



UNIVERSITY OF TRENTO
PHYSICS DEPARTMENT
Ph.D. Thesis

Digital Quantum Computing for Many-Body Simulations

Ph.D candidate
Valentina Amitrano

Advisor
Francesco Pederiva

Co-advisor
Alessandra Bernardi

Final examination date: December 13, 2023

*Ai miei genitori,
per il loro infinito supporto*

Outline

This thesis is structured as follows. In Chapter 1 (*Introduction*) an elementary foundation for the understanding of quantum computing is laid out. Starting with the basic definition and representation of single and multi-qubit states, the argument progresses through the concept of quantum superposition in the context of measurement, culminating in the exploration of quantum algorithms and their formulation as quantum circuits. Special emphasis is placed on the intricate process of quantum gate decomposition, an elementary lynchpin that underpins the development of quantum algorithms and plays a crucial role in this research. In particular, this concerns the implementation of quantum algorithms designed to simulate the dynamic evolution of multi-particle quantum systems, the so-called Hamiltonian simulations.

Chapter 2 (*Quantum gate decomposition and circuits*) introduces the concept of quantum gate decomposition and quantum circuit optimisation, which allows to decompose the entire qubit transformation into a sequence of elementary gates available in the quantum computing architecture. The specific decomposition of a unitary transformation plays a crucial role in fault-tolerant quantum computing. An optimal implementation of a quantum gate is essential to efficiently perform a quantum simulation, especially for near-term quantum computers. In fact, the number of gates we use to implement a given evolution affects the execution time, and the fidelity of the simulation decreases as the simulation time increases. In general, there is no universal procedure that is optimal for every situation, but their knowledge can help in choosing and defining the best way to implement a particular quantum algorithm. In this chapter a bunch of useful information is gathered about quantum gates, summarising properties and equivalences that can be used to optimise, manipulate and generate quantum circuits. A simple tool that we can use to construct a naive compiler is then presented. The chapter ends with an overview of the current methods that can be used to perform quantum gate decomposition, trying to motivate the fact that at this stage of quantum computing there is no optimal method for doing this, but the choice strongly depends on the problem we are approaching.

The quantum gate decomposition has to take into account not only the particular physical problem we are tackling, but also the quantum machine we are using. This point can be summarised in what is called *machine aware compilation*. Chapter 3 (*Trapped-ion based quantum computing*) introduces the special architecture based on trapped ions and highlights its advantages. This type of hardware is used in the last part of this thesis for a practical application of a simulation of many-body quantum systems.

The aim of Chapter 4 (*An explicit tensor notation for quantum computing*) is to develop a new explicit tensor notation, which, at the best of my knowledge, has never been used in any work related to quantum computing developed by physicists, but which may be more intuitive for mathematicians. Two notations are commonly used in the literature to describe quantum computing theory. The first is the Dirac notation and the second is based on what is known as the computational basis. The main disadvantage of the latter is the exponential growth of vector and matrix dimensions, which makes classical computing resources insufficient to predict the evolution of a number of qubits just above ten. The second disadvantage of this notation

is that it hides some relevant quantum properties of the operations by increasing the apparent number of independent variables. This chapter presents a third possible notation which is explicitly tensorial with particular attention on its advantages in terms of detection of separable and entangled systems, and in terms of quantum gate representations. The notation is applied to the description of the remarkable quantum teleportation algorithm. This chapter is the one that most diverges and differs from the rest of the work presented in this text and comes from a transdisciplinary collaboration with the mathematics department.

The last two chapters present a practical application of quantum computing to simulate a quantum many-body system governed by weak interactions. Chapter 5 (*Collective flavour oscillation*) presents the basic concept behind the physics of neutrinos in high-density situations, and the detailed derivation of the Hamiltonian governing the collective flavour oscillation phenomena. These phenomena are of particular interest in extreme astrophysical settings such as core-collapse supernovae, neutron star mergers and the early universe, as demonstrated by previous research. The dynamics are particularly interesting and hard to solve on a classical computer, and therefore provide a useful context in which to explore the benefits of quantum computing.

Finally, the aim of Chapter 6 (*Quantum algorithm and results*) is to design, optimise, apply and analyse a quantum algorithm allowing the dynamics of collective flavour oscillations to be simulated on a trapped-ion-based quantum computer. This is an application of quantum computing to the so-called Hamiltonian simulation, which requires the quantum gate decomposition of the real-time unitary propagator generated by the system Hamiltonian. After the quantum gate decomposition and the machine-aware optimisation, the quantum circuit implementing the algorithm has been run on the *Quantinuum* machine and the results are presented and discussed. The chapter ends with a section giving a comprehensive analysis of the Trotter error and evaluating the complexity scaling of the algorithm, whose explicit calculations are reported in the Appendix A (*Full Trotter error calculations*).

Contents

Outline	5
1 Introduction	11
1.1 Quantum computing in a nutshell	12
1.1.1 Qubits with binary mapping to states	12
1.1.2 Measurement, quantum superposition and entanglement	13
1.1.3 Quantum algorithms and circuits	14
1.1.4 Operation on qubits and quantum gate decomposition	15
1.2 Hamiltonian simulation	16
1.2.1 Efficiency of a quantum algorithm	17
1.2.2 Ingredients for the Hamiltonian simulation	17
2 Quantum gate decomposition and circuits	19
2.1 Notation	20
2.2 Quantum circuit equivalences	21
2.2.1 Identity gates	21
2.2.2 Inverse rotations	21
2.2.3 Rotation mergers	21
2.2.4 Hadamard properties	22
2.2.5 Commutation properties	22
2.2.6 Control-target inversion	22
2.2.7 CNOT properties	23
2.2.8 Local action of an entangling circuit	24
2.3 Naive compiler	25
2.3.1 Euler decomposition for single qubit gates	25
2.3.2 Fully controlled gate decomposition	26
2.3.3 Two-level matrix decomposition	30
2.3.4 Compiler	33
2.4 Methods for quantum gate decomposition	34
2.4.1 An overview	34
2.4.2 The magic basis	36
2.4.3 Implementation of Pauli string-generated operations	40
2.4.4 Walsh basis decomposition	41
3 Trapped-ion based quantum computing	45
3.1 Trapped-ion system and Hamiltonian	45
3.1.1 Free term	46
3.1.2 Trap term	46
3.1.3 Atom-light interaction term	46

3.1.4	The Lamb-Dicke regime	47
3.1.5	The rotating wave approximation	48
3.2	Di Vincenzo criteria for trapped-ion QC	51
3.2.1	Qubit definition	51
3.2.2	Initialization	51
3.2.3	Coherence	51
3.2.4	Universal quantum gate set	52
3.2.5	Measurement	53
3.3	General features of a trapped-ion machine	54
4	An explicit tensor notation for quantum computing	55
4.1	Mathematical background	56
4.1.1	Tensor product space	56
4.1.2	Operations on tensors	57
4.1.3	Natural maps defined by tensors	58
4.1.4	Tensor rank decomposition	58
4.2	Quantum computing in standard notation	58
4.2.1	Qubit states as vectors	58
4.2.2	Quantum gates as matrices	59
4.2.3	Local gate detection	60
4.2.4	Controlled gates	60
4.3	Quantum computing in tensor notation	61
4.3.1	Qubit states as tensors	62
4.3.2	Separable and entangled states	63
4.3.3	Quantum local gates as multilinear maps	64
4.3.4	Controlled gates as quasi-multilinear maps	69
4.3.5	Quantum non-local gates	72
4.3.6	Measurement interpretation	73
4.4	The quantum teleportation algorithm in tensor formalism	73
4.4.1	Quantum state evolution	74
4.4.2	Measurement and classical communication	76
4.5	Circuit complexity and quantum gate rank	76
4.5.1	Rank and number of CNOTs	76
4.5.2	Further analysis	79
5	Collective flavour oscillation	81
5.1	Neutrinos in a high-density environment	81
5.1.1	Neutrino mixing	82
5.1.2	Supernovae neutrinos	83
5.1.3	Why we care about supernovae neutrinos	85
5.2	Hamiltonian formalism in two-flavor approximation	86
5.2.1	Vacuum term	87
5.2.2	Matter term	88
5.2.3	Neutrino-neutrino term	90
5.2.4	Total isospin Hamiltonian	91

6	Quantum algorithm and results	93
6.1	System model and theoretical evolution	94
6.1.1	Encoding map	95
6.1.2	Ideal unitary propagator	96
6.2	Trotter decomposition of unitary propagator	98
6.2.1	Pair order and qubit topology	100
6.2.2	Pair order and Trotter error	101
6.3	Optimal quantum gate decomposition	105
6.4	Machine aware compilation	110
6.4.1	Quantinuum system model	111
6.4.2	Optimal quantum circuit	112
6.5	Results	113
6.5.1	Statistical error	114
6.5.2	Single Trotter step evolution	115
6.5.3	Multiple Trotter step evolution	119
6.5.4	Machine-aware vs automatic compilation	121
6.6	Error scaling and gate cost	121
6.6.1	First order Trotter error for the interaction Hamiltonian	121
6.6.2	Second order Trotter error for the interaction Hamiltonian	124
6.6.3	Qubitisation scaling	126
6.6.4	Real numerical cost	128
	Conclusions	129
	Bibliography	135
	Appendices	141
A	Full Trotter error calculations	143
A.1	Explicit calculations for the first order Trotter error	143
A.2	Second order Trotter error for the interaction Hamiltonian	145

Chapter 1

Introduction

Quantum computing uses the principles of quantum mechanics to revolutionise the way we process and manipulate information. Unlike classical computers, which operate on bits of information $b \in \mathbb{Z}_2 = \{0, 1\}$ represented as either 0 or 1, quantum computers use quantum bits $|q\rangle \in \mathbb{C}^2$, or qubits, which can exist in superpositions of $|0\rangle$ and $|1\rangle$ at the same time. At the heart of quantum computing are the amazing phenomena of quantum superposition and entanglement. Superposition allows qubits to occupy multiple states simultaneously, exponentially increasing computational power compared to classical bits. Entanglement, on the other hand, allows qubits to be correlated in such a way that the state of one qubit depends on the state of another, regardless of their physical distance.

The power of quantum computing lies in its ability to perform certain calculations and solve complex problems exponentially faster than classical computers. This potential has profound implications for a wide range of fields, including cryptography, optimisation, materials science, artificial intelligence and particle physics. However, realising the potential of quantum computing does not come without challenges. Quantum systems are highly sensitive and prone to noise and decoherence, which can lead to computational errors. Building and maintaining stable qubits and developing robust error correction techniques are critical areas of research in this field. Several approaches are being pursued in the development of quantum computers. One of the most prominent is based on the use of trapped ions (as Quantinuum and IonQ), where individual ions are manipulated to represent qubits. Other approaches include superconducting circuits (as Google and IBM), topological qubits, photon-based systems and neutral atoms (as QuEra and Pasqal). Each approach has its own advantages and challenges, and researchers are actively exploring different ways to build practical and scalable quantum computers. In recent years, quantum computing has made significant progress, even with claims of having demonstrated quantum supremacy - the ability of a quantum computer to solve a problem that would be practically impossible for classical computers. However, quantum computing is still in its infancy and many technological and theoretical hurdles must be overcome before it becomes a widespread reality. Despite these challenges, quantum computing is an exciting and rapidly evolving field that holds great promise for the future of computing. Its ability to harness the principles of quantum mechanics to process information in fundamentally new ways opens up a realm of possibilities for solving complex problems that are beyond the reach of classical computers.

This chapter introduces the basic concepts of quantum computing (Sec. 1.1), focusing on the representation of single and multi-qubit states, the concept of quantum superposition in the context of measurement, and the quantum circuit formulation of quantum algorithms. A key concept is quantum gate decomposition, which is related to the implementation of quantum algorithms designed to simulate the dynamic evolution of multi-particle quantum systems -

the so-called Hamiltonian simulations (Sec. 1.2). This always requires two main ingredients, an encoding map and a unitary implementation, which must take into account the physical problem and aim to produce an efficient quantum algorithm.

1.1 Quantum computing in a nutshell

1.1.1 Qubits with binary mapping to states

We can define a qubit as a two-level quantum system belonging to the Hilbert space $\mathcal{H} = \mathbb{C}^2$. A qubit state is identified, in Dirac notation, by a unit vector $|\varphi\rangle \in \mathbb{C}^2$ which satisfies the normalisation condition $\| |\varphi\rangle \| = 1$ and the equivalence up to a global phase $|\varphi\rangle \cong e^{i\phi} |\varphi\rangle$, reflecting the fact that it is physically impossible to access the global phase of a wavefunction. One of the main properties of a qubit with respect to a classical bit is the so-called quantum superposition, according to which the state of a single qubit can be in any linear combination of the basis elements $\mathcal{B} = \{|0\rangle, |1\rangle\} \subset \mathbb{C}^2$ of the Hilbert space, namely

$$|\varphi\rangle = \alpha |0\rangle + \beta |1\rangle, \quad (1.1.1)$$

where $\alpha, \beta \in \mathbb{C}$. The linear combination must satisfy the normalisation condition $|\alpha|^2 + |\beta|^2 = 1$, which guarantees $\| |\varphi\rangle \| = 1$, and the equivalence up to the global phase. These two conditions together allow us to identify a state with only two parameters according to the parameterisation

$$|\varphi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle, \quad (1.1.2)$$

where $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi)$. This parameterisation is consistent with the Bloch sphere representation of a single qubit, according to which there is a one-to-one correspondence between the single qubit states and the points on the surface of the S_2 sphere in \mathbb{R}^3 with radius $r = 1$. The parameterisation is the same as in Eq. (1.1.2) if we place the basis elements along the z axis and identify the zenith angle with θ and the azimuth angle with ϕ , as shown in the Figure 1.1. Using this notation, a qubit state can also be represented as a three-dimensional unit vector

$$|\varphi\rangle = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \sin \theta \cos \phi \\ \sin \theta \sin \phi \\ \cos \theta \end{pmatrix}, \quad (1.1.3)$$

expressed in Cartesian or spherical coordinates. On the other hand, one of the most common way to express the state of a qubit is the one based on the canonical basis

$$\mathcal{B} = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\} \subset \mathbb{C}^2, \quad (1.1.4)$$

also called computational basis, on which a qubit state is a two-dimensional normalised vector

$$|\varphi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \cos \theta/2 \\ e^{i\phi} \sin \theta/2 \end{pmatrix} \in \mathbb{C}^2. \quad (1.1.5)$$

It is well known that the physics of a two-level quantum system can be described by the $SU(2)$ group, whose algebra $\mathfrak{su}(2)$ is generated by the Pauli matrices, which on the canonical basis correspond to the following hermitian, unitary, traceless, (-1)-determinant matrices

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (1.1.6)$$

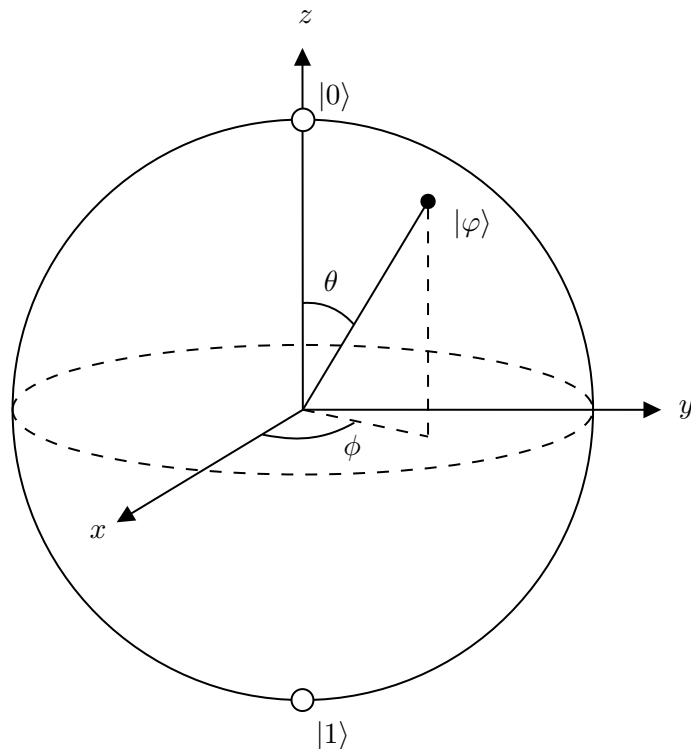


Figure 1.1: Bloch sphere representation of a single-qubit state.

It is easy to prove that the Cartesian coordinates in Eq. (1.1.3) correspond to the expectation values of the Pauli matrices on the qubit state expressed as a two-dimensional vector, i.e. $k = \langle \varphi | \sigma_k | \varphi \rangle$, where $k \in \{x, y, z\}$, due to the isomorphism $\mathcal{SU}(2) \cong \mathcal{SO}(3)/\mathbb{Z}_2$.

When considering a system composed of n qubits, the states characterising it belong to the tensor product Hilbert space $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$ of dimension $\dim(\mathcal{H}) = 2^n$, whose states are generally identified by linear combination of all basis elements

$$|\varphi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle, \quad (1.1.7)$$

where $\alpha_i \in \mathbb{C}$ are the complex coefficients of the linear combination, $\sum_i |\alpha_i|^2 = 1$, and $|i\rangle$ are the basis elements of

$$\mathcal{B} = \{|i\rangle, \text{ s.t. } i \in \{0, \dots, 2^n - 1\}\}, \quad (1.1.8)$$

and can be expressed in binary or decimal notation. $|i\rangle$ is in decimal notation if $i \in \{0, \dots, 2^n - 1\}$ and $||[i]\rangle$ is in binary notation if $[i]$ is the bit string representation of the decimal number i with the most significant bit on the left. Using the common computational basis, each basis element $|i\rangle$ corresponds to a 2^n dimensional vector with only one non-zero entry at position i .

1.1.2 Measurement, quantum superposition and entanglement

In the transition from classical to quantum computing, there is an evident increase in the amount of information stored in the information unit. In fact, a bit can take either of the two possible values 0 and 1, instead a qubit can be in any possible linear combination of the basis elements $|0\rangle$ and $|1\rangle$. There is another crucial difference between the information stored in a bit and that stored in a qubit, and it relates to the probabilistic nature of the measurement of a qubit state. When we look at a bit state, we observe with absolute certainty a certain value in $\mathbb{Z}_2 = \{0, 1\}$.

On the other hand, when we measure a qubit, we observe the state $|0\rangle$ with probability $|\alpha|^2$ and the state $|1\rangle$ with probability $|\beta|^2$. This means that we can only get probabilistic information about the state by measuring the same qubit state $|\varphi\rangle$ several times and sampling its probability distribution. The post-measurement state changes due to the measurement itself and collapses to a well-defined basis element. For this reason the measurement procedure can be seen as a projection along the basis and it physically causes the wave function collapse in the obtained basis element. This means that if we measure $|\varphi\rangle$ and get $|0\rangle$, a later measurement will give $|0\rangle$ with probability 1. This characterises the measurement procedure as an irreversible operation.

Of particular interest is the concept of measuring a many-qubit system. In this case, the measurement is not only related to the quantum superposition property, but also to the entanglement property. It is well known that when we measure a qubit that is entangled with another one, we automatically cause the collapse of the state of the other qubit that is entangled with the first one, even if we haven't touched it. This brings into the field of quantum computing a subtle difference between the concept of a qubit and that of a qudit. Let's consider three qubits $|q_0\rangle$, $|q_1\rangle$ and $|q_2\rangle$ and a general state indicated by $|\varphi\rangle_{qqq}$. The whole system belongs to the Hilbert space $\mathcal{H} = \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2$ with dimension $\dim(\mathcal{H}) = 2^3 = 8$. Using the isomorphism $(\mathbb{C}^2)^{\otimes 3} \cong \mathbb{C}^8$, we can describe the three-qubit state with a single qudit of dimension 8 (i.e a three-level quantum system). This is true in general, but we must be careful when dealing with partial measurements of the system. Indeed, when we measure the state of the qudit, we collapse the state to a basis element $|i\rangle_{qudit}$, where $i \in \{0, \dots, 7\}$, which corresponds to a basis element of three qubits. On the other hand, if we're dealing with three different qubits, we can assume that only a subset of them can be measured. Let's consider, for example, the three-qubit system in the state

$$|\varphi\rangle_{qqq} = \frac{|000\rangle + |110\rangle}{\sqrt{2}} := |\Phi^+\rangle \otimes |0\rangle \in (\mathbb{C}^2)^{\otimes 3}, \quad (1.1.9)$$

where the first two qubits are in an entangled (Bell) state and the third is separated. The same state can be described by the following qudit state

$$|\varphi\rangle_{qudit} = \frac{|0\rangle + |6\rangle}{\sqrt{2}} \in \mathbb{C}^8, \quad (1.1.10)$$

which is a superposition of two basis elements. If we measure it, we can either get $|0\rangle$ or $|6\rangle$, which corresponds to $|000\rangle$ and $|110\rangle$ in the space $(\mathbb{C}^2)^{\otimes 3}$, with the same probability 1/2. If we measure the third qubit of the three-qubit system instead, we won't change the overall state, and the first two qubits will be in an entangled state $|\Phi^+\rangle$ again. This is a link between quantum superposition and entanglement. In the first case we observe a partially entangled state and in the second we have a system in a superposition state. The description is exactly the same in both cases, but the states behave differently under a partial measurement because quantum superposition and entanglement are physically different properties.

1.1.3 Quantum algorithms and circuits

As first hypothesized by Feynman [1], quantum computers can in principle help to fight the exponential growth of resources needed to simulate quantum systems of increasing size. To appreciate that, consider the classical memory required to store a fully general state $|\varphi\rangle$ of n qubits. The Hilbert space $\mathcal{H}_{qubit} = (\mathbb{C})^{\otimes n}$ is spanned by 2^n orthogonal states, and the n qubits can be in a superposition of all of them. To store this description of the quantum state in a classical computer, we need to store all 2^n complex numbers of the linear combination. The inefficiency of a classic computer to describe a quantum system is evident. But if a system can simulate its own evolution it is possible to arrive at the solution of a certain class of problems

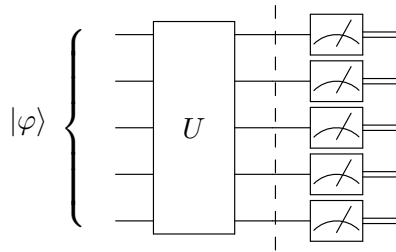


Figure 1.2: Schematic representation of a quantum circuit.

exponentially faster than any classical computation can do. It was Deutsch [2] that first formalized a standardized quantum computer model, and defined the classes of problems for which it would become dominant. According to the Deutsch model, given a system in a pure state $|\varphi\rangle \in \mathcal{H}_{qubit}$, a quantum algorithm consists of a unitary transformation which produces a certain final state $|\varphi'\rangle = U|\varphi\rangle \in \mathcal{H}_{qubit}$, also represented by

$$|\varphi\rangle \xrightarrow{U} |\varphi'\rangle. \quad (1.1.11)$$

It is then possible to perform quantum measurements yielding the probability of finding the systems in a given state of a given basis, probability that constitutes the result of the calculation. So, a quantum algorithm is by definition a unitary transformation U of an initial qubit state $|\varphi\rangle$ into a final one $|\varphi'\rangle = U|\varphi\rangle$ according to quantum mechanical rules and we usually represent this by a quantum circuit as the one in Fig. 1.2. We can identify three main steps in a quantum algorithm:

- (i) **Initialisation** of the qubits in a desired initial state $|\varphi\rangle$. The horizontal lines are called quantum wires and the set of qubits involved is the quantum register. If n is the number of qubits in the system, the states belong to the Hilbert space $\mathcal{H} = (\mathbb{C}^2)^{\otimes n} \ni |\varphi\rangle$ and can be written as in Eq. (1.1.7).
- (ii) **Unitary evolution** of the initial state, at the end of which we have a final quantum state $|\varphi'\rangle = U|\varphi\rangle$. This part is the heart of the quantum algorithm and is usually the most difficult step in developing simulations. If the system contains n qubits, the unitary transformation is an element of the $\mathcal{U}(2^n) \ni U$ group. The final state can be expressed in a similar way as the one in Eq. (1.1.7) with different coefficients α'_i .
- (iii) **Measurement** of the final state $|\varphi'\rangle$ that allows us to extract probabilistic information about the evolved state and constitutes the result of the simulation. To extract the probability distribution of $|\varphi'\rangle$, the simulation must be run several times. After the measurement, all the qubits collapse to a basis element (see Eq. (1.1.8)) carrying classical information in the form of an n -bit string. After the measurement, each qubit is identified by a classical wire, denoted by two horizontal lines, the set of which constitutes the classical register.

1.1.4 Operation on qubits and quantum gate decomposition

In general, quantum computers can be divided into two main categories: analog quantum machines, which is a type of quantum computing architecture that operates using continuous variables, and digital quantum machines, also known as gate-based quantum computers. The latter is a quantum computing architecture in which computation is performed by applying a finite sequence of quantum logic gates to manipulate and transform the qubit state. Each quantum logic gate is defined as a unitary transformation of a small set of qubits, usually one or two, belonging

to a library, i.e. a universal gate set, proper to a particular quantum testbed. Quantum gates that act on a single qubit $|q_i\rangle$ belong to the group $\mathcal{U}(2) \ni G_i$, and are represented, on the computational basis, by 2×2 unitary matrices. Instead, operations acting on two qubits $|q_i\rangle \otimes |q_j\rangle$ belong to the unitary group $\mathcal{U}(4) \ni G_{ij}$ and can be represented by 4×4 unitary matrices. The main property of a two-qubit gate belonging to the library is the ability to generate entanglement between the qubits on which it acts. Note that each library contains a particular subset of $\mathcal{U}(2)$ and $\mathcal{U}(4)$ that can be engineered by the quantum hardware. When designing a quantum algorithm for a digital quantum computer, the entire unitary transformation $U \in \mathcal{U}(2^n)$ must be broken down into a finite sequence of elementary gates that can be implemented by the quantum computer. This process is known as quantum gate decomposition, and it is the most important and difficult step in any digital quantum algorithm design.

The Solovay-Kitaev theorem (see for example Refs. [3, 4]) guarantees that a finite sequence of operations can be found to implement U with an arbitrarily small error, and that, for a given error, the length of such sequence scales logarithmically with the inverse of the error itself. More formally, for any $\varepsilon > 0$ and $U \in \mathcal{U}(2^n)$, there exists a constant c and a gate sequence \tilde{U} of length $\mathcal{O}(\log^c(1/\varepsilon))$ which approximates U with precision ε , namely

$$\|\tilde{U} - U\|_{op} \leq \varepsilon, \quad (1.1.12)$$

where $\|\cdot\|_{op}$ is the operator norm defined by:

$$\|\mathcal{A}\|_{op} := \sup_{|\varphi\rangle \neq 0} \frac{\|\mathcal{A}|\varphi\rangle\|}{\|\varphi\rangle}. \quad (1.1.13)$$

Quantum circuits are prone to errors arising from various sources, including different types of noise, decoherence and imperfect gate operations. Optimisation strategies are therefore needed to improve the fidelity and accuracy of quantum computation. Since every operation performed on a qubit introduces an error, a necessary strategy to ensure the robustness of a quantum algorithm is to optimise the quantum gate decomposition procedure. This consists of finding the shortest possible gate sequence that implements the desired unitary.

We define a *layer* as an n -qubit global operation composed of smaller operations that can be performed simultaneously. For example, we can have a separable layer like $U_l = G_0 \otimes \dots \otimes G_{n-1}$, where $G_i \in \mathcal{U}(2)$, or an entangling layer that can act non-trivially on two or more pairs of qubits. For example, $U_l = G_{01} \otimes \mathbb{I}_{n-2}$ which contains only one two-qubit gate, or $U_l = G_{01} \otimes G_{23} \otimes \mathbb{I}_{n-4}$, which entangles two pairs. All qubits not involved in the action are acted on by a tensor product of identity operators. An optimal quantum circuit involves the smallest possible number of layers and the maximum simplicity of each layer, for example by preferring single qubit gates to entangled ones.

1.2 Hamiltonian simulation

For the physics community, one of the key areas of focus in quantum computing is the simulation of complex quantum systems, in particular the simulation of Hamiltonians. The Hamiltonian plays a fundamental role in quantum mechanics, representing the energy operator that governs the behaviour of quantum systems. It describes the dynamics and evolution of physical systems by capturing the relationships between different quantum states. The simulation of Hamiltonians is crucial for understanding and predicting the behaviour of quantum systems, allowing researchers to study the properties of molecules, materials and other quantum phenomena. The simulation of Hamiltonians becomes particularly challenging as the size and complexity of the

quantum system increases. Classically, the simulation of such systems becomes incredibly difficult due to the exponential growth in the number of parameters required to describe the quantum states. Important exceptions to this behavior are found, for instance, in stabilizer states [5, 6] or in systems with low levels of bipartite entanglement [7, 8]. However, quantum computers offer a promising way to efficiently simulate Hamiltonians by exploiting the inherent parallelism and superposition properties of quantum mechanics [1, 9]. The advantage of this is that we do not need an exponential amount of computational resources as the size of the system increases. For example, consider a physical system consisting of n two-level quantum systems (such as n spins). The wave function $|\psi\rangle \in (\mathbb{C}^2)^{\otimes n}$ of this system can be a general linear combination of all the basis elements as in Eq. (1.1.7). To simulate the evolution of this system we need to store 2^n complex coefficients and the needed resource increases exponentially by increasing n ¹. On the other hand, using a machine that is intrinsically quantum, we would need a linear amount of qubits. This property is called *exponential speedup*.

The most natural use of quantum computers is to implement Hamiltonian simulation algorithms aiming to efficiently implement the time evolution of quantum systems governed by the Schrödinger equation

$$|\psi(t)\rangle = e^{-iHt} |\psi(0)\rangle, \quad (1.2.1)$$

where H is the Hamiltonian of the system. The fact that the operator $e^{-iHt} := U(t)$ is unitary allows us to design a suitable quantum algorithm to implement it. Hamiltonian simulation is thus at the heart of quantum computing, providing a powerful tool for exploring and understanding quantum systems by naturally predicting their evolution and taking into account all quantum properties such as superposition, interference and entanglement.

1.2.1 Efficiency of a quantum algorithm

When using a classical approach to solve the dynamical problem in Eq. (1.2.1), classical computers become increasingly expensive as the size of the system increases, due to both memory requirements and operational costs. On the other hand the application of quantum computing to solve quantum system dynamics presents an exponential speedup showing advantage in the memory required and in the computational cost of the propagation that classically would require to perform $2^n \times 2^n$ matrix multiplications. For this reason, a large part of the scientific community is investing efforts in the development of quantum algorithms that predict the evolution of multi-body physical systems (see, e.g., Ref. [10] for a recent review).

The efficiency of such algorithms is defined according to various criteria, which include different characteristics of the algorithms. First, the memory required to store the degrees of freedom of interest within the qubit states plays a fundamental role, since it places a limit on the size of the systems that we can describe with a well-defined set of qubits. The second fundamental aspect to be analysed in order to define an algorithm as efficient is the study of the complexity of the algorithm itself. By complexity we mean the number of elementary operations that a quantum circuit contains, and this obviously influences the quality of the results obtained from a simulation, since each operation introduces errors. Therefore, both properties (memory requirements and circuit complexity) have to be analysed in relation to the size of the physical system to be described.

1.2.2 Ingredients for the Hamiltonian simulation

In order to carry out a quantum Hamiltonian simulation one always needs two ingredients.

¹For example, if we fix $n = 27$ and use 32 bits to store each float number α_i , we would need $2^{27} \cdot 32$ bits = 2^{27} bytes = 8^9 bytes, which is almost 1 Gbyte of memory just to store one wavefunction in a classical computer.

- (i) **State encoding map** $|\psi\rangle \mapsto |\varphi\rangle$ which makes it possible to identify the states of the system $|\psi\rangle \in \mathcal{H}_{system}$ we want to describe with the states of the system of qubits $|\varphi\rangle \in \mathcal{H}_{qubit}$ allowing for initialization and measurement interpretation of the quantum simulation. The number of qubits required to store the wave function of the physical system quantifies the complexity of the algorithm in terms of the memory required, and generally scales linearly, depending on the chosen mapping, with the number of particles in the physical system.
- (ii) **Operator-gate compilation** $U(t) \mapsto \tilde{U}$, also called quantum gate decomposition in the case of digital quantum computing, that consists of encoding of the real-time propagator

$$U(t) = e^{-iHt} \tag{1.2.2}$$

in a language understandable by the quantum machine being used. This procedure consists in decomposing $U(t)$ into a sequence of quantum gates (grouped in layers) from a fixed set. It is the most challenging part of creating quantum algorithms and more importantly it depends on the machine used. The time evolution operator $U(t)$ is represented on the computational basis by a $2^n \times 2^n$ unitary matrix, whose dimension obviously grows exponentially with the number of qubits. The quantum gate decomposition of this type of operation would generally require an exponential number of elementary operations if one used a standard universal way to compile it. This would make any quantum algorithm inefficient due to the enormous complexity in terms of the number of operations. To avoid this, one has to use smart expedients to reduce the complexity to polynomial scaling. For example, taking into account the physics of the system, the presence of possibly sterile degrees of freedom, the presence of symmetries in the Hamiltonian, or the presence of reducible subspaces in the Hilbert space could help to reduce the number of layers required. If an approximate implementation is required, the Solovay-Kitaev theorem ensures that a finite sequence exists and the length can be reduced by accepting a larger error. In the latter case, the analysis of the scaling of the error is necessary to prove the efficiency of the algorithm.

Chapter 2

Quantum gate decomposition and circuits

Quantum gate decomposition is a procedure that consists of translating a unitary $U \in \mathcal{U}(2^n)$, acting on n qubits, into a language understandable by a quantum computer. This requires breaking down the entire operation U into a sequence of simpler and more elementary gates, arranged in layers. Qubit evolutions are represented by $2^n \times 2^n$ unitary matrices, which describe the transformations of quantum states according to quantum mechanics. However, as mentioned in Chapter 1, not all unitary matrices can be implemented directly by physical quantum hardware. It is therefore essential to decompose them into a sequence of basic operations that can be easily implemented by the machine. The process of quantum gate decomposition is driven by the principles of quantum circuit synthesis and compilation. By using various decomposition techniques and algorithms, some of which are presented in this chapter, the large unitary can be expressed as a sequence of simpler gates that are available in the quantum computing architecture and therefore belong to the respective library. Moreover, the decomposition of quantum gates plays a crucial role in fault-tolerant quantum computing in the sense that an optimal implementation of a unitary is essential to efficiently perform a quantum simulation, especially for near-term quantum computers. In fact, they are limited by a finite fidelity of the implemented unitaries and a finite coherence time due to the unavoidable coupling between the qubit system and the environment. The fidelity of the simulation decreases as the simulation time increases, and the number of gates we use to implement a given evolution affects the execution time. Finding the optimal decomposition is therefore crucial to ensure the desired accuracy of a simulation.

In this chapter, after defining the notation used (Sec. 2.1), a list of useful circuit equivalences is given (Sec. 2.2). The concept of quantum gate decomposition is then explained in more detail, and the sections even provide a simple tool that we can use to construct a rudimentary compiler (Sec. 2.3). Note, however, that the main goal here is not to find the fastest way to decompose these quantum building blocks, but to show how important and complex this process is in the world of quantum computing. At the end of this chapter there is a tentative of overview of the current methods that can be used to perform quantum gate decomposition (Sec. 2.4), trying to motivate the fact that at this stage of quantum computing there is no optimal method for doing this, but the choice strongly depends on the problem we are approaching.

2.1 Notation

In the following sections and chapters we will use the following convention for one and two qubit gates. First, the Pauli matrices

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad (2.1.1)$$

also denoted as σ_k for $k \in \{x, y, z\}$, form a complete basis for the $SU(2)$ group and can therefore be used as generators of its algebra $\mathfrak{su}(2)$. Any single qubit rotation of an angle $\alpha \in \mathbb{R}$ can be expressed as

$$R_{\hat{n}}(\alpha) = e^{i\frac{\alpha}{2}\hat{n}\cdot\sigma} = \cos\left(\frac{\alpha}{2}\right)\mathbb{I} + i\sin\left(\frac{\alpha}{2}\right)\hat{n}\cdot\sigma, \quad (2.1.2)$$

where $\sigma = (X, Y, Z)$ is the vector of Pauli matrices and \hat{n} is the unit 3-dimensional vector indicating the direction of the Bloch sphere with respect to which we are performing the rotation. By setting the unit vector \hat{n} to $(1, 0, 0)$, $(0, 1, 0)$ or $(0, 0, 1)$ we can define the elementary rotations with respect to the x , y and z axes respectively (see Fig. 1.1 of Chapter 1). On the computational basis the elementary rotations correspond to the unitary matrices

$$R_x(\alpha) = \begin{pmatrix} \cos(\alpha/2) & i\sin(\alpha/2) \\ i\sin(\alpha/2) & \cos(\alpha/2) \end{pmatrix}, \quad (2.1.3)$$

$$R_y(\alpha) = \begin{pmatrix} \cos(\alpha/2) & \sin(\alpha/2) \\ -\sin(\alpha/2) & \cos(\alpha/2) \end{pmatrix} \quad (2.1.4)$$

and

$$R_z(\alpha) = \begin{pmatrix} e^{i\alpha/2} & 0 \\ 0 & e^{-i\alpha/2} \end{pmatrix}. \quad (2.1.5)$$

Other very common single-qubit gates are the Hadamard gate and the phase gate. The first is represented by the matrix

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad (2.1.6)$$

and is used to generate a superposition state starting from a basis state, i.e.

$$H : |0\rangle \mapsto \frac{|0\rangle + |1\rangle}{\sqrt{2}} := |+\rangle, \quad H : |1\rangle \mapsto \frac{|0\rangle - |1\rangle}{\sqrt{2}} := |-\rangle. \quad (2.1.7)$$

The second one corresponds to

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \quad (2.1.8)$$

and add a phase only on the $|1\rangle$ contribution. All single qubit gates are represented, in terms of quantum circuit, by a box

$$\text{---} \boxed{G} \text{---}, \quad (2.1.9)$$

where $G \in U(2)$. The most common two-qubit gate is the CNOT, which is the quantum equivalent of the classical control NOT, and applies an X gate to the target qubit depending on the state of the control qubit. In matrix notation this corresponds to

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad (2.1.10)$$

which acts on the second qubit and is controlled by the first one, and

$$\text{CNOT}_{inv} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad (2.1.11)$$

which acts on the first one depending on the state of the second. The corresponding quantum circuits are

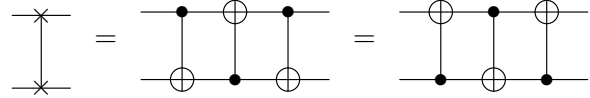


$$(2.1.12)$$

The SWAP operation that invert two qubit state is

$$\text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.1.13)$$

and in terms of quantum circuit it corresponds to



$$(2.1.14)$$

2.2 Quantum circuit equivalences

In this section we summarise some useful properties that can be used to manipulate, adapt to the qubit topology and optimise a quantum circuit. Some of these properties will be used later to optimise quantum circuits and to define quantum gate decomposition procedures. Some of the information presented here can be found in more specific texts such as [4] and also in Ref. [11], where the authors used some quantum circuit equivalences to obtain the quantum teleportation protocol, starting from the SWAP gate circuit and applying only equivalence properties.

2.2.1 Identity gates

Some special sequences of operations have a trivial effect and can therefore be replaced by an identity. For example, gates followed by their inverse $GG^\dagger = \mathbb{I}$, gates with the control qubit in the $|0\rangle$ state and gates acting on their positive eigenstate (for example $X|+\rangle = |+\rangle$ and $Z|0\rangle = |0\rangle$).

2.2.2 Inverse rotations

The rotation of single qubits along the axes y and z can be inverted if they are between two X gates according to the relations

$$XR_y(\alpha)X = R_y(-\alpha), \quad XR_z(\alpha)X = R_z(-\alpha). \quad (2.2.1)$$

2.2.3 Rotation mergers

A simple feature that can be used to reduce the number of single qubit operations is to merge similar qubit rotations that are performed sequentially. In fact, we have

$$R_k(\alpha + \beta) = R_k(\alpha)R_k(\beta) \quad (2.2.2)$$

for all $k \in \{x, y, z\}$.

2.2.4 Hadamard properties

Some relations coming from the Hadamard gate can be used to change the basis of a qubit state. Consider, for example,

$$HXH = Z, \quad HYH = -Y, \quad HZH = X, \quad (2.2.3)$$

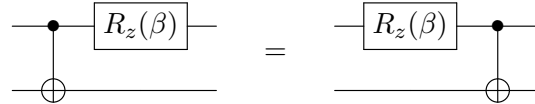
which more generally correspond to

$$HR_x(\alpha)H = R_z(\alpha), \quad HR_y(\alpha)H = R_y(-\alpha), \quad HR_z(\alpha)H = R_x(\alpha), \quad (2.2.4)$$

and can be used to go from one axis in the Bloch sphere to another.

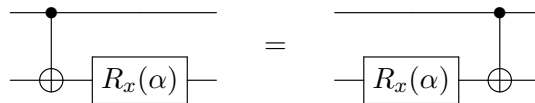
2.2.5 Commutation properties

Some equivalences can be used to change the order in which two operations act. For example, rotation with respect to the z axis always commutes with control because $R_z(\alpha)$ does not change the probability of measuring a state on the computational basis. So we have



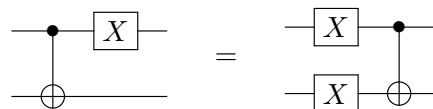
$$(2.2.5)$$

On the other hand, a rotation with respect to x always commutes with the target



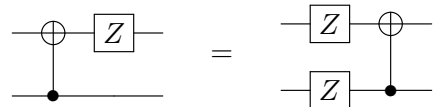
$$(2.2.6)$$

Note that the two properties above are derived from each other thanks to the introduction of Hadamard gates. This, together with the properties in Eq. (2.2.4), suggests a difference between the rotations R_x and R_z as opposed to R_y which does not have a similar equivalence. This is of course due to the use of the CNOT as the native two-qubit gate. Other interesting commutation properties include



$$(2.2.7)$$

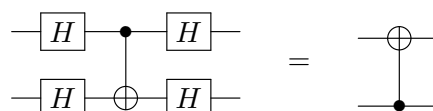
and



$$(2.2.8)$$

2.2.6 Control-target inversion

In some cases, reversing the action of a controlled gate can lead to a reduction in the quantum operation. The switching between the control and the target qubit can be done in several ways. For the CNOT case, we can go from the CNOT in Eq. (2.1.10) to that in Eq. (2.1.11) using Hadamard gates according to



$$(2.2.9)$$

or some similar equivalences¹, which can be proved using the identity in Eq. (2.2.3), according to which

$$\begin{array}{c} \bullet \\ | \\ \oplus \end{array} = \begin{array}{c} \bullet \\ | \\ \boxed{H} \boxed{Z} \boxed{H} \end{array} \quad (2.2.10)$$

together with the fact that the control-Z gate CZ, which applies a Z gate to the target qubit if the control qubit is in $|1\rangle$, is symmetric under the control and target exchange, namely

$$\begin{array}{c} \bullet \\ | \\ \boxed{Z} \end{array} = \begin{array}{c} \boxed{Z} \\ | \\ \bullet \end{array} , \quad (2.2.11)$$

since it changes the sign of the overall state only when both qubits are in the $|1\rangle$ state. Another similar property, which allows to invert the action performed on the target with that on the control qubit, is

$$\begin{array}{c} \bullet \\ | \\ \boxed{R_x(\alpha)} \\ | \\ \oplus \\ \boxed{R_z(\beta)} \end{array} = \begin{array}{c} \oplus \\ | \\ \boxed{R_z(\beta)} \\ | \\ \bullet \\ \boxed{R_x(\alpha)} \end{array} \quad (2.2.12)$$

for any $\alpha, \beta \in \mathbb{R}$. This can be easily proved by inserting two SWAPs and using the commutation properties in Eq. (2.2.5) and (2.2.6).

