state. But it is not the most general MPO, because the tensors describing the density matrix MPO now have a very peculiar structure, $\rho_{\tilde{\alpha}\tilde{\beta}}^{xx'} = \sum_y A_{\alpha\beta}^{xy} (A_{\alpha'\beta'}^{x'y})^*$, with the auxiliary indices $\tilde{\alpha} = (\alpha, \alpha')$ and $\tilde{\beta} = (\beta, \beta')$ originated by the fusion of the two auxiliary spaces. Notice that, in this representation, the normalisation of the purified state imply the usual normalisation of the density matrix $\langle \Phi_{SA} | \Phi_{SA} \rangle = \text{Tr}(\hat{\rho}) = 1$.

In practice, this positive ansatz can be utilized effectively in numerical simulations, as will be demonstrated in the following scenarios. Before proceeding, it is important to observe that, much like the “superket” formulation of a density operator, the purification ansatz simplifies the computation of expectation values of local operators (or MPOs in general). Specifically, given that the density matrix can be expressed as $\hat{\rho} = \text{Tr}_A (|\Phi_{SA}\rangle \langle \Phi_{SA}|)$, the expectation value of an operator \hat{O} with respect to the density matrix reduces to

the expectation value in the purified state: $\text{Tr}(\hat{O}\hat{\rho}) = \langle \Phi_{SA} | \hat{O}_S \otimes \hat{I}_A | \Phi_{SA} \rangle$, where the operator \hat{O}_S acts non-trivially only on the system indices, while \hat{I}_A acts as the identity on the ancillary indices. Graphically one has:



Thermal state with positive TN

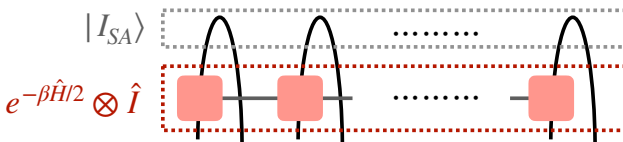
The purification ansatz is very natural for thermal states. Indeed, a purification of a Gibbs ensemble is intimately related to the METS we have seen in Section 5.2. In this scenario, the purified state reads

$$|\Phi(\beta)\rangle = \frac{1}{\text{Tr}(e^{-\beta\hat{H}})} \sum_k e^{-\beta E_k/2} |E_k\rangle_S |E_k\rangle_A \tag{5.53}$$

where the sum run over all the energy eigenstates, $\hat{H} |E_k\rangle = E_k |E_k\rangle$. Notice that such state can be rewritten as (a part from the normalisation)

$$|\Phi(\beta)\rangle \propto (e^{-\beta\hat{H}/2} \otimes \hat{I}) |I\rangle_{SA} \tag{5.54}$$

where basically the vector $|I\rangle_{SA}$ is a reshaping of the identity operator (as we already seen in the superket notation) and represents a maximally entangled state between system and ancilla. Notice that, we have decided to operate with the Hamiltonian on the system only, however, due to the equivalence of system and ancilla, we may have acted on the ancilla as well by splitting the imaginary evolution as $e^{-\beta\hat{H}/4} \otimes e^{-\beta\hat{H}/4}$. The procedure we have outlined above can be graphically represented as:

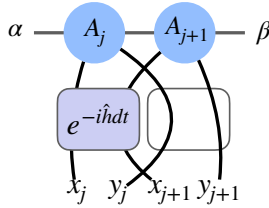


It is important to note that, within this Tensor Network (TN) representation, we assume the ability to express the operator $e^{-\beta\hat{H}/2}$ in Matrix Product Operator (MPO) form. However, as we are well aware, for both local and non-local Hamiltonians, it is often more practical to obtain the action of the exponentiated Hamiltonian $e^{-\beta\hat{H}/2}$ through imaginary-time evolution using techniques such as the Time-Evolving Block Decimation (TEBD) or the Time-Dependent Variational Principle (TDVP) schemes.

Open system dynamics with positive TN

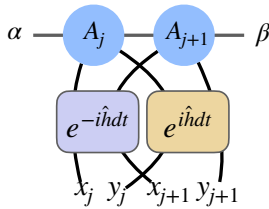
When dealing with Lindblad dynamics, the positive TN approach can have both advantages and disadvantages. We refer the reader to Eq. (5.40), where the dissipative part of the Lindblad generator is assumed to act locally on each lattice site (or qubit).

When integrating the **unitary part of the dynamics**, the purified state can be evolved from time t to $t + dt$ using the TEBD scheme, particularly for local Hamiltonians where $\hat{H} = \sum_j \hat{h}_{j,j+1}$. The update of two neighboring tensors is depicted as follows:



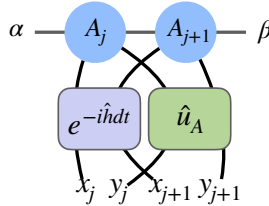
In this representation, the identity operator acts trivially on the ancillary states. As is typical in the TEBD algorithm, this operation is followed by a singular value decomposition (SVD) to reshape the updated tensors into their original MPS form.

It's important to note that the unitary part of Lindblad dynamics is often the primary contributor to entanglement spreading across the system, leading to an unbounded growth in the auxiliary bond dimension. To mitigate this issue, one can utilize the fact that the ancillary system can be evolved unitarily according to any operator. For instance, Karrasch et al. in ref. [KBM12] demonstrated that the entanglement growth inherent to any time-dependent calculation can be significantly reduced if the ancillary degrees of freedom, which purify the statistical operator, are time-evolved with the physical Hamiltonian but with time reversed. This approach is locally translated into the following TN contraction:



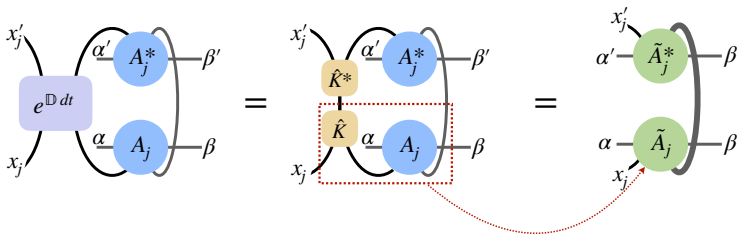
However, this method requires the ancillary system to be a copy of the physical one (thus having the same local physical dimensions), which may not be applicable once a local dissipator comes into play.

An even more effective approach, independent of the dimensions of the ancillary system, involves exploiting the fact that the physical density matrix is unaffected by *any* unitary operation performed on the ancillary system. In graphical notation, we can apply the local time-step operator $e^{-i\hat{h}dt} \otimes \hat{u}_A$ as follows:



Here, we seek the optimal unitary \hat{u}_A that minimizes the bipartite entanglement entropy, effectively acting as the best local disentangler for the purified state $|\Phi_{SA}\rangle$.

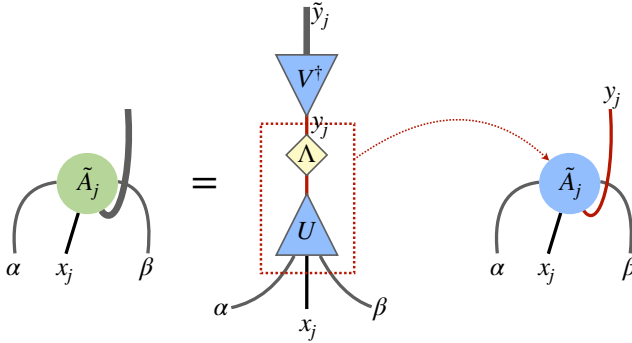
After implementing the coherent part of the evolution, the next step is to apply the **dissipative part**, which is typically assumed to be local in space. This means that each dissipative operator acts independently on each lattice site. The dissipative operator, denoted as $e^{\mathbb{D}dt}$, is **completely positive**, allowing it to be decomposed into a set of Kraus operators $\{\hat{K}_q\}$ such that $e^{\mathbb{D}dt} = \sum_{q=1}^k \hat{K}_q \otimes \hat{K}_q^*$. The action of this dissipative operator on the density matrix is realized through the contraction of the Kraus operators with the MPS tensors that describe the purified state. This process is represented graphically as follows:



Each time a dissipator layer is locally applied, the ancillary dimension — also known as the Kraus dimension κ — of the MPS increases by a factor of k .

The dissipative action tends to mix the state progressively. For example, the purified tensor network representation of a pure state (where the density matrix is a projector) has a Krauss dimension $\kappa = 1$, indicating no entanglement with the ancilla. However, as the dissipative part continues to act, this mixing increases the system-ancilla entanglement, leading to a rapid growth in the Kraus dimension κ .

To manage this growth, a dedicated compression procedure has been proposed, as outlined in ref. [Wer+16]. This method involves performing an SVD on the appropriately reshaped local tensor:



In this step, the largest singular values up to a given threshold are retained, while the isometry V^\dagger is discarded, as it is irrelevant to the definition of the density operator after tracing out the ancillary degrees of freedom.

5.4 Unravelling and monitored dynamics

One way of simulating Lindblad dynamics (5.40) is through a technique known as **unraveling** the Lindblad equation. In this approach, the evolution of the quantum system is described as an ensemble of individual quantum trajectories. Each trajectory represents a possible realization of the system’s dynamics. This method is computationally efficient for many systems since it avoids directly solving the full Lindblad master equation for the density matrix, which can be highly complex and resource-intensive, especially for large systems.

In the following we present two unravelings of the Lindblad equation for a monitored quantum dynamics.

5.4.1 Measurement protocols

Measurement in quantum mechanics is a fundamental aspect that significantly influences the state of a quantum system. The nature of quantum measurements can be broadly categorized into two types: projective (or strong) measurements, which are typically associated with quantum jumps, and weak measurements.

Quantum Jumps — Consider a quantum many-body system in one dimension, with the total Hilbert space $\mathcal{H} = \bigotimes_j \mathcal{H}_j$ being the tensor product of single-particle Hilbert spaces \mathcal{H}_j . If the system is isolated from the environment, it evolves according to the Schrödinger equation

$$|\psi(t)\rangle = e^{-i\hat{H}t} |\psi(0)\rangle, \quad (5.55)$$

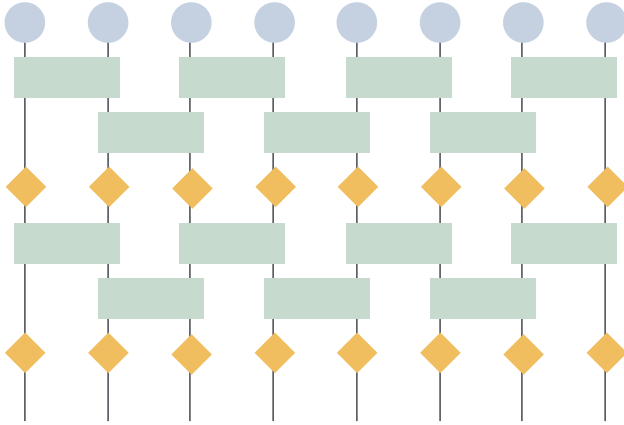
where \hat{H} is the full interacting Hamiltonian of the system and $|\psi(0)\rangle$ is the initial state. Assume that the unitary dynamics are sporadically disrupted by local measurements. Each local Hilbert space \mathcal{H}_j is briefly randomly coupled to a measurement apparatus that measures a local observable $\hat{O}_j = \sum_{k=1}^K o_k \Pi_j^{(k)}$, where o_k are the possible outcomes and $\Pi_j^{(k)}$ are the corresponding projection operators. These measurements occur at discrete time intervals dt with a characteristic rate γ . When a measurement occurs, the state $|\psi\rangle$ is projected according to Born's rule

$$|\psi\rangle \rightarrow \frac{\Pi_j^{(k)} |\psi\rangle}{\sqrt{\langle \psi | \Pi_j^{(k)} | \psi \rangle}}, \quad (5.56)$$

According to this protocol, the evolution of the many-body state $|\psi(t)\rangle$ depends on the series of measurement events and their results, featuring occasional quantum jumps that are sudden transitions in the quantum system. Notice that the quantum state is pure during the entire evolution.

This monitored evolution can be depicted through digitized quantum dynamics. The idea involves designing a circuit that unitarily evolves the wave function over a time interval dt . Afterwards, a measurement is carried

out as illustrated in the circuit below



$$\begin{array}{c}
 \text{[Green Box]} = U(dt) \quad \text{[Yellow Diamond]} = \left\{ \begin{array}{l} | \quad (1-p) \\ \begin{array}{c} \text{[Diamond with 0]} \times p(\text{[Diamond with 0]}) \\ \text{[Diamond with 1]} \times p(\text{[Diamond with 1]}) \end{array} \end{array} \right.
 \end{array}
 \tag{5.57}$$

following this protocol

$$|\psi_n\rangle = \prod_{k=1}^n \left[\mathcal{M} \prod_{i=1}^{N-1} U_{i,i+1}(dt) \right] |\psi_0\rangle
 \tag{5.58}$$

where \mathcal{M} implements the projective measurement on each site.

Weak-measurements — A weak measurement is a measurement that extracts partial information from a quantum system. The traditional way to describe a quantum measurement, that is a von Neumann projective measurements, is to write the state of the system into the eigenstates of a given observable O , namely

$$|\psi\rangle = \sum_a c_a |a\rangle, \quad O = \sum_n o_a |a\rangle\langle a|.
 \tag{5.59}$$

This measurement completely projects ρ into an eigenstate $|a\rangle$ of the observable with probability $|c_a|^2$, thus extracting maximal information.

Conversely, generalized measurements, which extract partial information, are described in terms of POVMs, i.e. positive operator-valued measures. Consider a set of operators L_a such that $\sum_a L_a^\dagger L_a = \mathbb{1}$. The measurement process is described by transforming

$$\rho \rightarrow \rho_f = \frac{L_a \rho L_a^\dagger}{\text{Tr}\{L_a \rho L_a^\dagger\}} \quad (5.60)$$

with probability $P(a) = \text{Tr}\{L_a \rho L_a^\dagger\}$. Let us present the following simple model to provide a clear illustration of a weak projective measurement. Consider a two-level ancilla, represented by the eigenstates $\{|+\rangle, |-\rangle\}$ of the Pauli matrix σ_z , initially prepared in the state

$$|a\rangle = \frac{|+\rangle + |-\rangle}{\sqrt{2}}. \quad (5.61)$$

The ancilla is coupled to the system of interest, represented by the state $|\psi_t\rangle$. Let both the ancilla and the system evolve over a time Δt under the unitary evolution operator, $\hat{U}_{S+A}(\Delta t)$

$$\hat{U}_{S+A}(\Delta t) |\psi_t\rangle |a\rangle = (L_+ |\psi_t\rangle) |+\rangle + (L_- |\psi_t\rangle) |-\rangle \quad (5.62)$$

where $L_\pm = \langle \pm | \hat{U}_{S+A}(\Delta t) | a \rangle$ act exclusively on the system's Hilbert space. Following this evolution, a projective measurement acts on the ancilla along the z -axis, resulting in the outcome $a = \pm 1$. As a result, the back-action of the measurement places the system in the state

$$|\psi_{t+\Delta t}\rangle = \frac{L_a |\psi_t\rangle}{\sqrt{\langle \psi_t | L_a^\dagger L_a | \psi_t \rangle}}. \quad (5.63)$$

Let us now consider the continuous limit $\Delta t \rightarrow 0$. We have to derive the explicit form of the operators L_\pm . If we want to measure the observable X , let us consider the coupling between $S + A$ of the form

$$H_{S+A} = H + \lambda X \sigma_y. \quad (5.64)$$

We now take the limit $\Delta t \rightarrow 0$, scaling lambda in such a way that $\gamma = \lambda^2 \Delta t$ is kept constant. Expanding the propagator U we have

$$\begin{aligned} U_{S+A} &= e^{-i(\Delta t H + \sqrt{\gamma \Delta t} X \sigma_y)} \\ &= 1 - i \Delta t H - i \sqrt{\gamma \Delta t} X \sigma_y - \frac{1}{2} \gamma \Delta t O^2 + O(\Delta t^{3/2}) \end{aligned} \quad (5.65)$$

We thus obtain for the L_{\pm}

$$L_{\pm} = \frac{1}{\sqrt{2}} \left(1 - i\Delta t H \mp \sqrt{\gamma\Delta t} X - \frac{1}{2}\gamma\Delta t X^2 \right). \quad (5.66)$$

In order to compute the norm, we expand

$$L_a^\dagger L_a = \frac{1}{2} - a\sqrt{\gamma\Delta t} X, \quad (5.67)$$

while the corresponding probabilities become

$$P(a) = \frac{1}{2} - a\sqrt{\gamma\Delta t} \bar{X}. \quad (5.68)$$

It follows that

$$\begin{aligned} |\psi_{t+\Delta t}\rangle &= |\psi_t\rangle - iH\Delta t |\psi_t\rangle - a\sqrt{\gamma\Delta t}(X - \langle X\rangle) |\psi_t\rangle \\ &\quad + \frac{3}{2}\gamma\Delta t \langle X\rangle^2 |\psi_t\rangle - \gamma\Delta t \langle X\rangle X |\psi_t\rangle - \frac{1}{2}\gamma\Delta t X^2 |\psi_t\rangle \end{aligned} \quad (5.69)$$

Finally, we have to put this in the form of a stochastic equation. To this aim we observe that the measurement outcome a is a random variable that satisfies

$$\bar{a} = -2\sqrt{\gamma\Delta t} \langle X\rangle, \quad \overline{a^2} = 1. \quad (5.70)$$

Let us thus introduce $Y_t = \sqrt{\Delta t} \sum_{t' \leq t} a$: in the limit $\Delta t \rightarrow 0$ this converges to a continuous stochastic variable such that

$$dY = -2\sqrt{\gamma}\langle X\rangle dt + d\xi, \quad (5.71)$$

where ξ_t is a real Wiener process (i.e. $\overline{d\xi} = 0$ and $\overline{d\xi^2} = dt$). Replacing $a \rightarrow dY/\sqrt{\Delta t}$ and using (5.71) in the limit $\Delta t \rightarrow 0$, we recover

$$d|\psi\rangle = -iHdt |\psi_t\rangle + \left(\sqrt{\gamma}(X - \langle X\rangle)d\xi - \frac{\gamma}{2}(X - \langle X\rangle)^2 dt \right) |\psi_t\rangle \quad (5.72)$$

References of Chapter 5

- [Ben+19] G. Benenti et al. *Principles of Quantum Computation and Information: A Comprehensive Textbook*. World Scientific Publishing Co Pte Ltd, 2019. ISBN: 9789813236909.
- [BP02] H. P. Breuer and F. Petruccione. *The Theory of Open Quantum Systems*. Oxford: Oxford University Press, 2002.

- [Cue+13] G. D. las Cuevas et al. “Purifications of multipartite states: limitations and constructive methods.” In: *New Journal of Physics* 15.12 (2013), p. 123021. DOI: [10.1088/1367-2630/15/12/123021](https://doi.org/10.1088/1367-2630/15/12/123021).
- [GZ00] C. W. Gardiner and P. Zoller. *Quantum Noise*. Berlin: Springer, 2000.
- [KBM12] C. Karrasch, J. H. Bardarson, and J. E. Moore. “Finite-Temperature Dynamical Density Matrix Renormalization Group and the Drude Weight of Spin-1/2 Chains.” In: *Phys. Rev. Lett.* 108 (22 2012), p. 227206. DOI: [10.1103/PhysRevLett.108.227206](https://doi.org/10.1103/PhysRevLett.108.227206).
- [Man20] D. Manzano. “A short introduction to the Lindblad master equation.” In: *AIP Advances* 10.2 (2020), p. 025106. ISSN: 2158-3226. DOI: [10.1063/1.5115323](https://doi.org/10.1063/1.5115323).
- [Sch89] E. Schrödinger. *Statistical Thermodynamics*. Based on 1952 second edition from Cambridge University Press, p. 5 and p. 89. New York: Dover Publications, 1989.
- [SW10] E. M. Stoudenmire and S. R. White. “Minimally entangled typical thermal state algorithms.” In: *New Journal of Physics* 12.5 (2010), p. 055026. DOI: [10.1088/1367-2630/12/5/055026](https://doi.org/10.1088/1367-2630/12/5/055026).
- [VGC04] F. Verstraete, J. J. García-Ripoll, and J. I. Cirac. “Matrix Product Density Operators: Simulation of Finite-Temperature and Dissipative Systems.” In: *Phys. Rev. Lett.* 93 (20 2004), p. 207204. DOI: [10.1103/PhysRevLett.93.207204](https://doi.org/10.1103/PhysRevLett.93.207204).
- [Wer+16] A. H. Werner et al. “Positive Tensor Network Approach for Simulating Open Quantum Many-Body Systems.” In: *Phys. Rev. Lett.* 116 (23 2016), p. 237201. DOI: [10.1103/PhysRevLett.116.237201](https://doi.org/10.1103/PhysRevLett.116.237201).
- [Whi09] S. R. White. “Minimally Entangled Typical Quantum States at Finite Temperature.” In: *Phys. Rev. Lett.* 102 (19 2009), p. 190601. DOI: [10.1103/PhysRevLett.102.190601](https://doi.org/10.1103/PhysRevLett.102.190601).
- [ZV04] M. Zwolak and G. Vidal. “Mixed-State Dynamics in One-Dimensional Quantum Lattice Systems: A Time-Dependent Superoperator Renormalization Algorithm.” In: *Phys. Rev. Lett.* 93 (20 2004), p. 207205. DOI: [10.1103/PhysRevLett.93.207205](https://doi.org/10.1103/PhysRevLett.93.207205).

Chapter 6

Tensor networks and quantum magic

*“But I don’t want to go among mad people” Alice remarked.
“Oh, you can’t help that” said the Cat:
“We’re all mad here. I’m mad. You’re mad.”*

Lewis Carroll

Due to their unique characteristics and wide-ranging applications, the Clifford group and stabilizer states play a crucial role in quantum information theory. In quantum error correction, for instance, they form a robust framework that enables efficient error detection and correction, enhancing the stability and reliability of quantum computations. Recent advancements have further expanded the importance of Clifford unitaries, leading to a variety of applications in quantum computation and the learning of unknown quantum states.

One of the most striking features of stabilizer states is that, despite their potential extensive entanglement, they can be described and manipulated using classical computational resources. This classical tractability is attributed to the fact that operations within the Clifford group, when applied to stabilizer states, do not increase the computational complexity beyond classical limits.

In this chapter, we present a concise theoretical introduction to the stabilizer formalism and the Clifford group, focusing on the concept of magic, which is the quantum resource associated with the use of non-Clifford gates (or non-stabilizer states). We will then explore how a certain measure of

magic, known as Stabilizer Rényi Entropies, can be computed for MPS states. Finally, we will discuss some recent proposals to incorporate the stabilizer formalism into Tensor Networks.

6.1 Introduction to Stabilizer Formalism and Magic

In this section, we introduce key concepts from the stabilizer formalism, a framework essential for understanding quantum error correction, efficient classical simulation of quantum systems, and the characterization of quantum resources beyond classical reach. These concepts underpin modern approaches to quantum computation and are fundamental for exploring the structure of quantum states.

6.1.1 Pauli group and stabilizer groups

Definition 6.1: Pauli group

The Pauli group \mathcal{P}_1 is the 16–elements group consisting of the following 2×2 matrices

$$\mathcal{P}_1 = \{\pm 1, \pm i1, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}.$$

The Pauli group \mathcal{P}_N is the group obtained by taking the tensor product of any N elements of \mathcal{P}_1 , i.e.

$$\mathcal{P}_N = \{\pm 1, \pm i1, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}^{\otimes N}.$$

Below is a list of key properties of Pauli operators. Here and in the following subsections, we use the symbols g (or h) to represent a generic Pauli string and highlight that these are elements of a group.

- The elements of \mathcal{P}_N are Hermitian ($g^\dagger = g$) or anti-Hermitian ($g^\dagger = -g$)
- The elements of \mathcal{P}_N have eigenvalues ± 1 (if Hermitian), or $\pm i$ (if anti-Hermitian)
- If $g \in \mathcal{P}_N$, then $g^2 = 1$ (if g is Hermitian) or $g^2 = -1$ (if g is anti-Hermitian)
- The elements of \mathcal{P}_N are unitary
- If $g, h \in \mathcal{P}_N$, then either they commute $[g, h] = 0$ or they anti-commute $\{g, h\} = 0$
- With the exception of $\pm 1, \pm i1$, the elements of \mathcal{P}_N are traceless

Definition 6.2: Stabilizers

We define *stabilizer group* any subgroup of \mathcal{P}_N that does not contain the element -1 . Given a stabilizer group \mathcal{S} , we define $V_{\mathcal{S}}$ as the set of N -qubits states $|\psi\rangle$ such that $g|\psi\rangle = |\psi\rangle, \forall g \in \mathcal{S}$ and $\forall |\psi\rangle \in V_{\mathcal{S}}$. $V_{\mathcal{S}}$ is by construction a subspace. It is called *stabilizer subspace*. If it is one-dimensional, it defines a single state called *stabilizer state*.

Notice that in order to have a non trivial subspace $V_{\mathcal{S}}$ it is necessary to have $-1 \notin \mathcal{S}$. Indeed otherwise we would have: $(-1)|\psi\rangle = |\psi\rangle, \forall |\psi\rangle \in V_{\mathcal{S}}$, which implies $V_{\mathcal{S}} = \{0\}$. Moreover, it is important to notice that any stabilizer group is an *abelian group*. Indeed, let us suppose that $\exists g, h \in \mathcal{S}$ such that $[g, h] \neq 0$. Because of the observation e), we have $\{g, h\} = 0 \Rightarrow gh = -hg$. Now, since $gh \in \mathcal{P}_N$, it can be either $(gh)^2 = 1$ or $(gh)^2 = -1$. In the second case, we can conclude since we break the initial hypothesis. In the first case, we have $(gh)(hg) = -(gh)(gh) = -(gh)^2 = -1$, and therefore we find again the absurd.

Thus, we conclude stabilizers are abelian groups of the Pauli group, where all elements are Hermitian and square to the identity. Therefore, any stabilizer group \mathcal{S} is generated by an independent set of k (commuting) elements of the Pauli group $g_1, \dots, g_k \in \mathcal{P}_N$, meaning that

$$\mathcal{S} = \left\{ \prod_{j=1}^k (g_j)^{n_j}, n_j \in \{0, 1\} \right\} = \langle g_1, \dots, g_k \rangle$$

The size of \mathcal{S} is therefore $|\mathcal{S}| = 2^k$.

Definition 6.3: Stabilizer projector

Given a stabilizer group \mathcal{S} , we define the operator

$$P_{\mathcal{S}} = \frac{1}{|\mathcal{S}|} \sum_{g \in \mathcal{S}} g .$$

It can be easily proven that $P_{\mathcal{S}}$ is nothing but the orthogonal projection on $V_{\mathcal{S}}$. In fact, given $g \in \mathcal{S}$, $P_g = (\mathbb{1} + g)/2$ is the orthogonal projector on its eigenspace with eigenvalue $+1$. The projector on \mathcal{S} is therefore

$$P_{\mathcal{S}} = \prod_{g \in \{g_1, \dots, g_k\}} P_g = \frac{1}{2^k} \sum_{\{n\}} \prod_{j=1}^k (g_j)^{n_j} = \frac{1}{|\mathcal{S}|} \sum_{g \in \mathcal{S}} g .$$

Let $|\psi\rangle$ be a stabilizer state. Its density matrix operator $\rho = |\psi\rangle\langle\psi|$ is the orthogonal projection on $|\psi\rangle$. Thus

$$\hat{\rho} = P_S = \frac{1}{|S|} \sum_{g \in S} g. \quad (6.1)$$

Lemma 6.1: Dimension of the stabilizer group

The dimension of the stabilizer subspace V_S of a stabilizer group generated by k elements of \mathcal{P}_N is 2^{N-k} . Indeed, since P_S is an orthogonal projection its eigenvalues are 0 or 1 and its trace is equal to the number of 1 eigenvalues, i.e. to the dimension of V_S . We have $\text{Tr}[P_S] = 2^{-k} \sum_{g \in S} \text{Tr}[g]$ and, since $\text{Tr}[g]$ is non zero only if $g = 1$, $\text{Tr}[P_S] = 2^{-k} 2^N$.