2.2.7 CNOT properties

Some manipulations can be used to adapt a network to a particular topology of the qubit architecture. In fact, different machines present a different connectivity between the qubits. In such a situation, two qubits to be entangled must be connected in hardware, and if this is not possible, other CNOTs can be added according to the equivalences

$$\begin{array}{c} \bullet \\ | \\ \oplus \end{array} = \begin{array}{c} \bullet \quad \bullet \\ | \quad | \\ \oplus \quad \oplus \\ | \quad | \\ \oplus \quad \oplus \end{array} = \begin{array}{c} \bullet \quad \bullet \\ | \quad | \\ \oplus \quad \oplus \\ | \quad | \\ \oplus \quad \oplus \end{array} , \quad (2.2.13)$$

from which we also obtain

$$\begin{array}{c} \bullet \quad \bullet \\ | \quad | \\ \oplus \quad \oplus \\ | \quad | \\ \oplus \quad \oplus \end{array} = \begin{array}{c} \bullet \\ | \\ \oplus \\ | \\ \oplus \end{array} . \quad (2.2.14)$$

Other useful equivalences are that two sequential CNOTs acting either on the same control qubit or on the same target one commute, namely:

$$\begin{array}{c} \bullet \quad \bullet \\ | \quad | \\ \oplus \quad \oplus \\ | \quad | \\ \oplus \quad \oplus \end{array} = \begin{array}{c} \bullet \quad \bullet \\ | \quad | \\ \oplus \quad \oplus \\ | \quad | \\ \oplus \quad \oplus \end{array} , \quad \begin{array}{c} \bullet \\ | \\ \oplus \\ | \\ \oplus \end{array} = \begin{array}{c} \bullet \\ | \\ \oplus \\ | \\ \oplus \end{array} . \quad (2.2.15)$$

¹such as

$$\begin{array}{c} \boxed{H} \bullet \boxed{H} \\ | \\ \oplus \end{array} = \begin{array}{c} \oplus \\ | \\ \boxed{H} \bullet \boxed{H} \end{array} , \quad \begin{array}{c} \bullet \boxed{H} \\ | \\ \oplus \end{array} = \begin{array}{c} \oplus \boxed{H} \\ | \\ \bullet \boxed{H} \end{array} .$$

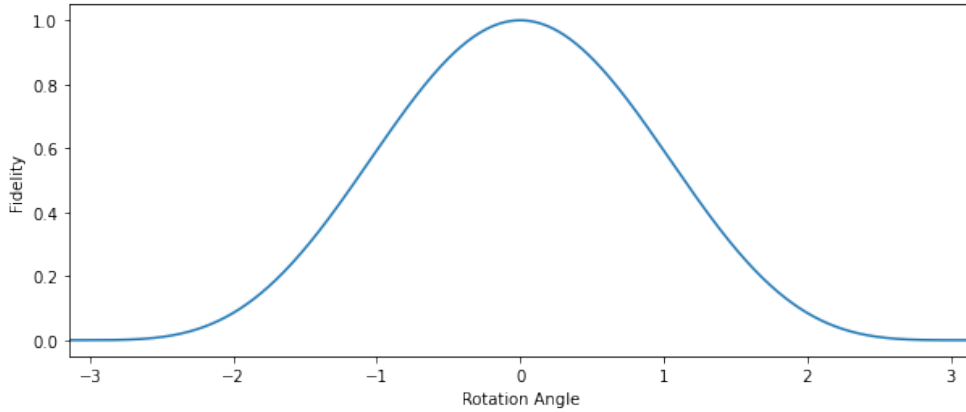


Figure 2.1: Fidelity, as a function of the angle θ , between a $R_z(\theta)$ rotation in the middle of two CNOTs and a single $R_z(\theta)$ rotation as suggested by Eq. (2.2.19). The fidelity between the two operations A and B , represented by matrices of dimension $d = 4$, is calculated as $\mathcal{F} = \text{tr}(A^\dagger B) \text{tr}(B^\dagger A)/d^2$.

If, instead, we want to swap two CNOTs, with one targeting the control of the other, we can use

$$\begin{array}{c} \bullet \\ \oplus \end{array} \begin{array}{c} \bullet \\ \oplus \end{array} = \begin{array}{c} \bullet \\ \oplus \end{array} \begin{array}{c} \bullet \\ \oplus \end{array} . \tag{2.2.16}$$

2.2.8 Local action of an entangling circuit

In some situations, a quantum circuit containing entangling gates can have the effective effect of a local operation. Some simple examples are:

$$\begin{array}{c} \bullet \\ \oplus \end{array} \begin{array}{c} \bullet \\ \oplus \end{array} \begin{array}{c} X \\ \oplus \end{array} = \begin{array}{c} X \\ \oplus \end{array} \begin{array}{c} X \\ \oplus \end{array} . \tag{2.2.17}$$

and

$$\begin{array}{c} \bullet \\ \oplus \end{array} \begin{array}{c} \bullet \\ \oplus \end{array} \begin{array}{c} Z \\ \oplus \end{array} = \begin{array}{c} Z \\ \oplus \end{array} \begin{array}{c} Z \\ \oplus \end{array} . \tag{2.2.18}$$

In Chapter 4 we will see a trick to detect whether a two-qubit operation is local or not. Another less trivial property in this context is the following approximation:

$$\begin{array}{c} \bullet \\ \oplus \end{array} \begin{array}{c} \bullet \\ \oplus \end{array} \begin{array}{c} R_k(\theta) \\ \oplus \end{array} \approx \begin{array}{c} R_k(\theta) \\ \oplus \end{array} . \tag{2.2.19}$$

for $k \in \{y, z\}$, whose error can be approximated by a Gaussian as suggested by the Figure 2.1, where we have plotted the fidelity between the operation in the left hand side of Eq. (2.2.19) and the one in the right hand side. This suggests that a small rotation in the middle of two CNOTs can be achieved by a small rotation alone. Of course, this is an approximation and introduces an angle-dependent error that must be taken into account. However, the theoretical error introduced by the approximation may be less than the error introduced by the hardware noise caused by the two CNOTs and the right part of Eq. (2.2.19) could therefore be advantageous.

2.3 Naive compiler

A very rudimentary compiler can be constructed by combining three decomposition methods. The first is the two-level matrix decomposition, which allows a large unitary to be decomposed into a product of more simple unitaries of the same dimension. These two-level matrices can be interpreted as fully controlled quantum gates, which can be decomposed into networks containing only CNOTs and a single qubit gate. Finally, the Euler decomposition can be used to implement each single-qubit operation in terms of elementary rotations. These three decompositions are presented in reverse order in the following sections.

2.3.1 Euler decomposition for single qubit gates

According to the Euler decomposition, also called zyz decomposition, we can write any special single-qubit quantum gate $SU \in \mathcal{SU}(2)$ as the composition of three elementary rotations

$$SU = R_z(\alpha)R_y(\theta)R_z(\beta), \quad (2.3.1)$$

where $\alpha, \theta, \beta \in \mathbb{R}$ are called Euler angles. In terms of quantum circuit, the single-qubit gate corresponds to the following

$$\text{---} \boxed{SU} \text{---} = \text{---} \boxed{R_z(\beta)} \text{---} \boxed{R_y(\theta)} \text{---} \boxed{R_z(\alpha)} \text{---} . \quad (2.3.2)$$

Every unitary matrix $U \in \mathcal{U}(2)$ can be written as the product of a phase for a special unitary matrix, namely

$$U = Ph(\delta)SU = e^{i\delta}SU, \quad (2.3.3)$$

where $Ph(\delta) = e^{i\delta}\mathbb{I}_2$. So the Euler property allows us to decompose any single qubit gate into a sequence of four elementary operations. The relationships between the Euler angles and the elements of the special unitary matrix can be easily determined from the definition:

$$SU = \begin{pmatrix} a & b \\ b^* & a^* \end{pmatrix} \quad (2.3.4)$$

and calculating the product of the three rotations in Eq. (2.3.1) to obtain the relations:

$$\alpha = \arctan\left(\frac{\text{Im } a}{\text{Re } a}\right) + \arctan\left(\frac{\text{Im } b}{\text{Re } b}\right), \quad (2.3.5)$$

$$\beta = \arctan\left(\frac{\text{Im } a}{\text{Re } a}\right) - \arctan\left(\frac{\text{Im } b}{\text{Re } b}\right), \quad (2.3.6)$$

$$\theta = 2 \arctan\left(\sqrt{\frac{\text{Im } b^2 + \text{Re } b^2}{\text{Im } a^2 + \text{Re } a^2}}\right), \quad (2.3.7)$$

$$\delta = \frac{1}{2} \arctan\left(\frac{\text{Im } \det(U)}{\text{Re } \det(U)}\right). \quad (2.3.8)$$

Note that if $\det(e^{-i\delta}U) = -1$ then we need to add a contribution $\pi/2$ to the phase, namely $\delta \mapsto \delta + \pi/2$. A very simple example is the Hadamard gate which can be decomposed as

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = e^{i\frac{\pi}{2}} R_z(-\pi) R_y(\pi/2) R_z(0). \quad (2.3.9)$$

2.3.2 Fully controlled gate decomposition

A fully controlled operation, also called generalised Deutsch-Toffoli, is a special quantum gate that acts on a single qubit according to the state of all the others. Following the convention used in the literature, and considering a total number of qubit equal to n , we refer to these operations as $\Lambda_{n-1}(U)$, where $U \in \mathcal{U}(2)$ acts on a single qubit according to the state of the other $n-1 := m$ control qubits, which define the subspace in which the gates U and the orthogonal one \mathbb{I} act. For example, for $n = 3$ the fully controlled operation acting on the last qubits is represented by the 8×8 unitary matrix

$$\Lambda_2(U) = \begin{pmatrix} \mathbb{I}_6 & 0 \\ 0 & U \end{pmatrix} \quad (2.3.10)$$

and the quantum circuit is



$$(2.3.11)$$

Some special cases are $\Lambda_0(U) = U$ which corresponds to a single qubit gate, $\Lambda_1(U) = CU$ which is a controlled operation with only one control qubit, the common CNOT = $\Lambda_1(X)$ and the Toffoli gate which is $\Lambda_2(X)$ and has two control qubits. In the work of A. Barenco et al. in Ref. [12], a basic method for decomposing a fully controlled quantum gate acting on several qubits simultaneously is introduced. The quantum gate decomposition is based on the universal gate set

$$\{\Lambda_0(U), \Lambda_1(X)\} = \{U, \text{CNOT}\} \quad (2.3.12)$$

for $U \in \mathcal{U}(2)$, and on the fact that every 2×2 special unitary matrix $SU \in \mathcal{SU}(2)$ can be written as

$$SU = C\sigma_x B\sigma_x A, \quad (2.3.13)$$

such that

$$A, B, C \in \mathcal{SU}(2) \text{ and } CBA = \mathbb{I}. \quad (2.3.14)$$

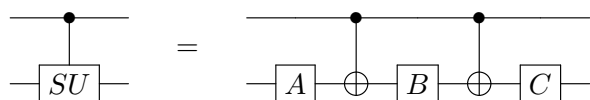
The proof is straightforward using the Euler decomposition. In fact we can always write $SU = R_z(\alpha)R_y(\theta)R_z(\beta) \in \mathcal{SU}(2)$ for some Euler angles, and express SU as in Eq. (2.3.13) using

$$C = R_z(\alpha)R_y\left(\frac{\theta}{2}\right), \quad B = R_y\left(-\frac{\theta}{2}\right)R_z\left(-\frac{\alpha+\beta}{2}\right), \quad A = R_z\left(\frac{\beta-\alpha}{2}\right), \quad (2.3.15)$$

which satisfy the two conditions in Eq. (2.3.14). Note that the choice of matrices A, B, C is not unique and one can also choose

$$C' = R_z\left(\frac{\alpha-\beta}{2}\right), \quad B' = R_z\left(-\frac{\alpha+\beta}{2}\right)R_y\left(-\frac{\theta}{2}\right), \quad A' = R_y\left(\frac{\theta}{2}\right)R_z(\beta). \quad (2.3.16)$$

Using the property in Eq. (2.3.13), we can directly implement the controlled gate $\Lambda_1(SU)$ using the following quantum circuit



$$(2.3.17)$$

The network implements the desired operation because if the first qubit is in the $|0\rangle$ state, the circuit applies $CBA = \mathbb{I}$, instead if it is in the $|1\rangle$ state it applies $C\sigma_x B\sigma_x A = SU$. If the

controlled gate is a phase gate the circuit is:

$$\begin{array}{c} \bullet \\ | \\ \text{---} \\ | \\ \text{---} \\ \boxed{Ph} \end{array} = \begin{array}{c} \boxed{E} \\ \text{---} \\ \text{---} \end{array} \quad (2.3.18)$$

where

$$E = R_z(-\delta)Ph \begin{pmatrix} \delta \\ 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\delta} \end{pmatrix}. \quad (2.3.19)$$

We conclude that any quantum gate $\Lambda_1(U)$, with $U \in \mathcal{U}(2)$, can be implemented as a two-qubit network containing four single-qubit gates Λ_0 (which are E, A, B, C) and two $\Lambda_1(X)$ gates (which are CNOTs). The complete quantum circuit is

$$\begin{array}{c} \bullet \\ | \\ \text{---} \\ | \\ \text{---} \\ \boxed{U} \end{array} = \begin{array}{c} \boxed{E} \\ \text{---} \\ \bullet \\ | \\ \oplus \\ \text{---} \\ \boxed{A} \end{array} \begin{array}{c} \bullet \\ | \\ \text{---} \\ | \\ \oplus \\ \text{---} \\ \boxed{B} \end{array} \begin{array}{c} \bullet \\ | \\ \text{---} \\ | \\ \oplus \\ \text{---} \\ \boxed{C} \end{array}, \quad (2.3.20)$$

which, using only elementary rotations, corresponds to

$$\begin{array}{c} \bullet \\ | \\ \text{---} \\ | \\ \text{---} \\ \boxed{Ph \left(\frac{\delta}{2} \right)} \end{array} \begin{array}{c} \bullet \\ | \\ \text{---} \\ | \\ \oplus \\ \text{---} \\ \boxed{R_z(-\delta)} \end{array} \begin{array}{c} \bullet \\ | \\ \text{---} \\ | \\ \oplus \\ \text{---} \\ \boxed{R_z \left(\frac{\beta-\alpha}{2} \right)} \end{array} \begin{array}{c} \bullet \\ | \\ \text{---} \\ | \\ \oplus \\ \text{---} \\ \boxed{R_z \left(-\frac{\alpha+\beta}{2} \right)} \end{array} \begin{array}{c} \bullet \\ | \\ \text{---} \\ | \\ \oplus \\ \text{---} \\ \boxed{R_y \left(-\frac{\theta}{2} \right)} \end{array} \begin{array}{c} \bullet \\ | \\ \text{---} \\ | \\ \oplus \\ \text{---} \\ \boxed{R_y \left(\frac{\theta}{2} \right)} \end{array} \begin{array}{c} \bullet \\ | \\ \text{---} \\ | \\ \oplus \\ \text{---} \\ \boxed{R_z(\alpha)} \end{array}. \quad (2.3.21)$$

The same property can be used to implement a two-qubit controlled gate, where the control qubit is the second and the target qubit is the first, by simply applying the operations to the opposite qubit, or where the operation is applied when the control qubit is in the state $|0\rangle$ using:

$$\begin{array}{c} \circ \\ | \\ \text{---} \\ | \\ \text{---} \\ \boxed{U} \end{array} = \begin{array}{c} \boxed{X} \\ \text{---} \\ \bullet \\ | \\ \text{---} \\ \boxed{X} \end{array} \begin{array}{c} \bullet \\ | \\ \text{---} \\ | \\ \oplus \\ \text{---} \\ \boxed{U} \end{array}. \quad (2.3.22)$$

In addition, if the operator U satisfies some special properties, then the network that implements $\Lambda_1(U)$ can be further simplified. For example, if we have $W = R_z(\alpha)R_y(\theta)R_z(\alpha)$, then the $\Lambda_1(W)$ operation is reduced to

$$\begin{array}{c} \bullet \\ | \\ \text{---} \\ | \\ \text{---} \\ \boxed{W} \end{array} = \begin{array}{c} \bullet \\ | \\ \text{---} \\ | \\ \oplus \\ \text{---} \\ \boxed{A'} \end{array} \begin{array}{c} \bullet \\ | \\ \text{---} \\ | \\ \oplus \\ \text{---} \\ \boxed{B'} \end{array} \begin{array}{c} \bullet \\ | \\ \text{---} \\ | \\ \oplus \\ \text{---} \\ \boxed{C'} \end{array}, \quad (2.3.23)$$

where

$$A' = R_y \left(\frac{\theta}{2} \right) R_z(\alpha), \quad B' = R_z(-\alpha) R_y \left(-\frac{\theta}{2} \right), \quad C' = \mathbb{I} \quad (2.3.24)$$

and

$$A = \mathbb{I}, \quad B = R_y \left(-\frac{\theta}{2} \right) R_z(-\alpha), \quad C = R_z(\alpha) R_y \left(\frac{\theta}{2} \right). \quad (2.3.25)$$

The property obviously holds for $R_y(\xi) = R_z(0)R_y(\xi)R_z(0)$ and $R_z(\xi) = R_z(\xi/2)R_y(0)R_z(\xi/2)$ (but not for $R_x(\xi) = R_z(\pi/2)R_y(\xi)R_z(-\pi/2)$), whose circuit is as follows

$$\begin{array}{c} \bullet \\ | \\ \text{---} \\ | \\ \text{---} \\ \boxed{R_k(\xi)} \end{array} = \begin{array}{c} \bullet \\ | \\ \text{---} \\ | \\ \oplus \\ \text{---} \\ \boxed{R_k(\xi/2)} \end{array} \begin{array}{c} \bullet \\ | \\ \text{---} \\ | \\ \oplus \\ \text{---} \\ \boxed{R_k(-\xi/2)} \end{array} = \begin{array}{c} \bullet \\ | \\ \text{---} \\ | \\ \oplus \\ \text{---} \\ \boxed{R_k(-\xi/2)} \end{array} \begin{array}{c} \bullet \\ | \\ \text{---} \\ | \\ \oplus \\ \text{---} \\ \boxed{R_k(\xi/2)} \end{array} \quad (2.3.26)$$

for any $\xi \in \mathbb{R}$ and for $k \in \{y, z\}$. Another special case is when we can write the operation as $W' = R_z(\alpha)R_y(\theta)R_z(\alpha)\sigma_x$ and the control gate $\Lambda_1(W')$ reduces to a network with only one CNOT, namely

$$\begin{array}{c} \bullet \\ | \\ \square{W'} \end{array} = \begin{array}{c} \bullet \\ | \\ \square{A} \oplus \square{B} \end{array} . \quad (2.3.27)$$

This property holds e.g. for $Y = R_z(\pi/2)R_y(2\pi)R_z(\pi/2)\sigma_x$, $Z = R_z(0)R_y(\pi)R_z(0)\sigma_x$ and of course for X , and is obviously related to the choice of CNOT for the elementary two-qubit gate.

The extension of this quantum gate decomposition to a fully controlled gate acting on three qubits is straightforward using the following network equivalence between a single $\Lambda_2(U)$ and a sequence of $\Lambda_1(V)$ and CNOTs:

$$\begin{array}{c} \bullet \\ | \\ \bullet \\ | \\ \square{U} \end{array} = \begin{array}{c} \bullet \\ | \\ \bullet \oplus \bullet \\ | \\ \square{V} \square{V^\dagger} \square{V} \end{array} , \quad (2.3.28)$$

where $V \in \mathcal{U}(2)$ and $V^2 = U$. Obviously, a similar network also holds for different target qubits, e.g.

$$\begin{array}{c} \bullet \\ | \\ \square{U} \\ | \\ \bullet \end{array} = \begin{array}{c} \bullet \\ | \\ \bullet \\ | \\ \square{V} \square{V^\dagger} \square{V} \\ | \\ \bullet \oplus \bullet \end{array} , \quad (2.3.29)$$

or different control conditions. The proof is trivial: if only one of the first two qubits is in the $|0\rangle$ state, the circuit applies $VV^\dagger = \mathbb{I}$ or $V^\dagger V = \mathbb{I}$. If both qubits are in $|0\rangle$, the circuit does not perform any operations, and if instead both qubits are in $|1\rangle$, it performs $VV = U$. We can decompose it again by implementing all the two-qubit controlled operators $\Lambda_1(V)$ and $\Lambda_1(V^\dagger)$ in single-qubit gates and CNOTs using the network in Eq. (2.3.20) and the adjoint quantum circuit:

$$\begin{array}{c} \bullet \\ | \\ \square{V^\dagger} \end{array} = \begin{array}{c} \bullet \\ | \\ \bullet \\ | \\ \square{C^\dagger} \oplus \square{B^\dagger} \oplus \square{A^\dagger} \\ | \\ \square{E^\dagger} \end{array} . \quad (2.3.30)$$

In summary, each $\Lambda_2(U)$ operation can be implemented using two CNOTs and three $\Lambda_1(V)$, which can be implemented using two CNOTs and four Λ_0 . However, by sequencing them, some gates can be combined into a single gate. For example, C of the first $\Lambda_1(V)$ and C^\dagger of the second one become an identity (see property in Sec. 2.2.1). Similarly, A^\dagger of the second $\Lambda_1(V^\dagger)$ and A of the third $\Lambda_1(V)$. The final number is eight CNOTs and eight Λ_0 , as shown in the final quantum circuit below


$$\begin{array}{c} \bullet \\ | \\ \bullet \\ | \\ \square{U} \end{array} = \begin{array}{c} \bullet \\ | \\ \bullet \\ | \\ \square{E} \oplus \square{E^\dagger} \\ | \\ \square{A} \oplus \square{B} \oplus \square{B^\dagger} \oplus \square{B} \oplus \square{C} \end{array} . \quad (2.3.31)$$

This is the result obtained in Ref. [12]. However, the quantum circuit can be further optimised by reducing the number of CNOTs, which are usually the most noisy in a real quantum machine.

First of all, we can see that the E gates commute with the control of the CNOT (but not with the target). In fact:

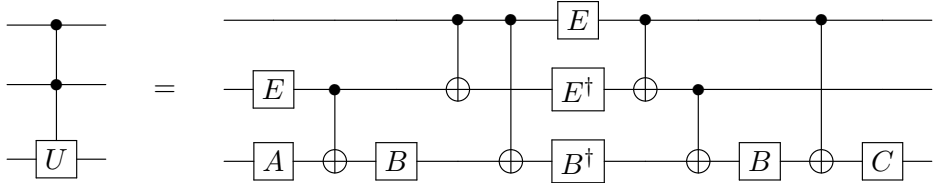
$$\text{CNOT}(E \otimes \mathbb{I}) = \begin{pmatrix} \mathbb{I} & 0 \\ 0 & X \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 \\ 0 & Ph(\delta) \end{pmatrix} = \begin{pmatrix} \mathbb{I} & 0 \\ 0 & XPh \end{pmatrix} \quad (2.3.32)$$

and, since $[X, Ph(\delta)] = 0$, we have the circuit equivalence:



$$\begin{array}{c} \text{---} E \text{---} \bullet \text{---} \\ | \\ \oplus \end{array} = \begin{array}{c} \bullet \text{---} E \text{---} \\ | \\ \oplus \end{array} . \quad (2.3.33)$$

So we can identify two triplets of CNOTs for which we can use the property in Eq. (2.2.14). Both of them can be implemented with two CNOTs each and the final count for $\Lambda_2(U)$ is six CNOTs and eight Λ_0 , as shown below



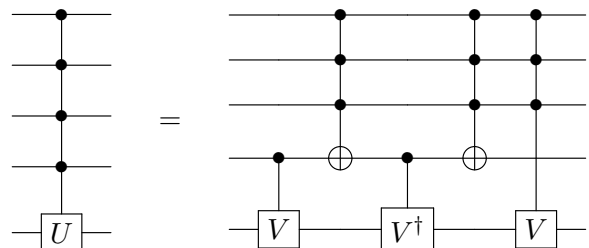
$$\begin{array}{c} \bullet \text{---} \\ | \\ \bullet \text{---} \\ | \\ \oplus \text{---} U \text{---} \end{array} = \begin{array}{c} \bullet \text{---} E \text{---} \bullet \text{---} \bullet \text{---} E \text{---} \bullet \text{---} \\ | \quad | \quad | \\ \oplus \quad \oplus \quad \oplus \\ \oplus \text{---} A \text{---} \oplus \text{---} B \text{---} \oplus \text{---} B^\dagger \text{---} \oplus \text{---} B \text{---} \oplus \text{---} C \text{---} \end{array} . \quad (2.3.34)$$

We can extend this procedure to any arbitrary $\Lambda_{n-1}(U)$ gate. Using the construction described in detail in Ref. [12], we find that the total number of elementary quantum gates needed to implement a general $\Lambda_{n-1}(U)$ is

$$2^n \Lambda_0 + (3 \cdot 2^{n-1} - 6) \text{CNOT} = \mathcal{O}(2^n), \quad (2.3.35)$$

which scales exponentially with the number of qubits and is therefore inefficient. Note that to obtain the function in Eq. (2.3.35) we considered the cancellation of $\Lambda_0(U)$ gates, as done in the previous work, and also the optimisation of the CNOT triplets done here, which reduces the number of CNOTs by two, with respect to the result in the previous work, but keeps the scaling exponential.

As shown in Ref. [12], we can make the gate decomposition efficient and obtain polynomial scaling by allowing the use of generalised Toffoli gates $\Lambda_{n-1}(X)$. In fact, any $\Lambda_{n-1}(U)$ operation can be implemented by a network containing two $\Lambda_1(V)$, two $\Lambda_{n-2}(X)$ and one $\Lambda_{n-2}(V)$, as shown below for $n = 5$



$$\begin{array}{c} \bullet \text{---} \\ | \\ \bullet \text{---} \\ | \\ \bullet \text{---} \\ | \\ \bullet \text{---} \\ | \\ \oplus \text{---} U \text{---} \end{array} = \begin{array}{c} \bullet \text{---} \bullet \text{---} \bullet \text{---} \\ | \quad | \quad | \\ \bullet \text{---} \bullet \text{---} \bullet \text{---} \\ | \quad | \quad | \\ \oplus \quad \oplus \quad \oplus \\ \oplus \text{---} V \text{---} \oplus \text{---} V^\dagger \text{---} \oplus \text{---} V \text{---} \end{array} , \quad (2.3.36)$$

where $V^2 = U$. The complexity of $\Lambda_1(V^\dagger)$ is $\mathcal{O}(1)$, that of $\Lambda_{n-2}(X)$ (which is not fully-controlled) is $\mathcal{O}(n)$ (by Lemmas 7.2 and 7.3 in Ref. [12]), so the cost of $\Lambda_{n-1}(U)$ is the cost of $\Lambda_{n-2}(V)$ plus a linear contribution for the generalised Toffoli. The recursion formula

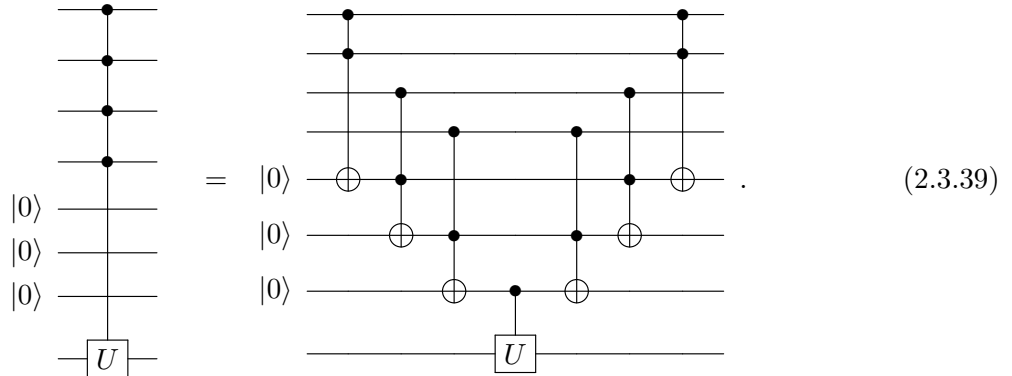
$$\mathcal{C}(\Lambda_{n-1}(U)) = \mathcal{C}(\Lambda_{n-2}(V)) + \mathcal{O}(n) \quad (2.3.37)$$

implies that the cost is quadratic, i.e.

$$\mathcal{C}(\Lambda_{n-1}(U)) = \mathcal{O}(n^2), \quad (2.3.38)$$

and therefore efficient.

Moreover adding a certain number of working qubits allows a controlled gate to be implemented more efficiently [4]. Let's take, for example, $\Lambda_{n-1}(U)$ for $n-1=4$, which has 4 control qubits. We add $n-2=3$ working qubits to the system initialized in the $|0\rangle$ state. At this point, the quantum gate is a non-fully controlled operation that can be implemented by the network



In fact, the last working qubit is the one that actually controls the action of U , so it must contain all the information about the state of the initial 4 control qubits. We assume that they are in a certain initial state $|q_0\rangle \otimes |q_1\rangle \otimes |q_2\rangle \otimes |q_3\rangle$, where $q_i \in \{0, 1\}$. The last working qubit must contain the information $q_0 \wedge q_1 \wedge q_2 \wedge q_3 = q_0 \cdot q_1 \cdot q_2 \cdot q_3$. The first Toffoli gate changes the state of the first working qubit to $|q_1 \cdot q_0\rangle$, the second changes the second working qubit to $|q_2 \cdot q_1 \cdot q_0\rangle$, the third produces $|q_3 \cdot q_2 \cdot q_1 \cdot q_0\rangle$ in the last working qubit. This applies U to the last qubit only if all the controlling qubits satisfy the AND condition. The following gates are used to restore the initial state of the circuit by returning the first 4 qubits to the state $|q_0\rangle \otimes |q_1\rangle \otimes |q_2\rangle \otimes |q_3\rangle$ and the working qubits to $|000\rangle$. This implementation requires a linear number of gates, since $\mathcal{O}(n)$ operations are needed to simulate the Toffolis and the complexity of $\Lambda_1(U)$ is $\mathcal{O}(1)$. So we conclude that

$$\mathcal{C}(\Lambda_{n-4}(U)) = \mathcal{O}(n) \quad (2.3.40)$$

using a sufficient number of auxiliary qubits.

2.3.3 Two-level matrix decomposition

We present here a way to decompose a unitary $U \in \mathcal{U}(N)$ as a sequence of two-level unitary matrices. The procedure is also given in Ref. [13] and can be used as a first step to implement a universal compiler as proposed in Sec. 2.3.4.

A two-level unitary matrix, also called a *Givens rotation*, is a unitary matrix which acts non-trivially only on two vector components and can therefore be obtained from the identity matrix by changing a 2×2 principal submatrix. Given a unitary matrix $U \in \mathcal{U}(N)$ there are $N-1$ two-level classes, each corresponding to a fixed 2×2 principal submatrix. Consider the vector $P = (k_1, k_2, \dots, k_N)$, where $k_i \in \{0, 1, \dots, N-1\}$, defined as a permutation of $(0, 1, \dots, N-1)$. A P-unitary matrix of type p is defined as a unitary matrix with a non-trivial principal submatrix in correspondence of columns and rows k_p and k_{p+1} . The unitary decomposition into two-level matrices can be very useful for quantum computation if we consider a special permutation $P := G_n$ called *Gray code* [14]. If the unitary matrix $U \in \mathcal{U}(N)$ corresponds to an

n -qubit quantum gate, then $N = 2^n$ and the Gray code allows us to decompose the quantum operation as a sequence of fully controlled quantum gates. By definition G_n is a permutation of (k_1, k_2, \dots, k_N) where each k_p corresponds to the decimal representation of the binary sequence of n bits $(x_{n-1}, \dots, x_1, x_0) = [k]$ where $x_i \in \{0, 1\}$:

$$k = x_0 2^0 + x_1 2^1 + x_2 2^2 + \dots + x_{n-1} 2^{n-1} = \sum_{i=0}^{n-1} x_i 2^i \quad (2.3.41)$$

such that two neighboring elements k_j and k_{j+1} (and also the two elements k_1 and k_N) differ only by the value of one bit in binary notation. The Gray code can be constructed using a recursive procedure. Given the first element $G_1 := (0, 1)$, any Gray code of $n + 1$ qubits can be constructed as

$$G_{n+1} = (0[k_1], \dots, 0[k_N], 1[k_N], \dots, 1[k_1]) = (0G_n, 1G'_n), \quad (2.3.42)$$

for all $n \geq 1$, where we have defined the inverted vector $G'_n = (k_N, \dots, k_1)$. Note that adding a qubit doubles the length of the Gray code, and the first N elements start with 0 and are followed by the previous Gray code, while the next N elements start with 1 and are followed by the previous Gray code inverted from right to left. For example:

$$G_2 = (0G_1, 1G'_1) = (0(0, 1), 1(1, 0)) = (00, 01, 11, 10) = ([0], [1], [3], [2]). \quad (2.3.43)$$

The next Gray code is

$$G_3 = (0G_2, 1G'_2) = (000, 001, 011, 010, 110, 111, 101, 100) \\ = ([0], [1], [3], [2], [6], [7], [5], [4]), \quad (2.3.44)$$

and similarly for any n . It is easy to prove that, given a Gray code G_n , a generic $U \in \mathcal{U}(2^n)$, where $N = 2^n$, can be written as the product of m two-level matrices with $m \leq N(N-1)/2$. If $U \in \mathcal{SU}(2^n)$ then each of the m matrices has a determinant equal to 1. This property follows from the fact that a $N \times N$ unitary matrix has N^2 elements, but the unitarity condition automatically fixes a number of elements equal to the elements of the triangular matrix (including the diagonal), and then removes $N(N+1)/2$ degrees of freedom. This leaves $N^2 - N(N+1)/2 = N(N-1)/2$ elements. The combination of the concept of two-level unitary matrices and the Gray code can be used to decompose any quantum gate $U \in \mathcal{U}(2^n)$ as the product of no more than $2^{n-1}(2^n-1)$ two-level matrices, corresponding to a fully controlled single-qubit quantum gate $\Lambda_{n-1}(V)$, where $V \in \mathcal{U}(2)$, with $n-1$ control qubits.

For example, consider a two-qubit gate $U \in \mathcal{U}(4)$, to which we associate the Gray code G_2 in Eq. (2.3.43). There are $N-1 = 3$ classes of two-level matrices defined by G_2 which are of type 1 (with non-trivial submatrices in columns and rows 0 and 1), of type 2 (with non-trivial submatrices in positions 1 and 3), and of type 3 (with non-trivial entries in 3 and 2). All three types correspond to a controlled gate $CV = \Lambda_1(V)$. Type 1 is

$$\begin{pmatrix} v_0 & v_1 & 0 & 0 \\ v_2 & v_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{array}{c} \text{---} \circ \text{---} \\ | \\ \text{---} \square V \text{---} \end{array}, \quad (2.3.45)$$

type 2 is

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & v_0 & 0 & v_1 \\ 0 & 0 & 1 & 0 \\ 0 & v_2 & 0 & v_3 \end{pmatrix} = \begin{array}{c} \text{---} \square V \text{---} \\ | \\ \text{---} \bullet \text{---} \end{array}, \quad (2.3.46)$$

and type 3 is

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & v_0 & v_1 \\ 0 & 0 & v_2 & v_3 \end{pmatrix} = \begin{array}{c} \bullet \\ | \\ \square V \\ | \\ \bullet \end{array}, \quad (2.3.47)$$

where the single-qubit gate is

$$V = \begin{pmatrix} v_0 & v_1 \\ v_2 & v_3 \end{pmatrix}. \quad (2.3.48)$$

In the case of a three-qubit quantum gate $U \in \mathcal{U}(8)$ there are $N - 1 = 7$ classes of two-level matrices generated by G_3 in Eq. (2.3.44), and thus 7 corresponding types of fully controlled operations. For example, type-1 matrices have the non-trivial submatrix in position (0, 1), type-2 matrices in (1, 3), and so on. Below are the quantum circuits of type 1 to 7 from left to right

$$(2.3.49)$$

The procedure for decomposing a unitary $U \in \mathcal{U}(4)$ is described in detail in Ref. [13] and can be easily extended to any number of qubits. In the case of $n = 2$ the number of two-level matrices is $2^{n-1}(2^n - 1) = 6$ and they can be found by diagonalising U according to the Gray code order, which fixes the element on the triangular matrix to be set to zero and the two-level matrix type from $2^n - 1$ to 1. At the end of the procedure we have

$$\left(\prod_{k=1}^{\frac{2^n(2^n-1)}{2}} U_k \right) U = \mathbb{I} \quad (2.3.50)$$

where the product is on the left. For example, for $n = 2$ we have $U_6 \dots U_1 U = \mathbb{I}$ which gives $U = U_1^\dagger \dots U_6^\dagger$. The type of each two-level matrix U_k , for $k \in \{1, \dots, 2^{n-1}(2^n - 1)\}$, is determined by the table in Fig. 2.2a, which produces the controlled gates $U_k^\dagger = CV_k = \Lambda_1(V_k)$ in the following quantum circuit

$$(2.3.51)$$

This decomposition can be improved by noting that two consecutive gates with the same target qubit can be merged. In fact

$$(2.3.52)$$

because

$$\begin{pmatrix} A & 0 \\ 0 & \mathbb{I} \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 \\ 0 & B \end{pmatrix} = \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix} = \begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 \\ 0 & A^{-1}B \end{pmatrix}. \quad (2.3.53)$$

This reduces the number of fully controlled operations of $(2^n - 2)/2$ by adding the same number of non-fully controlled gates. The same property holds for the case $n = 3$. The decomposition

Type	1	2	3
	U_3	U_2	U_1
		U_5	U_4
			U_6

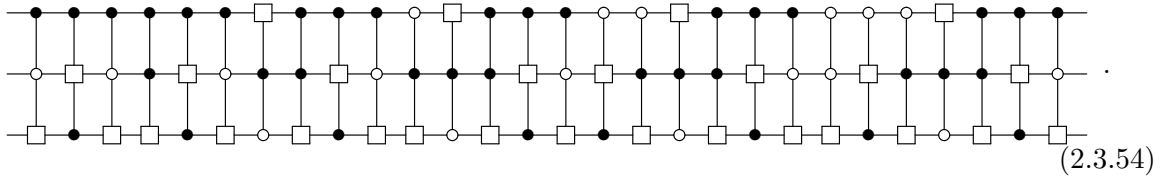
(a)

Type	1	2	3	4	5	6	7
	U_7	U_6	U_5	U_4	U_3	U_2	U_1
		U_{13}	U_{12}	U_{11}	U_{10}	U_9	U_8
			U_{18}	U_{17}	U_{16}	U_{15}	U_{14}
				U_{22}	U_{21}	U_{20}	U_{19}
					U_{25}	U_{24}	U_{23}
						U_{27}	U_{26}
							U_{28}

(b)

Figure 2.2: The two tables show the type of two-level matrices obtained from the diagonalisation of a unitary $U \in \mathcal{U}(2^n)$. Panel (a) for $n = 2$ and panel (b) for $n = 3$.

procedure now follows the order in the table in Fig. 2.2b, from which we obtain a network of the form



Here we can use the simplification property

$$\begin{array}{c} \bullet \\ \text{---} \\ \circ \\ \text{---} \\ \square \\ \text{---} \end{array} \begin{array}{c} \circ \\ \text{---} \\ \bullet \\ \text{---} \\ \square \\ \text{---} \end{array} = \begin{array}{c} \bullet \\ \text{---} \\ \circ \\ \text{---} \\ \square \\ \text{---} \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \\ \text{---} \\ \square \\ \text{---} \end{array}, \quad (2.3.55)$$

which corresponds to $U_k^\dagger U_{k+1}^\dagger$ for $k + 1 = 8$ of type 7 and $k = 7$ of type 1. An equivalent property can be used for $k = 7, 18, 25$. The optimal network was found in Ref. [15]. The number of controls that can be removed is equal to the number of different consecutive controls in similar controlled gates.

2.3.4 Compiler

Putting together the three decomposition methods just presented: (1) the two-level matrix decomposition in Sec. 2.3.3, (2) the fully controlled gate decomposition in Sec. 2.3.2, and (3) the zyz decomposition for single-qubit gate in Sec. 2.3.1, we can build a very naive compiler. We first decompose the matrix $U \in \mathcal{U}(2^n)$ into $2^{n-1}(2^n - 1)$ two-level matrices, each corresponding to a fully controlled gate $\Lambda_{n-1}(V_k)$. Then these fully controlled operations can be implemented by a network containing only CNOTs and single-qubit rotations, and finally each single-qubit gate can be implemented using at most three elementary rotations.

This compiler potentially works for any number n of qubits and for any general unitary $U \in \mathcal{U}(2^n)$, but the number of gates needed grows exponentially with the size of the system. In fact, we need $\mathcal{O}(4^n)$ fully controlled gates $\Lambda_{n-1}(V)$ (two-level matrices). Each of these requires $\mathcal{O}(n)$ for the Toffoli gates and $\mathcal{O}(n^2)$ for the non fully controlled gates. So the total complexity is:

$$\mathcal{C}(U \in \mathcal{U}(2^n)) = \mathcal{O}(n^3 4^n). \quad (2.3.56)$$

By adding auxiliary qubits, we can implement the $\Lambda_{n-1}(V)$ gates with a complexity of $\mathcal{O}(n)$, reducing the total complexity to

$$\mathcal{C}(U \in \mathcal{U}(2^n)) = \mathcal{O}(n^2 4^n), \quad (2.3.57)$$

that is still larger than the theoretical lower bound that is² $\mathcal{O}(4^n)$ as stated in Ref. [16]. This naive compiler can be optimised by reducing the complexity scaling to the theoretical lower bound, as proposed in Ref. [15], where the authors used the approaches analysed so far and combined them to a further optimisation that allows to remove the superfluous controls. The complexity is obviously still exponential.

2.4 Methods for quantum gate decomposition

There are many different methods in the literature for decomposing unitary transformations into elementary gates. In general, there is no universal procedure that is optimal for every situation. However, the information we have about the properties of the problem to be solved can help in choosing and defining the best way to implement a particular quantum algorithm. Note that the exponential scaling of complexity with the number of qubits n is intrinsic to the universal decomposition of quantum gates, and in order to avoid this, which would make quantum computation inefficient, a scientist must take into account the greatest number of properties that can reduce complexity, such as symmetry, conservation laws, scaling of the error in the approximations, reducible Hilbert subspaces and any other property that can help reduce the number of operations required to implement a desired unitary. After the quantum gate decomposition, some circuit optimisations can be easily performed by exploiting the circuit equivalences presented in Sec. 2.2.

2.4.1 An overview

To give an attempted (though not exhaustive) overview of quantum gate decomposition techniques, we list below some well-known methods. Some of them will be analysed in detail and sometimes improved in the following sections.

1. The **zyz-decomposition** implements arbitrary single-qubit gates $U \in \mathcal{U}(2)$ with three elementary rotations using the Euler angles. This simple decomposition was presented in Sec. 2.3.1, where we also gave a practical way to implement it as code.
2. The **magic-basis** method gives an optimal decomposition procedure to implement a generic two-qubit gate $U \in \mathcal{U}(4)$ with three CNOTs, which is the theoretical lower bound to generate all elements in $\mathcal{U}(4)$. The procedure is presented in Refs. [16–18] and verified in detail in Sec. 2.4.2, where we also give an alternative derivation and results.
3. The **Qiskit function** `.decompose()` is a practical and open source implementation of the unitaries $U \in \mathcal{U}(4)$ decomposition, which returns a quantum circuit with no more than three CNOTs. However, the function has some limitations because it does not work for certain special gates, which should be even simpler, because it performs a division by a determinant of a submatrix that could be zero. It also involves a random search for the solution, following an over-fitting parameter approach, which gives high fidelity but makes the simulation very susceptible to noise. In other words, to achieve maximum fidelity, it sets values for the rotation angles that theoretically produce an equivalent transformation,

²The number of needed CNOTs for implementing $U \in \mathcal{U}(2^n)$ is $\frac{4^n - 3n - 1}{4}$.

but which are less accurate experimentally. Finally, it produces a circuit that is not optimal in terms of the number of single-qubit gates, as will be clear in Sec. 2.4.2. Note that the Qiskit package uses a different convention for elementary rotations³ than the one used in this thesis (Eqs. (2.1.3), (2.1.4) and (2.1.5)).

4. The implementation of quantum gates generated by **Pauli strings** (see e.g. Ref. [4]) allows an efficient and analytical implementation of a certain set of unitaries, which can be used whenever the unitary U coincides with an operator generated by a simple tensor of Pauli matrices, but can also be used to implement an operation generated by a sum of Pauli strings, provided that a Trotter error (see Sec. 6.2 for more details) due to the separation of the different exponents is accepted. Since any Hermitian operator can be written as a sum of Pauli strings, this method can be used to decompose any time propagator generated by a Hamiltonian. The procedure is analysed in Sec. 2.4.3.
5. The **fully controlled gate decomposition** method proposed in Ref. [12], allows to decompose a fully controlled single qubit gate as a network containing only CNOTs and single qubit gates. This method has been analysed and implemented in Sec. 2.3.2.
6. The **unentangling method**, first proposed in Ref. [19] and presented in other works such as Refs. [20, 21], exploits the action of a unitary $U \in \mathcal{U}(2^n)$ on the basis state vectors. This procedure consists in decomposing the unitary as a set of multiple controlled gates, which disentangles the last qubit of the system. The procedure is applied recursively so that the final state is separable.
7. The method based on **two-level matrices** decomposition, or Givens rotations, allows to decompose a unitary into a sequence of fully controlled operations. This method is proposed in Ref. [13], analysed and implemented also in Sec. 2.3.3, and optimised in Ref. [15]. As we have seen, this method, together with the decomposition of a fully controlled operation, can be used to implement a simple and naive compiler, presented in Sec. 2.3, which would work for any number n of qubits.
8. The **cosine-sine decomposition** (CSD) method, proposed and implemented in Refs. [22, 23], is a way to decompose a unitary into three parts: a block-real diagonal matrix, containing sine and cosine rotations, and a left and right uniformly controlled operations, also called multiplexer⁴. The recursive application allows the unitary to be decomposed as a

³The single qubit gate definitions used by Qiskit are

$$R_x(\alpha) = \begin{pmatrix} \cos(\theta/2) & -i \sin(\theta/2) \\ -i \sin(\theta/2) & \cos(\theta/2) \end{pmatrix}, R_y(\theta) = \begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix}, R_z(\alpha) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix}. \quad (2.4.1)$$

⁴The quantum multiplexer gates are uniformly controlled operations, i.e. unitary operations that act on the target qubits in a way determined by any possible value of the control qubits. The matrix representation of an n -qubit multiplexer is:

$$U(2^n) = \begin{pmatrix} U_0(2^{n-1}) & 0 \\ 0 & U_1(2^{n-1}) \end{pmatrix}, \quad (2.4.2)$$

where $U_{0,1}$ acts on $n - 1$ qubits. When the control qubit (the first of the system) is in the $|0\rangle$ state, the operator acts with U_0 while, when it is in the $|1\rangle$ state it acts with U_1 . To identify a multiplexor we can also use the direct-sum notation

$$U = U_0 \oplus U_1. \quad (2.4.3)$$

These gates implement the *if-then-else* condition in the context of classical computation. In the quantum field, on the other hand, the control qubit can be in a superposition of *if* and *else*, then the multiplexer creates a linear combination of these two actions. The most simplest example of a multiplexer is $\text{CNOT} = \mathbb{I} \oplus X$.

sequence of multiplexers. An alternative application of CSD using eigenvalue decomposition is the **Quantum Shannon Decomposition** (QSD) method proposed in Ref. [21] and then optimised in Ref. [20].

9. The **Quantum Signal Processing** method [24, 25], also called *Qubitisation*, allows to implement the unitary time propagator $U(t) = e^{-iHt}$ generated by a sparse matrix H by transferring the eigenvalues of H to the state of an ancilla qubit, using the ancilla to perform a controlled unitary rotation, finally projecting the ancilla at the end of the procedure.
10. Several methods have been proposed in the literature to implement a diagonal unitary. Some of the seminal ones are the method based on the **Walsh basis**, e.g. presented in Ref. [26] and summarised in Sec. 2.4.4, and the so-called **phase kick-back** method presented in Ref. [27]. Regarding the Walsh-based method, Sec. 2.4.4 present a connection between the results in Ref. [26] and another result proposed in Ref. [28], where the authors identify a certain subset of diagonal operators that are also local. In addition, Sec. 2.4.4 proposes a generalisation of this last result by defining a more general set of diagonal operators that can be implemented by a local quantum circuit.

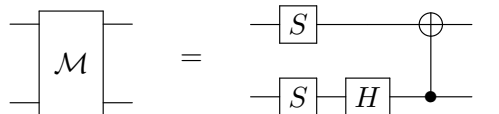
2.4.2 The magic basis

Given a two-qubit gate $U \in \mathcal{U}(4)$ with a matrix representation one can prove that three CNOTs and 15 elementary rotations are always sufficient to implement it as a quantum circuit. The theory behind this statement can be found e.g. in Refs. [16–18].

Let's define the magic basis, also called *maximally entangled*, as:

$$\mathcal{M} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i & 0 & 0 \\ 0 & 0 & i & 1 \\ 0 & 0 & i & -1 \\ 1 & -i & 0 & 0 \end{pmatrix}, \quad (2.4.4)$$

that can be implemented by the quantum circuit



$$\text{Circuit with } \mathcal{M} \text{ gate} = \text{Circuit with } S, H, \text{ and CNOT gates}, \quad (2.4.5)$$

where the S gate is a π phase operation (matrix in Eq. (2.1.8)) and H is the Hadamard (matrix in Eq. (2.1.6)). The proof of Eq. (2.4.5) is trivial by performing the matrix product between the

operations⁵. The operation \mathcal{M} maps the elements of the two-qubit basis onto the so-called Bell states, i.e.

$$\begin{aligned}\mathcal{M} : |00\rangle &\mapsto \frac{|00\rangle + |11\rangle}{\sqrt{2}} := |\Phi^+\rangle \\ \mathcal{M} : |01\rangle &\mapsto \frac{|00\rangle - |11\rangle}{\sqrt{2}} := |\Phi^-\rangle \\ \mathcal{M} : |10\rangle &\mapsto \frac{|01\rangle + |10\rangle}{\sqrt{2}} := |\Psi^+\rangle \\ \mathcal{M} : |11\rangle &\mapsto \frac{|01\rangle - |10\rangle}{\sqrt{2}} := |\Psi^-\rangle.\end{aligned}\tag{2.4.7}$$

It is known that any unitary matrix $U \in \mathcal{U}(4)$ can be written as

$$U = (A_1 \otimes A_2)\mathcal{N}(\alpha, \beta, \gamma)(A_3 \otimes A_4)\tag{2.4.8}$$

where $A_i \in \mathcal{U}(2)$ are single qubit gates and

$$\mathcal{N}(\alpha, \beta, \gamma) = e^{i(\alpha X \otimes X + \beta Y \otimes Y + \gamma Z \otimes Z)}\tag{2.4.9}$$

for $\alpha, \beta, \gamma \in \mathbb{R}$. From this property we have the following quantum circuit implementation of U :

$$\begin{array}{c} \text{---} \boxed{U} \text{---} \\ \text{---} \end{array} = \begin{array}{c} \text{---} \boxed{A_3} \text{---} \boxed{\mathcal{N}} \text{---} \boxed{A_1} \text{---} \\ \text{---} \boxed{A_4} \text{---} \boxed{\mathcal{N}} \text{---} \boxed{A_2} \text{---} \end{array}.\tag{2.4.10}$$

We can prove that the gate \mathcal{N} expressed on the magic basis is a diagonal matrix. In fact the exponent is

$$\alpha X \otimes X + \beta Y \otimes Y + \gamma Z \otimes Z = \begin{pmatrix} \gamma & 0 & 0 & \alpha - \beta \\ 0 & -\gamma & \alpha + \beta & 0 \\ 0 & \alpha + \beta & -\gamma & 0 \\ \alpha - \beta & 0 & 0 & \gamma \end{pmatrix} := \mathcal{N}',\tag{2.4.11}$$

⁵Proof of Eq. (2.4.5)

$$\begin{aligned}\mathcal{M} &= \text{CNOT}'(\mathbb{I} \otimes H)(S \otimes S) \\ &= \text{CNOT}_{inv} \begin{pmatrix} H & 0 \\ 0 & H \end{pmatrix} \left[\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \right] \\ &= \text{CNOT}_{inv} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & i & 0 & 0 \\ 1 & -i & 0 & 0 \\ 0 & 0 & i & -1 \\ 0 & 0 & i & 1 \end{pmatrix} \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i & 0 & 0 \\ 0 & 0 & i & 1 \\ 0 & 0 & i & -1 \\ 1 & -i & 0 & 0 \end{pmatrix}.\end{aligned}\tag{2.4.6}$$

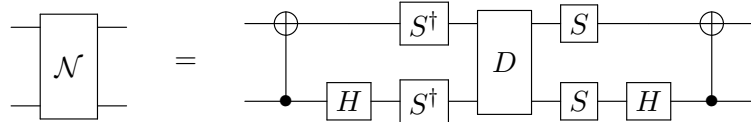
which, on the magic basis, is equivalent to

$$\mathcal{M}^\dagger \mathcal{N}' \mathcal{M} = \begin{pmatrix} \alpha - \beta + \gamma & 0 & 0 & 0 \\ 0 & -\alpha + \beta + \gamma & 0 & 0 \\ 0 & 0 & \alpha + \beta - \gamma & 0 \\ 0 & 0 & 0 & -\alpha - \beta - \gamma \end{pmatrix}. \quad (2.4.12)$$

At this point, the exponential of a diagonal matrix is still diagonal with the exponentials on the diagonal, namely

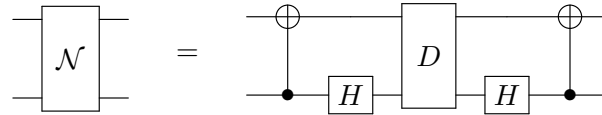
$$D = \mathcal{M}^\dagger \mathcal{N} \mathcal{M} = \begin{pmatrix} e^{i(\alpha-\beta+\gamma)} & 0 & 0 & 0 \\ 0 & e^{-i(\alpha-\beta-\gamma)} & 0 & 0 \\ 0 & 0 & e^{i(\alpha+\beta-\gamma)} & 0 \\ 0 & 0 & 0 & e^{-i(\alpha+\beta+\gamma)} \end{pmatrix} := \begin{pmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & d \end{pmatrix}, \quad (2.4.13)$$

where we have defined a, b, c, d for the sake of simplicity. This gives $\mathcal{N} = \mathcal{M} D \mathcal{M}^\dagger$, which corresponds to the circuit



$$\text{Circuit } \mathcal{N} = \text{CNOT}_{1 \rightarrow 2} \left(S^\dagger \text{ on } 1, H, S^\dagger, S, H \text{ on } 2 \right) D \left(S, H \text{ on } 2 \right) \text{CNOT}_{2 \rightarrow 1}. \quad (2.4.14)$$

One can easily see that $(S \otimes S) D (S^\dagger \otimes S^\dagger) = D$ is still equal to the original diagonal matrix, so we have



$$\text{Circuit } \mathcal{N} = \text{CNOT}_{1 \rightarrow 2} D \text{CNOT}_{2 \rightarrow 1} \left(H \text{ on } 2 \right). \quad (2.4.15)$$

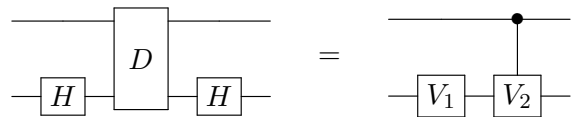
The gate in the middle of the two CNOTs is

$$(\mathbb{I} \otimes H) D (\mathbb{I} \otimes H) = \frac{1}{2} \begin{pmatrix} a+b & a-b & 0 & 0 \\ a-b & a+b & 0 & 0 \\ 0 & 0 & c+d & c-d \\ 0 & 0 & c-d & c+d \end{pmatrix}, \quad (2.4.16)$$

that is in the form of a single qubit gate V_1 applied to the second qubit, followed by a controlled one $\Lambda_1(V_2) = C V_2$, i.e.

$$C V_2 (\mathbb{I} \otimes V_1) = \begin{pmatrix} \mathbb{I} & 0 \\ 0 & V_2 \end{pmatrix} \begin{pmatrix} V_1 & 0 \\ 0 & V_1 \end{pmatrix} = \begin{pmatrix} V_1 & 0 \\ 0 & V_2 V_1 \end{pmatrix}, \quad (2.4.17)$$

which corresponds to the circuit:



$$\text{Circuit } \mathcal{N} = \text{CNOT}_{1 \rightarrow 2} \left(H, D, H \text{ on } 2 \right) = \text{CNOT}_{1 \rightarrow 2} \left(V_1, V_2 \text{ on } 2 \right), \quad (2.4.18)$$

where

$$V_1 = \frac{1}{2} \begin{pmatrix} a+b & a-b \\ a-b & a+b \end{pmatrix} = e^{i\gamma} \begin{pmatrix} \cos(\alpha-\beta) & i \sin(\alpha-\beta) \\ i \sin(\alpha-\beta) & \cos(\alpha-\beta) \end{pmatrix} \quad (2.4.19)$$

and

$$V_2 = \frac{1}{2} \begin{pmatrix} c+d & c-d \\ c-d & c+d \end{pmatrix} V_1^\dagger = e^{-2i\gamma} \begin{pmatrix} \cos(2\beta) & i \sin(2\beta) \\ i \sin(2\beta) & \cos(2\beta) \end{pmatrix}. \quad (2.4.20)$$

We can now decompose V_1 using Euler angles obtaining

$$\begin{aligned} V_1 &= e^{i\gamma} R_z(-\pi/2) R_y(2(\beta - \alpha)) R_z(\pi/2) \\ &= e^{i\gamma} R_z(\pi/2) R_y(2(\alpha - \beta)) R_z(-\pi/2), \end{aligned} \quad (2.4.21)$$

and similar for V_2 getting

$$\begin{aligned} V_2 &= e^{-2i\gamma} R_z(\pi/2) R_y(4\beta) R_z(-\pi/2) \\ &= e^{-2i\gamma} R_z(-\pi/2) R_y(-4\beta) R_z(\pi/2). \end{aligned} \quad (2.4.22)$$

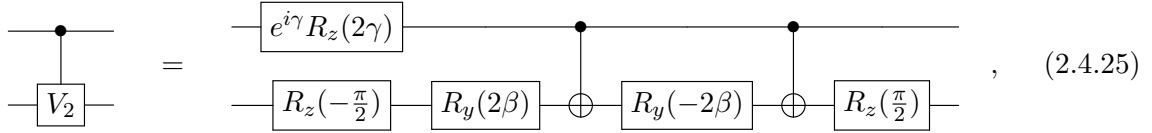
We now use the property in Eq. (2.3.13) to write

$$V_2 = e^{-2i\gamma} C \sigma_x B \sigma_x A, \quad (2.4.23)$$

where

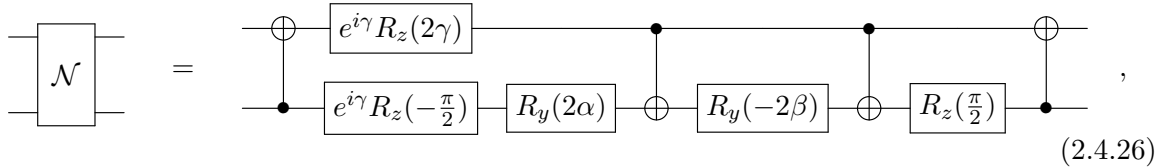
$$A = R_y(2\beta) R_z(-\pi/2), \quad B = R_y(-2\beta), \quad C = R_z(\pi/2), \quad (2.4.24)$$

and implement the gate CV_2 using the circuit in Eq. (2.3.20), which reads



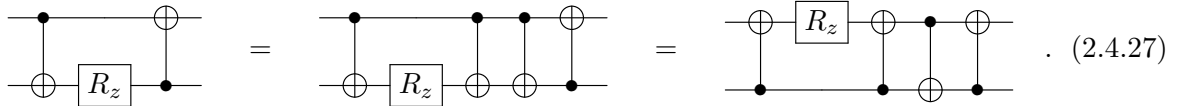
$$= \text{Circuit with } e^{i\gamma} R_z(2\gamma) \text{ on top qubit, } R_z(-\frac{\pi}{2}), R_y(2\beta), \text{CNOT, } R_y(-2\beta), \text{CNOT, } R_z(\frac{\pi}{2}) \text{ on bottom qubit.} \quad (2.4.25)$$

where the controlled phase is implemented with a single qubit gate $E = e^{i\gamma} R_z(2\gamma)$ on the first qubit. Combining, merging, and eliminating the maximum number of gates, we have



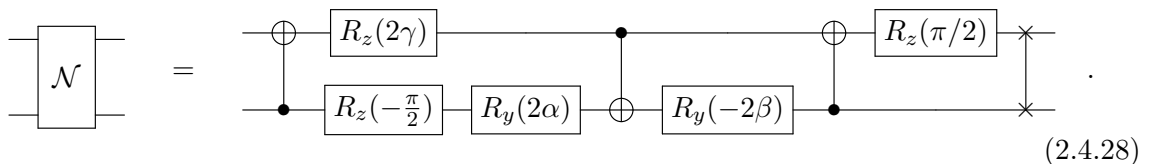
$$= \text{Circuit with } e^{i\gamma} R_z(2\gamma) \text{ on top qubit, } e^{i\gamma} R_z(-\frac{\pi}{2}), R_y(2\alpha), \text{CNOT, } R_y(-2\beta), \text{CNOT, } R_z(\frac{\pi}{2}) \text{ on bottom qubit.} \quad (2.4.26)$$

where the two phases $e^{i\gamma}$ can be eliminated because they represent a global phase that is not observable. Now we can manipulate the last part of the circuit by inserting an identity formed by two CNOTs and inverting the operation between the two qubits according to the property in Eq. (2.2.12)



$$= \text{Circuit with } R_z \text{ on top qubit.} \quad (2.4.27)$$

This means that we can remove a CNOT from the circuit in Eq. (2.4.26) because the action is simply to swap the two qubits. One gets



$$= \text{Circuit with } R_z(2\gamma) \text{ on top qubit, } R_z(-\frac{\pi}{2}), R_y(2\alpha), \text{CNOT, } R_y(-2\beta), \text{CNOT, } R_z(\frac{\pi}{2}) \text{ on bottom qubit.} \quad (2.4.28)$$

This result is slightly different from the one obtained in the literature (Refs. [16–18]) that is

$$(2.4.29)$$

up to a global phase of $e^{-i\pi/4}$. Noticing that the two external R_z rotations can be merged with the gates A_1 and A_4 in Eq. (2.4.10), we obtain that any unitary $U \in \mathcal{U}(4)$ can be implemented with a quantum circuit with three CNOTs and up to 15 elementary rotations (or 7 general single-qubit rotations):

$$(2.4.30)$$

where $A'_1 := A_1 R_z(-\pi/2)$ and $A'_4 := R_z(\pi/2) A_4$. This is an optimal decomposition because there is no other decomposition made from the same library and with fewer quantum gates.

2.4.3 Implementation of Pauli string-generated operations

An interesting and useful quantum gate decomposition is that of a unitary operator generated by a Pauli string \mathcal{P} , namely an n -qubit separable operator composed of single qubit operations $\sigma_i \in \{\mathbb{I}, \sigma_x, \sigma_y, \sigma_z\}$, where i indicates the qubit $|q_i\rangle$ to which they are applied, i.e.

$$\mathcal{P} = \bigotimes_{i=0}^{n-1} \sigma_i = \sigma_0 \otimes \sigma_1 \otimes \cdots \otimes \sigma_{n-1}. \quad (2.4.31)$$

The Pauli string decomposition can be used for the quantum circuit implementation of the unitary evolution operator $U(t) = e^{-iHt}$, by first decomposing the Hamiltonian H as a sum of Pauli strings

$$H = \sum_k c_k \mathcal{P}_k := \sum_k h_k, \quad (2.4.32)$$

where h_k is a Hamiltonian term acting on all n qubits in a separable way. The exponential can be decomposed into a product of individual Pauli string exponentials

$$U(t) = e^{-it \sum_k h_k} = \left(e^{-i \sum_k h_k t/r} \right)^r \approx \left(\prod_k e^{-ih_k t/r} \right)^r, \quad (2.4.33)$$

at the cost of introducing a Trotter error $\sim \mathcal{O}(t^2/r)$. The main advantage is that each exponential can be implemented exactly by a cascade of CNOTs before and after a R_z rotation on the last qubit [4]. Consider a single term

$$e^{-ih_k t} := e^{-i\frac{\theta}{2} \sigma_0 \otimes \cdots \otimes \sigma_{n-1}}, \quad (2.4.34)$$

where $\theta := 2t$, the quantum gate decomposition procedure is as follows:

- (i) For each qubit $|q_i\rangle$, where $i \in \{0, \dots, n-1\}$, on which we act with $\sigma_i \neq \mathbb{I}$, rotate the basis in the z axes, namely if $\sigma_i = \sigma_z$ apply an identity, if $\sigma_i = \sigma_x$ apply a Hadamard gate (see Eq. (2.2.3)), and if $\sigma_i = \sigma_y$ apply a Hadamard followed by a $R_z(\pi/2)$ rotation (see the relation $Y = R_z(-\pi/2) H Z H R_z(\pi/2)$).

- (ii) For each qubit $|q_i\rangle$, where $i \in \{0, \dots, n-1\}$, to which we apply $\sigma_i \neq \mathbb{I}$, we add a CNOT gate controlled in $|q_i\rangle$ and targeted to the first $|q_j\rangle$ for $j > i$ such that $\sigma_j \neq \mathbb{I}$.
- (iii) Apply $R_z(\theta)$ to the most significant qubit $|q_k\rangle$ to which we apply $\sigma_k \neq \mathbb{I}$. An equivalent way is to target all CNOTs in the last significant qubit.
- (iv) Apply the cascade of CNOTs in reverse order and recover the initial bases using the adjoint single qubit operators of the first step.

A practical example of this quantum gate decomposition is the following

$$e^{i\frac{\theta}{2}X \otimes Z \otimes \mathbb{I} \otimes X} = \begin{array}{c} \text{---} [H] \text{---} \bullet \text{---} \text{---} \bullet \text{---} [H] \text{---} \\ | \quad | \quad | \quad | \\ \oplus \quad \bullet \quad \bullet \quad \oplus \\ | \quad | \quad | \quad | \\ \text{---} [H] \text{---} \oplus \text{---} [R_z(\theta)] \text{---} \oplus \text{---} [H] \text{---} \end{array}, \quad (2.4.35)$$

which can also be achieved by applying the targets to the last qubit:

$$e^{i\frac{\theta}{2}X \otimes Z \otimes \mathbb{I} \otimes X} = \begin{array}{c} \text{---} [H] \text{---} \bullet \text{---} \text{---} \bullet \text{---} [H] \text{---} \\ | \quad | \quad | \quad | \\ \oplus \quad \oplus \quad \oplus \quad \oplus \\ | \quad | \quad | \quad | \\ \text{---} [H] \text{---} \oplus \text{---} [R_z(\theta)] \text{---} \oplus \text{---} [H] \text{---} \end{array}. \quad (2.4.36)$$

If we quantify the complexity of the circuit as the number of two-qubit gates, then the number of CNOTs required is equal to

$$\# \text{ CNOTs} = 2(p-1), \quad (2.4.37)$$

where p is the number of qubits to which we apply $\sigma_i \neq \mathbb{I}$,

2.4.4 Walsh basis decomposition

In the work in Ref. [26], the authors present an efficient way to implement a diagonal unitary D on n qubits with a quantum circuit using $2^{n+1} - 3$ one- and two-qubit gates. The method is based on the connection between the Walsh functions and the basis of diagonal operators, and the problem of finding the shortest quantum circuit implementing $D = e^{if}$, where f is diagonal and hermitian, coincides with the problem of finding the shortest series of Walsh functions w_i that reconstruct f according to

$$f = \sum_{k=0}^{2^n-1} a_k w_k. \quad (2.4.38)$$

The key point of this approach is that there is a connection between the Walsh basis and the diagonal operator basis, also called Walsh operators. Let's consider a separable n -qubit state $|q_0 \dots q_{n-1}\rangle$, for $q_i \in \{0, 1\}$, and define the Pauli Z_i operator acting on the qubit $|q_i\rangle$ as

$$Z_i |q_0 \dots q_{n-1}\rangle = (-1)^{q_i} |q_0 \dots q_{n-1}\rangle. \quad (2.4.39)$$

A Walsh operator of order $k \in \{0, \dots, 2^n - 1\}$ is a string of Z_i Pauli operators, for all $i \in \{0, \dots, n-1\}$, i.e.

$$W_k = \bigotimes_{i=0}^{n-1} (Z_i)^{k_i} = Z_0^{k_0} \otimes Z_1^{k_1} \otimes \dots \otimes Z_{n-1}^{k_{n-1}}, \quad (2.4.40)$$

where $[k] = (k_{n-1} \dots k_1 k_0)$ is the binary notation of k (note the reverse order with respect to the qubit labelling). If $k_i = 0$ then the operator applied to $|q_i\rangle$ is $Z_i^{k_i} = Z_i^0 = \mathbb{I}$ and if $k_i = 1$ then $Z_i^{k_i} = Z_i^1 = Z_i$ is applied to $|q_i\rangle$. So using any possible bit string $[k]$ we identify all the possible combinations of Z and \mathbb{I} . The set $\{W_k\}_{k=0, \dots, 2^n-1}$ forms a complete orthonormal basis of diagonal operators on n qubits. The diagonal unitary operator is therefore

$$D = e^{if} = e^{i \sum_{k=0}^{2^n-1} a_k W_k} = \prod_{k=0}^{2^n-1} e^{ia_k W_k} := \prod_{k=0}^{2^n-1} D_k, \quad (2.4.41)$$

where we have used the fact that each Walsh operator commutes with all the others so that we can split the exponential into a product of exponentials. Each term D_k is in the form of an exponential of a Walsh operator:

$$D_k = e^{ia_k w_k} := e^{-i \frac{\theta_k}{2} (Z_0^{k_0} \otimes \dots \otimes Z_{n-1}^{k_{n-1}})}, \quad (2.4.42)$$

where $\theta_k := -2a_k$, and is a special case of the Pauli string-generated operator introduced in Sec. 2.4.3. The circuit implementing this type of diagonal gate has a $R_z(\theta)$ rotation on the qubit $|q_l\rangle$ corresponding to the most significant bit ($k_l = 1$), preceded and followed by two cascades of CNOTs targeted on the same qubit $|q_l\rangle$ and with controls on the qubits $|q_i\rangle$ corresponding to $k_i = 1$.

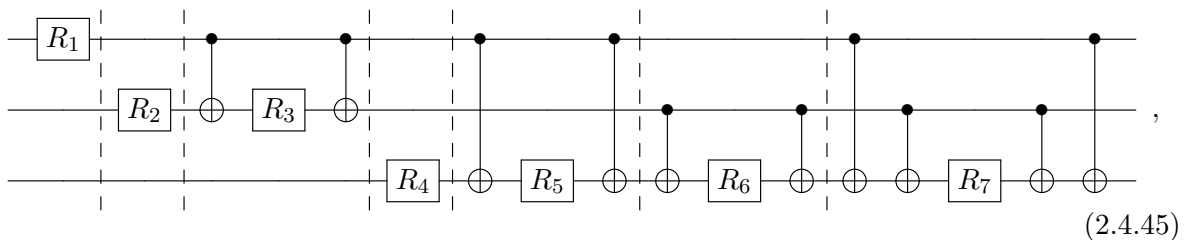
For example, consider a system of $n = 3$ qubits. The Walsh operators can be found using the binary strings $[0] = 000$, $[1] = 001$, $[2] = 010$, $[3] = 011$, $[4] = 100$, $[5] = 101$, $[6] = 110$ and $[7] = 111$, which give

$$\begin{aligned} W_0 &= \mathbb{I} \otimes \mathbb{I} \otimes \mathbb{I}, & W_1 &= Z \otimes \mathbb{I} \otimes \mathbb{I}, & W_2 &= \mathbb{I} \otimes Z \otimes \mathbb{I}, & W_3 &= Z \otimes Z \otimes \mathbb{I}, \\ W_4 &= \mathbb{I} \otimes \mathbb{I} \otimes Z, & W_5 &= Z \otimes \mathbb{I} \otimes Z, & W_6 &= \mathbb{I} \otimes Z \otimes Z, & W_7 &= Z \otimes Z \otimes Z. \end{aligned} \quad (2.4.43)$$

If we write the diagonal unitary D as

$$D = D_7 D_6 \dots D_0 = e^{-i \frac{\theta_7}{2} W_7} \dots e^{-i \frac{\theta_0}{2} W_0}, \quad (2.4.44)$$

the quantum circuit is the sequence of networks implementing each Walsh operator



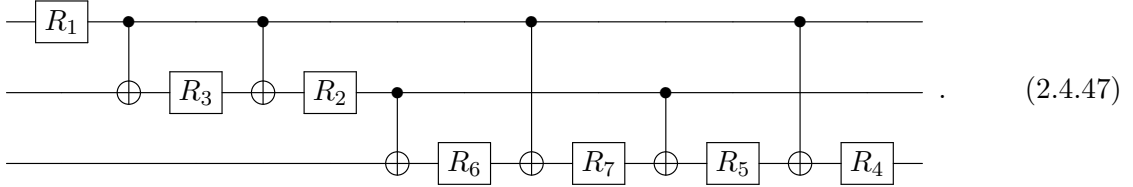
where vertical dashed lines highlight different Walsh operators and we have used the compact form $R_z(\theta_k) := R_k$. Using this approach, the number of Walsh operators required to implement D on n qubits is $2^n - 1$. Each of them requires one $R_z(\theta_k)$ rotation and up to $n - 1$ CNOTs. The complexity is therefore

$$\mathcal{C}(D) \sim \mathcal{O}(n2^n), \quad (2.4.46)$$

which scales exponentially and is not efficient.

We can optimise this approach by noting that each Walsh operator commutes with all the others, and we can order them to reduce the number of CNOTs. In particular, when two Walsh operators are sequenced, the cascade of CNOTs can be simplified by cancelling the sequential CNOTs applied to the same control qubit (see the CNOTs between R_6 and R_7 in the circuit of Eq. (2.4.45)). This follows from the circuit equivalence in Eq. (2.2.15). The CNOTs remaining

between two Walsh circuits W_k and W_l are those applied to the qubits $|q_i\rangle$ whose corresponding bits in $[k]$ and $[l]$ are different, namely $k_i \neq l_i$. This means that the optimal order is the one that minimises the flipped bits between two orders k and l of the sequential Walsh operator, which by definition coincides with the Gray code. For the case $n = 3$, the Gray code is $G_3 = ([0], [1], [3], [2], [6], [7], [5], [4])$ (see Eq. (2.3.44)), and the optimal quantum circuit is:



The Walsh operator W_k , where the most significant bit is k_0 , has only one rotation. If the second bit k_1 is the most significant, there are two Walsh operators with a total of two rotations and two CNOTs. If the third bit k_3 is the most significant, there are four Walsh operators with a total of four rotations and four CNOTs. So if k_i is the most significant bit, the number of rotations applied to $|q_i\rangle$ is 2^i and the number of CNOTs applied to $|q_i\rangle$ is 2^i if $i > 1$ and 0 if $i = 0$. Summing all the Walsh operators up to n qubits, we have

$$\# \text{ CNOTs} = \sum_{i=1}^{n-1} 2^i = 2^n - 2, \quad (2.4.48)$$

and the number of rotations is

$$\# R_z = \sum_{i=0}^{n-1} 2^i = 2^n - 1, \quad (2.4.49)$$

which gives a total number of quantum operations equal to

$$\# \text{ of operations} = 2^{n+1} - 3. \quad (2.4.50)$$

We conclude that the number of operations grows exponentially with the number of qubits, even for the optimal Walsh-based method. As proposed in Ref. [26], the number of operations can be reduced at the cost of only approximately implementing the propagator. The strategy uses a truncated Walsh series.

Related to the Walsh basis decomposition there is another interesting result which allows to implement a certain set of diagonal unitaries using only single qubit gates. Therefore, by definition, this set contains separable operations. In the work in Ref. [28], the authors identify a particular set of diagonal unitaries called *diagonal evenly-spaced* (DES) operators, which can be expressed as the exponential of a diagonal matrix A whose elements depend on a linear function on the computational basis, namely

$$D = e^{iA} = e^{i \sum_{i=0}^{2^n-1} g(i) |i\rangle\langle i|} = \sum_{i=0}^{2^n-1} e^{ig(i)} |i\rangle\langle i|, \quad (2.4.51)$$

where $g(i)$ is a linear function of i . The authors proved this statement by considering a particular operator: the bosonic number one. In this case $g(i) = i$, which gives the evenly-spaced diagonal

$$A = \sum_{i=0}^{2^n-1} i |i\rangle\langle i| = \text{diag}(0, 1, 2, \dots, 2^n - 1). \quad (2.4.52)$$

Considering the case $n = 2$, the binary notation of i , and using the relations

$$|0\rangle\langle 0| = \frac{\mathbb{I} + Z}{2}, \quad |1\rangle\langle 1| = \frac{\mathbb{I} - Z}{2}, \quad (2.4.53)$$

it is easy to verify that the operator A becomes a sum of local gates, each acting on a single qubit. In fact

$$\begin{aligned}
A &= |01\rangle\langle 01| + 2|10\rangle\langle 10| + 3|11\rangle\langle 11| \\
&= \frac{\mathbb{I} + Z}{2} \otimes \frac{\mathbb{I} - Z}{2} + 2\frac{\mathbb{I} - Z}{2} \otimes \frac{\mathbb{I} + Z}{2} + 3\frac{\mathbb{I} - Z}{2} \otimes \frac{\mathbb{I} - Z}{2} \\
&= \frac{1}{4}(\mathbb{III} - \mathbb{IZ} + Z\mathbb{I} - ZZ) + \frac{2}{4}(\mathbb{III} + \mathbb{IZ} - Z\mathbb{I} - ZZ) + \frac{3}{4}(\mathbb{III} - \mathbb{IZ} - Z\mathbb{I} + ZZ) \\
&= \frac{3}{2}\mathbb{III} - \frac{1}{2}\mathbb{IZ} - Z\mathbb{I}.
\end{aligned} \tag{2.4.54}$$

This result holds of course for any function $g(i) = \alpha i$ where $\alpha \in \mathbb{C}$ and the generated operators A can be expressed as a sum of a subset of Walsh operators W_k where $k = 2^j$ for $j \in \{0, \dots, n-1\}$ plus potentially W_0 which is a trivial phase contribution. The Walsh series corresponding to an evenly-spaced diagonal unitary is

$$\begin{aligned}
f &= a_0 w_0 + a_1 w_1 + a_2 w_2 + a_4 w_4 + \dots + a_{2^{n-1}} w_{2^{n-1}} \\
&= \left(\alpha \sum_{j=0}^{n-1} 2^j \right) w_0 + \sum_{j=0}^{n-1} (-\alpha 2^j) w_{2^j}.
\end{aligned} \tag{2.4.55}$$

For $n = 2$ and $\alpha = 1/2$ we recover the bosonic operator (since $a_1 = -\alpha = -1/2$, $a_2 = -2\alpha = -1$ and $a_0 = 3\alpha = 3/2$), but we can do better than that. Indeed, the result suggests that the locality property can be extended to any Walsh function containing at most n contributions in the list $\{w_0, w_1, w_2, w_4 \dots w_{2^n}\}$, all of which correspond to Walsh operators W_k where $k = 2^j$ for some j and whose bit string $[k]$ contains all zeros except a single bit. Thus

$$f = a_0 w_0 + \sum_{j=0}^{n-1} a_{2^j} w_{2^j} \tag{2.4.56}$$

which includes the evenly spaced diagonal unitaries, but also other diagonal unitaries that can be expressed using the subset of Walsh operators $\{W_{2^j}\}$. In particular, for the case $n = 2$ we have

$$e^{if} = e^{ia_2 W_2} e^{ia_1 W_1} e^{ia_0 W_0} = e^{ia_2 \mathbb{I} \otimes Z} e^{ia_1 Z \otimes \mathbb{I}} e^{ia_0 \mathbb{I} \otimes \mathbb{I}}. \tag{2.4.57}$$

In matrix form, this diagonal unitary can be expressed by e^{iB} , where

$$B = \sum_{i=0}^{2^n-1} g(i) |i\rangle\langle i| \iff g(i) = \sum_{j=0}^{n-1} \beta_j (-1)^{i_j} + \text{const}, \tag{2.4.58}$$

for each $\beta_j \in \mathbb{C}$, where i_j is the i -th bit of the string identifying $|j\rangle$. The evenly-spaced diagonal is a special case of the function $g(i)$ where

$$g(i) = \sum_{j=0}^{n-1} \beta_j (-1)^{i_j} = i = \sum_{j=0}^{n-1} j 2^{i_j} \implies \beta_j = j (-2)^{i_j}. \tag{2.4.59}$$

Chapter 3

Trapped-ion based quantum computing

The scientific community is investing energy and resources in developing and improving different types of quantum machines. As quantum computing research has progressed, several different types of quantum computers have emerged, each with its own unique architecture, strengths and challenges. Different testbeds are generally based on different definitions of qubit systems and different implementations of their evolution (see, for example, Ref. [29] for a recent review). These quantum computing platforms include superconducting, trapped-ion, neutral atoms, topological qubits, photonic quantum computers and quantum annealers. Each of these testbeds must fulfil certain characteristics, defined by Di Vincenzo's criteria [30], in order to be defined as efficient. The necessary characteristics of a quantum machine are (1) a scalable physical system with well-defined qubits, (2) the possibility to initialise it, (3) a long time coherence allowing not to lose the quantum superposition during the computation, (4) the definition and implementation of a universal quantum gate set, and finally (5) the possibility to make measurements of final states allowing to obtain information.

An interesting and promising quantum testbed is the one based on trapped ions, which will be used to test the quantum algorithm presented in the last part of this thesis. Before analysing the Di Vincenzo's criteria fulfilled by this type of quantum machine (Sec. 3.2), we will briefly present the physical description of a trapped-ion system (Sec. 3.1) and, at the end of the chapter, summarise the main advantages of this type of machine (Sec. 3.3). For a more detailed study, we refer the reader to more specific texts such as the Nielsen-Chuang manuscript [4] or the works in the Refs. [31–33].

3.1 Trapped-ion system and Hamiltonian

The qubit system in a trapped-ion based machine is defined by a set of two orthogonal quantum states implemented by two atomic levels of nuclei, such as $^{40}\text{Ca}^+$ and $^{171}\text{Yb}^+$, with a closed shell plus a valence electron. Two of the possible states in which the valence electron can be found are used as basic orthogonal states $|0\rangle$ and $|1\rangle$ for the computational definition of the qubit state. The chosen states must have a long lifetime and not be easily perturbed. A simple way to encode the memory of a qubit at the atomic level of the ion is to take a ground and an excited state, defining a so-called *optical qubit* as an alternative to the *hyperfine qubit* type, which stores the information in two stable states in a hyperfine structure¹.

¹A hyperfine structure is an energy split caused by an electromagnetic field (generated by a multipole between the charges in the nucleus and the electron) between two atomic levels that would otherwise be degenerate.

The total Hamiltonian of the valence electron used as a qubit contains three terms:

$$H_{ion} = H_0 + H_{trap} + H_{int}, \quad (3.1.1)$$

where H_0 is the free part, H_{trap} is the external potential used to create the trap, and H_{int} is the atom-light interaction term, which describes the way we will act on the ion with a laser beam.