From this lemma, we conclude that the stabilizer group of an N -qubit stabilizer state is generated by exactly N mutually commuting Pauli operators. The next question is whether stabilizer states span the entire N -qubit Hilbert space. The answer is no. In fact, since \mathcal{P}_N has finite cardinality, the number of N mutually commuting Pauli operators is also finite. For instance, the stabilizer states for $N = 1$ are just six, namely

$$|0\rangle, |1\rangle, \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \frac{|0\rangle - |1\rangle}{\sqrt{2}}, \frac{|0\rangle + i|1\rangle}{\sqrt{2}}, \frac{|0\rangle - i|1\rangle}{\sqrt{2}}.$$

In general, it can be shown that the cardinality of the set of pure stabilizer states is $|\text{Stab}| = 2^N \prod_{k=0}^{N-1} (2^{N-k} + 1)$.

6.1.2 The Clifford group

Definition 6.4: Clifford group

The Clifford group \mathcal{C}_N is the group of N -qubits unitaries that map elements of the Pauli group into elements of the Pauli group

$$\mathcal{C}_N = \{U \in \mathcal{U}_{2^N \times 2^N} \text{ s.t. } U \mathcal{P}_N U^\dagger = \mathcal{P}_N\}.$$

It can be shown that the Hadamard gate H , the phase gate S and the CNOT,¹ i.e.

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \boxed{\text{H}} \quad \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{pmatrix} = \boxed{\text{S}}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \text{CNOT} \tag{6.2}$$

generate the entire Clifford group \mathcal{C}_N . Nevertheless, another gate is necessary to construct an universal set of gates, i.e. to span all the N -qubits unitaries. This is usually chosen to be

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} = \boxed{\text{T}} \tag{6.3}$$

which is non Clifford, since for instance $T\hat{X}T^\dagger = \frac{1}{\sqrt{2}}(\hat{X} + \hat{Y})$.

Importantly, the uniform distribution on the Clifford group is a 3–design, meaning that the average of any third order polynomial over \mathcal{C}_N equals the average over the entire unitary group U_N . Consequently, Clifford group can be used for the simulation of random unitary circuits, allowing the extraction of information up to the third order moments.

The following result gives a connection between the Clifford group and the set of stabilizer states.

Lemma 6.2: Preparation of Stabilizer states

A state $|\psi\rangle$ is a stabilizer if and only if it is equal to $C|0\rangle^{\otimes N}$, for some $C \in \mathcal{C}_N$. The second implication is very easy to show. Indeed, $|0\rangle^{\otimes N}$ is clearly a stabilizer state having as generators

$$g_1 = \hat{Z} \otimes 1 \otimes \dots \otimes 1$$

$$g_2 = 1 \otimes \hat{Z} \otimes \dots \otimes 1$$

$$\dots$$

$$g_N = 1 \otimes 1 \otimes \dots \otimes \hat{Z}.$$

Since the group $C \langle g_1, g_2 \dots g_N \rangle C^\dagger$ stabilizes $|\psi\rangle = C|0\rangle^{\otimes N}$, we conclude that $|\psi\rangle$ is a stabilizer. The converse can also be shown easily.

¹CNOT can be replaced with controlled \hat{Z} .

Consider now any stabilizer subspace $V_{\mathcal{S}}$ and an unitary operator $U \in \mathcal{C}_N$. The action of U maps $V_{\mathcal{S}}$ to $UV_{\mathcal{S}}$. It is not difficult to realize that $UV_{\mathcal{S}}$ is the stabilizer subspace for the group USU^\dagger . Indeed, $\forall g' = UgU^\dagger, \forall |\psi'\rangle = U|\psi\rangle \in UV_{\mathcal{S}}$ we have $g'|\psi'\rangle = g'U|\psi\rangle = UgU^\dagger U|\psi\rangle = Ug|\psi\rangle = U|\psi\rangle = |\psi'\rangle$. This means that we can keep track of the action of any Clifford unitary simply by applying it to the generator set of \mathcal{S} .

6.1.3 The tableau representation

A simple but useful mapping exists between elements of \mathcal{P}_1 and the binary vector space $(\mathbb{Z}_2)^2$. The mapping is given by the following tableau

$$\begin{cases} r(\mathbb{1}) = (0|0) \\ r(\hat{X}) = (1|0) \\ r(\hat{Y}) = (1|1) \\ r(\hat{Z}) = (0|1) \end{cases}$$

and neglects the overall phases ± 1 or $\pm i$ (thus, $r(-i\hat{X}) = r(\hat{X})$). It is not difficult to show that $r(gh) = r(g) \oplus r(h)$, \oplus denoting the sum modulo 2. The binary representation can be easily generalized to \mathcal{P}_N . Indeed, if we split $\mathbf{r}(g)$ as $(\mathbf{r}^{(1)}(g) | \mathbf{r}^{(2)}(g))$, where $\mathbf{r}^{(1)}$ and $\mathbf{r}^{(2)}$ are vectors of length N , we can assign the value 1 to the j th component of $\mathbf{r}^{(1)}(g)$ ($j = 1, 2 \dots N$) iff the j th Pauli matrix in g is \hat{X} or \hat{Y} , and the value 1 to the j th component of $\mathbf{r}^{(2)}(g)$ iff the j th Pauli matrix in g is \hat{Y} or \hat{Z} . Furthermore, if we define the matrix

$$\Lambda_N = \begin{pmatrix} \mathbf{0}_{N \times N} & \mathbf{1}_{N \times N} \\ \mathbf{1}_{N \times N} & \mathbf{0}_{N \times N} \end{pmatrix}$$

of dimension $2N \times 2N$, we can realize that the symplectic inner product of two binary vectors evaluates whether the matrices commute or anticommute

$$(\mathbf{r}(g))^T \Lambda_N \mathbf{r}(h) = \begin{cases} 0 & \text{if } g, h \text{ commute} \\ 1 & \text{if } g, h \text{ anticommute} \end{cases} .$$

Definition 6.5: Generator matrix

Given a stabilizer state, fully specified by $\mathcal{S} = \langle g_1, g_2 \dots g_N \rangle$, its *generator matrix* is the $N \times 2N$ matrix

$$G = \begin{pmatrix} \mathbf{r}(g_1) \\ \mathbf{r}(g_2) \\ \dots \\ \mathbf{r}(g_N) \end{pmatrix}.$$

This matrix only specifies the generators up to an overall phase.

Notice that the stabilizer generators must be independent, meaning that no product of them can produce 1 . Since $\mathbf{r}(g_i g_j) = \mathbf{r}(g_i) \oplus \mathbf{r}(g_j)$, no rows of G can sum to zero and therefore G must have full rank over the finite field \mathbb{Z}_2 . Furthermore, since generators must commute they must satisfy $G^T \Lambda_N G = 0$.

Let us notice that the Hadamard gate H and the phase gate S act as follows on the Pauli matrices

$$\begin{cases} H\hat{X}H^\dagger = \hat{Z} \\ H\hat{Y}H^\dagger = -\hat{Y} \\ H\hat{Z}H^\dagger = \hat{X} \end{cases} \quad \begin{cases} S\hat{X}S^\dagger = \hat{Y} \\ S\hat{Y}S^\dagger = -\hat{X} \\ S\hat{Z}S^\dagger = \hat{Z} \end{cases}$$

In other words, H interchanges \hat{X} and \hat{Z} , giving \hat{Y} a phase, and S interchanges \hat{X} and \hat{Y} , possibly adding a phase, and leaves \hat{Z} unchanged. In the tableau representation, this means:

$$\begin{aligned} \left(\dots r_j^{(1)} \dots \mid \dots r_j^{(2)} \dots \right) &\xleftrightarrow{H_j} \left(\dots r_j^{(2)} \dots \mid \dots r_j^{(1)} \dots \right) \\ \left(\dots r_j^{(1)} \dots \mid \dots r_j^{(2)} \dots \right) &\xleftrightarrow{S_j} \left(\dots r_j^{(1)} \dots \mid \dots (r_j^{(1)} \oplus r_j^{(2)}) \dots \right). \end{aligned}$$

Regarding the CNOT gate, we have ($\text{CNOT} = \text{CNOT}^\dagger$)

$$\begin{cases} \text{CNOT}(1 \otimes 1) \text{CNOT} = 1 \otimes 1 \\ \text{CNOT}(1 \otimes \hat{X}) \text{CNOT} = 1 \otimes \hat{X} \\ \text{CNOT}(1 \otimes \hat{Y}) \text{CNOT} = \hat{Z} \otimes \hat{Y} \\ \text{CNOT}(1 \otimes \hat{Z}) \text{CNOT} = \hat{Z} \otimes \hat{Z} \\ \text{CNOT}(\hat{X} \otimes 1) \text{CNOT} = \hat{X} \otimes \hat{X} \\ \text{CNOT}(\hat{X} \otimes \hat{X}) \text{CNOT} = \hat{X} \otimes 1 \\ \text{CNOT}(\hat{X} \otimes \hat{Y}) \text{CNOT} = \hat{Y} \otimes \hat{Z} \\ \text{CNOT}(\hat{X} \otimes \hat{Z}) \text{CNOT} = -\hat{Y} \otimes \hat{Y} \end{cases} \quad \begin{cases} \text{CNOT}(\hat{Y} \otimes 1) \text{CNOT} = \hat{Y} \otimes \hat{X} \\ \text{CNOT}(\hat{Y} \otimes \hat{X}) \text{CNOT} = \hat{Y} \otimes 1 \\ \text{CNOT}(\hat{Y} \otimes \hat{Y}) \text{CNOT} = -\hat{X} \otimes \hat{Z} \\ \text{CNOT}(\hat{Y} \otimes \hat{Z}) \text{CNOT} = \hat{X} \otimes \hat{Y} \\ \text{CNOT}(\hat{Z} \otimes 1) \text{CNOT} = \hat{Z} \otimes 1 \\ \text{CNOT}(\hat{Z} \otimes \hat{X}) \text{CNOT} = \hat{Z} \otimes \hat{X} \\ \text{CNOT}(\hat{Z} \otimes \hat{Y}) \text{CNOT} = 1 \otimes \hat{Y} \\ \text{CNOT}(\hat{Z} \otimes \hat{Z}) \text{CNOT} = 1 \otimes \hat{Z} \end{cases}$$

This table can be summarized in following update rule for a CNOT where the qubit k control the target qubit j ($j, k \in \{1, 2, \dots, N\}$)

$$\left(\dots r_j^{(1)} \oplus r_k^{(1)} \dots r_k^{(1)} \dots \mid \dots r_j^{(2)} \dots r_k^{(2)} \oplus r_j^{(2)} \dots \right) \xleftrightarrow{\text{CNOT}_{kj}} \left(\dots r_j^{(1)} \dots r_k^{(1)} \dots \mid \dots r_j^{(2)} \dots r_k^{(2)} \dots \right).$$

As a simple example, let us consider a system with $N = 2$ qubits. The initial state is $|00\rangle$, which has $\{\mathbb{1}_1 \hat{Z}_2, \hat{Z}_1 \mathbb{1}_2\}$ as stabilizer generators. The stabilizer matrix is therefore

$$G = \left(\begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right).$$

If we apply the gate H on the first qubit we can update the matrix G as follows

$$G' = \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right).$$

Then, we apply a CNOT, that gives

$$G'' = \left(\begin{array}{cc|cc} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{array} \right).$$

Indeed, we have

$$\text{CNOT}_{12}(H_1 |00\rangle) = \text{CNOT}_{12} \frac{|00\rangle + |10\rangle}{\sqrt{2}} = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

and the final state has $\{\hat{X}_1 \hat{X}_2, \hat{Z}_1 \hat{Z}_2\}$ as stabilizer generators, which are exactly the operators encoded in the tableau G'' .

Other simple rules can be found to keep track of the overall sign ± 1 and to update the generator matrix after a Pauli measurements. These observations lead to the following important Theorem [NC10].

Theorem 6.1: Gottesman-Knill

Suppose we perform a quantum computation which involves only state preparations in the computational basis, Hadamard gates H , phase gates S , CNOT gates, Pauli gates and measurements of observables in the Pauli group, together with the possibility of classical control conditioned on the outcome of such measurements. Such a computation may be *efficiently* simulated on a classical computer, meaning that there exists an Algorithm to do this classically in a polynomial time, specifically at cost $\mathcal{O}(N^2 M)$ operations, where N is the number of qubits and M the number of operations.