3.1.1 Free term

The free Hamiltonian is given by

$$H_0 = \hbar\omega_0 |1\rangle\langle 1|, \quad (3.1.2)$$

where ω_0 is the frequency distance between the ground state $|0\rangle$ and the excited one $|1\rangle$ (in the case of optical qubits). The frequency distance is of the order of $\omega_0 \sim 100$ THz (optical range) which is compatible with a laser frequency that can be used to manipulate the qubit state, but of course depends on the particular ion used to define the qubit.

3.1.2 Trap term

Because the ions are charged, it is possible to apply an electromagnetic field to them, but according to Earnshaw's theorem based on Gauss's law, an electric potential alone cannot trap a charged particle ($\nabla \cdot \mathbf{E} = 0$). A static field cannot actually create a 3D confining potential, only a saddle potential, so the trap is constructed using an electric field plus a magnetic field (Penning trap) or a dynamic electric field (Paul trap). Both technologies create a rotating saddle that confines in all radial directions, and the ion is forced into a periodic motion of very small amplitude around an equilibrium point. Considering the dimensional scale of the ions ~ 100 pm and the distance scale of different ions in the trap $\sim \mu\text{m}$, we can treat the ions in the trap as classical charged points and the Hamiltonian of this system can be modelled by a harmonic oscillator in which the ions, of mass m , perform phonon modes:

$$H_{trap} = \frac{p^2}{2m} + \frac{1}{2}m\omega_{ph}^2 x^2. \quad (3.1.3)$$

Due to the confining we can consider here the vibration degree of freedom in only one direction. Replacing the operators p and x by the ladder operators according to

$$x = \sqrt{\frac{\hbar}{2m\omega_{ph}}} (a + a^\dagger), \quad p = i\sqrt{\frac{m\omega_{ph}\hbar}{2}} (a^\dagger - a), \quad (3.1.4)$$

one gets

$$H_{trap} = \hbar\omega_{ph} \left(a^\dagger a + \frac{1}{2} \right), \quad (3.1.5)$$

where ω_{ph} is the frequency and $a^{(\dagger)}$ are the annihilation (creation) operators of the harmonic oscillator levels.

3.1.3 Atom-light interaction term

The interaction with the valence electron is usually performed by a laser beam, which can be described by a time-dependent electric field

$$\mathbf{E}(t) = E_0 \boldsymbol{\varepsilon} \cos(k_L x - \omega_L t + \varphi), \quad (3.1.6)$$

where E_0 is the amplitude, $\boldsymbol{\varepsilon}$ is the polarisation vector, k_L is the wave vector, ω_L is the frequency and φ is the phase. Note that the bold symbols here indicate three-dimensional vectors. The Hamiltonian for the interaction of an electron in an electric field is

$$H_{int} = e(\boldsymbol{r} \cdot \boldsymbol{E}) = \boldsymbol{D} \cdot \boldsymbol{E}, \quad (3.1.7)$$

where $\boldsymbol{D} := e\boldsymbol{r}$ is the dipole operator which is defined by the position distance \boldsymbol{r} . If the electron can occupy two possible states, the interaction Hamiltonian can be written as

$$H_{int} = (|0\rangle\langle 0| + |1\rangle\langle 1|) (\boldsymbol{D} \cdot \boldsymbol{E}) (|0\rangle\langle 0| + |1\rangle\langle 1|) = \sum_{i,j \in \{0,1\}} |i\rangle\langle i| (\boldsymbol{D} \cdot \boldsymbol{E}) |j\rangle\langle j|, \quad (3.1.8)$$

where we have inserted two identities $\mathbb{I} = |0\rangle\langle 0| + |1\rangle\langle 1|$. We can see that the terms with $i = j$ are zero due to parity properties, because atomic orbitals $\psi(\boldsymbol{r})$ are in general symmetric (like S-wave) or antisymmetric (like P-wave), which means that the parity operator mapping $P : \boldsymbol{r} \mapsto -\boldsymbol{r}$ transforms them into $P\psi(\boldsymbol{r}) = \pm\psi(\boldsymbol{r})$. Since P is self-adjoint and unitary, we have that $P^2 = \mathbb{I}$ and we can calculate the matrix elements of the operator as

$$\langle i|\boldsymbol{D} \cdot \boldsymbol{\varepsilon}|i\rangle = \langle i|PP\boldsymbol{D} \cdot \boldsymbol{\varepsilon}PP|i\rangle = \langle i|P(\boldsymbol{D} \cdot \boldsymbol{\varepsilon})P|i\rangle = -\langle i|\boldsymbol{D} \cdot \boldsymbol{\varepsilon}|i\rangle := 0, \quad (3.1.9)$$

where we used $P|i\rangle = \pm|i\rangle$ and $P(\boldsymbol{D} \cdot \boldsymbol{\varepsilon})P = -\boldsymbol{D} \cdot \boldsymbol{\varepsilon}$. So the interaction Hamiltonian contains only off-diagonal elements

$$\begin{aligned} H_{int} &= \langle 0|\boldsymbol{D} \cdot \boldsymbol{E}|1\rangle |0\rangle\langle 1| + \langle 1|\boldsymbol{D} \cdot \boldsymbol{E}|0\rangle |1\rangle\langle 0| \\ &= E_0 \cos(k_L x - \omega_L t + \varphi) (\langle 0|\boldsymbol{D} \cdot \boldsymbol{\varepsilon}|1\rangle |0\rangle\langle 1| + \langle 1|\boldsymbol{D} \cdot \boldsymbol{\varepsilon}|0\rangle |1\rangle\langle 0|) \\ &:= \hbar\Omega \cos(k_L x - \omega_L t + \varphi) (|1\rangle\langle 0| + |0\rangle\langle 1|) \\ &= \hbar\Omega \cos(k_L x - \omega_L t + \varphi) \sigma_x, \end{aligned} \quad (3.1.10)$$

where we defined $d_{10} := \langle 1|\boldsymbol{D} \cdot \boldsymbol{\varepsilon}|0\rangle = \langle 0|\boldsymbol{D} \cdot \boldsymbol{\varepsilon}|1\rangle$ and the Rabi frequency

$$\Omega := \frac{d_{10}E_0}{\hbar}. \quad (3.1.11)$$

3.1.4 The Lamb-Dicke regime

The Lamb-Dicke parameter η is a measure of the ratio between the oscillation energy induced by the laser beam and that of the trap:

$$\eta := \sqrt{\frac{\hbar^2 k_L^2 / 2m}{\hbar\omega_{ph}}}. \quad (3.1.12)$$

The numerator meant to produce the electron transition (recoil energy from photon absorption) and the denominator corresponds to the energy distance between different levels of the harmonic oscillator. Using the first relation in Eq. (3.1.4) we also have the equivalence

$$k_L x = \eta (a + a^\dagger). \quad (3.1.13)$$

The Lamb-Dicke regime corresponds to the condition that $\eta \ll 1$. In this limit the interaction term in the Hamiltonian can be approximated by neglecting $k_L x \sim 0$ and the interaction contribution becomes

$$\begin{aligned} H_{int} &= \hbar\Omega \cos(-\omega_L t + \varphi) (|1\rangle\langle 0| + |0\rangle\langle 1|) \\ &= \frac{\hbar\Omega}{2} \left(e^{i(-\omega_L t + \varphi)} + e^{-i(-\omega_L t + \varphi)} \right) (|1\rangle\langle 0| + |0\rangle\langle 1|). \end{aligned} \quad (3.1.14)$$

Physically, the Lamb-Dicke limit creates a condition where the energy phonon (the vibration of the ion in the harmonic potential) is not sufficient to trigger the flip from the ground to the excited state, i.e. it cannot change the state of the qubit, and the two external manipulations, namely the one that keeps the ion trapped and the one that manipulates the state of the valence electron, do not interfere with each other. This is a fundamental requirement for trapped-ion quantum computing because it allows precise and controlled manipulation of the qubits, leading to precise quantum operations. This establishes a point to which the trapped ions should be cooled. In general, the Lamb-Dicke parameter η is small but not negligible, and this is also a necessary condition to couple the electron levels with the phonon motion in the trap. We can write the interaction Hamiltonian considering the relation in Eq. (3.1.13) and obtaining

$$\begin{aligned} H_{int} &= \frac{\hbar\Omega}{2} \left(e^{i(-\omega_L t + \varphi)} e^{i\eta(a^\dagger + a)} + e^{-i(-\omega_L t - \varphi)} e^{-i\eta(a^\dagger + a)} \right) (|1\rangle\langle 0| + |0\rangle\langle 1|) \\ &\approx \frac{\hbar\Omega}{2} \left(e^{i(-\omega_L t + \varphi)} (1 + i\eta(a^\dagger + a)) + e^{-i(-\omega_L t - \varphi)} (1 - i\eta(a^\dagger + a)) \right) (|1\rangle\langle 0| + |0\rangle\langle 1|), \end{aligned} \quad (3.1.15)$$

where we have expanded the relation for small values of η .

3.1.5 The rotating wave approximation

If we consider a single ion in the trap, we can describe its internal state using the free plus interaction Hamiltonian in the Lamb-Dicke regime (see Eq. (3.1.14)) which coincides with

$$H_0 + H_{int} = \hbar\omega_0 |1\rangle\langle 1| + \frac{\hbar\Omega}{2} \left(e^{i(-\omega_L t + \varphi)} + e^{-i(-\omega_L t + \varphi)} \right) (|1\rangle\langle 0| + |0\rangle\langle 1|). \quad (3.1.16)$$

Note that the trap term in the Hamiltonian is not needed if we want to describe only the electron state in the ion. In the rotating frame of the free term H_0 , one obtains a Hamiltonian in the interaction picture according to the relation

$$H' = U_0^\dagger H U_0 - i\hbar U_0^\dagger \frac{dU_0}{dt} = U_0^\dagger H_{int} U_0, \quad (3.1.17)$$

where here $H = H_0 + H_{int}$ and $U_0 = e^{-iH_0 t}$. This eliminates the first term in Eq. (3.1.16) adding a rotation with a new frequency, namely

$$\begin{aligned} H' &= \frac{\hbar\Omega}{2} \left(e^{i(-\omega_L t + \varphi)} + e^{-i(-\omega_L t + \varphi)} \right) \left(U_0^\dagger |1\rangle\langle 0| U_0 + U_0^\dagger |0\rangle\langle 1| U_0 \right) \\ &= \frac{\hbar\Omega}{2} \left(e^{i(-\omega_L t + \varphi)} + e^{-i(-\omega_L t + \varphi)} \right) \left(e^{i\omega_0 t} |1\rangle\langle 0| + e^{-i\omega_0 t} |0\rangle\langle 1| \right) \\ &= \frac{\hbar\Omega}{2} \left(e^{i\varphi} e^{-i(\omega_L - \omega_0)t} + e^{-i\varphi} e^{i(\omega_L + \omega_0)t} \right) |1\rangle\langle 0| + \\ &\quad + \frac{\hbar\Omega}{2} \left(e^{i\varphi} e^{-i(\omega_L + \omega_0)t} + e^{-i\varphi} e^{i(\omega_L - \omega_0)t} \right) |0\rangle\langle 1|, \end{aligned} \quad (3.1.18)$$

where we have used

$$U_0^\dagger |1\rangle\langle 0| U_0 = e^{i\omega_0 t} |1\rangle\langle 0|, \quad U_0^\dagger |0\rangle\langle 1| U_0 = e^{-i\omega_0 t} |0\rangle\langle 1|. \quad (3.1.19)$$

The Rotating Wave Approximation (RWA) allows us to neglect terms that define oscillations with a much faster frequency than the Rabi oscillation. In particular $\omega_L + \omega_0 \sim \text{GHz}$ while $\Omega \sim \text{MHz}$ so we can neglect the first oscillations and keep only the contribution with $\omega_L - \omega_0 := \delta$, called detuning, because the fast rotating terms average out before the system has had time to react. The Hamiltonian is then

$$H' = \frac{\hbar\Omega}{2} \left(e^{i\varphi} e^{-i\delta t} |1\rangle\langle 0| + e^{-i\varphi} e^{i\delta t} |0\rangle\langle 1| \right). \quad (3.1.20)$$

When considering more than one ion in the trap we need to combine both the internal and the external degrees of freedom. The space describing this system is then spanned by the space of the valence electron basis $|i\rangle_q$ coupled to the harmonic oscillator basis $|n\rangle_{ph}$, namely

$$\mathcal{B} = \text{span} \left(|n\rangle_{ph} \otimes |i\rangle_q \text{ s.t. } i \in \{0, 1\}, n \in \mathbb{R} \right), \quad (3.1.21)$$

where n denotes the harmonic oscillator levels. The Lamb-Dicke parameter contains the connection between the manipulation of the internal levels of the ion and the external degree of freedom of the harmonic potential. For this reason, to describe a system with more than one ion in the trap, we have to work in a situation where η is small but not negligible (see Eq. (3.1.15)). This allows us to couple the two degrees of freedom. The Hamiltonian is now $H_0 + H_{trap} + H_{int}$ where the interaction term is as in Eq. (3.1.15). This Hamiltonian in the rotating frame of $H_0 + H_{trap}$ is given by

$$H' = U^\dagger H U - i\hbar U^\dagger \frac{dU}{dt} = U^\dagger H_{int} U, \quad (3.1.22)$$

where now $H = H_0 + H_{trap} + H_{int}$ and $U = e^{-i(H_0 + H_{trap})t}$. Using Eq. (3.1.15) for the interaction part we have

$$H' = \frac{\hbar\Omega}{2} \left(e^{i(-\omega_L t + \varphi)} U^\dagger \left(1 + i\eta (a^\dagger + a) \right) \sigma_x U + e^{-i(-\omega_L t + \varphi)} U^\dagger \left(1 - i\eta (a^\dagger + a) \right) \sigma_x U \right). \quad (3.1.23)$$

Making the two Hilbert spaces explicit (ordered as in the basis (3.1.21)), the two contributions are of the form

$$U^\dagger \left(\mathbb{I} \otimes \sigma_x \pm i\eta (a^\dagger + a) \otimes \sigma_x \right) U \quad (3.1.24)$$

and the interaction picture Hamiltonian is

$$\begin{aligned} H' &= \frac{\hbar\Omega}{2} U^\dagger (\mathbb{I} \otimes \sigma_x) U (e^{-i\omega_L t} e^{i\varphi} + e^{i\omega_L t} e^{-i\varphi}) + \\ &\quad + \frac{\hbar\Omega}{2} i\eta U^\dagger \left((a^\dagger + a) \otimes \sigma_x \right) U (e^{-i\omega_L t} e^{i\varphi} - e^{i\omega_L t} e^{-i\varphi}) \\ &:= H'_q + H'_{ph \otimes q}. \end{aligned} \quad (3.1.25)$$

In the equation above, we defined H'_q as the term that can only act on the valence electron energy level, preserving the harmonic oscillation, and $H'_{ph \otimes q}$ as the term that couples the two Hilbert spaces. Calculating

$$\begin{aligned} U^\dagger (\mathbb{I} \otimes \sigma_x) U &= \mathbb{I} \otimes e^{iH_0 t/\hbar} \sigma_x e^{-iH_0 t/\hbar} \\ &= \mathbb{I} \otimes (e^{i\omega_0 t} |1\rangle\langle 0| + e^{-i\omega_0 t} |0\rangle\langle 1|), \end{aligned} \quad (3.1.26)$$

H'_q becomes explicitly

$$H'_q = \frac{\hbar\Omega}{2} (e^{-i\omega_L t} e^{i\varphi} + e^{i\omega_L t} e^{-i\varphi}) (\mathbb{I} \otimes (e^{i\omega_0 t} |1\rangle\langle 0| + e^{-i\omega_0 t} |0\rangle\langle 1|)) \quad (3.1.27)$$

and is dominant in the regime $\omega_L \sim \omega_0$, where the RWA keeps only the slowly rotating term containing $\delta = \omega_L - \omega_0$, giving the result obtained in Eq. (3.1.20). This transformation is represented by the black transition in the Figure 3.1. Let's calculate

$$\begin{aligned} U^\dagger \left((a^\dagger + a) \otimes \sigma_x \right) U &= e^{iH_{trap} t/\hbar} (a^\dagger + a) e^{-iH_{trap} t/\hbar} \otimes e^{iH_0 t/\hbar} \sigma_x e^{-iH_0 t/\hbar} \\ &= \left(e^{i\omega_{ph} t} a^\dagger + e^{-i\omega_{ph} t} a \right) \otimes (e^{i\omega_0 t} |1\rangle\langle 0| + e^{-i\omega_0 t} |0\rangle\langle 1|) \\ &= e^{i(\omega_{ph} + \omega_0)t} a^\dagger \otimes |1\rangle\langle 0| + e^{i(\omega_{ph} - \omega_0)t} a^\dagger \otimes |0\rangle\langle 1| + \\ &\quad + e^{-i(\omega_{ph} - \omega_0)t} a \otimes |1\rangle\langle 0| + e^{-i(\omega_{ph} + \omega_0)t} a \otimes |0\rangle\langle 1|, \end{aligned} \quad (3.1.28)$$

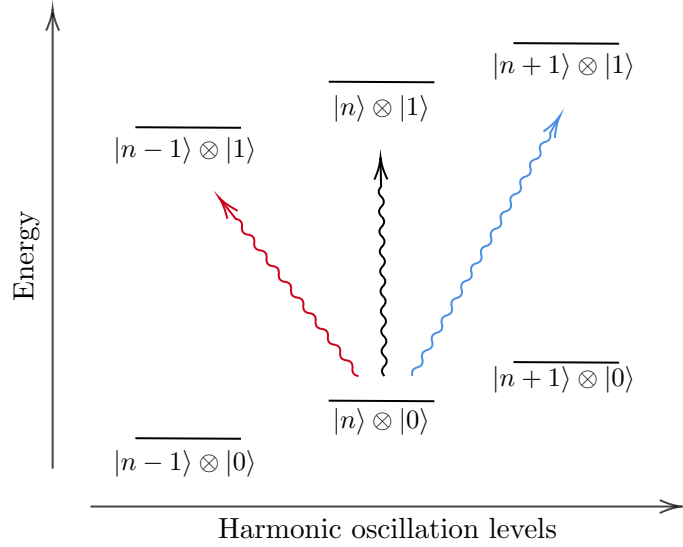


Figure 3.1: Sketch of the action of the interaction Hamiltonian H_{int} in the three main regimes. The red and blue transitions are generated by the red and blue regimes of the Hamiltonians in Eq. (3.1.32) and (3.1.33), and the black one describes the transition of the qubit state generated by Eq. (3.1.27).

where we have used the substitutions

$$a^\dagger \mapsto e^{i\omega_{ph}t} a^\dagger, \quad a \mapsto e^{-i\omega_{ph}t} a \quad (3.1.29)$$

obtained from the annihilation and creation operators in the rotating frame of the H_{trap} ². The term $H'_{ph\otimes q}$ becomes

$$H'_{ph\otimes q} = \frac{\hbar\Omega}{2} i\eta \left(e^{-i\omega_L t} e^{i\varphi} - e^{i\omega_L t} e^{-i\varphi} \right) \left(e^{i(\omega_{ph}+\omega_0)t} a^\dagger \otimes |1\rangle\langle 0| + e^{i(\omega_{ph}-\omega_0)t} a^\dagger |0\rangle\langle 1| + e^{-i(\omega_{ph}-\omega_0)t} a \otimes |1\rangle\langle 0| + e^{-i(\omega_{ph}+\omega_0)t} a \otimes |0\rangle\langle 1| \right). \quad (3.1.31)$$

This contribution is dominant in two different regimes. In the first one, called **red sideband**, when $\omega_L \sim \omega_0 - \omega_{ph}$, the Hamiltonian reads

$$(H'_{ph\otimes q})^{red} = \frac{\hbar\Omega}{2} i\eta \left(e^{i\varphi} e^{-i(\omega_L+\omega_{ph}-\omega_0)t} a \otimes |1\rangle\langle 0| - e^{-i\varphi} e^{i(\omega_L+\omega_{ph}-\omega_0)t} a^\dagger \otimes |0\rangle\langle 1| \right), \quad (3.1.32)$$

where we have neglected the fast rotating terms. This physically described the effect of increasing the electron energy level and decreasing the harmonic oscillator one, or decreasing the first and increasing the second (red transition in Fig. 3.1). The second regime corresponds to $\omega_L \sim \omega_0 + \omega_{ph}$ and is called **blue sideband**. In this case the Hamiltonian is

$$(H'_{ph\otimes q})^{blue} = \frac{\hbar\Omega}{2} i\eta \left(e^{i\varphi} e^{-i(\omega_L-\omega_{ph}-\omega_0)t} a^\dagger \otimes |1\rangle\langle 0| - e^{-i\varphi} e^{i(\omega_L-\omega_{ph}-\omega_0)t} a \otimes |0\rangle\langle 1| \right), \quad (3.1.33)$$

²The explicit calculation for the creation operator in the rotating frame of the trap is

$$\begin{aligned} e^{iH_{trap}t/\hbar} a^\dagger e^{-iH_{trap}t/\hbar} |n\rangle &= e^{i\omega_{ph}a^\dagger at} a^\dagger e^{-i\omega_{ph}a^\dagger at} |n\rangle \\ &= e^{i\omega_{ph}a^\dagger at} a^\dagger |n\rangle e^{-i\omega_{ph}nt} \\ &= e^{i\omega_{ph}a^\dagger at} \sqrt{n+1} |n+1\rangle e^{-i\omega_{ph}nt} \\ &= e^{i\omega_{ph}(n+1)t} \sqrt{n+1} |n+1\rangle e^{-i\omega_{ph}nt} \\ &= e^{i\omega_{ph}t} a^\dagger |n\rangle. \end{aligned} \quad (3.1.30)$$

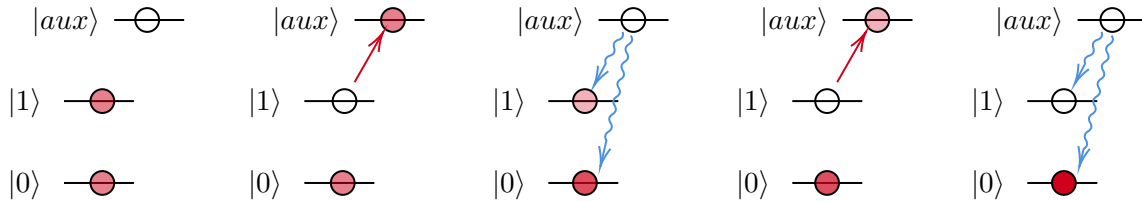


Figure 3.2: Sketch of the initialisation process. The transparency of the dots represents the occupation probability of the state, the red arrows are the transformations performed by X_{ini} and the blue ones represent the spontaneous decay. We repeat the step $X_{ini} + \text{decay}$ until all occupation probabilities are in the state $|0\rangle$.

and has the physical effect of raising or lowering the energy levels of both the electron and the harmonic oscillator (blue transition in Fig. 3.1). This Hamiltonian term is the necessary ingredient to implement a two-qubit gate, where the electrons of two ions can communicate with each other through the trap.

3.2 Di Vincenzo criteria for trapped-ion QC

3.2.1 Qubit definition

The first requirement that makes a two-level quantum system a good candidate for a qubit is that it has two orthogonal well-defined states that can be mapped to $|0\rangle$ and $|1\rangle$. As seen in the previous section, we can identify two energy levels for the valence electron of the ion, which can represent the two basis elements. The system is scalable by simply adding more ions to the trap. The typical number of ions is of the order of $\sim (1 - 100)$ with a geometry that is generally linear and with a typical scale distance of $\sim (1 - 10) \mu\text{m}$.

3.2.2 Initialization

The state of the valence electron is initialised and changed by a laser beam. As far as we are concerned at this stage, let us imagine that we have an additional state $|aux\rangle$ for the electron, and an operation that performs $X_{ini} |1\rangle = |aux\rangle$. The auxiliary state is chosen to satisfy some necessary properties: (1) it must be unstable, (2) it must have a typical decay time in state $|0\rangle$ that is shorter than the decay time in state $|1\rangle$, and (3) the frequency distance with $|1\rangle$ must be different from that with $|0\rangle$. In this way, given a certain unknown initial state $|\varphi\rangle = \alpha |0\rangle + \beta |1\rangle$, we can initialise it in the state $|0\rangle$ by repeatedly applying X_{ini} according to the procedure shown in Figure 3.2, obtaining a final state $|\varphi_0\rangle \approx |0\rangle$ with a fidelity of $\mathcal{F} = |\langle 0|\varphi_0\rangle|^2 > 99.9\%$.

3.2.3 Coherence

Quantum coherence is a key aspect of quantum computing as it ensures the conservation of quantum superposition. A two-level quantum system, such as the valence electron of the trapped ion, is a good definition of a qubit if it maintains coherence between the two possible levels for a time scale longer than the time required to perform the computation and obtain information from the measurement. In the specific case of an ion machine, the decay of the state from $|1\rangle$ to $|0\rangle$ has a typical time of the order of $\tau_{decay} \sim \text{s}$, which is orders of magnitude larger than the typical time of the gates that implement the simulation, which operate on a time scale of the order of $\tau_{gate} \sim (10 - 100) \mu\text{s}$. In fact, the main source of error in trapped-ion qubits is due to magnetic field fluctuations, which cause the frequency of the free Hamiltonian to depend on time, namely $H_0 = \hbar\omega_0(t) |1\rangle\langle 1|$. When acting on the basis states $e^{-i\omega_0(t)|1\rangle\langle 1|} |0\rangle = |0\rangle$

and $e^{-i\omega_0(t)|1\rangle\langle 1|} |1\rangle = e^{-i\omega_0(t)} |1\rangle$ it has the effect of a random σ_z gate which affects the phase information.

3.2.4 Universal quantum gate set

Another necessary requirement is the ability to perform single-qubit and two-qubit operations, which can be used to manipulate the qubit system and form a universal set of gates. Single-qubit gates are applied using the laser beam, which changes the qubit state depending on the duration of its application. To prove this, let's focus on the Hamiltonian of a single ion in the rotating frame of H_0 and in the Lamb-Dicke limit, as obtained in Eq. (3.1.20). Without losing generality, we can set the laser frequency ω_L such that $\delta \sim 0$, and obtain

$$H' = \frac{\hbar\Omega}{2} (e^{i\varphi} |1\rangle\langle 0| + e^{-i\varphi} |0\rangle\langle 1|) = \frac{\hbar\Omega}{2} \begin{pmatrix} 0 & e^{-i\varphi} \\ e^{i\varphi} & 0 \end{pmatrix}, \quad (3.2.1)$$

which, for $\varphi = 0$ and $\varphi = \pi/2$ respectively, corresponds to

$$H' = \frac{\hbar\Omega}{2} \sigma_x, \quad H' = \frac{\hbar\Omega}{2} \sigma_y. \quad (3.2.2)$$

Applying this Hamiltonian, the system evolves according to the unitary evolution described by

$$e^{iH't/\hbar} = e^{i\Omega\sigma_x t/2} = \cos(\Omega t/2)\mathbb{I} + i \sin(\Omega t/2)X = R_x(\Omega t), \quad (3.2.3)$$

for $\varphi = 0$, and by

$$e^{iH't/\hbar} = e^{i\Omega\sigma_y t/2} = \cos(\Omega t/2)\mathbb{I} + i \sin(\Omega t/2)Y = R_y(\Omega t), \quad (3.2.4)$$

for $\varphi = \pi/2$. By changing the duration of the laser, we can obtain any angle $\Omega t \in \mathbb{R}$, and the two single-qubit rotations form a universal single-qubit gate set. The typical time scale for a single-qubit rotation is of the order of $\sim 10 \mu\text{s}$.

The two-qubit entangling gate is implemented by the laser beam which couples the internal qubit state with the collective phonon motion. Thus, the basis used to implement this kind of operation is the one in Eq. (3.1.21). The Hamiltonian describing the coupling between the two degrees of freedom is given by Eq. (3.1.25) in the rotating frame of $H_0 + H_{trap}$. Using the RWA in the red sideband (Eq. (3.1.32)) and fixing the laser frequency to $\omega_L = \omega_0 - \omega_{ph}$ and the phase to $\varphi = -\pi/2$, we get

$$(H'_{ph\otimes q})^{red} = \frac{\hbar\Omega}{2} \eta \left(a \otimes |1\rangle\langle 0| + a^\dagger \otimes |0\rangle\langle 1| \right), \quad (3.2.5)$$

that generates the unitary

$$U_{ph\otimes q}(t) := e^{-iH'_{ph\otimes q}t/\hbar} = e^{-i\frac{\Omega\eta}{2}(a^\dagger \otimes |0\rangle\langle 1| + a \otimes |1\rangle\langle 0|)t}. \quad (3.2.6)$$

This maps

$$\begin{aligned} U_{ph\otimes q}(t) : |0\rangle_{ph} \otimes |1\rangle_q &\longmapsto \cos\left(\frac{\Omega\eta}{2}t\right) |0\rangle_{ph} \otimes |1\rangle_q - i \sin\left(\frac{\Omega\eta}{2}t\right) |1\rangle_{ph} \otimes |0\rangle_q \\ U_{ph\otimes q}(t) : |1\rangle_{ph} \otimes |0\rangle_q &\longmapsto \cos\left(\frac{\Omega\eta}{2}t\right) |1\rangle_{ph} \otimes |0\rangle_q - i \sin\left(\frac{\Omega\eta}{2}t\right) |0\rangle_{ph} \otimes |1\rangle_q. \end{aligned} \quad (3.2.7)$$

Using this operation, we can implement the so-called **Cirac-Zoller** gate, first proposed in Ref. [34] and demonstrated in Ref. [35]. For simplicity, let's consider two ions and two states

$\{|0\rangle_{q_1}, |1\rangle_{q_1}\}$ for the first one and three states $\{|0\rangle_{q_2}, |1\rangle_{q_2}, |aux\rangle_{q_2}\}$ for the second one, where the auxiliary state is never occupied. The basis elements are of the form

$$|n\rangle_{ph} \otimes |i\rangle_{q_1} \otimes |j\rangle_{q_2} \in \mathcal{L}^2(\mathbb{R}) \otimes \mathbb{C}^2 \otimes \mathbb{C}^2. \quad (3.2.8)$$

First we apply the operator in Eq. (3.2.6) to the first ion for a duration of $\Omega t = \pi$. This π pulse maps

$$\begin{aligned} U_1 : |0\rangle_{ph} \otimes |1\rangle_{q_1} \otimes |j\rangle_{q_2} &\longmapsto -i |1\rangle_{ph} \otimes |0\rangle_{q_1} \otimes |j\rangle_{q_2} \\ U_1 : |1\rangle_{ph} \otimes |0\rangle_{q_1} \otimes |j\rangle_{q_2} &\longmapsto -i |0\rangle_{ph} \otimes |1\rangle_{q_1} \otimes |j\rangle_{q_2}. \end{aligned} \quad (3.2.9)$$

We now apply a 2π pulse to the second ion at a frequency which includes $|0\rangle_{q_2}$ and $|aux\rangle_{q_2}$, namely

$$U_2 := e^{-i\frac{2\pi}{2}(a^\dagger \otimes |0\rangle\langle aux| + a \otimes |aux\rangle\langle 0|)}. \quad (3.2.10)$$

So there is only one state that changes under its action:

$$\begin{aligned} U_2 : |1\rangle_{ph} \otimes |i\rangle_{q_1} \otimes |0\rangle_{q_2} &\longmapsto - |1\rangle_{ph} \otimes |i\rangle_{q_1} \otimes |0\rangle_{q_2} \\ U_2 : |1\rangle_{ph} \otimes |i\rangle_{q_1} \otimes |1\rangle_{q_2} &\longmapsto |1\rangle_{ph} \otimes |i\rangle_{q_1} \otimes |1\rangle_{q_2}, \end{aligned} \quad (3.2.11)$$

and it acquires a phase of π . This is due to the use of the auxiliary state. In fact, using the excited $|1\rangle_{q_2}$ for this transition would also change the phase of the second line in the Eq. (3.2.11). Then we again apply the U_1 operator to the first ion and the chain of operators $U_1 U_2 U_1 := U_{CZ}$ is the so-called Cirac-Zoller gate which transforms, as follows, the states of the form $|0\rangle_{ph} \otimes |ij\rangle_{qq}$ where $|ij\rangle_{qq} := |i\rangle_{q_1} \otimes |j\rangle_{q_2}$:

$$\begin{array}{ccccccc} & U_1 & & U_2 & & U_1 & \\ |0\rangle_{ph} \otimes |00\rangle_{qq} & \longmapsto & |0\rangle_{ph} \otimes |00\rangle_{qq} & \longmapsto & |0\rangle_{ph} \otimes |00\rangle_{qq} & \longmapsto & |0\rangle_{ph} \otimes |00\rangle_{qq} \\ |0\rangle_{ph} \otimes |01\rangle_{qq} & \longmapsto & |0\rangle_{ph} \otimes |01\rangle_{qq} & \longmapsto & |0\rangle_{ph} \otimes |01\rangle_{qq} & \longmapsto & |0\rangle_{ph} \otimes |01\rangle_{qq} \\ |0\rangle_{ph} \otimes |10\rangle_{qq} & \longmapsto & -i |1\rangle_{ph} \otimes |00\rangle_{qq} & \longmapsto & i |1\rangle_{ph} \otimes |00\rangle_{qq} & \longmapsto & |0\rangle_{ph} \otimes |10\rangle_{qq} \\ |0\rangle_{ph} \otimes |11\rangle_{qq} & \longmapsto & -i |1\rangle_{ph} \otimes |01\rangle_{qq} & \longmapsto & -i |1\rangle_{ph} \otimes |01\rangle_{qq} & \longmapsto & - |0\rangle_{ph} \otimes |11\rangle_{qq}. \end{array} \quad (3.2.12)$$

This gate thus has the final effect of a $CZ = \Lambda_1(\sigma_z)$ gate, which is equivalent to a CNOT if we apply a Hadamard gate to the target qubit, changing the basis according to the circuit in Eq. (2.2.10) of Chapter 2. The action is in fact

$$\begin{array}{ccccccc} & \mathbb{I} \otimes H & & CZ & & \mathbb{I} \otimes H & \\ |0\rangle \otimes |0\rangle & \longmapsto & |0\rangle \otimes |+\rangle & \longmapsto & |0\rangle \otimes |+\rangle & \longmapsto & |0\rangle \otimes |0\rangle \\ |0\rangle \otimes |1\rangle & \longmapsto & |0\rangle \otimes |-\rangle & \longmapsto & |0\rangle \otimes |-\rangle & \longmapsto & |0\rangle \otimes |1\rangle \\ |1\rangle \otimes |0\rangle & \longmapsto & |1\rangle \otimes |+\rangle & \longmapsto & |1\rangle \otimes |-\rangle & \longmapsto & |1\rangle \otimes |1\rangle \\ |1\rangle \otimes |1\rangle & \longmapsto & |1\rangle \otimes |-\rangle & \longmapsto & |1\rangle \otimes |+\rangle & \longmapsto & |1\rangle \otimes |0\rangle \end{array} \quad (3.2.13)$$

and has a typical time scale of the order of $\sim (10 - 100) \mu\text{s}$.

3.2.5 Measurement

The ability to measure the qubit state is the final requirement for a machine to become an efficient quantum computer. In a trapped-ion based testbed, the measurement part is performed using a procedure similar to that used for initialisation. A transformation X_{meas} is applied, which

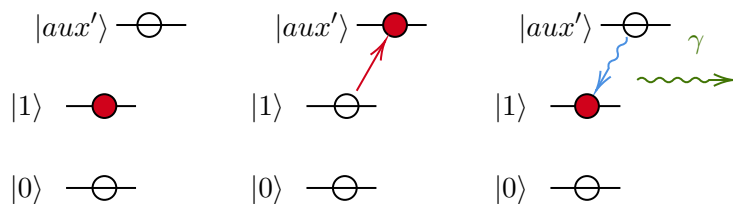


Figure 3.3: Sketch of the qubit measurement process. Red arrow is the transformation performed by X_{meas} and the blue one represents the spontaneous decay. A photon is only detected if the valence electron is measured in the $|1\rangle$ state.

transfers the occupancy probabilities from the state $|1\rangle$ to an auxiliary state $|aux'\rangle$. The decay of the new state $|aux'\rangle$ emits a photon with a time scale of the order of $\tau \sim$ ns, which is detected and we have the correspondence $|1\rangle \mapsto$ light and $|0\rangle \mapsto$ dark. A crucial aspect in the definition of the auxiliary state is that the transition $|0\rangle \mapsto |aux'\rangle$ must be strongly off-resonant. The procedure is shown in Fig. 3.3.

3.3 General features of a trapped-ion machine

Trapped-ion quantum computers offer several distinct advantages compared to other quantum computing architectures, such as superconducting qubits or topological qubits. These advantages arise from the unique properties of trapped ions and their interactions with electromagnetic fields. A key point is that trapped ions have relatively long qubit coherence times, which is related to how well quantum information is preserved before it decoheres. This extended coherence allows for more complex and longer quantum computations. In addition, trapped-ion qubits can be manipulated with extremely high fidelity using well-controlled laser interactions. This precision allows the implementation of high-quality quantum gates, which will also make near-term quantum devices suitable for relatively complex quantum algorithms, as well as a high-fidelity measurement technique using laser-induced fluorescence. The inherently stable and isolated nature of trapped ions allows the construction of large-scale quantum processors with relatively large numbers of qubits. A crucial and very useful aspect of this type of architecture is that entanglement can be created between arbitrarily distant ions using a variety of techniques, resulting in a fully connected qubit system in which all qubits can interact with all others. As we will see in Chapter 6, this feature makes a trapped-ion testbed suitable for implementing a class of quantum algorithms that describe the Hamiltonian evolution of quantum systems involving an all-to-all interaction term between the different degrees of freedom.

Chapter 4

An explicit tensor notation for quantum computing

Two notations are commonly used in the literature to describe quantum computing theory, where we establish a convention for the definition of qubits, gates and quantum algorithms. The first is the *Dirac notation*, which has its origins in the mathematical formalism of quantum mechanics and uses the *ket* symbol $|\varphi\rangle$ to denote the quantum state of the qubit system, and operators \hat{G} to denote quantum gates. This notation is purely physical and generally intuitive, and is useful for analysing algorithms that operate on many qubits.

The other standard formalism is based on the so-called *computational basis*, where qubit states are vectors and quantum gates are unitary matrices. This notation is very useful when we use quantum computation in physical applications, such as the description of many-body quantum systems. Indeed, it allows one to predict the results of a quantum simulation in a classical way by performing simple matrix multiplications. The main disadvantage of this notation is the exponential growth of vector and matrix dimensions, which makes classical computing resources insufficient to predict the evolution of a number of qubits just over ten. In fact, quantum operations are represented by potentially huge matrices in which the significant variables are often few. The second disadvantage of this notation is precisely that it hides some relevant quantum properties of the operations by increasing the apparent number of independent variables. As we shall see, this notation derives from the use of the Kronecker product \boxtimes , which is always denoted in literature (and also in all other chapters of this work) by the symbol of the tensor product \otimes — a instance of notational abuse.

The aim of this chapter is to develop a new explicit *tensorial notation*, which, to my knowledge, has never been used in any work developed by physicists related to quantum computing, but which could be more intuitive for mathematicians. This notation describes the qubit states as tensors and the quantum gate as multilinear (or quasi-multilinear) maps, which has the main advantage of preserving the internal structure of the qubit states and the operations on them. The second advantage is that it minimises the computational effort, allowing any quantum transformation to be performed classically by computing only 2×2 matrix multiplications.

After a summary of the mathematical background necessary to understand the concept of tensors in the field of quantum computing (Sec. 4.1) and a detailed description of the notations commonly used in quantum computing (Sec. 4.2), this chapter introduces an explicit tensorial notation to describe the state of multi-qubit systems and the quantum gates operating on them (Sec. 4.3). Some advantages of the tensorial notation of states with respect to the detection of separable and entangled states and with respect to the measurement procedure are shown. The special cases of local and control gates are analysed, showing the disadvantages of their description in the standard notation. Finally, the notation is applied to the description of the

remarkable quantum teleportation algorithm (Sec. 4.4). The chapter ends by pointing out a possible connection between the rank of quantum gates expressed in tensorial notation and the complexity of the quantum circuit implementing them (Sec. 4.5). This chapter, far from being conclusive, is only a starting point for a new powerful and potentially useful way of describing quantum computing, for understanding the physical transformation that happens along a quantum algorithm, and could lead us to new developments in quantum gate decomposition methods.

4.1 Mathematical background

In this section we summarise the basic mathematical concept needed to introduce the concept of tensors. For more details we refer the readers to any common text of linear algebra (see for example Ref. [36]).

We define a vector space V on a field $F \in \{\mathbb{R}, \mathbb{C}\}$ as the set of elements equipped with two internal operations: the addition $+: V \times V \rightarrow V$ and the scalar multiplication $\cdot: F \times V \rightarrow V$. Any linear combination of elements in V remains in V , namely $\alpha v + \beta w \in V$ for all $v, w \in V$ and $\alpha, \beta \in F$. Given a vector space, we define the *dual space* V^* as the space of maps $v^*: V \rightarrow F$. If we define the basis $\mathcal{B} = \{e_i\}$ of V and the dual basis $\mathcal{B}^* = \{e_j^*\}$ of V^* , their elements satisfy the property $e_i^*(e_j) = \delta_{ij}$. Note also that $\dim(V) = \dim(V^*)$. Given two vector spaces V and W , on a field F , with dimensions $\dim(V) = n$ and $\dim(W) = m$, we define a *linear map* the application $f: V \rightarrow W$. There is an isomorphism between linear maps and the space of $m \times n$ matrices, namely $\text{Hom}(V, W) \cong M_{m \times n}$, which means that we can represent any linear map by a matrix. Given three vector spaces V, W, K on a field F we define a *bilinear map* the application $\phi: V \times W \rightarrow K$ and the same definition can be extended to *multilinear maps* as $\phi: V_1 \times \dots \times V_d \rightarrow K$.

4.1.1 Tensor product space

The tensor product [37] is a map

$$\otimes: V \times W \rightarrow V \otimes W \quad (4.1.1)$$

that maps two elements of two vector spaces to another element of the tensor product space, namely

$$\otimes: (v, w) \mapsto v \otimes w \in V \otimes W, \quad (4.1.2)$$

for all $v \in V$ and $w \in W$. The space $V \otimes W$ is uniquely defined up to isomorphisms and its dimension is the product of the dimensions, namely $\dim(V \otimes W) = \dim(V) \cdot \dim(W) = n \cdot m$. The elements of the tensor product space of the form $v \otimes w \in V \otimes W$ are called *simple tensors* or *pure tensors*. The space of simple tensors is the image of the tensor product map, that is

$$V \otimes W = \text{span}(v \otimes w | v \in V, w \in W). \quad (4.1.3)$$

The basis of the tensor product space is the tensor product of the elements of the two bases, namely, given $\mathcal{B}_V = \{e_i\}$ the basis of V and $\mathcal{B}_W = \{f_j\}$ the basis of W , $\mathcal{B}_{V \otimes W} = \{e_i \otimes f_j\}$ is the natural basis of $V \otimes W$. A general tensor is a linear combination of simple tensors, i.e.

$$A = \sum_i \alpha_i v_i \otimes w_i \in V \otimes W, \quad (4.1.4)$$

for $v_i \in V$, $w_i \in W$ and $\alpha_i \in F$ and can be represented as a two-dimensional array, namely a $n \times m$ matrix, where $n = \dim(V)$ and $m = \dim(W)$. To express a tensor product of more than

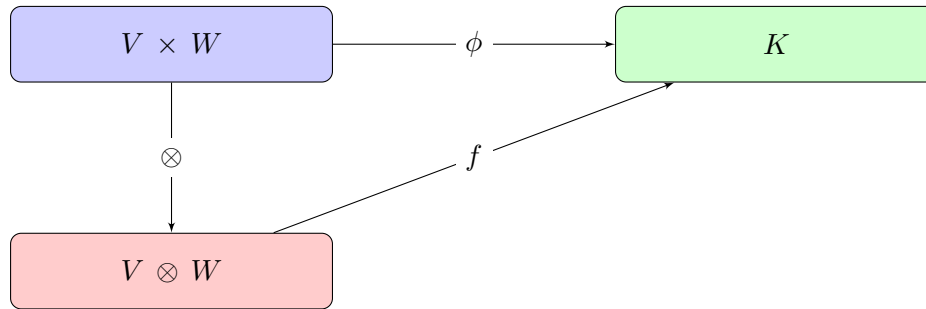


Figure 4.1: Sketch of the definition 4.1.1.

two elements as

$$A = \sum_i \alpha_i v_i^1 \otimes \cdots \otimes v_i^d, \quad (4.1.5)$$

one can use hypermatrices, which are the coordinate representation of tensors. The number d of spaces connected by the tensor product is called *order* of the tensor and will be denoted also as n in the rest of the chapter.

Property 4.1.1 (Associative property) *The tensor product is associative, namely $(V \otimes W) \otimes K = V \otimes (W \otimes K)$.*

Definition 4.1.1 (Bilinear maps and tensor algebra) *Given a bilinear map $\phi : V \times W \rightarrow K$, there exists a unique linear map $f : V \otimes W \rightarrow K$ such that*

$$\phi(v, w) = f \circ \otimes(v, w), \quad (4.1.6)$$

where $v \in V$, $w \in W$ and $\phi(v, w) \in K$.

This means that each bilinear map is a linear map after the correct embedding via the tensor product. This property can be extended to the multilinear maps $\phi : V_1 \times V_2 \times \cdots \times V_d \rightarrow K$ that can be represented by a linear map acting on the tensor product space, namely $f : V_1 \otimes V_2 \otimes \cdots \otimes V_d \rightarrow K$. The property is represented by the scheme in Figure 4.1.

4.1.2 Operations on tensors

A tensor A is transformed into another tensor B through the application of tensor products of linear maps (see Ref. [38] for more details), namely

$$B = M_1 \otimes \cdots \otimes M_d(A), \quad (4.1.7)$$

where M_k are linear maps and d is the order of tensors A and B , or equivalently under a multilinear map

$$B = (M_1, \dots, M_d) \cdot A. \quad (4.1.8)$$

This is related to the Property 4.1.1 that allows us to express the multilinear map (M_1, \dots, M_d) as a linear map $M_1 \otimes \cdots \otimes M_d$. Writing the tensor as a linear combination of pure tensors we have

$$\begin{aligned} B &= (M_1, \dots, M_d) \cdot \sum_i a_i^1 \otimes \cdots \otimes a_i^d \\ &= \sum_i (M_1 a_i^1) \otimes \cdots \otimes (M_d a_i^d) \\ &= \sum_i b_i^1 \otimes \cdots \otimes b_i^d. \end{aligned} \quad (4.1.9)$$

4.1.3 Natural maps defined by tensors

Given a tensor $A \in V_1 \otimes \dots \otimes V_d$ we define the following natural maps:

(i) **Multilinear map:**

$$V_1^* \otimes \dots \otimes V_d^* \ni (v_1, \dots, v_d) \mapsto (v_1, \dots, v_d)^T \cdot A \in F, \quad (4.1.10)$$

where T stands for *transpose*. This map corresponds to a bilinear map

$$V^* \otimes W^* \ni (v, w) \mapsto v^T A w \in F \quad (4.1.11)$$

if the tensor is of order $d = 2$.

(ii) **Sesquilinear map:**

$$V^* \otimes W \ni (v, w) \mapsto v^H A w \in F, \quad (4.1.12)$$

where H stands for *transpose conjugate*, also indicated as \dagger .

(iii) **Linear maps:**

$$V^* \ni v \mapsto A^T v \in W, \quad (4.1.13)$$

$$W^* \ni w \mapsto A w \in V. \quad (4.1.14)$$

4.1.4 Tensor rank decomposition

The tensor rank decomposition (TRD) consists of writing a tensor as the shortest linear combination of simple tensors:

$$A = \sum_{i=1}^r \alpha_i (v_i^1 \otimes \dots \otimes v_i^d), \quad (4.1.15)$$

where $r = \text{rank}(A)$, $\alpha_i \in F$ and $v_i^k \in V_k$. A simple tensor has $r = 1$ by definition.

Property 4.1.2 (Rank and multilinear maps) *The rank r of a tensor does not increase under multilinear multiplication, namely*

$$\text{rank}((M_1, \dots, M_d) \cdot A) \leq \text{rank}(A), \quad (4.1.16)$$

and remains the same if all matrices M_k have linearly independent columns.

4.2 Quantum computing in standard notation

4.2.1 Qubit states as vectors

The most common notation used in quantum computing is based on the Kronecker product and expresses the qubit states as exponentially large vectors. Given n qubits, whose states belong to the Hilbert space $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$, we use as basis a set of 2^n vectors of dimension 2^n , called the *computational basis*, which coincides with the canonical one

$$\mathcal{B} = \{\delta_i, \forall i \in \{0, \dots, 2^n - 1\}\}, \quad (4.2.1)$$

where δ_i is a vector with 0 entries in rows $j \neq i$ and 1 entry in row i . This basis comes from the isomorphism $(\mathbb{C}^2)^{\otimes n} \cong \mathbb{C}^{2^n}$ and the basis elements can be found by performing the Kronecker

product between n single qubit basis states. For example, the four basis states of a system of $n = 2$ qubits are

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad (4.2.2)$$

where we used the Kronecker product definition, for example

$$|01\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \boxtimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 0 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}. \quad (4.2.3)$$

The generalisation to n qubits is straightforward and the Kronecker product between two tensors increases the dimensions by keeping the order of the tensor constant. For this reason, we can think of an n qubit state as a tensor of order 1 and dimension 2^n , i.e. a vector.

4.2.2 Quantum gates as matrices

If we use vector notation to denote qubit states, we can act on them using matrix multiplication. In particular, a single-qubit gate $G_i \in \mathcal{U}(2)$ is a 2×2 unitary matrix which is the representation of a linear map $G_i : \mathbb{C}^2 \rightarrow \mathbb{C}^2$. This follows from the property $\text{Hom}(\mathbb{C}^2, \mathbb{C}^2) \cong M_{2 \times 2}$. In this sense, single-qubit gates belong to the space $\mathbb{C}^2 \otimes \mathbb{C}^2$. For the two-qubit case, quantum gates $G_{ij} \in \mathcal{U}(4)$ are 4×4 unitary matrices, and a general operation on n qubits can be expressed as a $2^n \times 2^n$ unitary. Again, this notation comes from the Kronecker product definition. For example, consider two single-qubit gates $G_1, G_2 \in \mathcal{U}(2)$ expressed as 2×2 matrices acting on two different qubits. The global two-qubit operation can be seen as a 4×4 matrix, resulting from the Kronecker product of the two, namely

$$G_1 \boxtimes G_2 = \begin{pmatrix} a_0 & a_1 \\ a_2 & a_3 \end{pmatrix} \boxtimes \begin{pmatrix} b_0 & b_1 \\ b_2 & b_3 \end{pmatrix} = \begin{pmatrix} a_0 \begin{pmatrix} b_0 & b_1 \\ b_2 & b_3 \end{pmatrix} & a_1 \begin{pmatrix} b_0 & b_1 \\ b_2 & b_3 \end{pmatrix} \\ a_2 \begin{pmatrix} b_0 & b_1 \\ b_2 & b_3 \end{pmatrix} & a_3 \begin{pmatrix} b_0 & b_1 \\ b_2 & b_3 \end{pmatrix} \end{pmatrix}. \quad (4.2.4)$$

So we can think of a two-qubit gate as an operation belonging to $(\mathbb{C}^2 \otimes \mathbb{C}^2) \boxtimes (\mathbb{C}^2 \otimes \mathbb{C}^2)$ and each two-qubit gate can be expressed as a sum of single Kronecker products as

$$G_{12} = \sum_{i=1}^r G_1^i \boxtimes G_2^i \in \mathcal{U}(4). \quad (4.2.5)$$

The Kronecker-based notation can be useful in several situations, such as when we classically predict the evolution of a qubit state during the Hamiltonian simulation by performing the multiplication between the matrix representing a gate and the vector representing a state. However, this is only possible for a small number of qubits because the dimension of the vectors (qubit states) and matrices (quantum gates) grows exponentially with the number n of qubits involved. A disadvantage of this notation is precisely the exponential growth of the computational resources required to perform this matrix multiplication. Another disadvantage is that the Kronecker product increases the non-trivial coefficients in the matrix and hides the local property of quantum gates, as explained in the following section.

4.2.3 Local gate detection

Given a random gate on n qubits, it is generally not trivial to determine whether the gate is local or entangled by looking at the matrix notation which has $2^n \times 2^n$ elements. On the other hand, we know that a tensor is separable if and only if it is a rank-1 tensor. At this point it is necessary to pay attention to the space in which the rank is computed and to what is meant by the separability of a tensor in relation to the separability of a gate. For example, consider a local two-qubit gate $G_L = G_1 \boxtimes G_2 \in \mathcal{U}(4)$ like the one in Eq. (4.2.4), which gives the matrix

$$G_L = G_1 \boxtimes G_2 = \begin{pmatrix} a_0 & a_1 \\ a_2 & a_3 \end{pmatrix} \boxtimes \begin{pmatrix} b_0 & b_1 \\ b_2 & b_3 \end{pmatrix} = \begin{pmatrix} a_0b_0 & a_0b_1 & a_1b_0 & a_1b_1 \\ a_0b_2 & a_0b_3 & a_1b_2 & a_1b_3 \\ a_2b_0 & a_2b_1 & a_3b_0 & a_3b_1 \\ a_2b_2 & a_2b_3 & a_3b_2 & a_3b_3 \end{pmatrix}. \quad (4.2.6)$$

If we calculate the rank we get $\text{rank}(G_L) = 4$ and the same applies to the identity gate \mathbb{I}_4 . Calculating the rank of a local quantum gate acting on n qubits in this notation would always give the result $\text{rank}(G_L) = 2^n$. The reason is that we are not using the correct space on which to define the separability of a multi-qubit gate. Going back to the two-qubit gate definition, it can be useful to define the tensor space of gates as $(\mathbb{C}^2 \otimes \mathbb{C}^2) \otimes (\mathbb{C}^2 \otimes \mathbb{C}^2) = V_1 \otimes V_2$ where $V_1 = \mathbb{C}^2 \otimes \mathbb{C}^2$ is the space of gates on the first qubit and $V_2 = \mathbb{C}^2 \otimes \mathbb{C}^2$ on the second. The locality property of a gate is the property of acting independently on qubits, so the separability of a two-qubit gate must be defined in the space $V_1 \otimes V_2$. Elements of $\mathbb{C}^2 \otimes \mathbb{C}^2$ are 2×2 matrices, but elements of V_1 are arrays of dimension 4 due to the isomorphism $\mathbb{C}^2 \otimes \mathbb{C}^2 \cong \mathbb{C}^4$. A tensor product of two 4-dimensional vectors $G_1 \in V_1$ and $G_2 \in V_2$ is a tensor of order 2, namely a 4×4 matrix:

$$G_L = G_1 \otimes G_2 = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} (b_0 \quad b_1 \quad b_2 \quad b_3) = \begin{pmatrix} a_0b_0 & a_0b_1 & a_0b_2 & a_0b_3 \\ a_1b_0 & a_1b_1 & a_1b_2 & a_1b_3 \\ a_2b_0 & a_2b_1 & a_2b_2 & a_2b_3 \\ a_3b_0 & a_3b_1 & a_3b_2 & a_3b_3 \end{pmatrix}, \quad (4.2.7)$$

whose columns are all linearly dependent. Using this notation, the separable gate satisfies the condition $\text{rank}(G_L) = 1$. The map between the standard notation and the one we should use to detect the separability is:

$$G_1 \boxtimes G_2 = \begin{pmatrix} a_0b_0 & a_0b_1 & a_1b_0 & a_1b_1 \\ a_0b_2 & a_0b_3 & a_1b_2 & a_1b_3 \\ a_2b_0 & a_2b_1 & a_3b_0 & a_3b_1 \\ a_2b_2 & a_2b_3 & a_3b_2 & a_3b_3 \end{pmatrix} \mapsto \begin{pmatrix} a_0b_0 & a_0b_1 & a_0b_2 & a_0b_3 \\ a_1b_0 & a_1b_1 & a_1b_2 & a_1b_3 \\ a_2b_0 & a_2b_1 & a_2b_2 & a_2b_3 \\ a_3b_0 & a_3b_1 & a_3b_2 & a_3b_3 \end{pmatrix} = G_1 \otimes G_2. \quad (4.2.8)$$

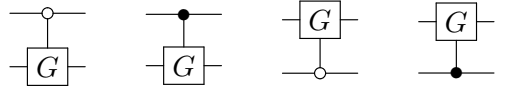
4.2.4 Controlled gates

A quantum controlled gate is a non-local operation that acts on the target qubits depending on the state of the control qubits. In the case of two-qubit operations acting on $|q_1\rangle \otimes |q_2\rangle$, there are four possible controlled gates, denoted by $\Lambda_1(G)$ according to the notation introduced in Chapter 2. The four types are $C_c G$ which act on $|q_2\rangle$ with G if the condition $|q_1\rangle = |c\rangle$ is satisfied, and $G C_c$ which act on $|q_1\rangle$ if $|q_2\rangle = |c\rangle$. The matrix notation results from the sum of

two Kronecker products as follows:

$$\begin{aligned}
C_0G &= |0\rangle\langle 0| \boxtimes G + |1\rangle\langle 1| \boxtimes \mathbb{I} \\
C_1G &= |0\rangle\langle 0| \boxtimes \mathbb{I} + |1\rangle\langle 1| \boxtimes G \\
GC_0 &= G \boxtimes |0\rangle\langle 0| + \mathbb{I} \boxtimes |1\rangle\langle 1| \\
GC_1 &= \mathbb{I} \boxtimes |0\rangle\langle 0| + G \boxtimes |1\rangle\langle 1|.
\end{aligned} \tag{4.2.9}$$

The corresponding quantum circuit representations are



$$\tag{4.2.10}$$

For example, the controlled gate C_1G acts with G on the second qubit if the first one is in the state $|1\rangle$ and with the identity if it is in $|0\rangle$. Writing the matrix explicitly, we have

$$\begin{aligned}
C_1G &= |0\rangle\langle 0| \boxtimes \mathbb{I} + |1\rangle\langle 1| \boxtimes G \\
&= \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \boxtimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \boxtimes \begin{pmatrix} a_0 & a_1 \\ a_2 & a_3 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a_0 & a_1 \\ 0 & 0 & a_2 & a_3 \end{pmatrix}.
\end{aligned} \tag{4.2.11}$$

Due to the fact that a controlled gate can be written as the sum of two simple Kronecker products, we conclude that a controlled two-qubit gate can also be represented by a rank-2 tensor if we replace \boxtimes by \otimes .

4.3 Quantum computing in tensor notation

We usually identify a multi-qubit state using Dirac notation and taking unit vectors $|\varphi\rangle$ in the Hilbert space $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$, where n is the number of qubits in the system. The dimension of the space is $\dim(\mathcal{H}) = 2^n$ and the natural basis is given by the tensor product of the single-qubit basis, namely

$$\mathcal{B} = \{|q_0\rangle \otimes \dots \otimes |q_{n-1}\rangle, \forall q_i \in \{0, 1\}\}. \tag{4.3.1}$$

Each basis element can be identified by $|i\rangle$ where $i \in \{0, \dots, 2^n - 1\}$ corresponds to the decimal notation of the binary number $(q_{n-1} \dots q_0)$ in reverse order, namely

$$i = q_0 + q_1 2^1 + \dots + q_{n-1} 2^{n-1}, \tag{4.3.2}$$

as defined in Eq. (1.1.8) of the Introduction 1. On this basis, a multi-qubit state is a normalised linear combination

$$|\varphi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle \text{ where } \alpha_i \in \mathbb{C} \text{ and } \sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1. \tag{4.3.3}$$

Note that according to Schmidt's theorem there exists an orthonormal basis $\{|\phi_k\rangle_1 \otimes \dots \otimes |\phi_k\rangle_n\}$ such that we can write the state as a linear combination of the minimum possible number of terms, namely

$$|\varphi\rangle = \sum_{k=1}^s \beta_k |\phi_k\rangle \otimes \dots \otimes |\phi_k\rangle \text{ where } \beta_k \in \mathbb{C} \text{ and } \sum_{k=1}^s |\beta_k|^2 = 1. \tag{4.3.4}$$

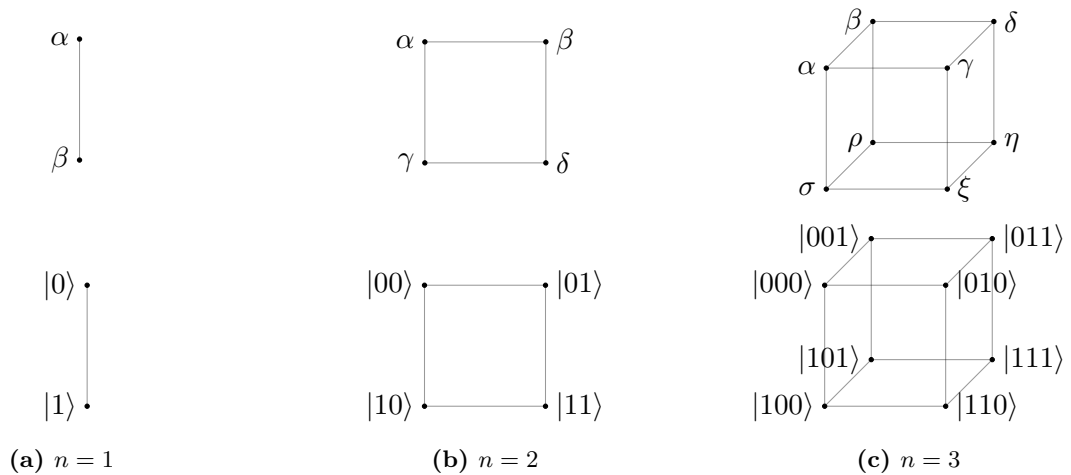


Figure 4.2: Representation of $n = 1, 2, 3$ qubits in the tensor product space $(\mathbb{C}^2)^{\otimes n}$ using an explicit tensor notation.

s is called Schmidt number and we have that $s = 1$ if and only if $|\varphi\rangle$ is a pure (separable) state and $s \geq 2$ implies that $|\varphi\rangle$ is entangled. So far we have two standard notations for representing a multi-qubit state: the Dirac and the vector notation. We now propose a third notation that exploits the explicit tensor formalism.

4.3.1 Qubit states as tensors

In this section we define an explicit tensor notation that can be used to represent multi-qubit states. Let's start by considering a system with $n = 1$. Single-qubit states on the computational basis are vectors of two elements $|\varphi\rangle \in \mathbb{C}^2$ as defined in Eq. (1.1.5) in Chapter 1. Since we are not doing a Kronecker product, the state is the same in vector and tensor notation, and we can interpret the vector as a rank-1 tensor. Using a pictorial representation, a general single-qubit state is a line with two vertices as shown in Fig. 4.2a, where the top vertex corresponds to $|0\rangle$ and the bottom vertex corresponds to $|1\rangle$. Increasing to $n = 2$, a two-qubit state belongs to the space $\mathcal{H} = \mathbb{C}^2 \otimes \mathbb{C}^2$, but in standard notation we use the homomorphism with \mathbb{C}^4 and therefore represent $|\varphi\rangle \in \mathbb{C}^4$ as a four-dimensional vector which is a linear combination of the four basis elements in Eq. (4.2.2) computed using the Kronecker product. If we replace the Kronecker product by the tensor product, we get instead an order-2 tensor $|\varphi\rangle \in \mathbb{C}^2 \otimes \mathbb{C}^2$, expressed as a linear combination of the basis elements

$$|00\rangle = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad |01\rangle = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad |10\rangle = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \quad |11\rangle = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \quad (4.3.5)$$

derived from the tensor product between two elements in \mathbb{C}^2 . For example

$$|01\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}. \quad (4.3.6)$$

Using a pictorial representation, a general two-qubit state can be represented as the square in Fig. 4.2b where we have added, to the vertical direction corresponding to the first qubit $|q_1\rangle$, an additional dimension (the horizontal one) representing the qubit $|q_2\rangle$. The left vertex corresponds

to $|q_2\rangle = |0\rangle$ and the right one to $|q_2\rangle = |1\rangle$. The map between the two representations is

$$\mathbb{C}^2 \boxtimes \mathbb{C}^2 \ni \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{pmatrix} \mapsto \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \in \mathbb{C}^2 \otimes \mathbb{C}^2. \quad (4.3.7)$$

Increasing the number of qubits to $n = 3$, a state $|\varphi\rangle \in \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2$ is represented on the computational basis as a vector $|\varphi\rangle \in \mathbb{C}^8$ with eight elements by using the Kronecker product. Replacing the Kronecker with the tensor product definition we get an order 3 tensor, i.e. a cube, as shown in Fig. 4.2c. The generalisation to an n qubit system is straightforward and gives an order- n tensor instead of a 2^n dimensional vector in the standard formalism.

4.3.2 Separable and entangled states

The explicit tensorial notation can be useful to identify separable states in a simpler way because it makes it clear which qubit of the system is in a particular state. For example, consider two separable three-qubit states $|\varphi\rangle = |\phi\rangle \otimes |0\rangle \otimes |0\rangle$ and $|\varphi'\rangle = |0\rangle \otimes |\phi\rangle \otimes |0\rangle$ (where $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$), which in vector notation are

$$|\varphi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \boxtimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \boxtimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \alpha \\ 0 \\ 0 \\ \beta \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |\varphi'\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \boxtimes \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \boxtimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \alpha \\ 0 \\ \beta \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \quad (4.3.8)$$

The corresponding order-3 tensors are instead

$$|\varphi\rangle = \begin{array}{c} \begin{array}{ccc} & 0 & \\ \alpha & \cdot & \\ & 0 & \\ \beta & \cdot & \\ & 0 & \end{array} \\ \begin{array}{ccc} & 0 & \\ & 0 & \\ & 0 & \\ & 0 & \end{array} \end{array}, \quad |\varphi'\rangle = \begin{array}{c} \begin{array}{ccc} & 0 & \\ \alpha & \cdot & \\ & \beta & \\ & 0 & \\ & 0 & \end{array} \\ \begin{array}{ccc} & 0 & \\ & 0 & \\ & 0 & \\ & 0 & \end{array} \end{array}, \quad (4.3.9)$$

which highlight the fact that for $|\varphi\rangle$ only the first qubit is in a linear combination and the other two are in $|0\rangle$, and for $|\varphi'\rangle$ the second is in a linear combination and the others are in $|0\rangle$. This can be seen because the only non-trivial entries for $|\varphi\rangle$ are in the intersection edge between the front face ($|q_3\rangle = |0\rangle$) and the left face ($|q_2\rangle = |0\rangle$). The vertical dimension (first qubit) instead represents a linear combination of the two states $|0\rangle$ (top face) and $|1\rangle$ (bottom face). On the other hand, the state $|\varphi'\rangle$ has non-trivial entries only in the direction of the second qubit, which corresponds to the horizontal one, and at the upper ($|q_1\rangle = |0\rangle$) front ($|q_3\rangle = |0\rangle$) edge.

This notation is particularly intuitive and useful for distinguishing between separable and entangled states. For example, consider the three-qubit entangled state, also called the GWZ

state, written in Dirac and vector notation as

$$|\text{GWZ}\rangle = \frac{|000\rangle + |111\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \quad (4.3.10)$$

The corresponding order-3 tensor is

$$|\text{GWZ}\rangle = \begin{array}{c} \begin{array}{ccc} 0 & & 0 \\ \frac{1}{\sqrt{2}} & & 0 \\ 0 & & 0 \end{array} \\ \begin{array}{ccc} 0 & & 0 \\ 0 & & \frac{1}{\sqrt{2}} \\ 0 & & 0 \end{array} \end{array}, \quad (4.3.11)$$

which has only two non-trivial elements, like the tensors in Eq. (4.3.9), but they belong to opposite vertices. The Figure 4.3 shows some examples of separable, partially entangled and fully entangled states, where non-trivial entries are represented by bold coloured dots. The first panel (Fig. 4.3a) highlights the fact that separable states are represented by cubes with non-trivial entries that can be connected by a line along an edge. On the other hand, the cubes representing the tensor of partially entangled states, such as the one in second panel (Fig. 4.3b), have at least two non-trivial vertices that must be connected along a diagonal. The surface to which the diagonal belongs defines the dimension along which the state is entangled. Each partially entangled state belongs to a plane which defines the separable qubit and its state. Finally, for tensors of fully entangled three-qubit states in third panel (Fig. 4.3c), we must cross each dimension along a diagonal to connect all non-trivial vertices. We conclude that the tensor state representation can provide a useful intuitiveness for the detection of separable and entangled three-qubit states. The number of non-trivial entries obviously corresponds to the number of basis elements in the linear combination, but the way in which we can connect them can be related to the degree of entanglement of the state.

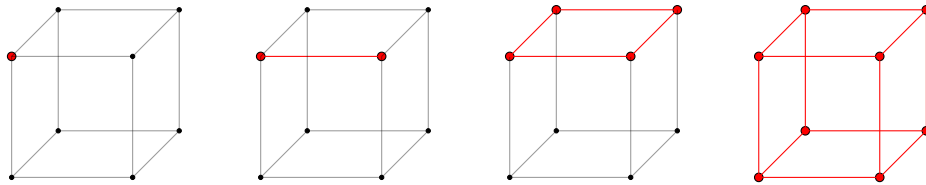
4.3.3 Quantum local gates as multilinear maps

In this section we define the action of separable quantum gates on multi-qubit states expressed in the tensor notation. The Fig. 4.4 shows a sketch about the action of quantum gates on tensor states of $n = 1, 2, 3$ qubits. Considering the case $n = 1$, single-qubit states are two-dimensional vectors and, in the common vector notation, operations on them are 2×2 unitary matrices $G \in \mathcal{U}(2)$. Since the state is the same using the computational basis or the tensor basis, then also the action of gates is the same and we can define single-qubit quantum gates as linear maps

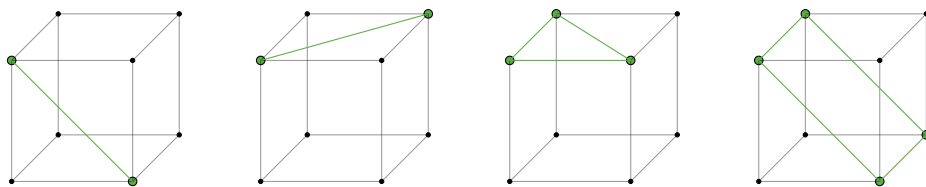
$$\mathcal{U}(2) \ni G : \mathbb{C}^2 \longrightarrow \mathbb{C}^2 \quad (4.3.12)$$

that can be represented by 2×2 unitary matrices according to the isomorphism $\text{Hom}(\mathbb{C}^2, \mathbb{C}^2) \cong M_{2 \times 2}$. The action of a single-qubit gate $G_1 \in \mathcal{U}(2)$ is therefore a 2×2 matrix multiplication

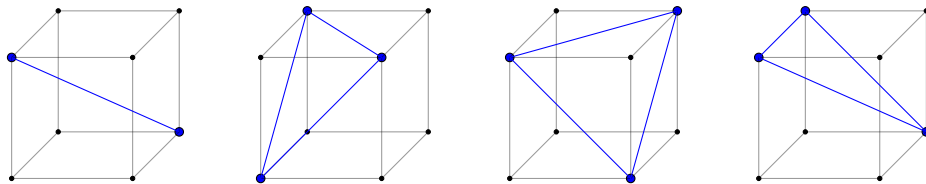
$$G_1 |\varphi\rangle = \begin{pmatrix} a_0 & a_1 \\ a_2 & a_3 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}. \quad (4.3.13)$$



(a) Tensor representation of the states $|0\rangle \otimes |0\rangle \otimes |0\rangle$, $|0\rangle \otimes |+\rangle \otimes |0\rangle$, $|0\rangle \otimes |+\rangle \otimes |+\rangle$ and $|+\rangle \otimes |+\rangle \otimes |+\rangle$ from left to right.



(b) Tensor representation of states $\frac{|00\rangle+|11\rangle}{\sqrt{2}} \otimes |0\rangle$, $|0\rangle \otimes \frac{|00\rangle+|11\rangle}{\sqrt{2}}$, $|0\rangle \otimes \frac{|00\rangle+|01\rangle+|11\rangle}{\sqrt{3}}$ and $\frac{|00\rangle+|11\rangle}{\sqrt{2}} \otimes |+\rangle$ from left to right.



(c) Tensor representation of states $\frac{|000\rangle+|111\rangle}{\sqrt{2}}$, $\frac{|001\rangle+|010\rangle+|100\rangle}{\sqrt{3}}$, $\frac{|000\rangle+|011\rangle+|110\rangle}{\sqrt{3}}$ and $\frac{|000\rangle+|001\rangle+|111\rangle}{\sqrt{3}}$ from left to right.

Figure 4.3: Some examples of tensor representations of three-qubit states, showing separable states (panel (a)), partially entangled states (panel (b)) and fully entangled states (panel (c)). The figure shows the relationship between the dimensionality of the entanglement and the connecting lines between non-trivial vertices.

The Figure 4.4a pictorially represents this operation.

In the case of a two-qubit system, local operations on states are described in standard notation by a 4×4 unitary matrices $G_{ij} \in \mathcal{U}(4)$, which can be written as a single Kronecker product of two single-qubit gates, namely $G_L = G_1 \boxtimes G_2$, as defined in Eq. (4.2.6), where $G_i \in \mathcal{U}(2)$ act on the qubit i . This local gate acts as a linear map $\mathcal{U}(4) \ni G_L : \mathbb{C}^4 \rightarrow \mathbb{C}^4$. The Property 4.1.1 tells us that a linear map is equivalent to a bilinear map after we have chosen the correct embedding of the space by the tensor product and in this case the embedding corresponds to the Kronecker product. This means that if we want to express G_L as an operation on order-2 tensors, it can be defined as a bilinear map

$$G_L : \mathbb{C}^2 \otimes \mathbb{C}^2 \rightarrow \mathbb{C}^2 \otimes \mathbb{C}^2. \quad (4.3.14)$$

However, there is an important property to consider: a local operation transforms a quantum state while keeping the degree of entanglement constant. This means that G_L cannot create or destroy entanglement between the two qubits. The gate locality can be interpreted, in this explicit tensor formalism, as the Property 4.1.2 corresponding to the action of a multilinear map preserving the rank of the tensors. In the two-qubit case, a separable state is represented by an order-2 tensor of rank 1, and after the G_L transformation it remains a rank-1 tensor. This is the reason why only local gates can be represented by multilinear maps. The general case of non-local gates will be analysed in Section 4.3.5. One of the main advantages of the bilinear definition is that it preserves the separability property of the operation. The common Kronecker product definition uses a 4×4 unitary matrix with 16 elements. However, only 8 of these are relevant and independently defined. In fact, the matrix must respect the separability property. However, this property is not manifest and furthermore the inverse step (from the 4×4 matrix G_L to the tensor product $G_1 \otimes G_2$ of two 2×2 matrices) is not immediate at all. On the other hand, if we use an explicit tensor notation, the gate is given by a set of two linear maps

$$G_L = (G_1, G_2) = \left(\begin{pmatrix} a_0 & a_1 \\ a_2 & a_3 \end{pmatrix}, \begin{pmatrix} b_0 & b_1 \\ b_2 & b_3 \end{pmatrix} \right), \quad (4.3.15)$$

and therefore the separability structure is preserved. Using the definition of bilinear map this local gate acts as defined by the natural map in Eq. (4.1.11), namely

$$G_L |\varphi\rangle = (G_1, G_2) \cdot \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} = G_1 \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} G_2^T. \quad (4.3.16)$$

Equivalently, we can consider the most general two-qubit state, which is by definition a rank-2 tensor $|\varphi\rangle = v_1 \otimes v_2 + w_1 \otimes w_2$ for $v_i, w_i \in \mathbb{C}^2$ and define the transformation $G_L |\varphi\rangle$ as

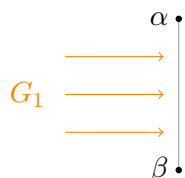
$$\begin{aligned} (G_1, G_2)(v_1 \otimes v_2 + w_1 \otimes w_2) &= G_1 v_1 \otimes G_2 v_2 + G_1 w_1 \otimes G_2 w_2 \\ &= G_1 v_1 (G_2 v_2)^T + G_1 w_1 (G_2 w_2)^T \\ &= G_1 (v_1 v_2^T + w_1 w_2^T) G_2^T \\ &= G_1 (v_1 \otimes v_2 + w_1 \otimes w_2) G_2^T. \end{aligned} \quad (4.3.17)$$

A pictorial representation of this action is represented in Figure 4.4b.

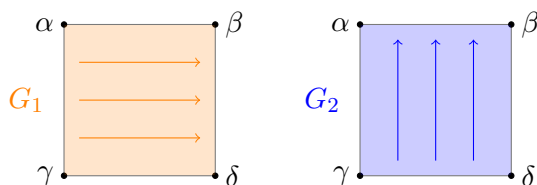
This definition can be extended to a three-qubit local gate, which in the standard notation is an 8×8 unitary matrix coming from the Kronecker product definition

$$G_L = \begin{pmatrix} a_0 & a_1 \\ a_2 & a_3 \end{pmatrix} \boxtimes \begin{pmatrix} b_0 & b_1 \\ b_2 & b_3 \end{pmatrix} \boxtimes \begin{pmatrix} c_0 & c_1 \\ c_2 & c_3 \end{pmatrix} \quad (4.3.18)$$

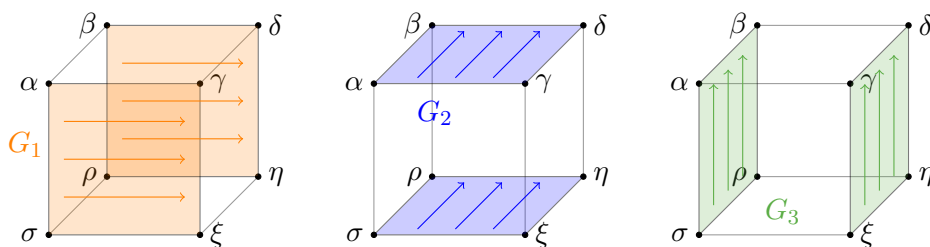
and containing 64 potentially non-trivial complex elements which is absolutely bigger than the number of free parameters after considering the unitary and separability constraints. In terms of



(a) Action of a single-qubit gate $G_1 \in \mathcal{U}(2)$ on the trivial system composed by $n = 1$ qubit.



(b) Action of the single-qubit gates $G_1, G_2 \in \mathcal{U}(2)$ on a system composed by $n = 2$ qubits. G_1 acts on the dimension of the first qubit $|q_1\rangle$, and G_2 acts vertically on the dimension of the second qubit $|q_2\rangle$.



(c) Action of single-qubit gates $G_1, G_2, G_3 \in \mathcal{U}(2)$ on a system composed by $n = 3$ qubits. Each G_i acts on the qubit $|q_i\rangle$ along the orthogonal direction of the corresponding dimension.

Figure 4.4: The figures highlight the dimension and the direction along which single-qubit gates act on states of $n = 1, 2, 3$ qubits expressed as tensors.

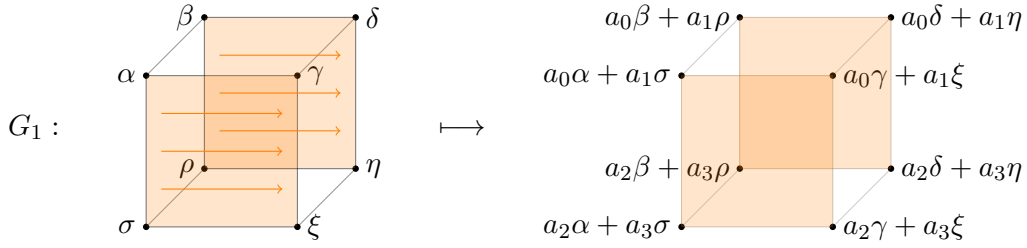


Figure 4.5: Action of a single qubit gate on the first qubit of a system composed by $n = 3$ qubits. The matrix representation of G_1 is defined in Eq. (4.3.21).

an operation on tensors, the same gate can be viewed as a single triplet expressing a multilinear map:

$$G_L : \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2 \longrightarrow \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2 \quad (4.3.19)$$

acting on an order-3 tensor and preserving its rank. Instead of a 8×8 unitary matrix we have now a triplet of 2×2 unitary matrices

$$G_L = (G_1, G_2, G_3) = \left(\begin{pmatrix} a_0 & a_1 \\ a_2 & a_3 \end{pmatrix}, \begin{pmatrix} b_0 & b_1 \\ b_2 & b_3 \end{pmatrix}, \begin{pmatrix} c_0 & c_1 \\ c_2 & c_3 \end{pmatrix} \right), \quad (4.3.20)$$

where each single-qubit gate $G_i \in \mathcal{U}(2)$ acts in the dimension corresponding to the qubit $|q_i\rangle$. To understand which kind of transformation is induced by a bilinear map on a order-3 tensor, consider a three-qubit gate acting non-trivially only on the first qubit, namely

$$G_L = (G_1, \mathbb{I}, \mathbb{I}) = \left(\begin{pmatrix} a_0 & a_1 \\ a_2 & a_3 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right). \quad (4.3.21)$$

The first qubit introduces the vertical dimension and is oriented downwards. This means that the gate G_1 is applied as a matrix multiplication in the orthogonal direction (from left to right), on both the front and back faces, and the tensor transformation is sketched in Fig. 4.5. In the same way, the second qubit introduces the horizontal dimension oriented from left to right and a gate G_2 applied on it, acts as a matrix multiplication in the orthogonal direction on both the bottom and top faces. Finally, the third qubit introduces the depth dimension and a gate G_3 acts as a matrix multiplication from the bottom to both the right and left faces. The Figure 4.4c summarizes them.

In addition to minimising the number of non-trivial elements in the gate representation, this explicitly tensorial formalism also minimises the number of matrix multiplications that must be performed to calculate the transformation of a state. To show it, consider the very famous gate $H \otimes H \otimes H$ (where H is the Hadamard gate as defined in Eq. (2.1.6) of Chapter 2) that generate the maximum superposition state, namely

$$H \otimes H \otimes H : |0\rangle \otimes |0\rangle \otimes |0\rangle \longmapsto |+\rangle \otimes |+\rangle \otimes |+\rangle = \frac{1}{\sqrt{8}} \sum_{i=0}^7 |i\rangle. \quad (4.3.22)$$

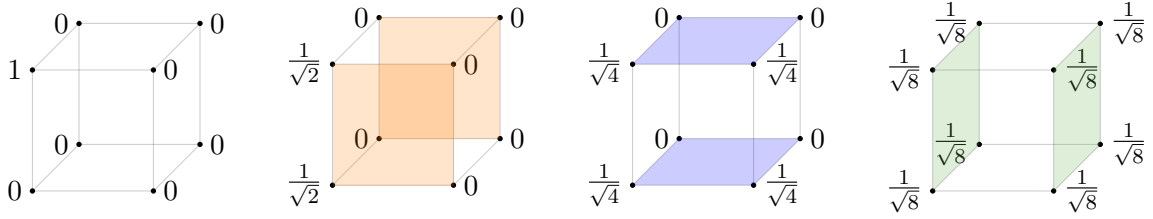


Figure 4.6: Action of the local gate $H \otimes H \otimes H$ and its explicit partial transformations of $(H, \mathbb{I}, \mathbb{I})$, $(\mathbb{I}, H, \mathbb{I})$ and $(\mathbb{I}, \mathbb{I}, H)$ respectively from left to right.

In the common matrix notation it would be

$$\begin{aligned}
 H \otimes H \otimes H &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \otimes \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \otimes \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\
 &= \frac{1}{\sqrt{8}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix}. \tag{4.3.23}
 \end{aligned}$$

Using the explicit tensor notation instead, we can represent each Hadamard gate as a 2×2 matrix multiplication on the correct dimension of the tensor. The transformation, making explicit the action on different qubits, is represented in Fig. 4.6. In order to apply a generic $G_L = G_1 \otimes G_2 \otimes G_3$ on a system with $n = 3$ qubits in vector notation, we have to perform a multiplication between an 8×8 matrix and a vector, i.e. it requires $8 \cdot 8 = 64$ multiplications and $7 \cdot 8 = 56$ sums. Calculating instead the action of the multilinear map $G_L = (G_1, G_2, G_3)$ requires $8 \cdot 6 = 48$ multiplications and $4 \cdot 6 = 24$ sums because for each qubit we need to act on two faces calculating two 2×2 multiplications and the number of faces is 6. Considering a generic number of qubit n , applying G_L in standard notation requires $2^n \cdot 2^n = 4^n$ multiplications and $(2^n - 1)2^n$ sums and using instead the tensor notation requires $8n$ multiplications and $4(2n)$ sums.