6.1.4 Entanglement

Let us suppose to have a pure stabilizer state $\rho = |\psi\rangle\langle\psi| = \frac{1}{2^N} \sum_{g \in \mathcal{S}} g$. We aim to assess the entanglement entropy $S_{A/B}$ associated to the partitioning of the system into a subsystem A consisting of N_A qubits and a complementary subsystem B containing $N_B = N - N_A$ qubits. We start by computing the reduced density matrix $\rho_A = \text{Tr}_B[\rho]$:

$$\hat{\rho}_A = \frac{1}{2^N} \sum_{g \in \mathcal{S}} \text{Tr}_B[g].$$

Only some elements g of the group \mathcal{S} give a non vanishing contribution, namely the Pauli strings that can be written as $g = g_A \otimes \mathbb{1}_B$. Notice that this subset \mathcal{S}_A is also a stabilizer group, since $(g_A \otimes \mathbb{1}_B)(g'_A \otimes \mathbb{1}_B) = (g_A g'_A) \otimes \mathbb{1}_B$. Let us denote k_A the number of independent generators of \mathcal{S}_A . Now we have

$$\hat{\rho}_A = \frac{1}{2^N} 2^{N_B} \sum_{g \in \mathcal{S}_A} g = \frac{1}{2^{N_A}} 2^{k_A} \frac{1}{2^{k_A}} \sum_{g \in \mathcal{S}_A} g.$$

The last part of the expression is the projector on the stabilizer subspace of \mathcal{S}_A , thus

$$\hat{\rho}_A = \frac{1}{2^{N_A - k_A}} P_{\mathcal{S}_A}. \quad (6.4)$$

Now, since eigenvalues of $P_{\mathcal{S}_A}$ are only 1 (with degeneracy $2^{N_A - k_A}$) and 0 (with degeneracy $2^{N_A} - 2^{N_A - k_A}$), we get

$$S_{A/B} = -2^{N_A - k_A} \frac{1}{2^{N_A - k_A}} \log \frac{1}{2^{N_A - k_A}} = (N_A - k_A) \log 2.$$

How to compute k_A in practice given the generator matrix G of $|\psi\rangle$? It is not difficult to show that k_A is simply the rank of the matrix obtained by setting to 0 all the entries of G corresponding to the subsystem B . Indeed, this operation corresponds to projecting the original generators on the space of operators having the form $g_A \otimes \mathbb{1}_B$.

6.1.5 Nonstabilizerness as a quantum resource

As we saw, a quantum computation involving only stabilizer states (or equivalently, Clifford circuits and measurements in the computational basis) can be simulated classically. Therefore, one has to consider nonstabilizer protocols to reach a true quantum advantage. In this sense, the “nonstabilizerness” can be considered as a key “resource” of the quantum world, making it more powerful of the classical one. This intuitive picture can be made mathematically

rigorous by exploiting the meta-theory of quantum resources. In this framework, one usually introduces a set of “free-operations” that by construction do not generate resource. In the case of entanglement these are the Local Operations and classical Communication (see Definition 2.5). In the case of the nonstabilizerness, also dubbed *quantum magic*, they are the following stabilizer operations:²

- Clifford unitaries, i.e. $\hat{\rho} \rightarrow \hat{U}\hat{\rho}\hat{U}^\dagger$, with $\hat{U} \in \mathcal{C}_N$
- Composition with stabilizer states, i.e. $\hat{\rho} \rightarrow \hat{\rho} \otimes \hat{\rho}_S$, where $\hat{\rho}_S$ is a stabilizer state
- Measurements in the computational basis, i.e. $\hat{\rho} \rightarrow \sum_k \hat{P}_k \hat{\rho} \hat{P}_k$, where \hat{P}_k are the projectors on the computational basis ($\sum_k \hat{P}_k = \mathbb{1}$)
- Discarding some qubits, i.e. $\hat{\rho} \rightarrow \text{Tr}_A[\hat{\rho}]$, where A is any subsystem
- The above operations conditioned on the outcomes of a measurement

The goal is typically to quantify the resources stored in a state $\hat{\rho}$ using a measure $\mathcal{M}(\hat{\rho})$, often referred to as a *monotone*. This measure must not increase under arbitrary free operations. Specifically, \mathcal{M} is considered a *magic monotone* if and only if it satisfies the condition $\mathcal{M}(\mathcal{E}(\hat{\rho})) \leq \mathcal{M}(\hat{\rho})$ for any quantum channel \mathcal{E} composed of the stabilizer operations listed above. If a monotone \mathcal{M} can be identified and calculated, it can be used to distinguish between states that can or cannot be prepared from an initial state using free operations. The resource needed to prepare a state can be injected in the initial state, for instance having access to many copies of the *magic state*

$$|T\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\frac{\pi}{4}}|1\rangle).$$

Examples of known genuine magic monotones are the following.

1. The robustness of magic [HC17; HG19], defined as

$$R(\rho) = \min \left\{ \sum_{\alpha=1}^{|\text{Stab}|} |x_\alpha| \text{ s.t. } \hat{\rho} = \sum_{\alpha} x_\alpha \hat{\rho}_S^{(\alpha)} \right\}, \quad (6.5)$$

where $\rho_S^{(\alpha)} = |\psi^{(\alpha)}\rangle\langle\psi^{(\alpha)}|$, $\alpha = 1, 2 \dots |\text{Stab}|$, are all the pure stabilizer states over N qubits.

²Sometimes an additional condition, known as *strong monotonicity*, is also imposed. It can be expressed as $\sum_k p_k \mathcal{M}(\hat{P}_k \hat{\rho} \hat{P}_k) \leq \mathcal{M}(\hat{\rho})$ and it ensures that \mathcal{M} does not increase, on average, when the experimenter has the ability to post-select multiple outcomes of a quantum measurement.

2. The min-relative entropy of magic [Vei+14], which for pure states reads

$$D(|\psi\rangle) = -\log\left(\max_{\alpha} |\langle\psi^{(\alpha)}|\psi\rangle|^2\right), \quad (6.6)$$

with the same notation as above.

However, both these quantity are very difficult to handle in practice, both in experiments and in numerical simulations, since their evaluation requires an optimization over an exponentially large space.

6.1.6 Stabilizer Rényi Entropies

In this Section, we will introduce the Stabilizer Rényi Entropies [LOH22], a simpler way of measuring the quantum magic. First, we introduce the projective Pauli group.

Definition 6.6: Projective Pauli group

The projective Pauli group is the standard Pauli group \mathcal{P}_N modulo global phases, i.e. $\tilde{\mathcal{P}}_N = \{\mathbb{1}, \hat{X}, \hat{Y}, \hat{Z}\}^{\otimes N}$.

From now on, we will indicate as $\hat{\sigma}$ a generic Pauli string belonging to $\tilde{\mathcal{P}}_N$.

Definition 6.7: Stabilizer Rényi Entropies

Given a N -qubits state $|\psi\rangle$ and a Rényi index $n > 0$, we define the *Stabilizer Rényi Entropy* (SRE) as

$$M_n(|\psi\rangle) = \frac{1}{1-n} \log\left(\sum_{\hat{\sigma} \in \tilde{\mathcal{P}}_N} \frac{1}{2^N} \langle\psi|\hat{\sigma}|\psi\rangle^{2n}\right). \quad (6.7)$$

To understand the relation with usual Rényi entropies, it is useful to introduce the density matrix $\hat{\rho} = |\psi\rangle\langle\psi|$ and the function $\Pi_{\rho}(\hat{\sigma}) = \frac{1}{2^N}(\text{Tr}[\hat{\rho}\hat{\sigma}])^2$. Let us observe that

$$\begin{aligned} \sum_{\hat{\sigma}} \Pi_{\rho}(\hat{\sigma}) &= \frac{1}{2^N} \sum_{\hat{\sigma}} \text{Tr}[\hat{\rho}\hat{\sigma}] \cdot \text{Tr}[\hat{\rho}\hat{\sigma}] = \\ &= \text{Tr}\left[\hat{\rho} \cdot \frac{1}{2^N} \sum_{\hat{\sigma}} \text{Tr}[\hat{\rho}\hat{\sigma}]\hat{\sigma}\right] = \text{Tr}[\hat{\rho}^2] = 1. \end{aligned}$$

We used the decomposition of the matrix ρ in terms of $\{\frac{\hat{\sigma}}{\sqrt{2^N}}\}_{\hat{\sigma} \in \tilde{\mathcal{P}}_N}$, which is a complete base set for the Hermitian matrices of size 2^N . We

also use the fact that for pure states $\text{Tr}[\rho^2] = 1$. Thus, we can consider $\Pi_\rho(\hat{\sigma})$ as a probability distribution. We have

$$M_n(|\psi\rangle) = \frac{1}{1-n} \log \left(\sum_{\hat{\sigma}} (\Pi_\rho(\hat{\sigma}))^n \right) - N \log 2, \quad (6.8)$$

that shows that $M_n(|\psi\rangle)$ is the n -Rényi entropy of $\Pi_\rho(\hat{\sigma})$, apart from a constant. Notice that, to extend the definition of M_n to arbitrary density matrices ρ , we have to normalize with the purity, i.e.

$$\Pi_\rho(g) = \frac{1}{2^N} \frac{(\text{Tr}[\rho \hat{\sigma}])^2}{\text{Tr}[\rho^2]}. \quad (6.9)$$

- a) $M_n(|\psi\rangle) = 0$ for stabilizer states. Indeed, it is easy to show that $\langle \psi | \hat{\sigma} | \psi \rangle = +1$ if $\hat{\sigma}$ belongs to the stabilizer group \mathcal{S} of $|\psi\rangle$, $\langle \psi | \hat{\sigma} | \psi \rangle = -1$ if $-\hat{\sigma} \in \mathcal{S}$, and $\langle \psi | \hat{\sigma} | \psi \rangle = 0$ in all other cases. Consequently, if $|\psi\rangle$ is a stabilizer state, $\langle \psi | \hat{\sigma} | \psi \rangle^{2n} = 1$ for 2^N strings, while it is 0 for the others, and $\sum_{\hat{\sigma}} \frac{1}{2^N} \langle \psi | \hat{\sigma} | \psi \rangle^{2n} = 1$.
- b) In general, $M_n(|\psi\rangle) = 0$ iff $|\psi\rangle$ is a stabilizer and $M_n(|\psi\rangle) > 0$ otherwise.
- c) The SRE are invariant under Clifford unitaries. Indeed, even if the Pauli strings are reshuffled by a Clifford unitary, the probability values $\Pi_\rho(\hat{\sigma})$ remain unchanged and consequently also the values of the Rényi entropies.
- d) The SRE are additive, i.e. $M_n(|\psi\rangle \otimes |\phi\rangle) = M_n(|\psi\rangle) + M_n(|\phi\rangle)$. This is a trivial consequence of the fact that the sum over Pauli strings on a system can be factorized into the sum of the Pauli strings of two subsystems.

These properties have recently attracted a lot of attention to SREs, considered as a potential simpler monotone for magic. However, it has been shown in ref. [HP23b] that, for Rényi indices in the range $0 \leq n < 2$, Stabilizer Rényi Entropies (SREs) fail to behave as monotones under stabilizer protocols involving measurements in the computational basis, even when considering pure states. Additionally, for any Rényi index, SREs do not adhere to the strict monotonicity criteria with respect to computational-basis measurements. Only recently, it has been definitively established in ref. [LB24] that stabilizer entropies are monotonic for $n \geq 2$ within the framework of magic-state resource theory, specifically for pure states. Furthermore, it has

been demonstrated that linear stabilizer entropies act as strong monotones in this context. Additionally, these stabilizer entropies have been extended to mixed states as valid magic monotones using convex roof constructions, ensuring their applicability beyond pure-state settings as well.

6.2 Measuring the magic of Tensor Network states

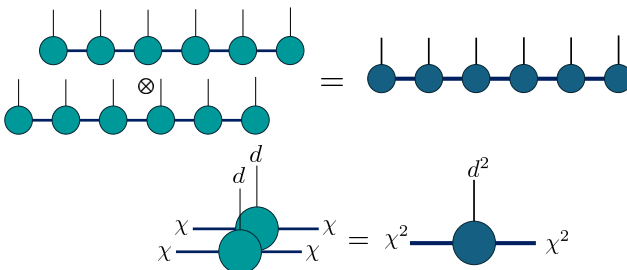
In this section, we explore various approaches to quantify the magic in tensor network states, with a particular emphasis on Matrix Product States (MPS). Among these, Stabilizer Rényi Entropies (SREs) stand out as the most viable option for tensor network representations. This is because, unlike other proposed measures of magic, SREs can be efficiently computed using tensor network techniques, making them especially suitable for large-scale many-body systems. While alternative methods have been explored, they tend to encounter significant computational challenges in the tensor network framework, making SREs the preferred tool for assessing nonstabilizerness in this setting — at least based on the current understanding and available techniques.

6.2.1 Replica methods

Replica methods are based on the simple fact that the expectation value of an operator to the power k , with k an integer number, can be seen as the expectation value of a replicated operator over a replicated state. For instance;

$$\langle \psi | \hat{\sigma} | \psi \rangle^k = \langle \psi | \hat{\sigma}^{\otimes k} (|\psi\rangle)^{\otimes k} \rangle. \quad (6.10)$$

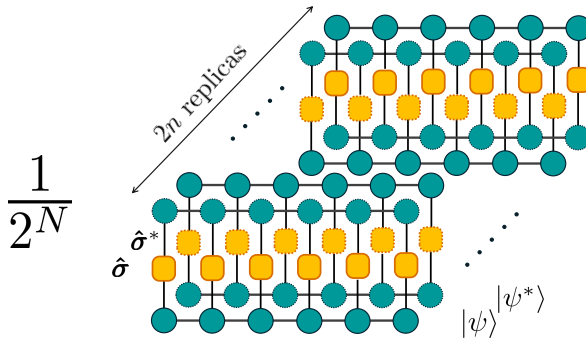
Following ref. [HP23a], we will explore how these methods can be applied to evaluate the SRE (we will restrict to integer $n > 1$). Notice that the number of replicas k equals $2n$. Now, if $|\psi\rangle$ is an MPS with bond dimension χ , the replicated state $|\psi\rangle^{\otimes k}$ is also an MPS with larger bond dimension $\chi' = \chi^k$ and physical dimension $d' = d^k$ ($d = 2$ for qubits). Indeed, one can simply merge together the k replicated MPS, as shown here for $k = 2$:



For the evaluation of the SRE, it is technically convenient to rewrite n of the $2n$ real-valued expectation values by taking their complex conjugates, i.e. $\langle \psi^* | \hat{\sigma}^* | \psi^* \rangle$. In this way, we can write the argument of the log in the SRE definition as

$$\frac{1}{2^N} \sum_{\sigma} (\langle \psi, \psi^* |)^{\otimes n} (\hat{\sigma} \otimes \hat{\sigma}^*)^{\otimes n} (|\psi, \psi^* \rangle)^{\otimes n}, \tag{6.11}$$

Which is graphically represented as follows



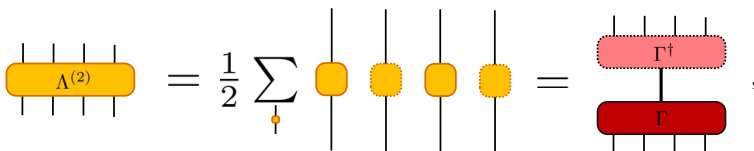
where we used dotted, lighter shapes for the complex conjugate tensors. It is now convenient to introduce the following object

$$\Lambda_j^{(n)} = \frac{1}{2} \sum_{\mu=0}^3 (\hat{\sigma}_j^{\mu} \otimes \hat{\sigma}_j^{\mu,*})^{\otimes n}, \tag{6.12}$$

which, if thought as a matrix, has shape $2^{2n} \times 2^{2n}$. One can easily show that $\Lambda^{(n)}$ is a positive matrix with rank 2^{2n-2} . Therefore, one can decompose $\Lambda^{(n)}$ as

$$\Lambda^{(n)} = \Gamma^{\dagger} \Gamma, \tag{6.13}$$

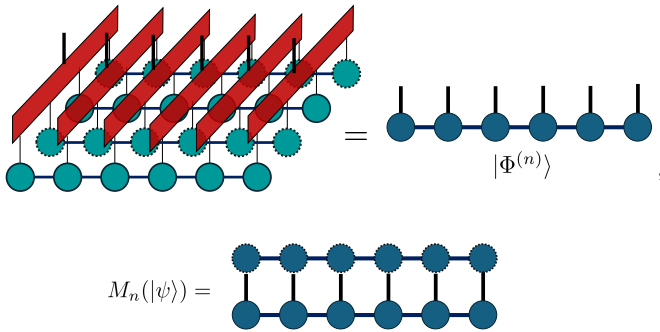
where Γ is a rectangular matrix of shape $2^{2n-2} \times 2^{2n}$. These steps are illustrated below for $n = 2$:



Now the Γ_j matrices can be incorporated in the MPS $(|\psi, \psi^*\rangle)^{\otimes n}$ by letting them act locally on the (replicated) physical spaces. The result is a new state $|\Phi^{(n)}\rangle = \left(\bigotimes_{j=1}^N \Gamma_j\right) (|\psi, \psi^*\rangle)^{\otimes n}$. Finally, the SRE can be computed by taking the state norm of $|\Phi^{(n)}\rangle$, namely

$$M_n(|\psi\rangle) = \frac{1}{1-n} \log \left(\langle \Phi^{(n)} | \Phi^{(n)} \rangle \right). \tag{6.14}$$

Last steps can be graphically understood as follows



The norm of the MPS $|\Phi^{(n)}\rangle$ in Eq. (6.14) can be computed exactly at a cost $\mathcal{O}(N\chi'^3 d') = \mathcal{O}(N\chi^{6n} d^{2n-2})$. This cost is often prohibitively high, but may be manageable for $n = 2, 3$ and small bond dimension χ . In particular, for $n = 2$ one can exploit a certain symmetry of Γ to reduce the computational cost (see [HP23a] for details).

Example 6.1: Replica Pauli-MPS

To efficiently calculate Stabilizer Rényi Entropies (SREs) using tensor networks, we can leverage the representation of quantum states $\hat{\rho} = |\psi\rangle\langle\psi|$ in the Pauli basis (using the density-matrix formalism of Chapter 5). This allows us to store Pauli expectation values compactly and compute them efficiently [Tar+24].

If a state is originally represented by a Matrix Product State (MPS) with bond dimension χ , its equivalent representation in the Pauli basis

as in Eq. (5.24), will have a bond dimension of χ^2 :

$$|\psi\rangle\langle\psi| = \sum_{\mu_1, \dots, \mu_N} \begin{array}{c} P_1 - P_2 - \dots - P_N \\ | \quad | \quad \quad \quad | \\ \mu_1 \quad \mu_2 \quad \quad \quad \mu_N \end{array} \hat{\zeta}^{\mu_1} \otimes \hat{\zeta}^{\mu_2} \otimes \dots \otimes \hat{\zeta}^{\mu_N}$$

$$\alpha - \begin{array}{c} P_j \\ | \\ \mu_j \end{array} - \beta = \alpha \left\{ \begin{array}{c} A_2^* \\ A_2 \\ \zeta \end{array} \right\} \beta \quad (6.15)$$

This representation facilitates the computation of SREs, where the SRE of index n is derived from the contraction of $2n$ replicas of the Pauli-MPS:

$$M_n(|\psi\rangle) = \frac{1}{1-n} \log \left(\begin{array}{c} \left. \begin{array}{c} P_1 - P_2 - \dots - P_N \\ | \quad | \quad \quad \quad | \\ \mu_1 \quad \mu_2 \quad \quad \quad \mu_N \end{array} \right\}^{n-1} \dots \left. \begin{array}{c} P_1^* - P_2^* - \dots - P_N^* \\ | \quad | \quad \quad \quad | \\ \mu_1 \quad \mu_2 \quad \quad \quad \mu_N \end{array} \right\}^{n-1} \right) - N \quad (6.16)$$

This approach offers significant advantages compared to earlier methods like the original replica trick presented in Section 6.2.1. One key benefit is the constant physical dimension d^2 for the intermediate MPS needed to compute the SRE, as opposed to the exponentially growing physical dimension $d^{2(n-1)}$ in the original method. While both methods maintain a bond dimension of χ^{2n} , the computational cost for exact contraction with the Pauli-MPS method scales as $\mathcal{O}(Nd^2\chi^{6n})$, a notable improvement over the $\mathcal{O}(Nd^{2(n-1)}\chi^{6n})$ cost of the replica trick.

Another advantage lies in the approximation scheme allowed by the Pauli-MPS method. As the contraction involves repeated MPO-MPS multiplications, it becomes possible to sequentially compress the intermediate MPS at each step using standard tensor network techniques. This controlled approximation, though not exact, is highly practical as it allows monitoring of truncation errors, thus enabling the method to be used beyond very small bond dimensions that exact contractions would require. The approximate contraction, assuming bond dimensions of order χ , scales as $\mathcal{O}(Nd^2\chi^4)$, offering significantly better efficiency compared to the exact approach.

6.2.2 Sampling approach

Stabilizer Rényi Entropies can be efficiently estimated using an MPS sampling algorithm [LC23; LC24], closely resembling the method described in Section 3.3 to sample from the computational basis. The starting point is the definition

$$M_n(|\psi\rangle) = \frac{1}{1-n} \log \left(\sum_{\sigma} (\Pi_{\rho}(\hat{\sigma}))^n \right) - N \log 2, \quad (6.17)$$

from which we see evaluating the SRE essentially requires computing $\sum_{\sigma} \Pi_{\rho}(\hat{\sigma})^n$, which is a sum over the 4^N possible strings of Pauli operators. This sum can be expressed as an average over the probability distribution $\Pi_{\rho}(\hat{\sigma})$ itself. In fact:

$$\sum_{\sigma} \Pi_{\rho}(\hat{\sigma})^n = \sum_{\sigma} \Pi_{\rho}(\hat{\sigma}) \cdot \Pi_{\rho}(\hat{\sigma})^{n-1} = \mathbb{E}_{\sigma \sim \Pi_{\rho}} [\Pi_{\rho}(\hat{\sigma})^{n-1}], \quad (6.18)$$

where $\mathbb{E}_{\sigma \sim \Pi_{\rho}} [\dots]$ denotes the expected value over the probability distribution Π_{ρ} . Thus, computing the SRE reduces to efficiently sampling Pauli string operators $\hat{\sigma}$ according to the probability distribution $\Pi_{\rho}(\hat{\sigma})$, and using these samples to estimate $\mathbb{E}_{\sigma \sim \Pi_{\rho}} [\Pi_{\rho}(\hat{\sigma})^{n-1}]$. One can simply use the sample average as the estimator. Specifically, if $\{\sigma_{\mu}\}_{\mu=1}^{N_{\text{samples}}}$ are the samples, the entropy can be estimated as follows:

$$\sum_{\sigma} \Pi_{\rho}(\hat{\sigma})^n \simeq \frac{1}{N_{\text{samples}}} \sum_{\mu=1}^{N_{\text{samples}}} \Pi_{\rho}(\hat{\sigma}_{\mu})^{n-1}. \quad (6.19)$$

Now, how to sample Pauli strings σ exactly with probability $\Pi_{\rho}(\sigma)$? The idea is to exploit the following decomposition Π_{ρ} in terms of conditional and prior

(or marginal) probabilities

$$\Pi_\rho(\boldsymbol{\sigma}) = \pi(\sigma_1)\pi(\sigma_2|\sigma_1)\pi(\sigma_3|\sigma_1, \sigma_2) \dots \pi(\sigma_N|\sigma_1, \sigma_2 \dots \sigma_{N-1}), \quad (6.20)$$

where

$$\pi_\rho(\sigma_j|\sigma_1 \dots \sigma_{j-1}) = \frac{\pi_\rho(\sigma_1 \dots \sigma_j)}{\pi_\rho(\sigma_1 \dots \sigma_{j-1})} \quad (6.21)$$

is the probability that the Pauli matrix σ_j occurs at position j given that the string $\sigma_1 \dots \sigma_{j-1}$ has already occurred at positions $1 \dots j-1$, no matter the occurrences in the rest of the system (i.e. marginalising over all possible Pauli strings for the remaining qubits $j+1 \dots N$). Specifically, one has

$$\pi_\rho(\sigma_1 \dots \sigma_j) = \sum_{\boldsymbol{\sigma} \in \mathcal{P}_{N-j}} \frac{1}{2^N} \text{Tr}[\rho \sigma_1 \dots \sigma_j \boldsymbol{\sigma}]^2. \quad (6.22)$$

In other terms, the conditional probability at the step j , i.e. $\pi_\rho(\sigma_j|\sigma_1 \dots \sigma_{j-1})$, can be thought as the probability $\pi_{\rho_{j-1}}(\sigma_j)$ of getting σ_j in the partially projected state

$$\rho_{j-1} \equiv \frac{\rho|_{\sigma_1 \dots \sigma_{j-1}}}{\pi_\rho(\sigma_1 \dots \sigma_{j-1})^{1/2}} \quad (6.23)$$

where we have defined the state

$$\rho|_{\sigma_1 \dots \sigma_{j-1}} \equiv \frac{1}{2^N} \sum_{\boldsymbol{\sigma} \in \mathcal{P}_{N-j+1}} \sigma_1 \dots \sigma_{j-1} \boldsymbol{\sigma} \text{Tr}[\rho \sigma_1 \dots \sigma_{j-1} \boldsymbol{\sigma}] \quad (6.24)$$

where, in the Pauli matrices decomposition of ρ , we are only keeping the contribution with fixed $\sigma_1 \dots \sigma_{j-1}$. Notice that such state is not normalized, however $\text{Tr}[\rho_{j-1}^2] = 1$, and the probability that the remaining string $\boldsymbol{\sigma} \in \mathcal{P}_{N-j+1}$ occurs is exactly given by $\pi_\rho(\boldsymbol{\sigma}|\sigma_1 \dots \sigma_{j-1})$. From the definition in Eq. (6.23), we can easily get the recursive relation $\rho_j = \pi_{\rho_{j-1}}(\sigma_j)^{-1/2} \rho_{j-1} |_{\sigma_j}$. Thanks to that, we can generate the outcomes (and the probabilities of that outcomes) by iterating over each single qubits, and sampling each local Pauli matrix according to the conditional probabilities. Once a local outcome occurs, the state is updated accordingly, and the iteration proceeds until all qubits are sampled. At the end of this procedure, as a result of Eq. (5.33), we generated configurations $\boldsymbol{\sigma}$ with probability $\Pi_\rho(\boldsymbol{\sigma})$.