4.3.4 Controlled gates as quasi-multilinear maps

We have seen in Sec. 4.2.4 that controlled gates are rank-2 operations because we identify two orthogonal states of the control qubit that influence the action on the target qubit. For example, the CNOT case can be written as

$$\text{CNOT} = |0\rangle\langle 0| \otimes \mathbb{I} + |1\rangle\langle 1| \otimes X, \tag{4.3.24}$$

and acts on the two-qubit space $\mathcal{H} = \mathbb{C}^2 \otimes \mathbb{C}^2$ flipping the state of the second qubit if the first one is in $|0\rangle$. If we want to define its action on an order-2 tensor, we can use a sum of bilinear maps as follows:

$$\text{CNOT} = (M_0, \mathbb{I}) + (M_1, X) = \left(\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right) + \left(\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \right), \tag{4.3.25}$$

$$(4.3.28)$$

Figure 4.7: Action of the CNOT gate on the order-2 tensor representing a two-qubit state.

where we have defined $M_i := |i\rangle\langle i|$. We stress that the sum of bilinear maps is not the bilinear map of the sums and it is not even generally a bilinear map. If we now consider a general two-qubit state, the action is

$$\begin{aligned} \text{CNOT}|\varphi\rangle &= M_0 \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \mathbb{I}^T + M_1 \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} X^T \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}^T + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}^T \\ &= \begin{pmatrix} \alpha & \beta \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ \delta & \gamma \end{pmatrix} = \begin{pmatrix} \alpha & \beta \\ \delta & \gamma \end{pmatrix}. \end{aligned} \tag{4.3.26}$$

However, we can do better than this by noting that if the control condition is not satisfied, the action is trivial. Therefore, we can compute the action by performing the matrix multiplication only on the dimension corresponding to the target qubit and in the half of the tensor that satisfies the control condition. This decreased by a factor of 2 the number of operations. Considering an order-2 tensor representing a two-qubit state, to fix the condition $|q_1\rangle = |1\rangle$ means that we restrict the transformation to the lower part of the square. So the CNOT gate can be written as the sum of the bilinear maps, as in Eq. (4.3.25), or as a local operation X acting only on the half tensor that satisfies the condition. Let's represent it with two elements: the first one C_1 is the condition $|q_1\rangle = |1\rangle$ and the second one is the operation X acting on $|q_2\rangle$, namely

$$\text{CNOT} = (C_1, X) = \left(C_1, \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \right). \tag{4.3.27}$$

The action of the CNOT on the order-2 tensor is pictorially represented in Fig. 4.7. We can generalise the controlled two-qubit operation for any control condition C_i for $i \in \{0, 1\}$ and for any operation G on the target qubit. The four types of controlled gate actions C_1G , C_0G , GC_1 , GC_0 are represented in Fig. 4.8.

In the case of three qubits the approach is the same and the control condition fixes the part of the tensor on which we have to act. If we have a single control qubit, the control condition fixes a face. For example, the gate $\text{CNOT} \otimes \mathbb{I}$, represented by the quantum circuit

$$(4.3.30)$$

acts with the X gate on the second qubit only if the first one is in $|1\rangle$, i.e. we do the matrix multiplication only on the bottom face as shown in Fig. 4.9. The same operation, in standard matrix notation, would be represented by an 8×8 unitary matrix multiplication where the

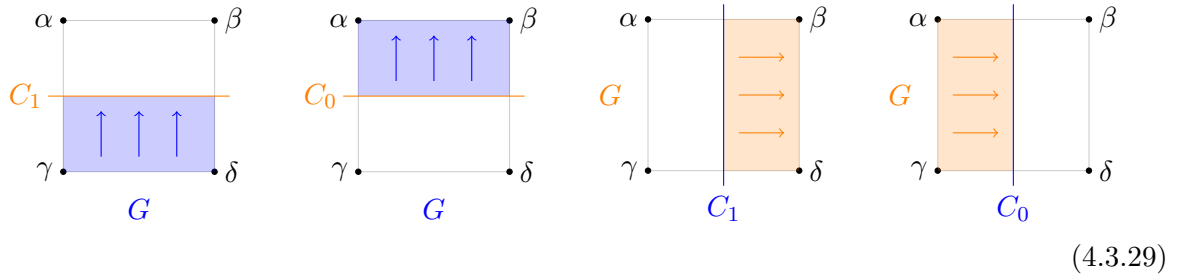


Figure 4.8: Action of control gates C_1G , C_0G , GC_1 , GC_0 , from left to right, on an order-2 tensor representing a two-qubit state.

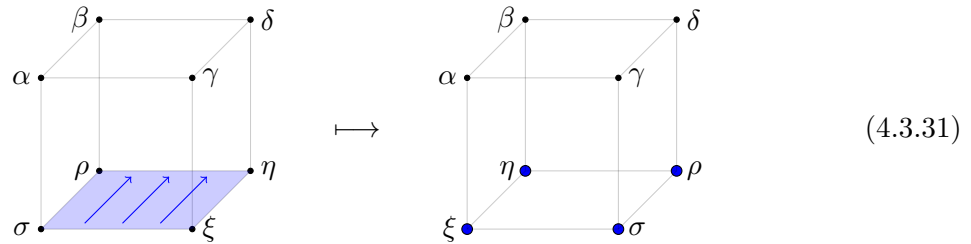


Figure 4.9: Action of $\text{CNOT} \otimes \mathbb{I}$ on a system composed by $n = 3$ qubits.

matrix can be obtained by performing the Kronecker products

$$\text{CNOT} \otimes \mathbb{I} = |0\rangle\langle 0| \otimes \mathbb{I} \otimes \mathbb{I} + |1\rangle\langle 1| \otimes X \otimes \mathbb{I}. \tag{4.3.32}$$

The multilinear representation is instead

$$\begin{aligned} G &= (M_0, \mathbb{I}, \mathbb{I}) + (M_1, X, \mathbb{I}) \\ &= \left(\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right) + \left(\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right) \end{aligned} \tag{4.3.33}$$

and to identify the operation on the cube as in Fig. 4.9 we have defined it as

$$G = (C_1, X, \mathbb{I}) = \left(C_1, \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right), \tag{4.3.34}$$

where C_1 identifies the control condition $|q_1\rangle = |1\rangle$.

Another case of controlled operations are the so-called fully controlled gates, also introduced in Section 2.3.2 of Chapter 2. They are defined as operations $\Lambda_{n-1}(G)$ acting on a single qubit with $G \in \mathcal{U}(2)$ depending on the state of all other $n - 1$ qubits. To fix the condition for $n - 1$ qubits means that we restrict the action of the controlled gate on a single edge of the order- n tensor. A very famous example of a fully controlled three-qubit gate is the Toffoli gate $\Lambda_2(X)$ corresponding to the quantum circuit



(4.3.35)

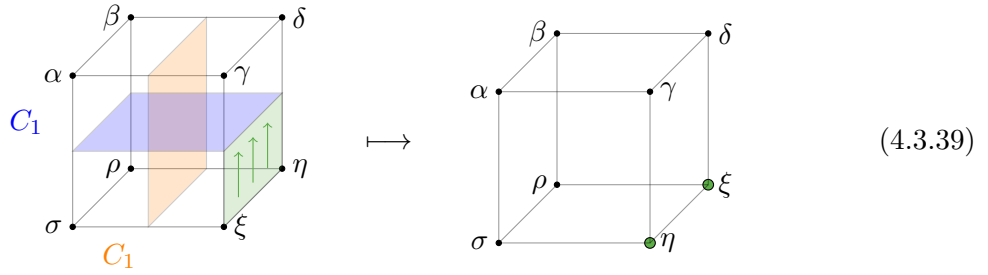


Figure 4.10: Action of the Toffoli gate on a system composed by $n = 3$ qubits. The X gate is applied on the third qubit but only on the edge fixed by the intersection of the two conditions that are $|q_1\rangle = |1\rangle$ (bottom face) and $|q_2\rangle = |1\rangle$ (right face).

and which acts on the third qubit if the first two satisfy $|q_1\rangle \otimes |q_2\rangle = |1\rangle \otimes |1\rangle$. In standard notation the operation would be represented as a rank-4 gate because we have four possible states for the first two qubits $|q_1\rangle$ and $|q_2\rangle$, namely

$$\Lambda_2(X) = |0\rangle\langle 0| \otimes |0\rangle\langle 0| \otimes \mathbb{I} + |0\rangle\langle 0| \otimes |1\rangle\langle 1| \otimes \mathbb{I} + |1\rangle\langle 1| \otimes |0\rangle\langle 0| \otimes \mathbb{I} + |1\rangle\langle 1| \otimes |1\rangle\langle 1| \otimes X, \quad (4.3.36)$$

that can be expressed as the following sum of multilinear maps

$$\Lambda_2(X) = (M_0, M_0, \mathbb{I}) + (M_0, M_1, \mathbb{I}) + (M_1, M_0, \mathbb{I}) + (M_1, M_1, X). \quad (4.3.37)$$

Writing it as a quasi-multilinear map, $\Lambda_2(X)$ becomes a single operation with two conditions:

$$\Lambda_2(X) = (C_1, C_1, X), \quad (4.3.38)$$

which acts on the order-3 tensor as in Fig. 4.10. The interesting thing is that the more the control conditions and the fewer multiplication we have to do.

4.3.5 Quantum non-local gates

As we have seen in Sec. 4.2.2, a quantum gate acting on two qubits is represented, on the computational basis, by a 4×4 unitary matrix $G_{ij} \in \mathcal{U}(4)$ thanks to the Kronecker product convention. This is equivalent to defining the operation as a linear map $G_{ij} : \mathbb{C}^4 \rightarrow \mathbb{C}^4$ which transforms four-dimensional vectors. Using explicit tensor notation instead, we want to define a two-qubit gate as an operation on the order-2 tensor representing a two-qubit state. Equivalent to what is defined in Eq. (4.2.5), we can always write a two-qubit gate as a sum of simple tensors

$$G_{12} = \sum_{i=1}^r G_1^i \otimes G_2^i \in \mathcal{U}(2) \otimes \mathcal{U}(2) \quad (4.3.40)$$

where $r \leq 4$ is the rank. This can be thought of as a linear map resulting from the embedding of the space by the Kronecker product, and coincides with a bilinear map if we use the Property 4.1.1. Therefore, we can define the two-qubit gate as the sum of linear maps

$$G_{12} = \sum_{i=1}^r (G_1^i, G_2^i), \quad (4.3.41)$$

which cannot be written in general as a single bilinear map, and acts on a two-qubit state as follows

$$G_{12} |\varphi\rangle = \sum_{i=1}^r G_1^i \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} (G_2^i)^T. \quad (4.3.42)$$

In the case of a three-qubit gate, it becomes a sum of triplets, and the generalisation to n -qubit gates is straightforward and gets

$$G_n = \sum_{i=1}^r (G_1^i, \dots, G_n^i). \quad (4.3.43)$$

If we define a general rank- s tensor representing an n -qubit system according the Schmidt theorem (see Eq. (4.3.4)), the gate G_n acts as

$$G_n |\varphi\rangle = \sum_{k=1}^s \sum_{i=1}^r \beta_k (G_1^i |\phi_k\rangle_1 \otimes \dots \otimes G_n^i |\phi_k\rangle_n). \quad (4.3.44)$$

We can compare the number of calculations needed to obtain the action of a general n -qubit gate G_n . Using the standard notation it corresponds to a matrix multiplication which requires $2^n \cdot 2^n = 4^n$ multiplications and $(2^n - 1)2^n$ sums. Using instead the explicit tensor notation we have to apply at most 2^n linear maps, eventually summing the results, with a total of $4(2n)$ multiplications and $4(2n) + 2^n$ sums where $2n$ is the number of faces and 2^n is the number of multilinear maps.

4.3.6 Measurement interpretation

The interpretation of the measurement of the qubit state is also more intuitive in this explicit tensorial notation. Let's consider a three-qubit state, represented by the cube in Figure 4.2c. The probability of measuring the state $|000\rangle$ is $|\alpha|^2$, and in general the absolute value squared of the complex value on each vertex, is the probability of measuring the corresponding state. Moreover, if we measure two qubits out of the three, the state of the unmeasured one is highlighted in the edge corresponding to the intersection of the two faces selected by the measurement of the other two qubits. This means that when we measure a single qubit we collapse the tensor of order n to a tensor of order $n - 1$. Once we have measured all the qubits in the system, we are left with a single basis element $|i\rangle$ and the probability of measure it at the beginning of the measurement procedure was $|k|^2$, where k was the original value at the corresponding vertex.

4.4 The quantum teleportation algorithm in tensor formalism

This algorithmic protocol facilitates the transfer of quantum states between distant particles, not through the transfer of physical matter, but by exploiting entanglement and classical communication. The quantum teleportation algorithm, first formulated by Bennett et al. in Ref. [39], exemplifies the profound departure from classical notions of information transfer, involving the delicate interplay of entanglement, superposition and measurement, which give quantum particles the capacity to exist in multiple states simultaneously and to establish non-local correlations. Quantum teleportation allows to transmit an unknown quantum state $|\phi\rangle$ from a sender, conventionally referred to as Alice, to a receiver, designated as Bob who share an intermediary entangled pair of particles at the beginning and can use only local operations and classical communication.

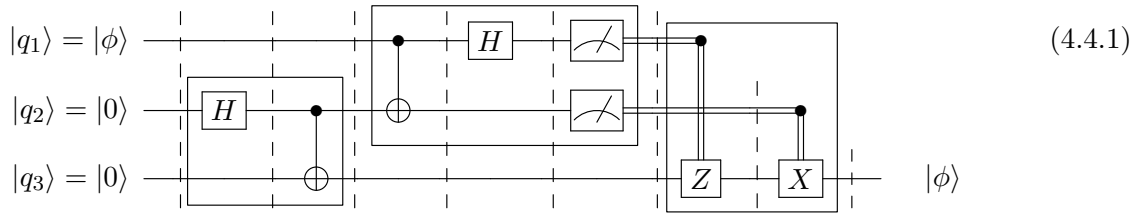


Figure 4.11: Quantum circuit implementing the teleportation algorithm. The first block represents the Bell’s state preparation, the second block is the local operation performed by Alice and finally the last block corresponds to the classical communication from Alice to Bob and the local operation performed by Bob.

The algorithm starts with the preparation of the entangled two-qubit state (first block in the quantum circuit of Fig. 4.11:

$$|\Phi^+\rangle = \frac{|0\rangle_A \otimes |0\rangle_B + |1\rangle_A \otimes |1\rangle_B}{\sqrt{2}}, \tag{4.4.2}$$

also called the Bell state. The two qubits can then be moved very far apart while maintaining the entanglement between them. Alice’s goal is to send another quantum state $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ to Bob, and she can perform operations and measurements on only her two qubits. Alice acts on her qubits using the quantum operations described by the second block in Fig. 4.11, eventually performing a joint measurement on both the sender’s particle and the intermediary particle. This measurement collapses the sender’s particle and the intermediary particle and instantly projects the receiver’s entangled particle (Bob qubit) into a particular state. Alice then classically transmits the measurement results, after which Bob performs a certain local operation that allows him to transform his qubit into $|\phi\rangle$ (last block in Fig. 4.11). Note that the quantum state is now in Bob’s qubit memory and Alice doesn’t have it anymore. This is consistent with the no cloning theorem [40].

4.4.1 Quantum state evolution

We now see the power of the explicit tensor notation in the study of the state evolution along the quantum teleportation algorithm described in Fig. 4.11. We label Alice’s qubits with $|q_1\rangle$ (the sender’s) and $|q_2\rangle$ (the intermediary’s) and Bob’s qubit with $|q_3\rangle$ (the receiver’s). The state evolution along the quantum circuit is described below in both Dirac and tensor notation, and we omit the vector formalism, which would consist of 8-dimensional vectors transformed by matrix multiplication and would require more computational effort. The order-3 tensor of the initial state is

$$|\varphi^{(0)}\rangle = \begin{matrix} & & 0 & & & & 0 \\ & & \cdot & & \cdot & & \\ & \alpha & \cdot & & \cdot & & 0 \\ & & & & & & \\ & & 0 & & \cdot & & 0 \\ & & \cdot & & \cdot & & \\ \beta & \cdot & & & \cdot & & \\ & & & & & & 0 \end{matrix}, \tag{4.4.3}$$

which has non-zero entries in the first qubit dimension and in the edge coming from the intersection between the face $|q_2\rangle = |0\rangle$ and the face $|q_3\rangle = |0\rangle$. In Dirac notation it’s $|\varphi^{(0)}\rangle = |\phi\rangle \otimes |0\rangle \otimes |0\rangle$ and has $\text{rank}(|\varphi^{(0)}\rangle) = 1$. The first operation is a local gate which acts with the Hadamard on the second qubit, and in terms of the multilinear map is $G^{(1)} = (\mathbb{I}, H, \mathbb{I})$. In terms

of an action on the tensor, it transforms the upper and lower faces, giving

$$|\varphi^{(1)}\rangle = \begin{array}{c} \begin{array}{ccc} 0 & & 0 \\ \frac{\alpha}{\sqrt{2}} & & \frac{\alpha}{\sqrt{2}} \\ \beta & & \beta \\ \frac{\beta}{\sqrt{2}} & & \frac{\beta}{\sqrt{2}} \end{array} \\ \cdot \end{array} \quad (4.4.4)$$

The operation $G^{(1)}$ is local, so the tensor preserves the rank, namely $\text{rank}(|\varphi^{(1)}\rangle) = 1$. In Dirac notation the state is $|\varphi^{(1)}\rangle = |\phi\rangle \otimes |+\rangle \otimes |0\rangle$. The first CNOT gate corresponds to the quasi-multilinear map $G^{(2)} = (\mathbb{I}, C_1, X)$, so we act with the X gate on the right face, which corresponds to the third qubit dimension and the condition $|q_2\rangle = |1\rangle$. We obtain:

$$|\varphi^{(2)}\rangle = \begin{array}{c} \begin{array}{ccc} 0 & & \frac{\alpha}{\sqrt{2}} \\ \frac{\alpha}{\sqrt{2}} & & 0 \\ 0 & & \beta \\ \frac{\beta}{\sqrt{2}} & & 0 \end{array} \\ \cdot \end{array} \quad (4.4.5)$$

The CNOT is a non-local gate that can change the rank and in this case we have transformed the rank-1 tensor into a rank-2 tensor. In Dirac notation the state is

$$|\varphi^{(2)}\rangle = |\phi\rangle \otimes \frac{|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle}{\sqrt{2}} = |\phi\rangle \otimes |\Phi^+\rangle. \quad (4.4.6)$$

The preparation block is concluded and now Alice transforms her qubits by first applying a CNOT which acts with X on the down face corresponding to the second qubit dimension and the condition $|q_1\rangle = |1\rangle$. The quasi-multilinear map is $G^{(3)} = (C_1, X, \mathbb{I})$ and we get

$$|\varphi^{(3)}\rangle = \begin{array}{c} \begin{array}{ccc} 0 & & \frac{\alpha}{\sqrt{2}} \\ \frac{\alpha}{\sqrt{2}} & & 0 \\ \frac{\beta}{\sqrt{2}} & & 0 \\ 0 & & \frac{\beta}{\sqrt{2}} \end{array} \\ \cdot \end{array} \quad (4.4.7)$$

that corresponds to a rank-4 tensor. In Dirac notation the state is

$$\begin{aligned} |\varphi^{(3)}\rangle &= \frac{\alpha}{\sqrt{2}} |0\rangle \otimes (|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle) + \frac{\beta}{\sqrt{2}} |1\rangle \otimes (|1\rangle \otimes |0\rangle + |0\rangle \otimes |1\rangle) \\ &= \frac{1}{\sqrt{2}} (\alpha |000\rangle + \alpha |011\rangle + \beta |110\rangle + \beta |101\rangle). \end{aligned} \quad (4.4.8)$$

We then apply the local gate H on the first qubit, which consists of the linear map $G^{(4)} = (H, \mathbb{I}, \mathbb{I})$, and get

$$|\varphi^{(4)}\rangle = \begin{array}{c} \begin{array}{ccc} \frac{\beta}{2} & & \frac{\alpha}{2} \\ \frac{\alpha}{2} & & \frac{\beta}{2} \\ -\frac{\beta}{2} & & \frac{\alpha}{2} \\ \frac{\alpha}{2} & & -\frac{\beta}{2} \end{array} \\ \cdot \end{array} \quad (4.4.9)$$

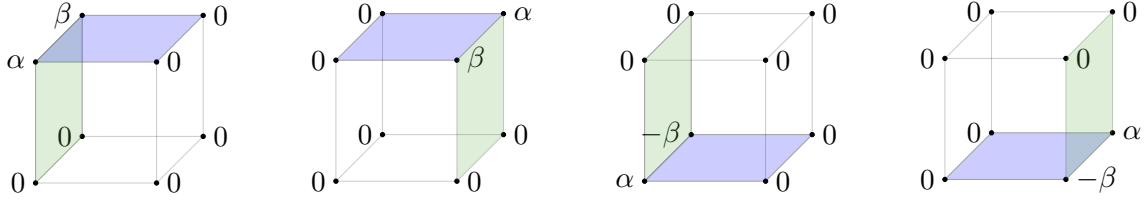


Figure 4.12: Tensor representation of the four possible post-measure states obtained by Alice at the end of the second block of the algorithm.

that corresponds to

$$\begin{aligned} |\varphi^{(4)}\rangle &= \frac{1}{\sqrt{2}} (\alpha |0\rangle \otimes |+\rangle \otimes |0\rangle + \alpha |0\rangle \otimes |-\rangle \otimes |1\rangle + \beta |1\rangle \otimes |-\rangle \otimes |0\rangle + \beta |1\rangle \otimes |+\rangle \otimes |1\rangle) \\ &= \frac{1}{2} (\alpha(|000\rangle + |010\rangle) + \alpha(|001\rangle - |011\rangle) + \beta(|100\rangle - |110\rangle) + \beta(|101\rangle + |111\rangle)), \end{aligned} \quad (4.4.10)$$

and is still a rank-4 tensor since $G^{(4)}$ is a local operation and a multilinear map.

4.4.2 Measurement and classical communication

Measuring two out of three qubits collapses the state to an edge corresponding to the intersection of the two results obtained. Let's focus on the final qubit state in Eq. (4.4.9). If we measure the first qubit in $|q_1\rangle = |0\rangle$ we collapse the system in the upper face and with $|q_2\rangle = |0\rangle$ we collapse the system in the left face. The two conditions together identify the upper edge on the left. This edge identifies the state for the third qubit, which in this case is $|q_3\rangle = |\phi\rangle = \alpha |0\rangle + \beta |1\rangle$. If we measure $|q_1\rangle = |0\rangle$ and $|q_2\rangle = |1\rangle$ we identify the upper edge on the right, which means $|q_3\rangle = \beta |0\rangle + \alpha |1\rangle$, so Bob has to apply X to recover $|\phi\rangle$. With $|q_1\rangle = |1\rangle$ we identify the lower face and with $|q_2\rangle = |0\rangle$ the left one, so the state is $|q_3\rangle = \alpha |0\rangle - \beta |1\rangle$ and Bob apply Z to recover $|\phi\rangle$ and finally measuring $|q_1\rangle = |q_2\rangle = |1\rangle$ the result is at the bottom right edge and the final state is $|q_3\rangle = -\beta |0\rangle + \alpha |1\rangle$ and using XZ Bob gets $|\phi\rangle$. The post-measurement states in the four cases are shown in Fig. 4.12.

This simple quantum algorithm demonstrates one of the potential features of an explicit tensor notation, which guarantees a more immediate visualisation of the state of the qubits and a smaller amount of computation required to calculate the transformations.

4.5 Circuit complexity and quantum gate rank

This section presents a tentative connection between the rank of the matrices representing a two-qubit gate (computed in the *correct* tensor space as discussed in Sec. 4.2.3) and the number of CNOTs needed to decompose it as a quantum circuit.

4.5.1 Rank and number of CNOTs

Some examples of common two-qubit circuit structures are reported here organised by rank, where white boxes represent a general single-qubit operation $\in \mathcal{U}(2)$. Note that in this section we denote a two-qubit gate as a tensor living in the space

$$G_{12} \in (\mathbb{C}^2 \boxtimes \mathbb{C}^2) \otimes (\mathbb{C}^2 \boxtimes \mathbb{C}^2) \cong \mathbb{C}^4 \otimes \mathbb{C}^4, \quad (4.5.1)$$

which means that it's represented by a 4×4 matrix with rank = 1 if it is a local operation. This notation is not the explicit tensorial notation proposed in the previous sections, where we use tensors to represent qubit states and multilinear or quasi-multilinear maps to represent quantum gates.

- (i) **Rank-1 circuits.** A rank-1 tensor is by definition a separable operation, and the most general one is $G_{12} = A \otimes B$ where $A, B \in \mathcal{U}(2)$ in the standard notation and $A, B \in \mathbb{C}^4$ in this section. The circuit of G_{12} is

$$\begin{array}{c} \square \\ \square \end{array} . \tag{4.5.2}$$

- (ii) **Rank-2 circuits.** Let's start with the simplest rank-2 quantum gates, namely the common CNOTs gate and it's generalization to C_1G :

$$\text{CNOT} = M_0 \otimes \mathbb{I} + M_1 \otimes X, \quad C_1G = M_0 \otimes \mathbb{I} + M_1 \otimes G. \tag{4.5.3}$$

Obviously we can add local operations before and after them without changing the rank (two quantum gates are called *locally equivalent* [41] if they differ only by local operations). The corresponding quantum circuits are

$$\begin{array}{c} \bullet \\ | \\ \oplus \end{array} , \quad \begin{array}{c} \square \\ | \\ \oplus \\ | \\ \square \end{array} , \quad \begin{array}{c} \bullet \\ | \\ \square \end{array} , \quad \begin{array}{c} \square \\ | \\ \bullet \\ | \\ \square \end{array} . \tag{4.5.4}$$

Using the approach presented in Sec. 2.3.2 of Chapter 2, and in particular the result in Eq. (2.3.20), we see that to decompose a controlled gate $\Lambda_1(G)$ two CNOTs are sufficient. We conclude that even a circuit with two CNOTs can be a rank-2 operation. For example, consider the following circuits

$$\begin{array}{c} \square \\ | \\ \bullet \\ | \\ \oplus \\ | \\ \square \end{array} , \quad \begin{array}{c} \square \\ | \\ \bullet \\ | \\ \oplus \\ | \\ \square \end{array} , \tag{4.5.5}$$

where the left one is derived from the decomposition in Eq. (2.3.20) and the right one can be obtained by inverting the two CNOTs by adding 4 Hadamards each as in Eq. (2.2.9). A similar but less trivial rank-2 circuit is

$$\begin{array}{c} \bullet \\ | \\ \oplus \\ | \\ \square \end{array} , \tag{4.5.6}$$

or also

$$\begin{array}{c} \square \\ | \\ \bullet \\ | \\ \oplus \\ | \\ \square \end{array} , \tag{4.5.7}$$

which are not in their more compact form. In fact, we can see that a chain of equally oriented CNOTs and arbitrary gates, always applied to the same qubit, constitutes a rank-2 operation that can be reduced to an optimal quantum circuit with at most 2 CNOTs. Obviously, it makes no difference whether the gates are applied to the control or the

4.5.2 Further analysis

There is much more to be explored in this area. Several entanglement measures for quantum gates have been proposed in the literature. These aim to quantify the average degree of entanglement they can generate when applied to separable states. The most well-known are Schmidt strength, linear entropy and concurrence. In the work in Ref. [41] the authors show that these three measures are equivalent for gates with Schmidt number equal to two (rank-2 operations), but not for operators with Schmidt number equal to four. It might be interesting to study these three measures in terms of the complexity of the circuit implementing the gate, rather than in terms of the entanglement generated. Indeed, the SWAP gate is classified as a maximally entangled operation by these measures. However, it does not produce any entanglement at all. If instead we interpret the entanglement measure as the number of CNOTs (or non-local operations in general) required to implement it, the result is consistent with what we observe, since the SWAP requires the maximum number of CNOTs to be implemented.

Another thing that would be interesting is to study these measures of entanglement (or complexity, in a sense) using different isomorphisms of Hilbert space. For example, in the case of two qubits we can use $\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2$, $(\mathbb{C}^2 \otimes \mathbb{C}^2) \boxtimes (\mathbb{C}^2 \otimes \mathbb{C}^2)$ (which is the one used in vector formalism), or even $(\mathbb{C}^2 \boxtimes \mathbb{C}^2) \otimes (\mathbb{C}^2 \boxtimes \mathbb{C}^2)$ (as suggested in Sec. 4.2.3). These different isomorphisms may highlight different properties of entanglement and have their roots in the tensor representation of quantum computation and in the different *flattering* modes of a tensor.

Chapter 5

Collective flavour oscillation

Collective flavour oscillations of neutrinos due to forward neutrino-neutrino scattering provide an intriguing many-body system for time evolution simulations on a quantum computer. These phenomena are of particular interest in extreme astrophysical settings such as core-collapse supernovae, neutron star mergers and the early universe, as demonstrated by previous researches (see Refs. [42–48]). Understanding the dynamics of flavor oscillations is paramount in such studies, since the behaviour of matter under extreme conditions shows strong flavor dependence [49, 50], and, furthermore, different energy spectra characterise different neutrino flavors [51]. A comprehensive description of the dynamic evolution of flavor in these processes is hampered by the substantial computational resources required for simulations involving large numbers of interacting neutrinos. To address this challenge, a common strategy is to use a mean-field approximation to the equation of motion, which allows the study of extensive systems with complicated geometries as done in Refs. [52–55]. However, a thorough treatment of correlation effects in the full many-body evolution can be achieved in relatively compact systems with about 10 neutrinos [56–58]. This is particularly feasible in scenarios with significant symmetries [59–62] or limited levels of bipartite entanglement, which can be effectively addressed using tensor network methods proposed in Refs. [63–65]. Another class of approaches, known as semiclassical methods, preserves certain correlations while maintaining computational efficiency. These methods have recently been applied to the neutrino problem in Ref. [66]. Quantum simulations offer an alternative way to study out-of-equilibrium flavor dynamics in regimes beyond the reach of classical approaches. Initial calculations have been performed on small systems containing up to four neutrinos, using both digital quantum computers [67, 68] and quantum annealers [69]. These studies highlight the need for careful algorithm design and implementation of error mitigation techniques when tackling this formidable challenge on current generation quantum devices. The aim of this chapter is to give an overview of the physics behind the collective neutrino oscillation phenomena.

This chapter begins with a detailed description of the physical phenomena and environments in which collective flavour oscillations occur (Sec. 5.1). Emphasis is placed on the supernova neutrino case, where the high density makes the neutrino-neutrino interaction important in the dynamical description. The supernova explosion mechanism is also summarised. Finally, the derivation of the Hamiltonian governing the evolution of flavor oscillations is detailed (Sec. 5.2).

5.1 Neutrinos in a high-density environment

Neutrinos are particles in the Standard Model (SM) that belong to the lepton family, along with the other two families of quarks and bosons. There are six types of lepton, divided into three generations according to flavour, each of which contains a charged lepton and its neutral

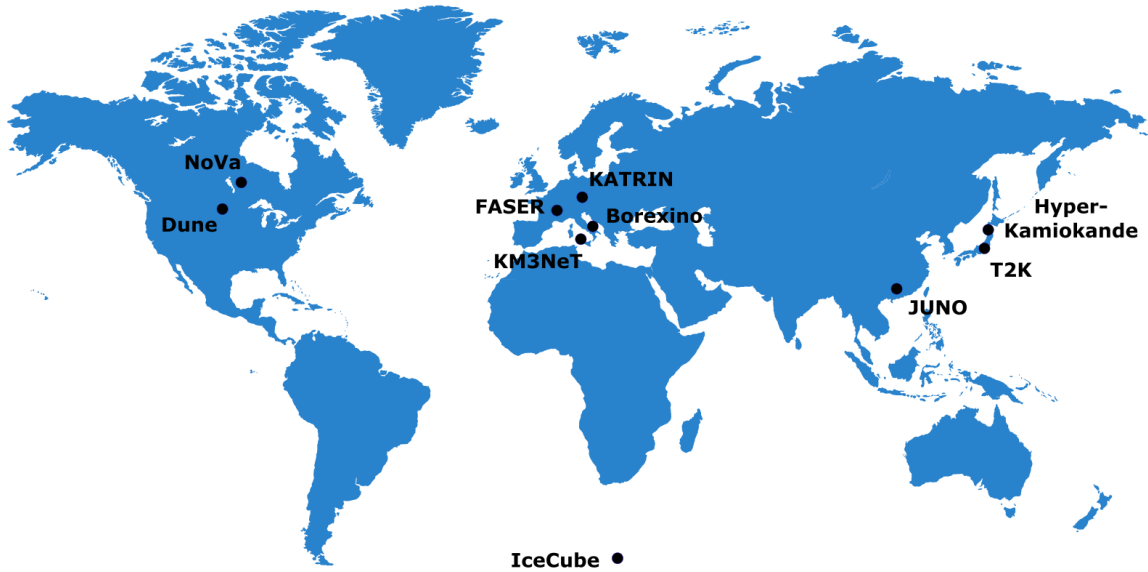


Figure 5.1: Some of the neutrino physics experiments conducted around the world.

counterpart, the neutrino, which is a fermionic stable particle with extremely low mass. The former interact through the electroweak force, the latter only through the weak nuclear force. We recall that, contrary to what the SM allows, neutrinos have a non-zero mass. This property was first observed experimentally thanks to β decays ($p \rightarrow n + e^+ + \nu_e$ and $n \rightarrow p + e^- + \bar{\nu}_e$) and is the cause of an important phenomenon called flavour oscillation. Nowadays, the physics community invests substantial efforts and resources in studying the physics of neutrinos. Fig. 5.1 shows a few of the past and currently active experiments worldwide.

5.1.1 Neutrino mixing

The main observation leading to the hypothesis that neutrinos have a non-zero mass was that of neutrino flavour oscillation. Neutrinos of the three generations ν_e, ν_μ, ν_τ do not have a definite mass, but they are rather linear combinations of three mass eigenstates ν_1, ν_2, ν_3 with a defined mass m_1, m_2, m_3 (of the order of $\sim 0.05 \text{ eV}/c^2$). This means that a neutrino of one flavour, after travelling a certain distance, develops the presence of other flavours that were not present at the beginning. This is due to the fact that wave functions, associated with a different mass eigenstates, will oscillate at different frequencies. We will therefore observe a relative phase between the components.

It is well known that the flavor of neutrinos changes under three contributions, each corresponding to different oscillation length and frequency scales, giving rise to different oscillation modes. The **vacuum term** describes the flavour oscillation caused by the mismatch between the mass and flavour eigenstates. This term gives rise to the so-called slow mode oscillations with an oscillation length of

$$l_{vac} \sim \frac{2E}{\delta m^2}, \quad (5.1.1)$$

which is of the order of $l_{vac} \sim (100 - 1000) \text{ m}$ using $E \sim 10 \text{ MeV}$, $\delta m^2 \sim (10^{-3} - 10^{-4}) \text{ eV}^2$ and introducing $\hbar c$ to the numerator.

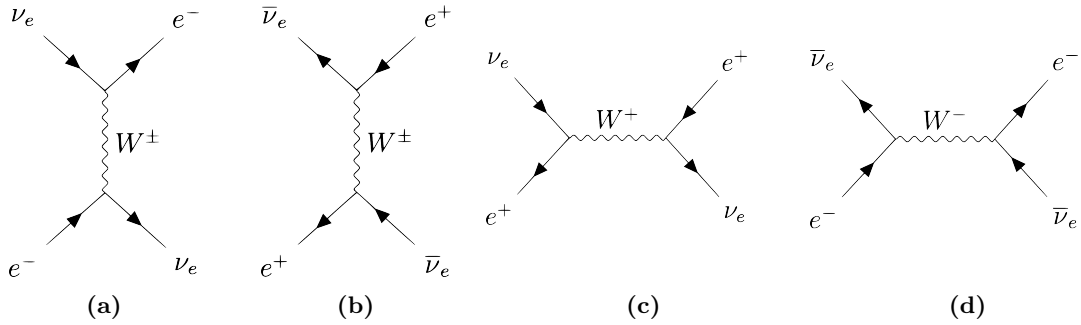


Figure 5.2: Scattering processes generating the MSW effect. Panels (a) and (c) for neutrinos and corresponding amplitude $\sqrt{2}G_F(n_e - \bar{n}_e)$ and panels (b) and (d) for antineutrinos and corresponding amplitude $-\sqrt{2}G_F(n_e - \bar{n}_e)$. Time runs from left to right. Panels (a) and (b) contain two diagrams each depending on the exchanged charged boson, which is fixed depending on the direction of the boson current.

The **matter term** describes the interaction with the surrounding matter through a boson exchange between the neutrino and the corresponding charged lepton. This happens mainly with the electrons, so we refer to this phenomenon as $\nu_e - e^-$, $\bar{\nu}_e - e^+$, $\nu_e - e^+$ and $\bar{\nu}_e - e^-$ scattering represented by the Feynman diagrams in Fig. 5.2. This interaction obviously depends on the density of the leptons and, according to the SM, on the Fermi coupling constant G_F . In particular, this contribution depends on the difference between the electron density n_e and the positron density \bar{n}_e (for neutrinos and with opposite sign for antineutrinos) and gives rise to the so-called MSW (Mikheyev-Smirnov-Wolfenstein) effect [70]. It is associated with the slow mode with an oscillation length of the order of

$$l_{mat} \sim \frac{1}{G_F(n_e - \bar{n}_e)}, \quad (5.1.2)$$

where the Fermi constant is measured in $\text{GeV} \cdot \text{m}^3$, the number density is in m^{-3} and recovering $\hbar c$ to the numerator we obtain a length.

Finally, the **neutrino-neutrino term** describes the $\nu - \nu$ scattering and depends on the Fermi constant and the difference between the neutrino density n_ν and the antineutrino density \bar{n}_ν . This type of oscillation is also called the fast mode and has an oscillation length of the order of

$$l_{\nu\nu} \sim \frac{1}{G_F(n_\nu - \bar{n}_\nu)}. \quad (5.1.3)$$

This term is responsible for the so-called collective flavour oscillations, a very interesting phenomenon that dominates at high density and is predicted to occur in extreme astrophysical environments such as core-collapse supernovae, neutron star mergers and the early universe.

5.1.2 Supernovae neutrinos

When very massive stars ($M \gtrsim 8M_\odot$) run out of fuel, a transformation begins that will eventually lead to a supernova explosion and the creation of a neutron star. During this mechanism, which takes a few seconds, a large amount of energy $\sim 10^{53}$ ergs¹ is released in the form of an enormous number of neutrinos $\sim 10^{58}$ (called supernova neutrinos SN), which escape with an energy of the order of $E_\nu \sim (10 - 30)$ MeV. This process, together with others such as the merging of massive stars, is one of the most interesting and not fully understood phenomena in the Universe. The explosion mechanism of supernovae is shown in Fig. 5.3 and can be summarised by the following steps. For a more in-depth study, we refer the reader to more specific texts, such as Ref. [71]

¹1 erg = 10^{-7} J.

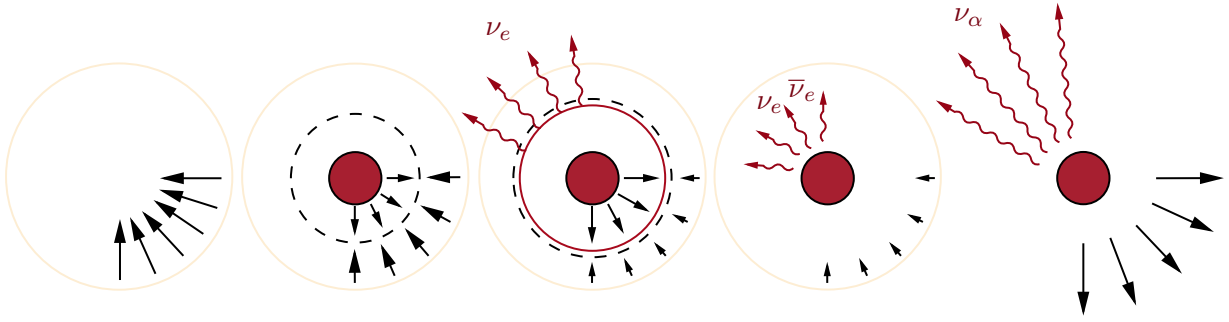
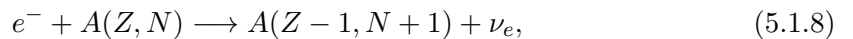


Figure 5.3: Scheme of the core collapse of a massive star. From left to right: first the collapse begins, increasing the density of the interior and forming the proto-neutron core. The falling matter reaches the wall, causing the so-called shock. When the inner neutrinos reach the neutrino sphere, they are emitted. The next step of neutrino emission cools the inner region and heats the outer region through the short mean free paths. This can revive the expansion and eventually cause the explosion, releasing a huge number of neutrinos.

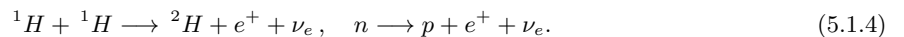
1. **Pre-collapse.** The steady state period of a star's life is on the order of $\sim 10^7$ years, during which different fusion states occur in sequence (hydrogen, helium, carbon, oxygen, silicon and finally iron)². During the nucleosynthesis process, pre-SN neutrinos are produced and emitted, cooling the star. In addition, when iron ^{56}Fe is produced by fusion inside a star, no heavier element can be produced with a corresponding release of energy. This element accumulates in the interior of the star, forming an inert core. The core is now supported only by the electron degeneracy pressure, as no further energy is available from fusion processes. When the mass of the iron core exceeds $\sim 1.44 M_{\odot}$, the degeneracy pressure is no longer sufficient to support the core. These two factors together (lack of energy and neutrino cooling) trigger the collapse.
2. **Collapse.** Due to the rapid increase in density and temperature, two main processes occur during this phase. The first is iron photodisintegration



which is an endothermic reaction that costs ~ 1.7 MeV/nucleon and reduces the energy in the nucleus, increasing the contraction. The neutrons remain stable due to the large e^- energy. Secondly, the atoms are neutronised by the electron capture reaction



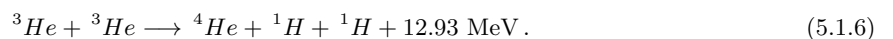
²An example of a nucleosynthesis reaction is the proton-proton chain, in which hydrogen nuclei are converted into helium nuclei in a series of reactions. In the first step, two hydrogen nuclei form a deuterium nucleus by β decay



Then the reaction



creates an isotope of helium. At this point three different cases can follow, depending on the temperature. One of the possibilities is the completion of the chain thanks to the reaction



where nucleons are released from nuclei and protons capture electrons by emitting neutrino beams. This causes electrons to disappear from the system, reducing the degeneracy pressure and further accelerating the implosion. The star is now in an extremely unstable equilibrium, influenced by the beta equilibrium

$$e^- + p \longleftrightarrow n + \nu_e. \quad (5.1.9)$$

At this stage, neutrinos dominate the energy emission of the star and play a crucial role in the explosion mechanism. The collapse dynamics are determined by the entropy and the lepton number, which are modified by the weak-dominated processes described above. When the equilibrium is broken, the star begins to collapse under its own weight due to gravity. Something important happened to the neutrinos in this process. Initially, the neutrinos produced by the electron capture reactions were free to flow out of the star, but once the matter density has reached $\sim 10^{12}$ g/cm³, their diffusion times become longer than the collapse time due to scattering with nuclei. Neutrinos in the high-density region are then trapped in a region called the neutrino-sphere.