Following this prescription, we start from the first term of the expansion in Eq. (5.33). This can be written as

$$\pi_\rho(\sigma_1) = \frac{1}{2^N} \sum_{\boldsymbol{\sigma} \in \mathcal{P}_{N-1}} \langle \psi | \sigma_1 \boldsymbol{\sigma} | \psi \rangle \langle \psi^* | \sigma_1^* \boldsymbol{\sigma}^* | \psi^* \rangle, \quad (6.25)$$

where we used the fact that the Pauli matrices are hermitian. Graphically this equation is represented as

$$\pi(\sigma_1) = \sum_{\sigma_2 \dots \sigma_N} \Pi_\rho(\sigma) = \frac{1}{2^N} \sum_{\sigma_2 \dots \sigma_N} \text{Diagram},$$

In terms of the operators $\Lambda_{\sigma_i} = \frac{1}{2} \sigma_i \otimes \sigma_i^*$ and $\Lambda_i = \frac{1}{2} \sum_{\sigma_i} (\sigma_i \otimes \sigma_i^*)$, each acting on the local Hilbert space given by a spin and its replica, the previous equation reads $\pi_\rho(\sigma_1) = [\langle \psi | \otimes \langle \psi^* |] \Lambda_{\sigma_1} \Lambda_2 \dots \Lambda_N [| \psi \rangle \otimes | \psi^* \rangle]$. Now, the following property can easily be proven

$$[\langle s'_i | \otimes \langle r'_i |] \Lambda_i [| s_i \rangle \otimes | r_i \rangle] = \delta_{s'_i, r'_i} \delta_{s_i, r_i}, \tag{6.26}$$

meaning that Λ_i is just two copies of the identity operator connecting the spin $|s_i\rangle$ and its replica (whose local computational basis is now indicated as $|r_i\rangle \in \{|0\rangle, |1\rangle\}$). Graphically this identity looks as follows (see Definition 5.2)

$$\frac{1}{2} \sum_{\sigma} \text{Diagram} = \text{Diagram} \tag{6.27}$$

Using Eq. (6.27) we get

$$\pi(\sigma_1) = \sum_{\sigma_2 \dots \sigma_N} \Pi_\rho(\sigma) = \text{Diagram}.$$

By applying the right-normalization condition of the MPS tensors, we can further simplify the calculation to a local tensor contraction as follows

$$\pi(\sigma_1) = \sum_{\sigma_2 \dots \sigma_N} \Pi_\rho(\sigma) = \text{Diagram}, \tag{6.28}$$

which as an equation reads:

$$\pi_\rho(\sigma_1) = \frac{1}{2} \sum_{s_1, s'_1, r_1, r'_1} (\mathbb{A}_1^{s'_1})^* \mathbb{A}_1^{r'_1}(\sigma_1)_{s'_1 s_1} (\sigma_1^*)_{r'_1 r_1} \mathbb{A}_1^{s_1} (\mathbb{A}_1^{r_1})^* .$$

After evaluating $\pi_\rho(\sigma_1)$ for $\sigma_1 \in \{\sigma^0, \sigma^1, \sigma^2, \sigma^3\}$, one can extract a sample from this distribution, obtaining the first element of the string. The information about the partially projected state Eq. (6.23) is encoded in an effective environment matrix \mathbb{L} initialized as follows:

$$\mathbb{L} = \frac{1}{\sqrt{2} \pi_\rho(\sigma_1)} \sum_{s_1, s'_1} (\mathbb{A}_1^{s'_1})^* (\sigma_1)_{s'_1 s_1} \mathbb{A}_1^{s_1} = \frac{1}{\sqrt{\pi(\sigma_1)}} \frac{1}{\sqrt{2}} \text{ [Diagram: A red circle labeled } \sigma_1 \text{ with two teal arrows pointing towards it from the right and two teal arrows pointing away from it to the right.]} .$$

The calculation of the next terms of Eq. (5.33) and the extraction of the remaining σ_i proceeds following the same line. In fact, it is not difficult to show that at a generic site i Eq. (6.28) becomes

$$\pi(\sigma_i | \sigma_1 \dots \sigma_{i-1}) = \text{ [Diagram: A green oval labeled } \mathbb{L} \text{ on the left and a green oval labeled } \mathbb{L}^* \text{ on the right. Two red circles are between them. Horizontal teal arrows point from } \mathbb{L} \text{ to the first red circle, from the first red circle to the second red circle, and from the second red circle to } \mathbb{L}^*. \text{ Vertical teal arrows point from the first red circle to the second red circle.]} . \tag{6.29}$$

After the extraction of σ_i the environment \mathbb{L} has to be updated as

$$\text{ [Diagram: A green oval labeled } \mathbb{L} \text{ with two horizontal lines extending to the right.]} \rightarrow \frac{1}{\sqrt{\pi(\sigma_i | \sigma_{i-1} \dots \sigma_1)}} \frac{1}{\sqrt{2}} \text{ [Diagram: A green oval labeled } \mathbb{L} \text{ with two horizontal lines extending to the right, and a red circle to its right with two teal arrows pointing towards it from the right and two teal arrows pointing away from it to the right.]} , \tag{6.30}$$

The full sampling recipe is summarized in the Algorithm 2.

Input: an MPS $|\psi\rangle$ of size N

- 1: Put the MPS in right-normalized form.
- 2: Initialize $\mathbb{L} = (1)$ and $\Pi = 1$
- 3: **for** ($i = 1, i = N, i ++$) **do**
- 4: Compute the probabilities $\pi(\alpha) = \pi_\rho(\sigma^\alpha |\sigma_1 \cdots \sigma_{i-1})$ for $\alpha \in \{0, 1, 2, 3\}$ as in Eq. (6.29).
- 5: Generate a random value of α according to $\pi(\alpha)$
- 6: Set $\sigma_i = \sigma^\alpha$, update $\Pi \rightarrow \Pi \cdot \pi(\alpha)$
- 7: Update \mathbb{L} as in Eq. (6.30).
- 8: **end for**

Output: a Pauli string σ and the probability $\Pi(\sigma)$

Algorithm 2. Pauli sampling from MPS.

Example 6.2: Stabilizer group of an MPS

One question which may be addressed using the approaches outlined previously is to identify the stabilizer group $\mathcal{S}(|\psi\rangle)$ of an MPS.

Pauli-mps approach — This can be done for example exploiting the fact that the stabilizer nullity $\nu = N - k$, where $|\mathcal{S}| = 2^k$, is given by $\nu = \lim_{n \rightarrow \infty} (n - 1)M_n$. Using the Pauli basis algorithm in ref. [Tar+24], on can find the fixed point of the power-series of the Pauli basis MPS, namely

$$|G(\psi)\rangle = \lim_{n \rightarrow \infty} \frac{1}{\mathcal{N}_n} \left[\begin{array}{cccc} & P_1 & P_2 & \dots & P_N \\ & | & | & & | \\ & P_1 & P_2 & \dots & P_N \\ & | & | & & | \\ \vdots & \vdots & \vdots & & \vdots \\ & P_1 & P_2 & \dots & P_N \\ & | & | & & | \\ & P_1 & P_2 & \dots & P_N \end{array} \right]$$

which is the normalized Pauli vector (hence the factor $1/\mathcal{N}_n$), compressed accordingly to the computation resources (usually $\chi_G \sim \chi^2$), and which can be used to extract the information about the stabilizer group of $|\psi\rangle$, since $\langle \mu | G(\psi) \rangle = 2^{k/2}$, if and only if $\hat{\sigma}^{\mu_1} \cdots \hat{\sigma}^{\mu_N} |\psi\rangle = \pm |\psi\rangle$, otherwise is zero. This can be achieved by sampling $|G(\psi)\rangle$ in the computational basis, as described in Chapter 3 and Chapter 5. The overall computational scaling of this approach is still subjected to the compression cost of of the Pauli MPS, i.e. $O(N\chi^4)$.

Biased Pauli-sampling approach — Another possibility is based on a clever biasing of the Pauli sampling techniques [LC24].

In this method, at each generic step i of the sweep, a set of K sub-strings $\{\sigma_{[1,i]}^\mu\}_{\mu=1}^K$ is stored, where $\sigma_{[1,i]}$ is shorthand for the string of Pauli operators $(\sigma_1 \dots \sigma_i)$. Along with these sub-strings, their associated partial probabilities $\{\pi_\rho(\sigma_{[1,i]}^\mu)\}_{\mu=1}^K$ are also retained. The partial probability $\pi_\rho(\sigma_1 \dots \sigma_i)$ is given by:

$$\pi_\rho(\sigma_1 \dots \sigma_i) = \sum_{\sigma \in \mathcal{P}_{N-i}} \frac{1}{2^N} \text{Tr} [\rho \sigma_1 \dots \sigma_i \sigma]^2, \quad (6.31)$$

where the sum runs over all possible Pauli strings acting on the remaining $N - i$ sites of the system. The ultimate goal is to track these sub-strings until the last step $i = N$, at which point one aims to identify those sub-strings that meet the stabilizer condition $\Pi_\rho(\sigma^\mu) = \pi_\rho(\sigma_{[1,N]}^\mu) = 1/2^N$.

However, storing all possible sub-strings quickly becomes computationally infeasible, as their number grows as 4^i . To manage this growth, efficient strategies for discarding certain sub-strings while maintaining a maximum allowable number \mathcal{N} of stored sub-strings have been developed. The challenge is to retain only those sub-strings that are more likely to form stabilizer strings at the end of the sweep. Two key strategies are used for this purpose:

1. **Threshold-based filtering:** It is observed that for any stabilizer string $\sigma \in \mathcal{S}(|\psi\rangle)$, the partial probability at step i satisfies a lower bound: $\pi_\rho(\sigma_{[1,i]}) \geq \frac{1}{2^i \chi_i}$, where χ_i represents the bond dimension at site i of the MPS representing $|\psi\rangle$. Thus, sub-strings for which $2^i \chi_i \pi_\rho(\sigma_{[1,i]}) < 1$ can be safely discarded.

Selective pruning: When the number K of stored sub-strings exceeds the limit of resources \mathcal{N} , the sub-strings are sorted based on their partial probabilities $\pi_\rho(\sigma_{[1,i]}^\mu)$ in descending order. Only the top \mathcal{N} sub-strings with the highest probabilities are retained. These sub-strings have the greatest potential to maximize the final stabilizer probability $\Pi_\rho(\sigma)$ at the end of the sweep. This approach ensures that the computational resources are focused on the most promising candidates.

These strategies collectively allow for efficient management of sub-string storage, ensuring that the computation remains feasible even for

large systems while retaining the most relevant sub-strings for stabilizer identification. Let us mention that, in most scenarios a single sweep is sufficient to extract the entire set of stabilizer generators, leading to a computational cost $O(N\chi^3)$; however further refined strategies have been proposed in ref. [LC24] to allow the algorithm to target unsampled regions of $\mathcal{S}(|\psi\rangle)$ with minor increasing of the computational effort.

6.3 Clifford enhanced matrix product states (CMPS)

Here, we show the possibility of improving tensor network algorithms by introducing a class of quantum states described by an MPS of bond dimension χ , where a Clifford unitary is applied to the state. We call this set of states Clifford enhanced matrix product states CMPS [MG24; LHN24]

$$\text{CMPS} = \{U_c |\psi_\chi\rangle, \quad \forall U_c \in \mathcal{C}_N, \quad |\psi_\chi\rangle \in \text{MPS bond dimension } \chi\}. \quad (6.32)$$

Note that, in general, $|\phi\rangle = U_c |\psi_\chi\rangle$ may describe a quantum state with higher entanglement, highlighting the potential of applying Clifford unitaries to MPS in order to represent more entangled quantum states.

Moreover, by combining stabilizer and tensor network methods, it is easy to figure out how to compute the expectation value of a Pauli string $\Sigma^\mu = \sigma^{\mu_1} \otimes \dots \otimes \sigma^{\mu_N}$ over a CMPS indeed

$$\langle \phi | \Sigma^\mu | \phi \rangle = \langle \psi_\chi | U_c^\dagger \Sigma^\mu U_c | \psi_\chi \rangle = \langle \psi_\chi | \Sigma^{\mu'} | \psi_\chi \rangle \quad (6.33)$$

where $\Sigma^{\mu'}$ is the new Pauli string under the action of the Clifford unitary U_c . Therefore, even though $|\phi\rangle$ can be highly entangled, the complexity of computing the expectation value of a Pauli string over $|\phi\rangle$ remains the same as computing it over $|\psi_\chi\rangle$.

The use of such hybrid tensor networks, which blend classical stabilizer elements with quantum resources, represents a cutting-edge strategy for simulating quantum systems that are otherwise too complex to handle using purely classical techniques. This approach underscores the importance of quantum magic in understanding the full computational potential of quantum systems and their simulation via tensor networks.

6.3.1 Hybrid stabilizer MPO

In general, any unitary operation can be decomposed into a set of Clifford unitaries and a magic quantum gate. In order to be as general as possible, we

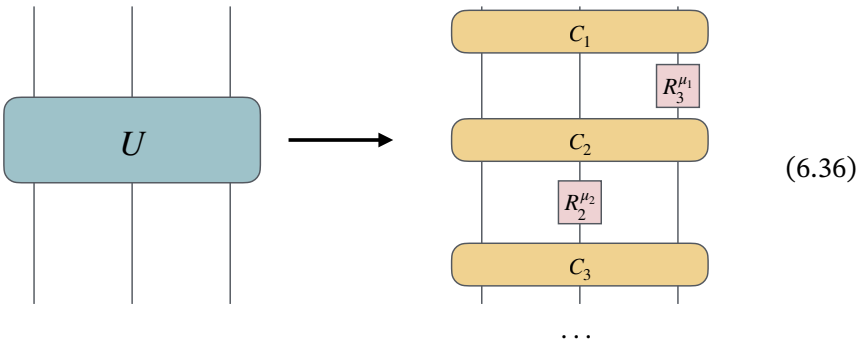
will consider local rotations as the set of non-Clifford operations that introduce non-stabilizerness to a quantum circuit. We define $R_l^\mu(\theta)$ as a local rotation gate acting on qubit l , rotating it by an angle θ around the μ -axis. This is given by the expression

$$R_l^\mu(\theta) = e^{-i\theta/2\sigma_l^\mu}. \tag{6.34}$$

A unitary acting on N qubits has the following representation

$$U = \prod_j R_{l_j}^{\mu_j}(\theta_j) C_j \tag{6.35}$$

where $R_{l_j}^{\mu_j}(\theta_j)$ are local rotations and $C_j \in \mathcal{C}_N$ Clifford unitaries.



We want to compute the action of U on a quantum state $|\psi_0\rangle$. In the following, we will provide a procedure to construct a \mathcal{C} MPS from $U|\psi_0\rangle = C|\phi\rangle$, such that the entanglement entropy of $|\phi\rangle$ for any bipartition is less than that of $U|\psi_0\rangle$. In particular, we notice that

$$C^\dagger R_l^\mu(\theta) C = \cos(\theta/2) \mathbb{1} \mp i \sin(\theta/2) \Sigma^\mu \tag{6.37}$$

where we used the fact that $C^\dagger \sigma_l^\mu C = \pm \Sigma^\mu$.

Before providing the full expression for the decomposition of U in Eq. (6.35), it is instructive to first examine how the transformation of U works for a small number of layers, as shown in the following example.

Example 6.3: Implementation of the stabilizer MPO

Here, we show an explicit implementation of the algorithm for two layers of U , that is

$$U = R_m^\mu(\theta_m) C_2 R_n^\nu(\theta_n) C_1 \tag{6.38}$$

first of all we insert the identity $C_1 C_1^\dagger$

$$U = R_m^\mu(\theta_m) C_2 C_1 C_1^\dagger R_n^\nu(\theta_n) C_1 \quad (6.39)$$

we define the since Clifford unitaries form a group we can define the composite Clifford $C_{12} = C_2 C_1$. We then insert the identity $C_{12} C_{12}^\dagger$ in order to transform the second local rotation

$$U = C_{12} C_{12}^\dagger R_m^\mu(\theta_m) C_{12} C_1^\dagger R_n^\nu(\theta_n) C_1 \quad (6.40)$$

now we can transform the local rotations and obtain

$$U = C_{12} [\cos(\theta_m/2) \mathbb{1} \mp i \sin(\theta_m/2) \Sigma^\mu] [\cos(\theta_n/2) \mathbb{1} \mp i \sin(\theta_n/2) \Sigma^\nu] \quad (6.41)$$

To write the decomposition of U we define $C_{1\dots j} \equiv C_j C_{j-1} \dots C_1$. By sequentially inserting the identities $C_{1\dots j} C_{1\dots j}^\dagger$ we obtain the following

$$U = C_{1\dots N} \prod_j^N C_{1\dots j}^\dagger R_{l_j}^{\mu_j}(\theta_j) C_{1\dots j} = C_{1\dots N} \prod_j^N (\cos(\theta_j/2) \mathbb{1} \mp i \sin(\theta_j/2) \Sigma^{\mu_j}). \quad (6.42)$$

We can now focus on constructing an MPO description of the terms

$$\cos(\theta/2) \mathbb{1} \mp i \sin(\theta/2) \Sigma^\mu = \mathbb{T}_1 \dots \mathbb{T}_N \quad (6.43)$$

Since we are summing two operators, i.e. $\mathbb{1}$ and Σ^μ , whose MPO representation has an auxiliary dimension $D = 1$, it is straightforward to see that their sum results in an MPO with an auxiliary dimension $D = 2$, such that

$$\mathbb{T}_j = \begin{pmatrix} \cos(\theta/2)^{1/N} \mathbb{1}_j & 0 \\ 0 & (\mp i \sin(\theta/2))^{1/N} \sigma_j^{\mu_j} \end{pmatrix} \quad (6.44)$$

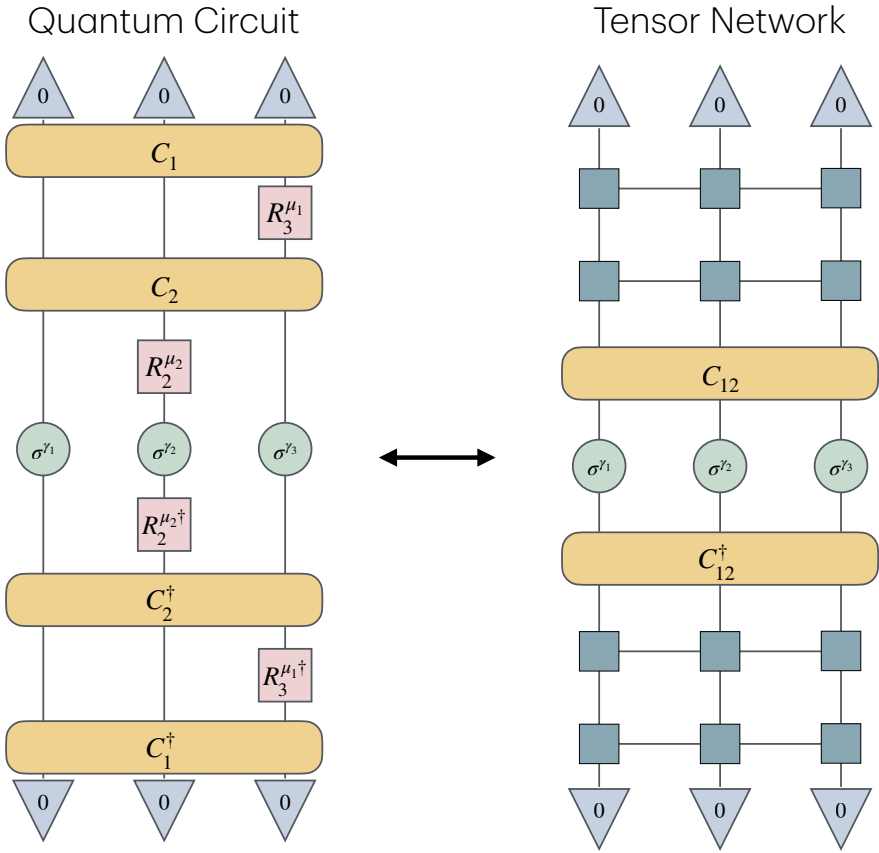
for $j = 2, \dots, N-1$, and with boundaries

$$\mathbb{T}_1 = \left(\cos(\theta/2)^{1/N} \mathbb{1}_1 \quad (\mp i \sin(\theta/2))^{1/N} \sigma_1^{\mu_1} \right), \quad \mathbb{T}_N = \begin{pmatrix} \cos(\theta/2)^{1/N} \mathbb{1}_N \\ (\mp i \sin(\theta/2))^{1/N} \sigma_N^{\mu_N} \end{pmatrix}. \quad (6.45)$$

We call this MPO with bond dimension $D = 2$, **stabilizer MPO** [MSC24].

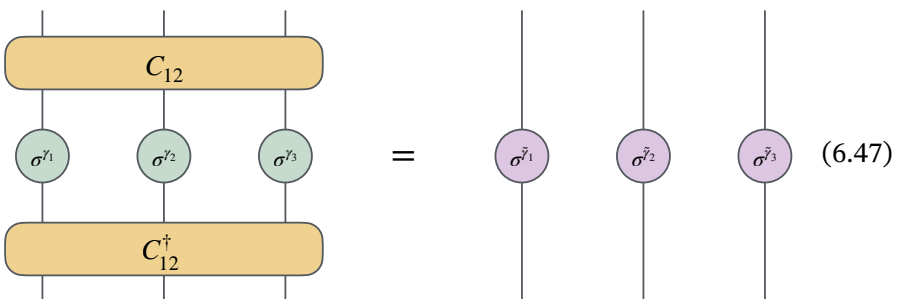
As an example, we can illustrate the expression $\langle \psi_0 | U^\dagger \Sigma^\nu U | \psi_0 \rangle$ for a two-layer unitary U by providing both its representation as a quantum circuit and

as a tensor network



(6.46)

Notice that the entangling Clifford C_{12} does not increase the complexity of the circuit, as it can be absorbed by conjugating the Pauli product $C_{12}^\dagger \Sigma^\gamma C_{12} = \Sigma^{\tilde{\gamma}}$, which, since it is a MPO with auxiliary dimension 1, does not increase the bond dimensions of the network, as shown in the diagram



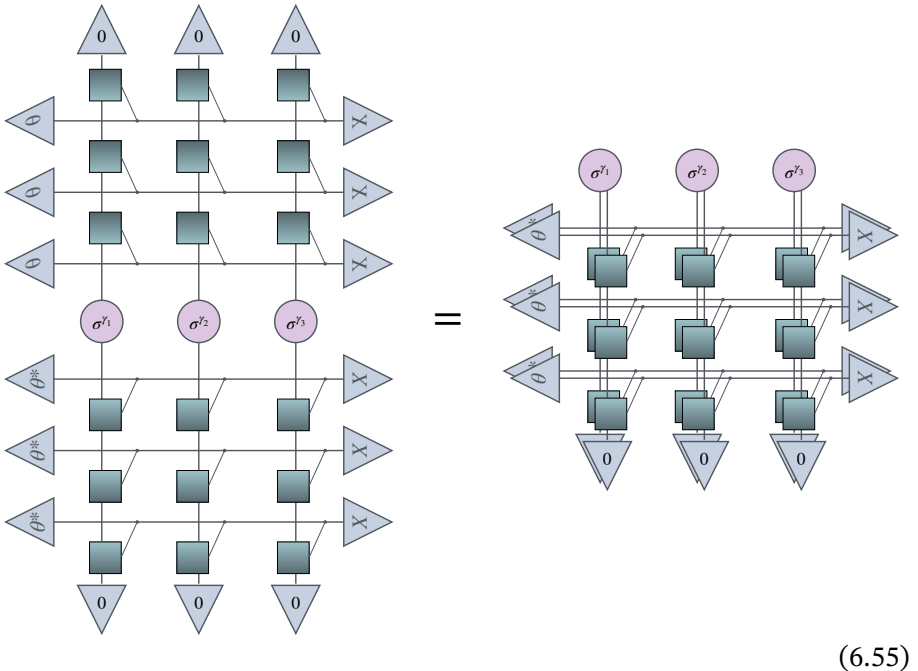
This relation enables us to express the evolution of the operator U as

$$U = C_{1\dots N} \prod_{m=1}^M \langle \theta_m | \mathbb{K}_1^m \dots \mathbb{K}_N^m | X \rangle \quad (6.53)$$

where $C_{1\dots N} = C_N \dots C_1$. Therefore we obtain the following

$$\langle \psi_0 | U^\dagger \Sigma^\gamma U | \psi_0 \rangle = \langle \psi_0 | \prod_{m=1}^N \langle \theta_m^* | \mathbb{K}_1^{m\dagger} \dots \mathbb{K}_N^{m\dagger} | X \rangle \Sigma^{\gamma\dagger} \prod_{m=1}^N \langle \theta_m | \mathbb{K}_1^m \dots \mathbb{K}_N^m | X \rangle | \psi_0 \rangle \quad (6.54)$$

This expression can be represented as a tensor network, which is shown below, and can be contracted efficiently in any direction

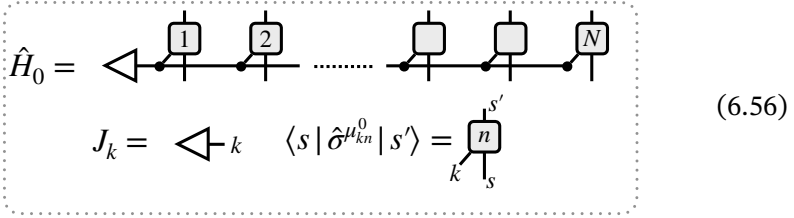


Furthermore, by folding the tensor network, as shown in the diagram, we can compute the evolution of Pauli strings in the Heisenberg picture.

6.3.2 Clifford-Dressed TDVP for Hamiltonian Dynamics

Another typical scenario which can be consider involves the dynamics, from a short-range correlated initial state (e.g., a product state), generated by the

Hamiltonian $\hat{H}_0 = \sum_k^{O(N)} J_k \hat{\Sigma}^{\mu_k}$, which in graphical notation looks like



$$\hat{H}_0 = \left\langle \left[\begin{array}{c} \boxed{1} \\ \bullet \end{array} \right] \left[\begin{array}{c} \boxed{2} \\ \bullet \end{array} \right] \dots \left[\begin{array}{c} \boxed{n} \\ \bullet \end{array} \right] \dots \left[\begin{array}{c} \boxed{N} \\ \bullet \end{array} \right] \right\rangle$$

$$J_k = \left\langle \left[\begin{array}{c} \bullet \\ \left[\begin{array}{c} \boxed{n} \\ \bullet \end{array} \right] \end{array} \right] \right\rangle_k \quad \langle s | \hat{\sigma}_{kn}^{\mu} | s' \rangle$$
(6.56)

which is a diagonal MPO with bond dimension equal to the number $O(N)$ of independent Pauli strings entering in the sum. Here the subscript zero is indicating that the Hamiltonian is in its “bare” form, meaning it has not yet been modified by any Clifford disentangler.

The algorithm that we are going to describe in this section has been recently proposed in ref. [Mel+24; QHQ24b]. A similar approach for ground-state search has been presented in ref. [QHQ24a].

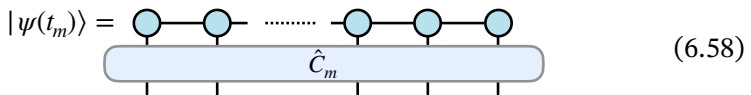
We are primarily interested in evolving the state and monitoring local observables that are experimentally relevant. This includes computing the expectation value of Pauli strings:

$$\langle \psi(0) | e^{i\hat{H}_0 t} \hat{\Sigma}^{\mu} e^{-i\hat{H}_0 t} | \psi(0) \rangle. \tag{6.57}$$

Typically, the time evolution is discretized into small intervals of length dt . The state, which is approximated as an MPS with bond dimension χ , is evolved using the single-site TDVP scheme. In this symplectic time-evolution integrator, the bond dimension χ is fixed from the beginning of the simulation.

However, the entanglement entropy typically grows rapidly, making the MPS representation inadequate after a relatively short time. This is where Clifford disentangling strategies, or “entanglement cooling” become essential.

At each time step $m \in \{1, \dots\}$, corresponding to time $t_m = m dt$, we apply a suitable Clifford transformation \hat{C}_m to the state $|\psi(t_m)\rangle$ to reduce the entanglement of the new state:



$$|\psi(t_m)\rangle = \left[\begin{array}{c} \bullet \quad \bullet \quad \dots \quad \bullet \quad \bullet \end{array} \right] \hat{C}_m$$
(6.58)

This induces a transformation of the Hamiltonian as well:

$$\hat{H}_{m-1} \leftarrow \begin{array}{c} \text{.....} \\ \text{.....} \\ \text{.....} \\ \text{.....} \\ \text{.....} \end{array} \begin{array}{c} \text{.....} \\ \text{.....} \\ \text{.....} \\ \text{.....} \\ \text{.....} \end{array} \hat{H}_m \quad (6.59)$$

Which can be efficiently performed using the stabilizer tableau formalism [Got97], and does not increase the number of Pauli strings involved in its definition. In practice, as depicted in the previous figure, the Clifford transformation preserves the diagonal structure of the MPO and only modifies the Pauli strings composing the operator (hence the different colors in the pictorial representation of the MPO).