3. **Shock.** At some point during the collapse phase, the density of the innermost core becomes so high ($\sim 10^{14}$ g/cm³) that the matter becomes almost incompressible, forming an impenetrable wall. As a result, the gravitational collapse suddenly stops and the core undergoes a bounce, creating a shock wave that propagates outwards as the outer layers continue to collapse. When the shock front reaches the neutrino sphere, it causes the neutrino burst emission in which neutrinos convert the high degeneracy energy to thermal energy.
4. **Explosion.** When the shock reaches the outer region can stop the collapse and cause the explosion, but can also happen that the explosion arrests due to energy loss. In this scenario the nascent neutron star contains degenerate electrons and trapped neutrinos which perform short mean free paths until they reach, in a fraction of second, the neutrino sphere with a thermal neutrinos emission. This has the important effect of cooling the inner parts and heating the outer parts (throughout $\nu_e + n \rightarrow e^- + p$ and $\bar{\nu}_e + p \rightarrow e^+ + n$) increasing the pressure and causing the explosion. The explosion releases $\sim 10^{51}$ ergs of energy, which is actually only 1% of the neutrino energy and 99% of the gravitational energy trapped by the collapse. Subsequently, the $\sim 10^{58}$ emitted neutrinos carry away 10^{53} ergs of energy in a time $\tau \sim (2 - 10)$ s after traversing the core by a random walk process with short mean free paths. After the collapse, the remaining core has a size of ~ 10 km and a mass comparable to that of the Sun.

5.1.3 Why we care about supernovae neutrinos

Observing and describing the evolution of neutrinos from supernovae and binary mergers is an area of great interest to modern physics, as it provides information about various phenomena that occur in them. The description of the flavour evolution is equally important, since these environments are dominated by weak interactions and are therefore very sensitive to the neutrino flavours. Some reasons why we care about the description of supernova neutrinos are listed below.

- (i) Neutrinos interact very weakly with the medium through which they travel and are therefore valuable astronomical messenger of information, together with gravitational waves, from very distant points in time and space in the Universe, where nature is described by physics under extreme conditions (high densities $\rho \sim 10^{15}$ g/cm³ and energies

$T \sim 10^{10} \text{ K}^3$).

- (ii) Neutrinos are emitted at different stages of the supernova explosion and provide different information about the life stage of the star. In fact, we can identify pre-collapse supernovae neutrinos emitted from nucleosynthesis, neutronisation neutrinos from the homonymous process and post-explosion neutrinos. Observing the neutrinos emitted during the supernova explosion is a useful way of obtaining information about the life stage of the star and the processes taking place within it. Note, for example, that the luminosity of the post-collapse neutrinos emitted exceeds that of photons, making neutrinos excellent candidates for observable particles from the supernova explosion.
- (iii) Neutrinos are the main carriers of the lepton number in the reactions that take place inside the nucleus. The main process is represented by nucleosynthesis reactions and electron capture. A crucial aspect is that these processes are governed by the weak interaction, so they involve neutrinos and are strongly flavour dependent [49, 50]. For example, consider that only electron neutrinos can change the neutron/proton ratio, depending on the equilibrium of the following processes:

$$\nu_e + n \longleftrightarrow p + e^-, \quad n \longleftrightarrow p + e^- + \bar{\nu}_e, \quad \bar{\nu}_e + p \longleftrightarrow n + e^+. \quad (5.1.10)$$

This is one reason why we are interested in the evolution of flavour oscillation.

- (iv) Neutrinos play a crucial role in the core collapse explosion mechanism for several reasons. They are responsible for the loss of entropy that occurs during the collapse, which can increase the instability and eventually cause the explosion [72], and they also heat and cool different regions of the inner part of the core, increasing the instability [71].
- (v) During core collapse, different neutrinos are emitted with different spectra [51]. In particular, electron neutrinos are emitted with generally lower energies than muonic and tauonic neutrinos $E_{\nu_e} < E_{\nu_x}$. This is because electronic neutrinos can decouple from matter later, giving them more time to cool. On the other hand, muonic and tauonic neutrinos, which come from an inner neutrino sphere⁴ at a higher temperature, can also decouple from matter earlier. However, some results suggest that at a certain distance from the emission point, the phenomenon known as spectrum splitting occurs, producing more energetic electronic neutrinos [54, 72].

5.2 Hamiltonian formalism in two-flavor approximation

To write the flavor Hamiltonian of a many-neutrino system emitted from a collapsing core and composed of N neutrinos, we first make the approximation of considering only two possible flavors, which allows us to exploit the algebra of $SU(2)$ and describe the flavor state as a flavor isospin. The two-flavour approximation is supported by evidence showing that this description can be quite suitable in some astrophysical situations where we can neglect the mixing angle between ν_1 and ν_3 (namely $\theta_{13} = 0$), as shown in [73]. In any case, it would certainly be interesting to add the third flavour, but at the cost of complicating the Hamiltonian and thus

³Temperatures are measured in energy units $k_B T$ where the Boltzmann constant is $k_b \approx 8.617 \times 10^{-5} \text{ eV/K}$. For example $10^{10} \text{ K} \approx 861700 \text{ eV} \sim 1 \text{ MeV}$.

⁴Note that while electronic neutrinos are also produced in beta decay, muonic and tauonic neutrinos are mainly produced from electron-positron pairs:

$$e^- + e^+ \longrightarrow \nu_\alpha + \bar{\nu}_\alpha. \quad (5.1.11)$$

the implementation of the quantum simulation, which in the case of qubits is intrinsically based on the $\mathfrak{su}(2)$ algebra. A recent paper analysing the role of the third flavor in the entanglement between neutrinos is, for example, the one in Ref. [74]. An important point is that studying the system in the two flavour approximation contains some physical properties that are also present in the three flavour case, and the $SU(2)$ model has the great advantage of making the physical description and quantum implementation feasible on a near-term quantum computer at a not too high price.

In the two-flavour context, each flavour state is a linear combination of two mass eigenstates, and we can write the relation between the annihilation (creation) operators on the flavour basis $\{|\nu_e\rangle, |\nu_x\rangle\}$ and on the mass one $\{|\nu_1\rangle, |\nu_2\rangle\}$ using a unitary transformation

$$\begin{pmatrix} a_e^{(\dagger)} \\ a_x^{(\dagger)} \end{pmatrix} = \begin{pmatrix} \cos(\theta_\nu) & \sin(\theta_\nu) \\ -\sin(\theta_\nu) & \cos(\theta_\nu) \end{pmatrix} \begin{pmatrix} a_1^{(\dagger)} \\ a_2^{(\dagger)} \end{pmatrix}, \quad (5.2.1)$$

where θ_ν is the mixing angle. The inverse relationship is

$$\begin{pmatrix} a_1^{(\dagger)} \\ a_2^{(\dagger)} \end{pmatrix} = \begin{pmatrix} \cos(\theta_\nu) & -\sin(\theta_\nu) \\ \sin(\theta_\nu) & \cos(\theta_\nu) \end{pmatrix} \begin{pmatrix} a_e^{(\dagger)} \\ a_x^{(\dagger)} \end{pmatrix}. \quad (5.2.2)$$

This unitary transformation is the two flavour version of the so-called PMNS matrix [75]. At this point we calculate the three main terms of the flavour Hamiltonian introduced in Sec. 5.1.1, as done in Refs. [72, 76], obtaining

$$H = H_{vac} + H_{mat} + H_{\nu\nu}. \quad (5.2.3)$$

In the above equation, H_{vac} is the vacuum mixing term, which describes the oscillations caused by the misalignment between the flavour and mass eigenstates, and H_{mat} is the matter term, which describes the interaction with the electrons of the surrounding matter causing the MSW effect. Both are mathematically represented by one-body interaction terms describing the effect of an external field. Finally, the $H_{\nu\nu}$ describes the neutrino-neutrino interaction generated by forward scattering, which causes the so-called collective flavour oscillations. The latter is expressed as two-body interactions and adds a non-linear contribution to the equations of motion, making the description of their dynamics both hard and interesting.

5.2.1 Vacuum term

On the mass basis, the free Hamiltonian of the mass eigenstates is obviously diagonal and can be expressed as

$$H_{vac} = E_1 a_1^\dagger a_1 + E_2 a_2^\dagger a_2, \quad (5.2.4)$$

where each eigenstate is associated with a fixed energy E_i and $a_i^{(\dagger)}$ are annihilation (creation) operators. Using the approximation

$$E_i = \sqrt{p^2 + m_i^2} = \sqrt{p^2 \left(1 + \frac{m_i^2}{p^2}\right)} \approx p \left(1 + \frac{m_i^2}{2p^2}\right) \approx p + \frac{m_i^2}{2p} \approx p + \frac{m_i^2}{2E} \quad (5.2.5)$$

we obtain

$$\begin{aligned} H_{vac} &= \frac{m_1^2}{2E} a_1^\dagger a_1 + \frac{m_2^2}{2E} a_2^\dagger a_2 + p \left(a_1^\dagger a_1 + a_2^\dagger a_2 \right) \\ &= \frac{m_1^2}{2E} a_1^\dagger a_1 + \frac{m_2^2}{2E} a_2^\dagger a_2 + p \mathbb{I}. \end{aligned} \quad (5.2.6)$$

We can express this Hamiltonian on the flavour basis by replacing the mass eigenstates in the above equation by the linear combination of flavour eigenstates in Eq. (5.2.2). We have:

$$H_{vac} = \frac{m_1^2}{2E} \left(\cos^2 \theta_\nu a_e^\dagger a_e + \sin^2 \theta_\nu a_x^\dagger a_x \right) + \frac{m_2^2}{2E} \left(\sin^2 \theta_\nu a_e^\dagger a_e + \cos^2 \theta_\nu a_x^\dagger a_x \right) + \\ - \frac{m_1^2}{2E} \left(\cos \theta_\nu \sin \theta_\nu \left(a_e^\dagger a_x + a_x^\dagger a_e \right) \right) + \frac{m_2^2}{2E} \left(\cos \theta_\nu \sin \theta_\nu \left(a_e^\dagger a_x + a_x^\dagger a_e \right) \right) + p\mathbb{I} \quad (5.2.7)$$

and after simple calculations:

$$H_{vac} = \cos^2 \theta_\nu \left(\frac{m_1^2}{2E} a_e^\dagger a_e + \frac{m_2^2}{2E} a_x^\dagger a_x \right) + \sin^2 \theta_\nu \left(\frac{m_1^2}{2E} a_x^\dagger a_x + \frac{m_2^2}{2E} a_e^\dagger a_e \right) + \\ + \sin \theta_\nu \cos \theta_\nu \left(a_e^\dagger a_x + a_x^\dagger a_e \right) \frac{m_2^2 - m_1^2}{2E} + p\mathbb{I}. \quad (5.2.8)$$

Now, using simple trigonometric properties⁵, defining the square mass difference $\delta m^2 := m_2^2 - m_1^2$ between mass eigenstates, and neglecting the terms proportional to the identity, we have

$$H_{vac} = \frac{\delta m^2}{4E} \sin(2\theta_\nu) \left(a_e^\dagger a_x + a_x^\dagger a_e \right) + \frac{\delta m^2}{4E} \cos(2\theta_\nu) \left(a_x^\dagger a_x - a_e^\dagger a_e \right). \quad (5.2.10)$$

In matrix form, $a_i^\dagger a_i$ and $a_i^\dagger a_j$ are diagonal and off-diagonal elements, respectively, and are related to Pauli matrices by

$$Z = a_e^\dagger a_e - a_x^\dagger a_x, \quad X = a_e^\dagger a_x + a_x^\dagger a_e \quad (5.2.11)$$

thanks to the quasi-spin algebra. Note that here and in the following we will use both the notation $\sigma^{(x)}, \sigma^{(y)}, \sigma^{(z)}$ and X, Y, Z to identify the Pauli matrices. The matrix form of the vacuum Hamiltonian is

$$H_{vac} = \frac{\delta m^2}{4E} (\sin(2\theta_\nu)X - \cos(2\theta_\nu)Z) = \frac{\delta m^2}{4E} \begin{pmatrix} -\cos(2\theta_\nu) & \sin(2\theta_\nu) \\ \sin(2\theta_\nu) & \cos(2\theta_\nu) \end{pmatrix} := \frac{\omega}{2} \mathbf{b} \cdot \boldsymbol{\sigma}, \quad (5.2.12)$$

where we have used bold symbols to denote 3-dimensional vectors, we have defined the external field

$$\mathbf{b} := (\sin(2\theta_\nu), 0, -\cos(2\theta_\nu)), \quad (5.2.13)$$

the energy scale $\omega := \delta m^2/2E$, and the vector of Pauli matrices $\boldsymbol{\sigma} = (X, Y, Z)$. It's also common in the literature to define the vacuum potential as $\Delta := \delta m^2/(4E)$.

5.2.2 Matter term

The matter term describes the scattering between the neutrinos and the surrounding matter. At low energies, neutrinos can only exchange a W^\pm boson with a charged lepton of the same family, i.e. they can only perform the elastic scattering shown in Fig. 5.4a. Since the only lepton present (in sufficient quantity) in a supernova environment is the electron e^- , only the electronic neutrino ν_e and antineutrino $\bar{\nu}_e$ interact with the surrounding matter via W^\pm , as shown in Fig. 5.2 (panel (a) and (d)). This is due to the fact that to exchange W^\pm with a lepton of a different family, the neutrino-lepton scattering should be anelastic, as the one shown

⁵

$$\cos^2(\theta) = \frac{1 + \cos(2\theta)}{2}, \quad \sin^2(\theta) = \frac{1 - \cos(2\theta)}{2}, \quad \cos(\theta) \sin(\theta) = \frac{\sin(2\theta)}{2}. \quad (5.2.9)$$

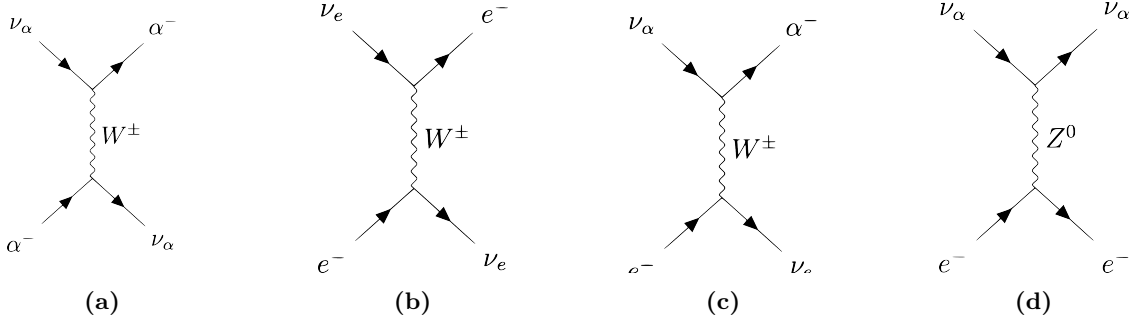


Figure 5.4: Feynman diagrams of neutrino-lepton scattering, with time going conventionally from left to right. Panel (a) shows the general W^\pm exchange between a neutrino and a lepton of the same family (the charged boson depends on the direction of the boson current). Panel (b) is the main process $\nu_e - e^-$ which produces the MSW effect throughout the exchange of charged bosons, panel (c) is the inelastic scattering in the case of $\alpha \in \{\mu, \tau\}$, which is suppressed at low energy, and panel (d) is the neutral boson exchange, which is flavour independent ($\alpha \in \{e, \mu, \tau\}$) and adds a term proportional to the identity.

in Fig. 5.4c compared to the case in Fig. 5.4b. This makes the matter contribution to the flavour dynamics dependent on the flavour itself. The $\nu_e - e^-$ interaction potential depends on the electron density n_e and the Fermi coupling constant G_F via the relation $A = \pm\sqrt{2}G_F n_e$ (where \pm stands for neutrinos and antineutrinos respectively), and this gives rise to a diagonal contribution Hamiltonian term which leads to the MSW effect. Note that neutrinos also interact with matter through the neutral Z^0 boson exchange (diagram in Fig. 5.4d). However, since this interaction affects neutrinos independently of flavor, it does not modify the flavor evolution and gives a contribution to the Hamiltonian that is proportional to the identity. The matter Hamiltonian is

$$H_{mat} = A \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = \frac{A}{2} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} + \frac{A}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \frac{A}{2} (Z + \mathbb{I}), \quad (5.2.14)$$

where we have rewritten it using the $\mathfrak{su}(2)$ algebra. The identity term can be neglected, as for the Z^0 boson exchange, and we end with

$$H_{mat} = \frac{G_F n_e}{\sqrt{2}} (a_e^\dagger a_e - a_x^\dagger a_x) = \frac{A}{2} Z. \quad (5.2.15)$$

We can interpret the vacuum plus matter Hamiltonian as an effective oscillation in the vacuum with different parameters, i.e. with a different external field. We have

$$\begin{aligned} H_{vac} + H_{mat} &= \frac{\delta m^2}{4E} \begin{pmatrix} -\cos(2\theta_\nu) & \sin(2\theta_\nu) \\ \sin(2\theta_\nu) & \cos(2\theta_\nu) \end{pmatrix} + \frac{A}{2} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \\ &= \frac{\delta m^2}{4E} \begin{pmatrix} -(\cos(2\theta_\nu) + \frac{A}{2} \frac{4E}{\delta m^2}) & \sin(2\theta_\nu) \\ \sin(2\theta_\nu) & (\cos(2\theta_\nu) - \frac{A}{2} \frac{4E}{\delta m^2}) \end{pmatrix}. \end{aligned} \quad (5.2.16)$$

To rewrite it in the same form as H_{vac} , we define

$$\cos(2\theta'_\nu) := \left(\cos(2\theta_\nu) - \frac{A}{2} \frac{4E}{\delta m^2} \right) C, \quad \sin(2\theta'_\nu) := \sin(2\theta_\nu) C, \quad \delta m'^2 = \frac{\delta m^2}{C}. \quad (5.2.17)$$

From the first two relations we get the equation

$$C^2 \left(\sin^2(2\theta_\nu) + \left(\cos(2\theta_\nu) - \frac{A}{2} \frac{4E}{\delta m^2} \right)^2 \right) = \cos^2(2\theta'_\nu) + \sin^2(2\theta'_\nu), \quad (5.2.18)$$

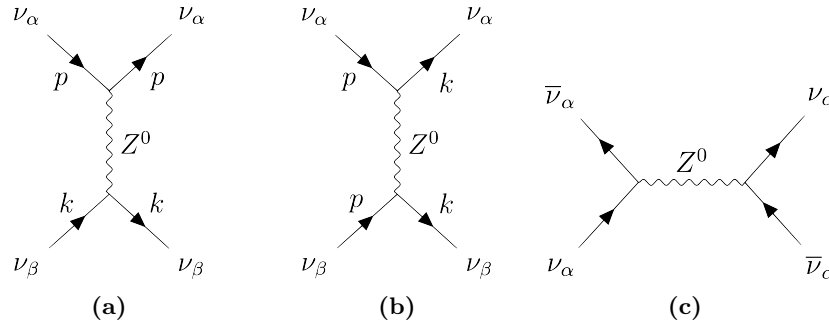


Figure 5.5: Feynman diagram for neutrino-neutrino scattering mediated by the neutral Z^0 boson. Time runs from left to right. Panel (a), where $\alpha, \beta \in \{e, \mu, \tau\}$ gives a diagonal contribution that does not affect the flavour mixing, while panel (b) represents the momentum exchange (if $\alpha \neq \beta$) between two neutrinos and is an off-diagonal term that contributes to the flavour oscillation. Panel (c) shows that the s-channel is also possible, but is flavour independent and therefore does not affect the flavour oscillation at low energies.

which gives

$$C = \sqrt{\sin^2(2\theta_\nu) + \left(\cos(2\theta_\nu) - \frac{A}{2} \frac{4E}{\delta m^2}\right)^2}. \quad (5.2.19)$$

The vacuum plus matter Hamiltonian becomes

$$H_{vac} + H_{mat} = \frac{\delta m'^2}{4E} \begin{pmatrix} -\cos(2\theta'_\nu) & \sin(2\theta'_\nu) \\ \sin(2\theta'_\nu) & \cos(2\theta'_\nu) \end{pmatrix}, \quad (5.2.20)$$

which is in the same form as the vacuum term alone but with a different parameter $\delta m'^2$ and a different external field $\mathbf{b}' := (\sin(2\theta'_\nu), 0, -\cos(2\theta'_\nu))$.

5.2.3 Neutrino-neutrino term

The $\nu - \nu$ scattering is described by a two-body interaction term depicted in Fig. 5.5. As for the neutrino-electron scattering, this contribution is proportional to the Fermi coupling constant G_F and the neutrino density $n_\nu = N/V$ by the amplitude $\eta := \sqrt{2}G_F n_\nu := \mu/N$. Note that N is the number of neutrinos, V is the volume of the system and we also defined the total energy $\mu := N\eta$. The $\nu - \nu$ interaction also depends on the relative angle of propagation between two different neutrinos ν_i and ν_j , which can be accounted for by the coupling constant

$$J_{ij} = (1 - \cos(\theta_{ij})), \quad (5.2.21)$$

where θ_{ij} is the scalar product between the two propagation directions according to

$$\frac{\mathbf{p}_i \cdot \mathbf{p}_j}{\|\mathbf{p}_i\| \|\mathbf{p}_j\|} = \cos(\theta_{ij}), \quad (5.2.22)$$

where \mathbf{p}_i is the momentum vector of the neutrino ν_i . This means that the neutrinos that interact the most are those that propagate in directions with a larger relative angle. Note that this result is consistent with what we would get by calculating the ultrarelativistic limit of the cross section. The Hamiltonian term describing the interaction between two different neutrinos ν_i and ν_j is then given by the $SU(2)$ invariant product of Pauli matrices

$$H_{ij} = \sqrt{2}G_F n_\nu \left(1 - \frac{\mathbf{p}_i \cdot \mathbf{p}_j}{\|\mathbf{p}_i\| \|\mathbf{p}_j\|}\right) \boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_j = \frac{\mu}{N} J_{ij} \boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_j, \quad (5.2.23)$$

where the expression $\boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_j$ is the scalar product between vectors of Pauli matrices, giving

$$\boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_j = X_i \otimes X_j + Y_i \otimes Y_j + Z_i \otimes Z_j. \quad (5.2.24)$$

In matrix form⁶, the interaction between two neutrinos is given by

$$H_{ij} = \frac{\mu}{N} \begin{pmatrix} J_{ij} & 0 & 0 & 0 \\ 0 & -J_{ij} & 2J_{ij} & 0 \\ 0 & 2J_{ij} & -J_{ij} & 0 \\ 0 & 0 & 0 & J_{ij} \end{pmatrix}. \quad (5.2.25)$$

5.2.4 Total isospin Hamiltonian

The complete Hamiltonian describing the flavour oscillation of a system of N neutrinos is given by the sum of the three contributions (Eqs. (5.2.12), (5.2.15) and (5.2.23)) summed over all the particles in the system, namely

$$H = \frac{1}{2} \sum_{i=0}^{N-1} \omega_i \mathbf{b} \cdot \boldsymbol{\sigma}_i + \frac{A}{2} \sum_{i=0}^{N-1} Z_i + \sum_{i<j}^{N-1} \frac{\mu}{N} J_{ij} \boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_j. \quad (5.2.26)$$

Note that in the above equation there are implicit tensor products with the particles that are not written explicitly. For example, in the one-body terms

$$\sigma_0^{(k)} = \sigma^{(k)} \otimes \mathbb{I} \otimes \cdots \otimes \mathbb{I}, \quad \sigma_1^{(k)} = \mathbb{I} \otimes \sigma^{(k)} \otimes \mathbb{I} \otimes \cdots \otimes \mathbb{I}, \quad (5.2.27)$$

for $k \in \{x, y, z\}$, and the two-body contribution contains terms of the form

$$\sigma_0^{(k)} \sigma_1^{(k)} = \sigma^{(k)} \otimes \sigma^{(k)} \otimes \mathbb{I} \otimes \cdots \otimes \mathbb{I}, \quad \sigma_0^{(k)} \otimes \sigma_2^{(k)} = \sigma^{(k)} \otimes \mathbb{I} \otimes \sigma^{(k)} \otimes \cdots \otimes \mathbb{I}. \quad (5.2.28)$$

The flavour isospin Hamiltonian in Eq. (5.2.26) is standard in studies of collective flavour oscillation as can be seen, for example, in Ref. [76]. To further illustrate the simple structure of the neutrino-neutrino interaction in this formulation, we observe that the last term in Eq. (5.2.26) contains a sum over all possible $N(N-1)/2$ pairs. The all-to-all character of the interaction is a product of two main ingredients: the consideration of neutrinos as plane-waves and the Z -mediated weak interaction as a contact term in coordinate space. Further extensions of the model that more realistically account for the spatial extension of neutrinos using wave packets, an important ingredient for inhomogeneous systems (see, e.g., Ref. [77]), are possible but are left for future work. In conclusion, the Hamiltonian of a many-neutrino system used in this work is equivalent to an all-to-all coupled spin system, and thus represents an interesting strongly coupled many-body quantum problem governed by the weak interaction, in which we can demonstrate the usefulness of quantum computing.

⁶We have used the standard notation which represents a composite system belonging to the tensor space with the Kronecker product convention indicating it with \otimes instead of \boxtimes .

Chapter 6

Quantum algorithm and results

Quantum computing can offer a significant advantage in the characterisation of multi-particle quantum systems, as anticipated in Sec. 1.2 of Chapter 1, without the strict need for computational resources to grow exponentially with the system's dimension. The global push to use quantum devices to explore the principles of Standard Model physics has sparked a worldwide effort to design algorithms and implement them on currently available quantum platforms. Given a quantum system in an initial state $|\psi(0)\rangle$, its time evolution governed by the Hamiltonian H is realised by the unitary operator $U(t) = e^{-iHt}$, which evolves the state as dictated by the time-dependent Schrödinger equation $|\psi(t)\rangle = U(t)|\psi(0)\rangle$. In general, a straightforward approach based on this interpretation encounters a growing computational cost on classical computers as the size of the system increases. This is due both to the large memory requirements for encoding the system states and to the operational overhead of performing matrix multiplications.

As mentioned in Sec. 1.2.2, the digital quantum simulation of the real-time evolution of a quantum system requires the decomposition of the propagator into a sequence of layers composed of elementary quantum gates, usually consisting of one- or two-qubit elementary operations. Quantum gate decomposition plays a crucial role in the use of near-term devices, where noise dominates the quality of the simulation. Optimisation of the quantum circuit, by minimising the number of operations to be performed and using the specific language of the machine, allows the quality of the results to be improved. All these aspects can be summarised in what is known as **machine-aware optimal compilation**. The aim of this chapter is to reproduce the evolution of the many-neutrino system presented in Chapter 5 using a quantum algorithm. The matrix dimension of the propagator grows exponentially with the number N of neutrinos, and in general would require an exponential number of operations to implement it exactly. This would make the quantum computing approach not really efficient in Hamiltonian simulations. To avoid exponential scaling, it is crucial to specialise the gate decomposition with respect to the particular physical problem we are tackling, for example by exploiting all the physical features that are present, such as the range of the interaction, the number of interacting bodies in each term, the Hamiltonian symmetries, the presence of reducible Hilbert spaces, and so on. The algorithm developed and the results presented in this chapter come from a joint work carried out with Dr Alessandro Roggero and can be found in the paper in Ref. [78].

This chapter introduces the approximation model used in the rest of the work for the dynamics of the collective flavour oscillation, and presents the theoretical evolution of a system with a limited number of neutrinos (Sec. 6.1). In order to cope with the computational complexity, the Trotter approximation of the time evolution operator is used (Sec. 6.2), which mitigates the exponential growth of the circuit complexity with special attention to the qubit topology and its error dependence. Then, the quantum algorithm designed to work on a trapped-ion-based testbed is presented in detail (Sec. 6.3), and the machine-aware optimisation is carried out

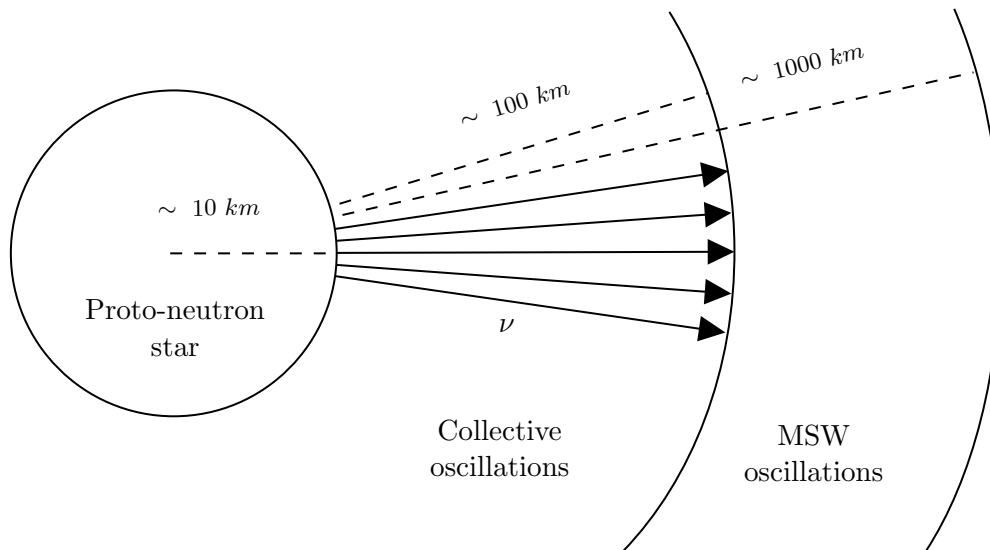


Figure 6.1: Scheme of the neutrino oscillation environment in a core-collapse supernova assuming spherical symmetry. The matter term is dominant in a large distance $r \sim 1000$ km from the core where the density of neutrinos is low enough but the density of matter is still relevant. The neutrino-neutrino term is dominant in a middle distance $r \sim (10 - 100)$ km from the core where the neutrino density is high.

(Sec. 6.4). The results obtained are finally presented (Sec. 6.5) for both single and multi-step evolution. The chapter ends with a comprehensive analysis of the Trotter error and evaluates the complexity scaling of the algorithm (Sec. 6.6).

6.1 System model and theoretical evolution

The study of extreme astrophysical environments is fundamental to understanding many aspects of astrophysics and cosmology, and requires a suitable model to describe many-neutrino systems. Based on the composition of the core-collapsed nucleus and the emitted neutrinos, layers are defined around the collapsing supernova core in which a particular behaviour of neutrinos dominates. A simple diagram showing the regions where different neutrino processes are active in a core-collapse supernova, assumed to have spherical symmetry, is shown in Fig. 6.1. Collective neutrino oscillations are generally expected to dominate in a region of intermediate distance from the core (~ 100 km) where the density of neutrinos is large, while the external lepton-electron density is not too large to suppress flavor oscillations [72]. The outer shell, where the neutrino density is lower, is instead dominated by vacuum oscillations and interactions with the surrounding matter, leading to the MSW effect.

The quantum simulation we want to implement corresponds to the collective flavor oscillation of a system of N neutrinos in a high-density environment. At this stage we decided to neglect the matter term in the Hamiltonian of Eq. (5.2.26)¹ and we fixed the neutrino energy to obtain the same coupling constant for the one-body and two-body energies, namely

$$\frac{\mu}{N} = \frac{\omega}{2}, \quad (6.1.1)$$

and measure time in units of μ^{-1} . Thus the quantum simulation aims to implement the evolution

¹This is mainly because the matter term is too large in magnitude and would be the dominant source of error in the Trotter approximation presented in the next sections. Methods to deal with this problem exist, such as the interaction picture approach, but are left for future work.

generated by:

$$H = \sum_{i=0}^{N-1} \mathbf{b} \cdot \boldsymbol{\sigma}_i + \sum_{i<j}^{N-1} J_{ij} \boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_j := \sum_{i=0}^{N-1} h_i + \sum_{i<j}^{N-1} h_{ij} := H^{(1b)} + H^{(2b)}, \quad (6.1.2)$$

where $H_{vac} := H^{(1b)}$ is the only one-body contribution and $H_{\nu\nu} := H^{(2b)}$ is the two-body one. We used, for the mixing angle in the external field \mathbf{b} of Eq. (5.2.13), the value $\theta_\nu = 0.195$, as previously used in the work [67], and we take a simple grid of angles

$$\theta_{ij} = \arccos(0.9) \frac{|i-j|}{(N-1)}, \quad (6.1.3)$$

meant to reproduce a narrow cone of forward peaked neutrinos in accordance to the geometry displayed in Fig. 6.1. With this choice of angular distribution and for even N , the neutrino Hamiltonian in Eq. (6.1.2) turns out to be symmetric under the particle exchange

$$\nu_k \longleftrightarrow \nu_{N-1-k}, \quad (6.1.4)$$

for $k \in \{0, \dots, N/2\}$. In this model, the geometry of the problem fixes the interaction between the neutrinos through the relative angles of propagation.

6.1.1 Encoding map

As mentioned in Sec. 1.2.2 of Chapter 1, two main ingredients are needed to perform a quantum Hamiltonian simulation: a state encoding map $|\psi\rangle \mapsto |\varphi\rangle$ of physical states $|\psi\rangle$ into device states $|\varphi\rangle$, and an operator to quantum gates mapping $e^{-iHt} \mapsto U$. In this section we see its implementation specialised in the case of a simulation of the collective flavor oscillation of a system of N neutrinos, where the Hamiltonian is the one defined in Eq. (6.1.2).

Due to the two-flavour approximation presented in Sec. 5.2, the flavour state of each neutrino corresponds to a flavour isospin. A single-neutrino state is generally given by the two-flavour superposition $|\psi\rangle = \alpha |\nu_e\rangle + \beta |\nu_x\rangle$ and can be completely described in terms of a qubit state by the mapping

$$|\nu_e\rangle \mapsto |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |\nu_x\rangle \mapsto |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad (6.1.5)$$

which is the simpler way to encode such a system. In this way, a system of N neutrinos can be completely characterised by N qubits, and the two spaces $\mathcal{H}_{qubit} = (\mathbb{C}^2)^{\otimes N}$ and $\mathcal{H}_{system} = (\mathbb{C}^2)^{\otimes N}$ have the same dimension. The initial state of the system used for all the simulations presented in this work consists in using an even value of N and setting the first $N/2$ neutrinos in the $|\nu_e\rangle$ flavour state and the other $N/2$ in the $|\nu_x\rangle$ state as follows

$$|\psi(0)\rangle = (|\nu_e\rangle)^{\otimes N/2} \otimes (|\nu_x\rangle)^{\otimes N/2}. \quad (6.1.6)$$

In this way, the initial state is symmetric under the composition of the particle exchange in Eq. (6.1.4), and the flavour inversion $|\nu_e\rangle \longleftrightarrow |\nu_x\rangle$, while the Hamiltonian is symmetric under the particle exchange alone. The encoding map is trivial, so the initial qubit state is

$$|\varphi(0)\rangle = (|0\rangle)^{\otimes N/2} \otimes (|1\rangle)^{\otimes N/2} \quad (6.1.7)$$

and corresponds to a 2^N dimensional vector with the only non-zero element at position $2^{N/2} - 1^2$. We emphasise that the algorithm we propose in this chapter works for any initial state $|\psi\rangle \in \mathcal{H}_{system}$, and different initial states allow the dynamics to be analysed at different points in the spectrum. For example, if the neutrinos were all the same, the system would be in the most energetic tail of the spectrum and, most importantly, we would not observe any dynamics. If, instead, the neutrinos are evenly distributed, as we have chosen, the system will be at an intermediate point in the spectrum, close to many other possible initial states. Moreover, we can interpret this initial state in another way, which may have a more physical meaning. This consists in interpreting the first half qubits in the register as fermions $|\nu_e\rangle, |\nu_x\rangle$, to which we assign $|\nu_e\rangle \mapsto |0\rangle$ and $|\nu_x\rangle \mapsto |1\rangle$, and the second half as antifermions $|\bar{\nu}_e\rangle, |\bar{\nu}_x\rangle$ with the same mapping. In this way we interpret the state differently depending on the position of the qubits. This has the advantage of allowing us to consider somewhat more realistic systems in which we have neutrinos and antineutrinos.

Finally, we would like to point out that the memory required by the quantum machine depends on the encoding map and is directly related to the size of the matrix representing e^{-iHt} and the vector representing $|\psi\rangle$. In general, the size of the quantum register varies depending on whether symmetries of the problem are considered or not. Including symmetries tends to reduce the number of qubits required to encode the problem, resulting in a smaller quantum register size. While this may suggest that the use of symmetries in quantum computing always offers advantages by providing a more concise representation that focuses on the relevant aspects, it is important to recognise that this advantage may not always be true if it leads to increased complexity in quantum algorithms. Consequently, the efficiency of a quantum algorithm depends on both the chosen encoding method and the design of the algorithm itself. It is worth noting that there are alternative encoding maps beyond the direct spin encoding in Eq. (6.1.5), which we leave for exploration in future research.

6.1.2 Ideal unitary propagator

Due to the fact that the encoding map is trivial, the real-time propagator expressed on the computational basis is the same as the one expressed on the flavour basis and is given by the exponential

$$U(t) = e^{-iHt} = e^{-i(\sum_i \mathbf{b} \cdot \boldsymbol{\sigma}_i + \sum_{i < j} J_{ij} \boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_j)t}, \quad (6.1.9)$$

which corresponds to a $2^N \times 2^N$ unitary matrix calculated as the exponential of the Hamiltonian in Eq. (6.1.2). One can obtain the exact time evolution by directly performing a matrix multiplication $|\psi(t)\rangle = U(t)|\psi(0)\rangle$. On the spin basis, the flavour content of an individual neutrino is obtained from the expectation value of the Pauli matrix $\sigma_i^{(z)} = Z_i$, namely

$$\langle Z_i(t) \rangle = \langle \psi(t) | Z_i | \psi(t) \rangle, \quad (6.1.10)$$

where we have left implicit the tensor products with identities related to the other particles. A neutrino in the $|\nu_e\rangle$ state has an expectation value of $\langle \nu_e | Z | \nu_e \rangle = 1$ and the flavour state $|\nu_x\rangle$

²For example, the initial state for the case of $N = 4$ neutrinos is given by a 16-dimensional vector with entry-1 at position 3, namely

$$|\varphi(0)\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \quad (6.1.8)$$

where we replaced the Kronecker product \boxtimes to the tensor one \otimes .

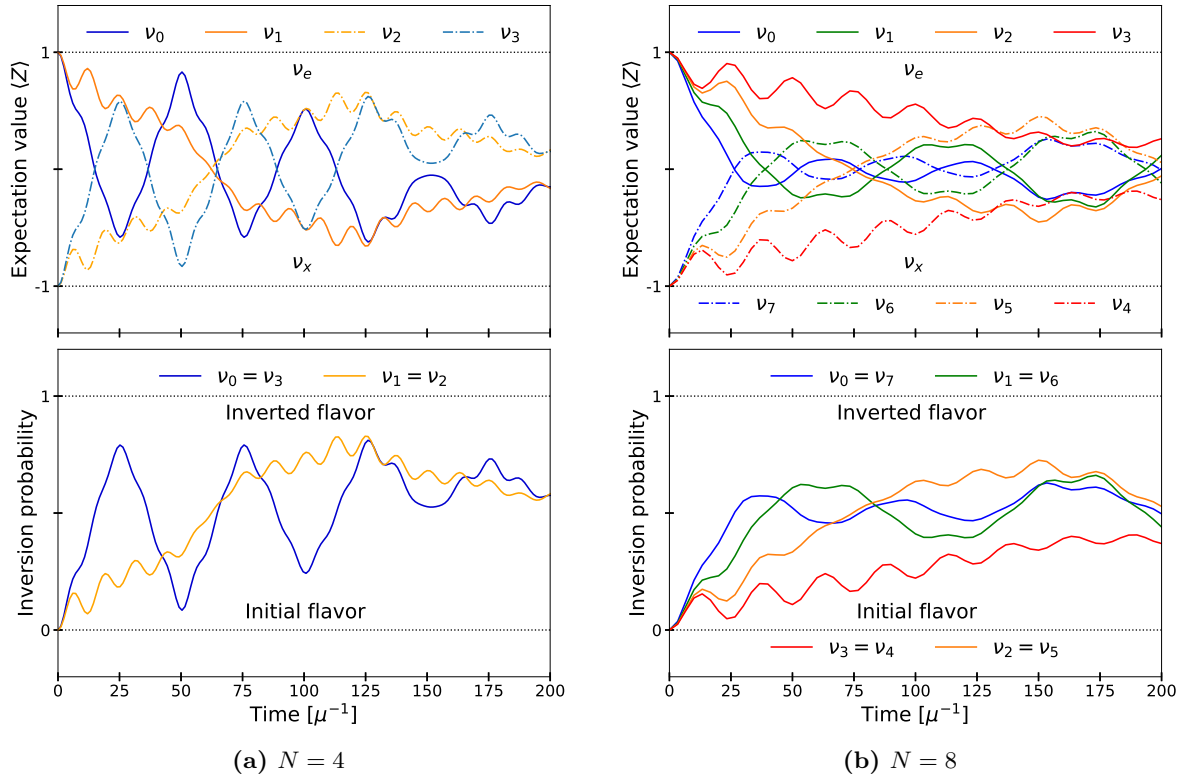


Figure 6.2: Exact evolution of a system of $N = 4$ (panel (a)) and $N = 8$ (panel (b)) with initial states $|\varphi(0)\rangle^{(4)} = |0011\rangle$ and $|\varphi(0)\rangle^{(8)} = |00001111\rangle$ respectively. The top panels show the evolution of the expectation value $\langle Z_i(t) \rangle$ for each neutrino $|\nu_i\rangle$, while the bottom panels show the evolution of the flavour inversion probability $P_i(t)$.

has the value $\langle \nu_x | Z | \nu_x \rangle = -1$. In a similar way, the inversion probability $P_i(t)$ can be expressed as

$$P_i(t) = \frac{|\langle Z_i(0) \rangle - \langle Z_i(t) \rangle|}{2}, \quad (6.1.11)$$

and represents the probability that a neutrino has inverted the flavour with respect to the initial one. We display the exact evolution of both quantities for systems of $N = 4$ and $N = 8$ neutrinos in Figures 6.2a and 6.2b respectively. In both figures, the top panel shows the results for $\langle Z_i(t) \rangle$, while the bottom panel shows the inversion probabilities $P_i(t)$. As expected from the exchange symmetry in Eq. (6.1.4) and the asymmetric choice of initial state in Eq. (6.1.6), the flavour evolution of neutrinos $|\nu_0\rangle$ and $|\nu_1\rangle$ is the mirror image of neutrinos $|\nu_3\rangle$ and $|\nu_2\rangle$ (respectively and for the case of 4 neutrinos). This is reflected in the equivalence of the inversion probabilities for these neutrinos. Due to the presence of this symmetry, in the following sections of this chapter we will only present results for the inversion probabilities. The Figure 6.3 shows the oscillation phenomenon carried by different contribution terms in the Hamiltonian: the inversion probability caused by H_{vac} alone in Fig. 6.3a, while the oscillation caused by $H_{\nu\nu}$ alone in Fig. 6.3b. Note that the vacuum oscillation occurs separately for each neutrino and also occurs for a system in which the neutrinos are all in the same flavour state. On the other hand, the $\nu - \nu$ interaction causes the flavour oscillation due to a flavour exchange mechanism, and therefore it cannot occur if all the neutrinos are in the same flavour state. Finally, the matter interaction that causes the MSW effect cannot produce flavour oscillation by itself, but only influences the vacuum oscillation, as proved in Sec. 5.2.2.