In practice, we are recasting the original time evolution,

$$\prod_{m=0}^{\lfloor t/dt \rfloor - 1} e^{-i\hat{H}_0 dt} |\psi(0)\rangle, \quad (6.60)$$

into a so called **Clifford-dressed** evolution

$$\prod_{m=0}^{\lfloor t/dt \rfloor - 1} e^{-i\hat{H}_m dt} \hat{C}_m |\psi(0)\rangle, \quad (6.61)$$

where $\hat{C}_0 = \hat{I}$ is the identity operator. The Clifford-dressed evolution iteratively constructs the final Clifford-enhanced MPS (cMPS), yielding:

$$|\psi(0)\rangle = \begin{array}{c} \text{.....} \\ \text{.....} \\ \text{.....} \\ \text{.....} \\ \text{.....} \end{array} \begin{array}{c} \text{.....} \\ \text{.....} \\ \text{.....} \\ \text{.....} \\ \text{.....} \end{array} \quad (6.62)$$

which however has been constructed iteratively from Eq. (6.61) by optimising the Clifford transformations $\{\hat{C}_1, \hat{C}_2, \dots\}$ at intermediate time steps and not just at the end of the full bare time evolution. Additionally, for any Pauli string $\hat{\Sigma}^\mu$, its time evolution must be transformed as

$$\hat{C}_m \dots \hat{C}_1 \hat{\Sigma}^\mu \hat{C}_1^\dagger \dots \hat{C}_m^\dagger, \quad (6.63)$$

which thanks to the Clifford properties, it is a trivial transformation.

This method ensures an efficient computation of the time-evolved states while maintaining the underlying structure of Clifford-enhanced Matrix Product States.

Two-site TDVP scheme — The Time-Dependent Variational Principle (TDVP) algorithm in the Matrix Product State (MPS) formalism traditionally supports a two-site integration scheme (see Chapter 4). While this modification abandons the symplectic nature of single-site integration, it compensates by allowing the bond dimension to dynamically adapt to the evolving entanglement structure. In this approach, the effective Hamiltonian acts on a two-site central block, updating the corresponding local MPS tensors. After the time evolution, a Singular Value Decomposition (SVD) is performed to extract the single-site tensors, with the rightmost tensor adapted through backward evolution.

Building upon this, a new two-site TDVP scheme enhanced by Clifford disentanglers has been proposed in ref. [QH24b]. Inspired by the disentangling techniques used in ref. [QH24a] for the Density Matrix Renormalization Group (DMRG), the 2-site method applies a local two-qubit Clifford gate before the SVD step. This “Clifford dressing” aims to further reduce entanglement by locally optimizing the two-site MPS tensors. After finding the optimal disentangler, the effective Hamiltonian is transformed under conjugation by the Clifford operator. This transformation remains localized, only impacting the Pauli strings corresponding to the two lattice sites currently being evolved.

The computational cost of this additional disentangling step should be modest, as it only involves optimizing local gates. However, a potential drawback of this approach is its propensity to get trapped in local entanglement minima. Without a global sweeping mechanism, the disentanglers may fail to optimally reduce global entanglement across the entire wave function. Nonetheless, this approach could offer a computationally efficient way to mitigate entanglement growth in systems where bond dimension scaling poses a challenge.

References of Chapter 6

- [Got97] D. Gottesman. *Stabilizer Codes and Quantum Error Correction*. 1997. DOI: [10.48550/ARXIV.QUANT-PH/9705052](https://doi.org/10.48550/ARXIV.QUANT-PH/9705052).
- [Hae+16] J. Haegeman et al. “Unifying time evolution and optimization with matrix product states.” In: *Phys. Rev. B* 94 (16 2016), p. 165116. DOI: [10.1103/PhysRevB.94.165116](https://doi.org/10.1103/PhysRevB.94.165116).

- [HC17] M. Howard and E. Campbell. “Application of a Resource Theory for Magic States to Fault-Tolerant Quantum Computing.” In: *Phys. Rev. Lett.* 118 (9 2017), p. 090501. DOI: [10.1103/PhysRevLett.118.090501](https://doi.org/10.1103/PhysRevLett.118.090501).
- [HG19] M. Heinrich and D. Gross. “Robustness of Magic and Symmetries of the Stabiliser Polytope.” In: *Quantum* 3 (2019), p. 132. ISSN: 2521-327X. DOI: [10.22331/q-2019-04-08-132](https://doi.org/10.22331/q-2019-04-08-132).
- [HP23a] T. Haug and L. Piroli. “Quantifying nonstabilizerness of matrix product states.” In: *Physical Review B* 107.3 (2023). ISSN: 2469-9969. DOI: [10.1103/physrevb.107.035148](https://doi.org/10.1103/physrevb.107.035148).
- [HP23b] T. Haug and L. Piroli. “Stabilizer entropies and nonstabilizerness monotones.” In: *Quantum* 7 (2023), p. 1092. ISSN: 2521-327X. DOI: [10.22331/q-2023-08-28-1092](https://doi.org/10.22331/q-2023-08-28-1092).
- [LB24] L. Leone and L. Bittel. *Stabilizer entropies are monotones for magic-state resource theory*. 2024. arXiv: [2404.11652](https://arxiv.org/abs/2404.11652) [[quant-ph](#)].
- [LC23] G. Lami and M. Collura. “Nonstabilizerness via Perfect Pauli Sampling of Matrix Product States.” In: *Phys. Rev. Lett.* 131 (18 2023), p. 180401. DOI: [10.1103/PhysRevLett.131.180401](https://doi.org/10.1103/PhysRevLett.131.180401).
- [LC24] G. Lami and M. Collura. “Unveiling the Stabilizer Group of a Matrix Product State.” In: *Phys. Rev. Lett.* 133 (1 2024), p. 010602. DOI: [10.1103/PhysRevLett.133.010602](https://doi.org/10.1103/PhysRevLett.133.010602).
- [LHN24] G. Lami, T. Haug, and J. D. Nardis. *Quantum State Designs with Clifford Enhanced Matrix Product States*. 2024. arXiv: [2404.18751](https://arxiv.org/abs/2404.18751) [[quant-ph](#)]. URL: <https://arxiv.org/abs/2404.18751>.
- [LOH22] L. Leone, S. F. E. Oliviero, and A. Hama. “Stabilizer Rényi Entropy.” In: *Phys. Rev. Lett.* 128 (5 2022), p. 050402. DOI: [10.1103/PhysRevLett.128.050402](https://doi.org/10.1103/PhysRevLett.128.050402).
- [Mel+24] A. F. Mello et al. *Clifford Dressed Time-Dependent Variational Principle*. 2024. arXiv: [2407.01692](https://arxiv.org/abs/2407.01692) [[quant-ph](#)].
- [MG24] S. Masot-Llima and A. Garcia-Saez. *Stabilizer Tensor Networks: universal quantum simulator on a basis of stabilizer states*. 2024. arXiv: [2403.08724](https://arxiv.org/abs/2403.08724) [[quant-ph](#)]. URL: <https://arxiv.org/abs/2403.08724>.
- [MSC24] A. F. Mello, A. Santini, and M. Collura. “Hybrid Stabilizer Matrix Product Operator.” In: *Phys. Rev. Lett.* 133 (15 2024), p. 150604. DOI: [10.1103/PhysRevLett.133.150604](https://doi.org/10.1103/PhysRevLett.133.150604).
- [NC10] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. DOI: [10.1017/CB09780511976667](https://doi.org/10.1017/CB09780511976667).

- [QH24a] X. Qian, J. Huang, and M. Qin. *Augmenting Density Matrix Renormalization Group with Clifford Circuits*. 2024. arXiv: [2405.09217 \[cond-mat.str-el\]](#).
- [QH24b] X. Qian, J. Huang, and M. Qin. *Clifford Circuits Augmented Time-Dependent Variational Principle*. 2024. arXiv: [2407.03202 \[cond-mat.str-el\]](#).
- [Tar+24] P. S. Tarabunga et al. “Nonstabilizerness via Matrix Product States in the Pauli Basis.” In: *Phys. Rev. Lett.* 133 (1 2024), p. 010601. DOI: [10.1103/PhysRevLett.133.010601](#).
- [Vei+14] V. Veitch et al. “The resource theory of stabilizer quantum computation.” In: *New Journal of Physics* 16.1 (2014), p. 013009. DOI: [10.1088/1367-2630/16/1/013009](#).

Index

- Bell states, 40
- Bloch sphere, 34
 - rotation, 34
- Clifford gates, 155
- CNOT, 14, 36, 155
 - implementation, 80
- Controlled-Z gate, 80
- Copy tensor, 11
- CPT maps, 133
- Density matrix, 27, 31, 116
 - MPDO, 140
 - superket formalism, 121
- Diagonalization, 18
- Digitized quantum annealing, 105
 - optimised compression, 111
- Entanglement, 38
 - entropy, 42
 - Schmidt decomposition, 37
 - stabilizer state, 159
- GHZ state, 68
- Gottesman-Knill theorem, 158
- Hadamard gate, 33
- Hilbert space, 25
 - many-body systems, 30
- Hopfield model, 106
- Identity tensor, 9
- Ising model, 103, 119
 - quantum annealing, 104
- Josephson junction, 77
- Jump operator, 135
- Kraus operator, 29
- Krylov, 94
- Lanczos, 94
- Lindblad, 135
 - MPO dissipator, 137
 - MPO formalism, 137
 - positive TN, 143
 - unraveling, 145
- Magic, 160
 - Monotones, 160
- Magic gate, 33
- Magic state, 160
- Many-body state, 30, 35, 45
 - matrix product state, 48
 - product state, 46, 65
- Measurement, 29, 73
 - bit-string sampling, 75
 - MPS scheme, 74
 - MPS update, 74
 - teleportation, 40
- METTS, 128
- Min-relative entropy of magic, 161
- MPO, 54
 - auxiliary-state diagram, 57
 - direct sum, 56

- Long-range Hamiltonian, 58
- Pauli basis, 124
- product, 56
- XXZ Hamiltonian, 58
- MPS, 20, 47
 - circuit representation, 67
 - gauge transformation, 50
 - left canonical, 51
 - mixed canonical, 51
 - norm, 49
 - overlap, 52
 - Pauli perfect sampling, 170
 - projective sampling, 75, 131
 - Pauli string average, 53
 - right canonical, 51
 - variational compression, 110
- Observables, 26, 30, 54
- Optical tweezers, 78
- Parity tensor, 13
- Partial trace, 31
- Partition function, 118, 120
 - Pauli basis, 136
 - superket, 123
- Pauli matrices, 32
 - completeness, 125, 169
 - orthogonality, 125
- Pauli string, 53
- Phase gate, 33
- Purification, 31, 141
- QR decomposition, 20, 50
- Quantum jumps, 146
- QUBO, 99
 - max-cut, 101
 - traveling salesman problem, 102
 - unweighted graphs, 99
 - weighted graphs, 100
- Reduced density matrix, 31
- Robustness of magic, 160
- Rydberg blockade, 78
- Schrödinger equation, 27
- Singular Value Decomposition, 19, 21, 37, 72
- Stabilizer Rényi Entropies, 161
 - Pauli MPS, 165
 - Pauli perfect sampling, 168
 - replica method, 165
- Stabilizer tensor network, 173
 - clifford-dressed TDVP, 180
 - stabilizer MPO, 175
- Statistical mixture, 26, 116
- Swap tensor, 12
- TDVP, 90, 182, 183
 - errors, 98
 - MPS equations, 93
 - projectors, 91
- TEBD, 70
- Tensors
 - contraction cost, 17
 - index contraction, 15
 - index fusion, 7
 - index splitting, 8
 - network diagram, 7
 - outer product, 8
- Thermal state, 118
 - positive TN, 142
- Transfer matrix, 49, 120
- Trapped ions, 81
- Trotter decomposition, 70, 105, 122, 136
- TTN, 22
- Unitary operator, 27
 - single qubit gate, 35, 79
- Vectorization, 121
- von Neumann equation, 118
- W state, 69
- Weak measurements, 147

Tensor Network Techniques for Quantum Computation

Mario Collura, Guglielmo Lami, Nishan Ranabhat, Alessandro Santini

This book serves as an introductory yet thorough guide to tensor networks and their applications in quantum computation and quantum information, designed for advanced undergraduate and graduate-level readers.

In Part I, foundational topics are covered, including tensor structures and network representations like Matrix Product States (MPS) and Tree Tensor Networks (TTN). These preliminaries provide readers with the core mathematical tools and concepts necessary for quantum physics and quantum computing applications, bridging the gap between multi-linear algebra and complex quantum systems.

Part II explores practical applications of tensor networks in simulating quantum dynamics, with a particular focus on the efficiency they offer for systems of high computational complexity. Key topics include Hamiltonian dynamics, quantum annealing, open system dynamics, and optimization strategies using TN frameworks. A final chapter addresses the emerging role of “quantum magic” in tensor networks. It delves into non-stabilizer states and their contribution to quantum computational power beyond classical simulability, featuring methods such as stabilizer-enhanced MPS and the Clifford-dressed TDVP.

In presenting tensor networks as tools for understanding quantum complexity, this book aims to foster a deeper collaboration between the many-body physics and quantum computing communities, inviting a broader audience to engage with these recent developments.



ISBN 9788898587049