We would like to emphasise that the number of neutrinos used in this work is of the order of

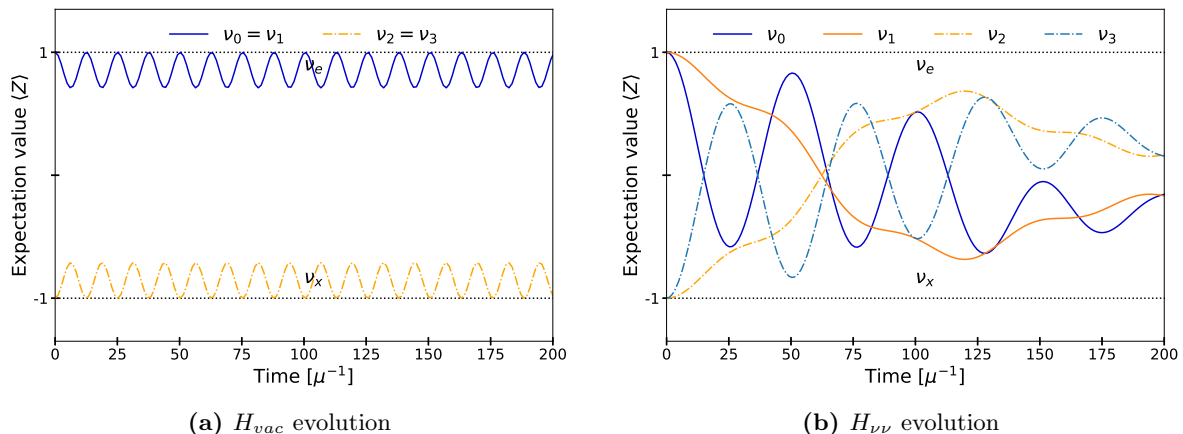


Figure 6.3: Exact evolution of the expectation value $\langle Z_i(t) \rangle$ for a system of $N = 4$ neutrinos generated by H_{vac} alone (panel (a)) and by $H_{\nu\nu}$ alone (panel (b)). The initial state is $|\varphi(0)\rangle = |0011\rangle$ also in this case.

$N \sim 10$, and improvements in quantum technology will probably allow us to simulate $N \sim 100$ in the next few years. In any case, a full quantum simulation of the flavour evolution of all 10^{58} neutrinos produced during a core-collapse supernova will always be beyond the reach of any simulation method. However, there are two main reasons to expect that a simulation of flavor evolution with a reasonable number of degrees of freedom would be valuable for understanding neutrino oscillations in supernovae. First, the simulation of a limited number of neutrinos can be useful to investigate purely quantum properties that cannot be included in mean-field approaches, such as the degree of entanglement. Previous work on this topic has clearly shown how the analysis of small-scale simulations can shed light on the conditions necessary for the emergence of collective flavor modes, as well as the role played by beyond mean-field effects (see, e.g., Refs. [58, 60, 62–64]). And second, the number of neutrinos that should be considered for a simulation is the number of neutrinos in a causally connected region of space, which may be much smaller than the whole region where the supernova explosion takes place. Furthermore, the underlying assumption that neutrinos are treated as exact plane-waves cannot be applied in a straightforward way when large regions of space are considered. A proper description that considers wave packets instead would be required [79]. We leave this extension for future work.

6.2 Trotter decomposition of unitary propagator

One way to decompose the operator in Eq. (6.1.9) is to explicitly exploit the actual interaction of the physical system. In fact, the most difficult part of the Hamiltonian in Eq. (6.1.2) is the all-to-all $\nu - \nu$ interaction term, which makes the problem nonlinear. However, this part of the interaction occurs in pairs and can therefore be implemented by considering only one pair of neutrinos (and hence qubits) at a time. The approach taken in the previous work (Ref. [67]) uses the exact pair propagator obtained in two steps. First, they symmetrize the one-body term and express the total Hamiltonian as a sum of two-body terms, namely

$$H = \sum_{i < j}^{N-1} \left(\mathbf{b} \cdot \frac{(\boldsymbol{\sigma}_i + \boldsymbol{\sigma}_j)}{N-1} + J_{ij} \boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_j \right) := \sum_{i < j}^{N-1} \hat{h}_{ij}, \quad (6.2.1)$$

where

$$\hat{h}_{ij} := \frac{h_i + h_j}{N-1} + h_{ij}. \quad (6.2.2)$$

Then they approximate the total propagator by a product of pair propagators accordingly:

$$U(t) \approx \tilde{U}^{sym}(t) := \prod_{i < j}^{N-1} e^{-i\hat{h}_{ij}t} := \prod_{i < j}^{N-1} \hat{u}_{ij}(t). \quad (6.2.3)$$

The approximation introduces an error which scales as $\sim \mathcal{O}(t^2)$ due to the non-commutativity of the symmetrized two-body terms

$$[\hat{h}_{ij}, \hat{h}_{ik}] = \frac{1}{N-1} ([h_i, h_{ik}] + [h_{ij}, h_i]) + [h_{ij}, h_{ik}] \neq 0. \quad (6.2.4)$$

A first improvement to this implementation can be made by considering that the entire one- and two-body terms commute, namely

$$[H^{(1b)}, H^{(2b)}] = 0 \quad (6.2.5)$$

in the case of a particle-independent external field (i.e. $\mathbf{b}_i = \mathbf{b}$ for all $i \in \{0, \dots, N-1\}$), and therefore their separation does not introduce any errors, namely

$$U(t) = e^{-iH^{(2b)}t} e^{-iH^{(1b)}t} := U^{(2b)}(t)U^{(1b)}(t). \quad (6.2.6)$$

At this point, the one-body contribution $U^{(1b)}(t)$ is a product of N one-body operators, since the one-body terms referring to different particles commute: $[\sigma_i, \sigma_j] = 0$ if $i \neq j$. We have:

$$U^{(1b)}(t) = e^{-iH^{(1b)}t} = e^{-i\sum_{i=0}^{N-1} h_i t} = \prod_{i=0}^{N-1} e^{-ih_i t} := \prod_{i=0}^{N-1} u_i(t), \quad (6.2.7)$$

where we have defined the single-neutrino propagator $u_i(t) = e^{-ih_i t}$ in terms of the one-body Hamiltonian h_i . Subsequently, the two-body term alone can be efficiently implemented by the pair decomposition

$$U^{(2b)}(t) \approx \tilde{U}^{(2b)}(t) := \prod_{i < j}^{N-1} e^{-ih_{ij}t} := \prod_{i < j}^{N-1} u_{ij}(t), \quad (6.2.8)$$

where we have defined the single-pair propagator $u_{ij}(t) = e^{-ih_{ij}t}$ of the neutrinos ν_i and ν_j . Again, this approximation introduces an error which scales as $\sim \mathcal{O}(t^2)$ but with a smaller prefactor, and the approximated propagator is now

$$U(t) \approx \tilde{U}(t) := \tilde{U}^{(2b)}(t)U^{(1b)}(t). \quad (6.2.9)$$

The implementation in Eq. (6.2.3) not only fails to exploit the commutativity of the two terms (Eq. (6.2.5)), but may also lead to an increase in the error due to the lack of commutativity between individual one- and two-body contributions (terms $[h_i, h_{ij}] \neq 0$ in Eq. (6.2.4)), and may lead to an explicit breaking of symmetry under particle exchange. This property will be analysed in detail in Sec. 6.2.2. However, the approach can still be useful in the case of a particle-dependent external field, where $\mathbf{b} \rightarrow \mathbf{b}_i$, as occurs when different neutrinos have different energies $E \rightarrow E_i$ (a necessary ingredient to observe spectral splits [58, 72]). In this case we can separate the external field of each neutrino into a mean part and a variation part, namely $\mathbf{b}_i := \bar{\mathbf{b}} + \delta\mathbf{b}_i$. At this point we can separate out the one-body term associated with the common external field

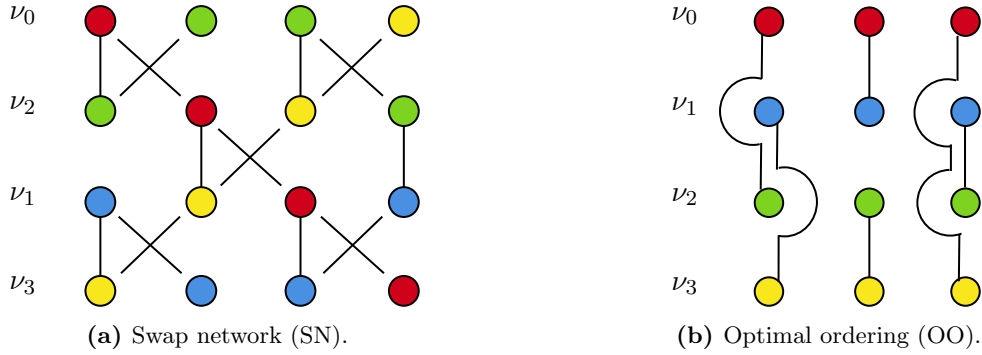


Figure 6.4: Panel (a) shows the swap network (SN) scheme for $N = 4$ qubits, which implements the two-body part $\tilde{U}^{(2b)}(t)$ using a chain of linearly connected qubits. Vertical lines represent the qubit interaction and crosses the SWAP operations. Panel (b) shows the network for the implementation of the two-body part using a system of fully connected qubits. This scheme implements the optimal pair order (OO) and minimises the decomposition error caused by the non-commutativity property.

$\bar{\mathbf{b}}$, which commutes with everything else, and use the exact pair approximation in Eq. (6.2.3) only for the variation part $\delta\mathbf{b}_i$. This approach is equivalent to using the Hamiltonian:

$$\begin{aligned}
 H &= \sum_{i=0}^{N-1} \bar{\mathbf{b}} \cdot \boldsymbol{\sigma}_i + \sum_{i<j}^{N-1} \left(\frac{(\delta\mathbf{b}_i \cdot \boldsymbol{\sigma}_i + \delta\mathbf{b}_j \cdot \boldsymbol{\sigma}_j)}{N-1} + J_{ij} \boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_j \right) \\
 &:= \sum_{i=0}^{N-1} h_i + \sum_{i<j}^{N-1} \hat{h}_{ij},
 \end{aligned} \tag{6.2.10}$$

where \hat{h}_{ij} is the two-body Hamiltonian that includes the $\nu - \nu$ interaction and the one-body interaction term related to the energy differences.

6.2.1 Pair order and qubit topology

Once we have encoded each neutrino state into a qubit and interpreted the $U^{(2b)}(t)$ propagator as a product of pair interactions, we need to consider the effect of the qubit topology, which determines the ability of a quantum machine to make two qubits interact. The Hamiltonian in Eq. (6.1.2) contains an all-to-all interaction term $H^{(2b)}$, which means that all neutrinos interact with all others during evolution. This means that in order to implement the total interaction between the N neutrinos of the system, we have to implement all $\binom{N}{2} = N(N-1)/2$ pair interactions, and in terms of the quantum algorithm, the implementation of $\tilde{U}^{(2b)}(t)$ in Eq. (6.2.8) requires that all N qubits interact with all others at least once during the simulation. In quantum computing, this means that the algorithm design must be adapted to the particular topology of the quantum device used for the simulation. As shown in Ref. [67], it is possible to implement the two-body propagator using a linearly connected chain of qubits with N layers using the so-called *swap network* (SN) (the same scheme was later adopted for tensor network simulations in Ref. [63]). The algorithm consists of applying the $u_{ij}(t)$ propagator to a pair of qubits, followed by a SWAP gate that exchanges the qubit states. The new unitary $w_{ij}(t)$ is thus

$$w_{ij}(t) := \text{SWAP} \cdot u_{ij}(t), \tag{6.2.11}$$

where in the computational basis the SWAP unitary is represented by the matrix in Eq. (2.1.13) of Chapter 2. For example, the swap network required for the case $N = 4$ is shown in Fig. 6.4a, where the vertical lines represent the interaction terms $u_{ij}(t)$ and the crosses represent the SWAP

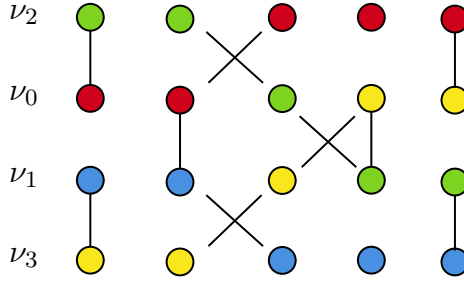


Figure 6.5: Optimal ordering for $N = 4$ implemented in a linearly connected chain of qubits, showing that five layers are needed to implement $U^{(2b)}(t)$ as a product of pairs ordered according to the OO scheme and using only unitaries as in Eq. (6.2.11).

operations. The structure of the swap network, dictated by the available qubit topology, imposes a constraint on the possible ordering of the pair propagators. Once the network structure is fixed, the order can be chosen by varying the initial encoding of each neutrino in the qubits. For our case, we found that the ordering $(q_0, q_1, q_2, q_3) = (\nu_0, \nu_2, \nu_1, \nu_3)$ (the same as in Ref. [67]) is the best. In principle, any order could be achieved by adding additional SWAP gates, but at the cost of increasing the complexity of the scheme. For our Hamiltonian and in the case of $N = 4$ we found that the best interaction order would be that described by the network in Fig. 6.4b for the encoding map $(q_0, q_1, q_2, q_3) = (\nu_0, \nu_1, \nu_2, \nu_3)$. This best decomposition order cannot be expressed as a swap network of only four layers when restricted to linear qubit connectivity, but at least five layers would be needed, as explicitly shown in Fig. 6.5. However, with all-to-all connectivity, this algorithm can be implemented in one layer less than the swap network, and each layer is full in the sense that we perform the maximum number of possible operations simultaneously. As we will see in the next sections, this type of qubit topology allows to reduce the theoretical Trotter error of the unitary implementation (Sec. 6.2.2) to obtain more accurate and longer simulations, and to reduce the complexity of the quantum circuit (Sec. 6.3).

6.2.2 Pair order and Trotter error

Importantly, the error introduced by the approximation in Eq. (6.2.8) also depends on the order in which the pairs interact, since the error is given by a sum of the commutators $[h_{ij}, h_{kl}]$ with a given order³. To see this, consider that for any number N of neutrinos, we can always group them into triplets (ν_i, ν_j, ν_k) to which we assign the two-body commutators $[h_{ij}, h_{ik}]$, $[h_{ij}, h_{jk}]$ and $[h_{ik}, h_{jk}]$ which are identified by $2^N \times 2^N$ matrices with the same non-trivial elements and with magnitudes depending on J_{ij} , J_{ik} and J_{jk} . An optimal order allows to maximise the cancellations between the commutators while minimising the decomposition error, and this is exactly what happens in the OO scheme of Fig. 6.4b. For example, the Figure 6.6 shows the three commutators associated with the triplet (ν_0, ν_1, ν_2) in the case of $N = 4$ neutrinos. If we decompose $e^{-i(h_{01}+h_{12}+h_{02})t} \approx e^{-ih_{01}t}e^{-ih_{12}t}e^{-ih_{02}t}$, we minimise the error due to the fact that the first commutator $[h_{01}, h_{02}]$ and the last one $[h_{12}, h_{02}]$ in the figure are opposite in magnitude and cancel each other out. Determining the optimal ordering for large systems is not feasible in general as this would require a super-exponential cost in the system size N . There are, in fact, $\binom{N}{2}$ pairs and $\binom{N}{2}!$ possible orderings in total, which is an upperbound because lots of them are overcounted since we can swap neighboring pairs in a list if the indices do not match (ie. the two-body terms commute). However the possible orders that one should test is still super-exponential in N , and for large systems a randomization procedure for the order could prove valuable to control the error [80, 81].

³The approximation $e^{A+B+C} \approx e^A e^B e^C$ introduces an error of $[A, B] + [A, C] + [B, C]$ at first order.

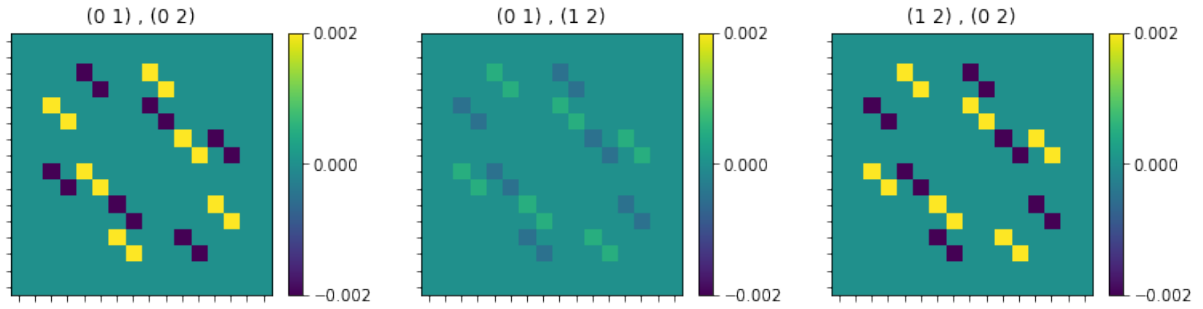


Figure 6.6: Commutator matrices of the triplet of neutrinos ν_0, ν_1 and ν_2 of a system of $N = 4$ particles. From left to right: $[h_{01}, h_{02}]$, $[h_{01}, h_{12}]$ and $[h_{12}, h_{02}]$.

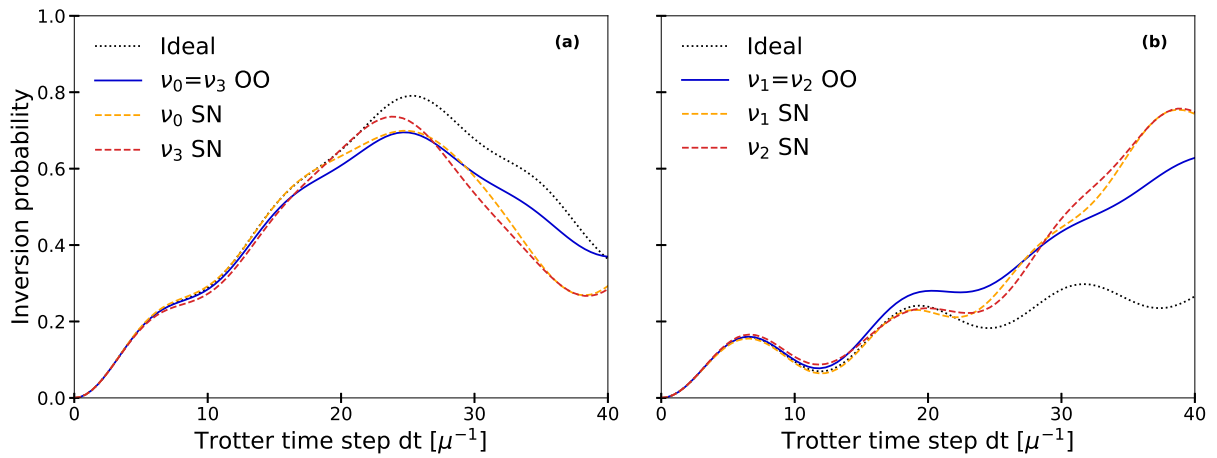


Figure 6.7: The two figures show the inversion probability $P_i(dt)$ after single Trotter step evolutions for different time steps $dt \in [0, 40] \mu^{-1}$ of a system with $N = 4$ neutrinos. Panel (a) shows the evolution for neutrinos ν_0 and ν_3 , and panel (b) for ν_1 and ν_2 . Dotted black lines are the ideal perfect evolutions obtained by applying e^{-iHdt} to the initial state $|\psi(0)\rangle = |0011\rangle$ (as in the first time interval in the bottom panel of Fig. 6.2a). The solid blue lines are the evolution obtained using the implementation proposed in this work (Eqs. (6.2.6) - (6.2.8)) and the scheme in Fig. 6.4b. The dashed orange and red lines are the evolution using the implementation proposed in Ref. [67] and described by Eqs. (6.2.1) - (6.2.3) and scheme 6.4a.

The two panels in Fig. 6.7 show the inversion probability, for the 4 neutrinos system, for a single Trotter step evolution using times $dt \in [0, 40] \mu^{-1}$ for different implementations of the propagator: (1) the exact one $U(dt)$, (2) the one proposed in the previous work (Ref. [67]), and (3) the optimal one implemented in the present work. The ideal data were obtained by applying the exact propagator $U(dt) = e^{-iHdt}$ to $|\varphi(0)\rangle = |0011\rangle$ through 16×16 matrix multiplications. The data corresponding to the SN model used in Ref. [67] were obtained by applying to $|\varphi(0)\rangle$ the exact pairwise propagator in Eq. (6.2.3) and following the swap network ordering in Fig. 6.4a. While the data denoted by OO are obtained using the approach proposed in this work, which uses the separation of the one- and two-body parts as in Eq. (6.2.9) and the best pair ordering as shown in Fig. 6.4b for the two-body part. The result highlights that the implementation proposed here always preserves the exchange symmetry of Eq. (6.1.4). In this case, this feature is given by the separation of the one- and two-body propagators $U^{(1b)}(dt)$ and $U^{(2b)}(dt)$, but the relationship between optimal order and symmetry preservation is actually more complicated. In fact, using the OO scheme in Fig. 6.4b the evolution preserves symmetry even without the one- and two-body separation, while using the SN scheme in Fig. 6.4a the separation is necessary to

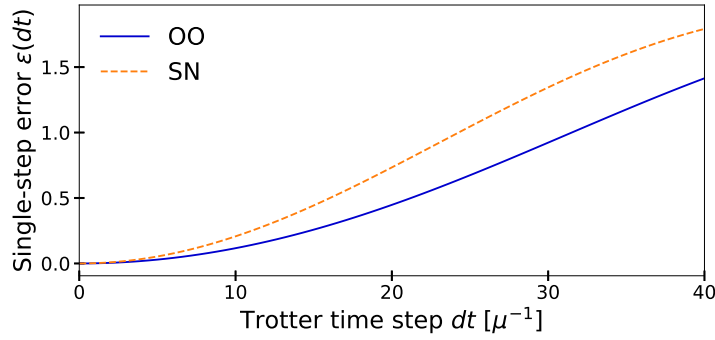


Figure 6.8: The figure shows the infinite norm distance between the ideal propagator and the pair approximated one in function of Trotter step dt for the best possible swap network (dashdot orange line) and the optimal ordering (solid blue line).

restore symmetry. This is due to the fact that when using the OO and the exact propagator in pairs (Eq. (6.2.3)) the spurious commutators of type $[h_i, h_{ij}]$ cancel each other out.

The Fig. 6.7 seem to suggest that the SN best approximates evolution for time steps less than $dt < 30 \mu^{-1}$. This is mostly an effect of choosing a particular initial state $|\varphi(0)\rangle = |0011\rangle$ and a particular observable $P_i(t)$. In Fig. 6.8 we show the error of the effective approximated propagator $\tilde{U}(dt)$, as a function of the Trotter step dt , in the cases of the two implementations: $\tilde{U}(dt) = \tilde{U}^{sym}(dt)$ as in Eq. (6.2.3) and with SN order, and $\tilde{U}(dt) = \tilde{U}^{(2b)}(dt)U^{(1b)}(dt)$ as in Eqs. (6.2.7) - (6.2.8) and with the OO scheme. We calculate the error using the infinite (spectral) norm distance

$$\varepsilon(dt) = \left\| \tilde{U}(dt) - U(dt) \right\|_{\infty}, \quad (6.2.12)$$

which is the largest error we can have for the eigenstates associated with the largest eigenvalue. For general initial states and observables, the error displayed in Fig. 6.8 shows that the OO approximation indeed has the smallest worst case error for all time steps. We note that the Trotter error bound scales with $\mathcal{O}(dt^2)$, but Fig. 6.8 shows that the error actually scales sub-quadratically with dt . The deviation from the quadratic shown in the figure is most likely only due to the fact that the maximum error in the spectral norm is bounded as $\varepsilon \leq 2$, since we have a difference of unitary operators.

The usual way to reduce the Trotter error is to set a sufficiently small value for the time step dt and to iteratively apply the approximated propagator, namely

$$|\psi(kdt)\rangle = \tilde{U}(dt)^k |\psi(0)\rangle. \quad (6.2.13)$$

In particular, if we want to evolve the state up to $T = rdt$, the error will scale as $\mathcal{O}(T^2) = \mathcal{O}(r^2 dt^2)$ if we apply $U(T)$ once, and will instead scale as $\mathcal{O}(rdt^2)$ using r steps in which we apply $U(dt)$. The reduction of the Trotter error is particularly advantageous when the aim of the simulation is to represent the evolution of the system over a long total time T , during which the accumulation of errors predominates. To show the consistency of the error accumulation, we plot in Fig. 6.9 the evolution of the 4-neutrinos system with $dt = 16 \mu^{-1}$ up to $T = 1200 \mu^{-1}$ using the two approaches OO and SN. In the first case, the multiple Trotter step evolution is obtained as follows

$$|\psi(kdt)\rangle^{(OO)} = (\tilde{U}^{(2b)}(dt))^k (U^{(1b)}(dt))^k |\psi(0)\rangle, \quad (6.2.14)$$

while in the second case we used

$$|\psi(kdt)\rangle^{(SN)} = \tilde{U}^{sym}(dt)^k |\psi(0)\rangle. \quad (6.2.15)$$

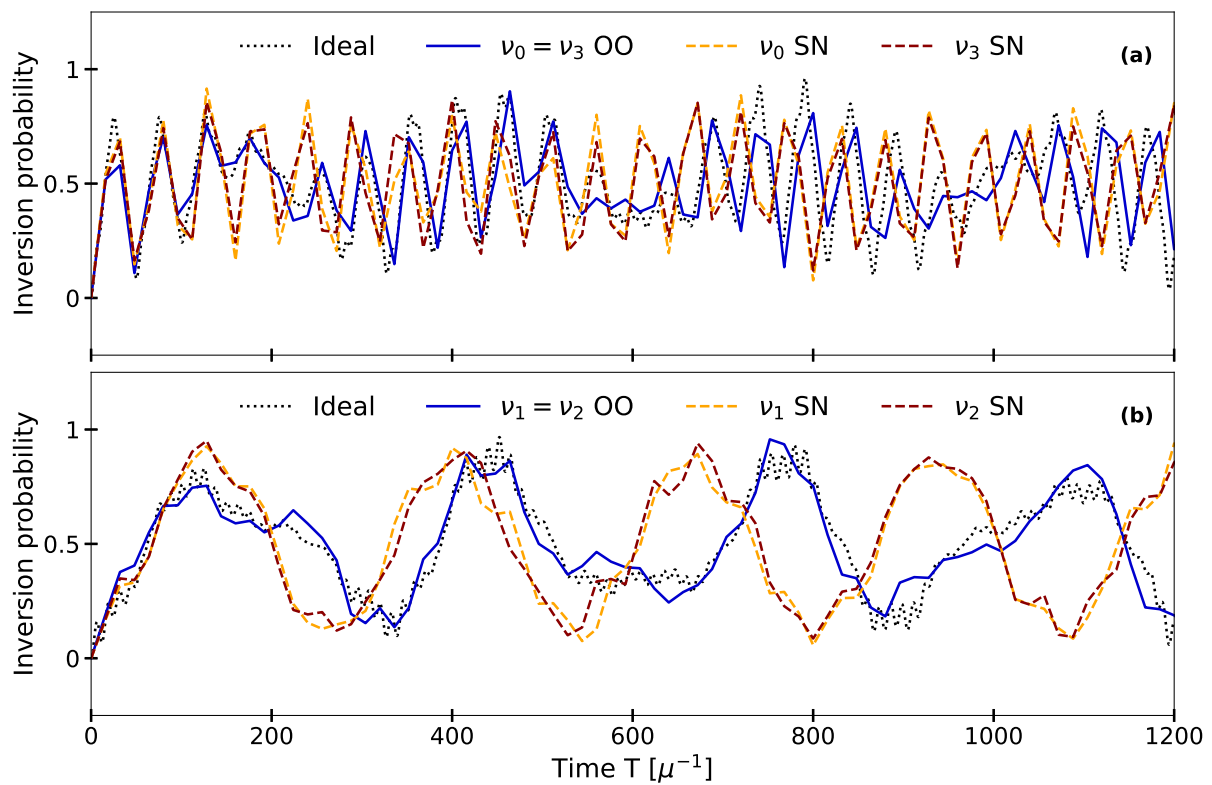


Figure 6.9: Long-time evolution of a system composed of $N = 4$ neutrinos. Panel (a) for neutrinos ν_0 and ν_3 and panel (b) for ν_1 and ν_2 . Dotted black lines represent the ideal perfect evolution. The solid blue lines are the evolution obtained by applying Eq. (6.2.14) and the OO scheme, and the dashed orange and red lines are the evolution using Eq. (6.2.15) and the SN scheme. Both for the same time step $dt = 16 \mu^{-1}$.

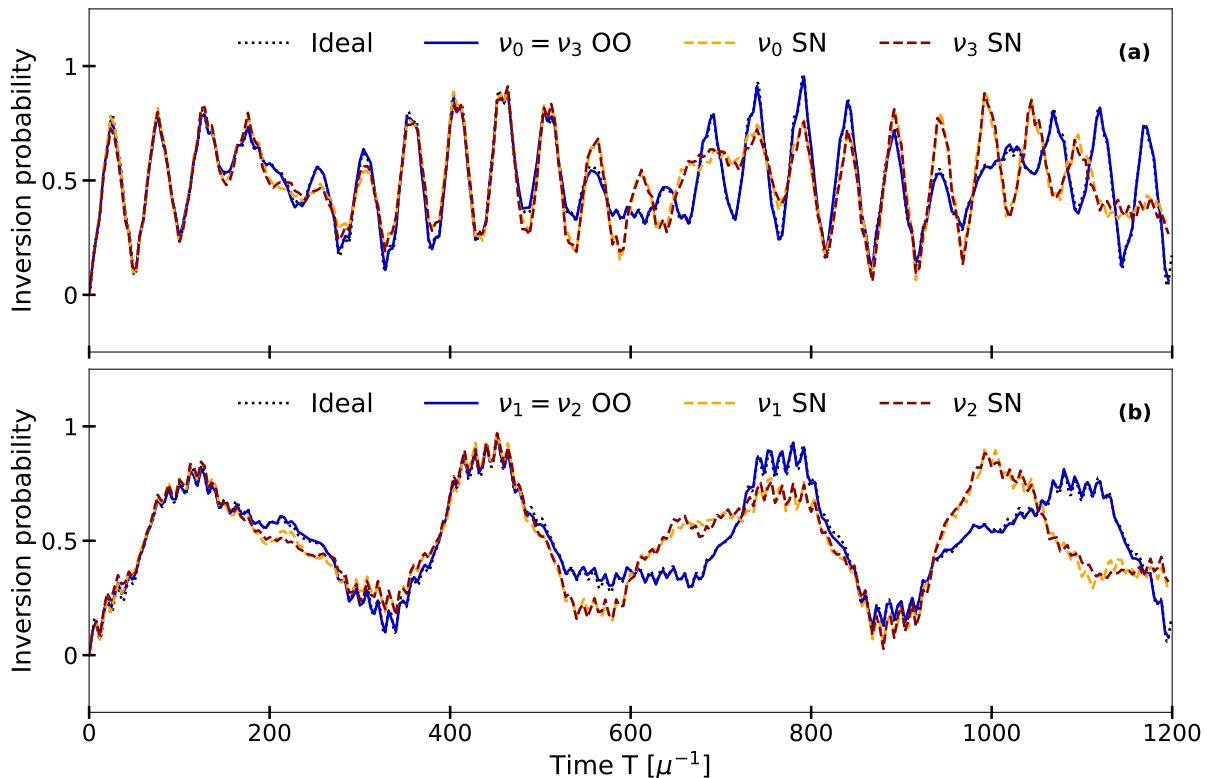


Figure 6.10: Long time evolution of a system composed of $N = 4$ neutrinos using $dt = 4\mu^{-1}$. The lines correspond to the different implementations and follow the same convention of Fig. 6.9. Here the ideal data, shown as a black dotted line, is almost overwritten by the blue solid line.

We note that by using the OO scheme instead of the SN one, we can fix a larger value of dt and still obtain accurate results for longer time evolution, since we reduce the implementation error $\varepsilon(dt)$ (see Fig. 6.8) and consequently its accumulation. With a sufficiently small time step the two implementations are almost equal at the beginning of the evolution, but eventually the accumulation of error becomes relevant again. The evolution with $dt = 4\mu^{-1}$ for the two implementations compared to the ideal one is shown in Fig. 6.10. To compare the error induced by the two implementations, we have plotted in Fig. 6.11 the accumulated error calculated using

$$\varepsilon(T) = \left\| \tilde{U}(dt)^k - U(T) \right\| \quad (6.2.16)$$

where $T = kdt$, $k \in \{0, \dots, 10\}$ and $dt = 4\mu^{-1}$, and we compare it with the linear bound

$$\varepsilon(T) \leq k\varepsilon(dt). \quad (6.2.17)$$

6.3 Optimal quantum gate decomposition

In the previous sections we have seen the Trotter approximation used in the development of this quantum algorithm, and we have analysed its connection with the qubit topology and with the theoretical simulation error. We now specialise the algorithm in the case of a fully connected quantum machine performing the quantum gate implementation of the algorithm. We show quantum circuits for the case of $N = 4$ and its extension for $N = 8$, or other values of N , can be

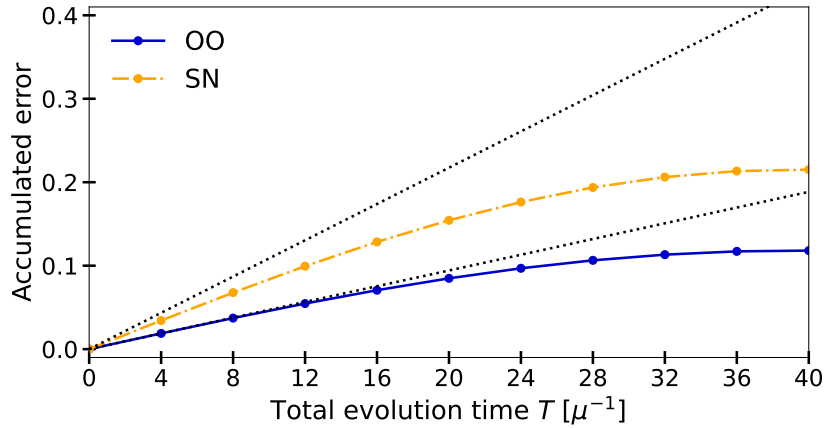


Figure 6.11: Accumulated error for the two implementations OO and SN, calculated as in Eq. (6.2.16).

done in a similar way. In terms of quantum circuits, the commutation property in Eq. (6.2.5) is

$$U(dt) = U^{(1b)}(dt) U^{(2b)}(dt) \quad (6.3.1)$$

At this point, we can implement the one-body propagator in Eq. (6.2.7) as a tensor product of single-qubit gates

$$U^{(1b)}(dt) = \bigotimes_{i=0}^{N-1} u_i(dt), \quad (6.3.2)$$

since it is a product of operators that can be implemented by layers that act non-trivially on only a single neutrino (e.g. $\mathbb{I} \otimes u_1(dt) \otimes \mathbb{I} \otimes \mathbb{I}$ act only on ν_1) and they also commute between each other, namely $[u_i(t), u_j(t)] = 0$. So, in terms of a quantum circuit, we have

$$U^{(1b)}(dt) = u_0(dt) u_1(dt) u_2(dt) u_3(dt), \quad (6.3.3)$$

and each $u_i(t)$ can be implemented with at most three elementary rotations according to the zyz decomposition presented in Sec. 2.3.1 of the Chapter 2. The two-body propagator, according to the approximation in Eq. (6.2.8), becomes a product of $\binom{N}{2}$ terms, which act non-trivially on two qubits. In addition, using the OO scheme in Fig. 6.4b, we can run two terms simultaneously in each layer, giving a total of $N - 1 = 3$ layers for the case of four neutrinos. Following the optimal order, the pair decomposition allows us to implement the $U(t)$ propagator using the quantum circuit in Fig. 6.12. The first four blocks represent single-qubit gates implementing $u_i(dt)$, while the following six wider blocks represent the two-qubit gates implementing the terms $u_{ij}(dt)$. The pair decomposition procedure can be extended to any value of N . For example, in the case of $N = 8$ we can use the quantum circuit shown in Fig. 6.13⁴, which again implements the propagator $\tilde{U}^{(2b)}(dt)$ with $N - 1$ layers.

⁴This network is an improvement on the one used in Ref. [78].

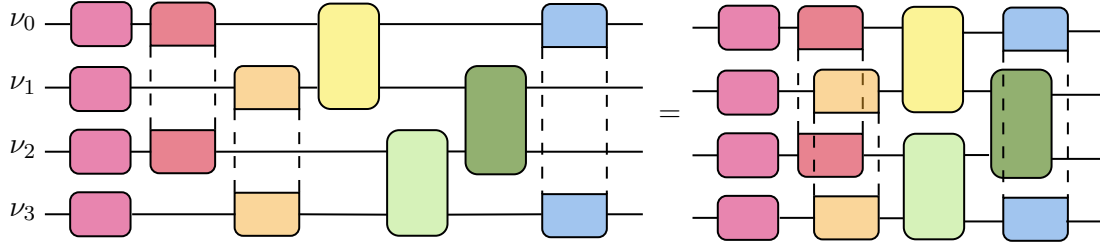


Figure 6.12: Scheme of the quantum circuit implementing the propagator $\tilde{U}(dt)$ after the one- and two-body separation in (6.3.1), the implementation of $U^{(1b)}(dt)$ in (6.3.3) and the pair decomposition in Eq. (6.2.8) with the OO scheme in Fig. 6.4b corresponding to the pair order $P^{(4)} = (02, 13, 01, 23, 12, 03)$. The right-hand side highlights the possibility of performing simultaneous interaction terms within a layer, as they involve different qubits. The total number of layers is 3 and each layer is full.

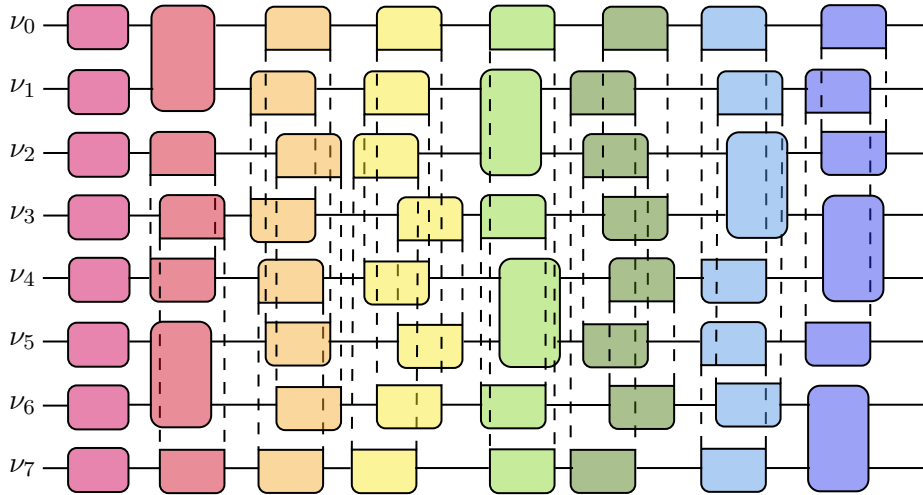


Figure 6.13: Scheme of the quantum circuit implementing the propagator $\tilde{U}(dt)$, after one- and two-body separation in Eq. (6.3.1), for the case of $N = 8$ neutrinos. The pair-order decomposition is $P^{(8)} = (01, 24, 37, 56, 05, 13, 26, 47, 06, 14, 27, 35, 07, 12, 36, 45, 03, 17, 25, 46, 04, 16, 23, 57, 02, 15, 34, 67)$, which again corresponds to $N - 1 = 7$ layers.

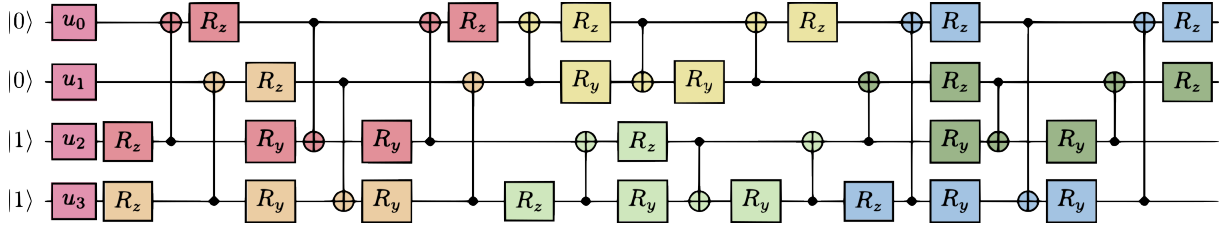


Figure 6.14: Optimal CNOT-based quantum circuit implementing $\tilde{U}(dt) = \tilde{U}^{(2b)}(dt)U^{(1b)}(dt)$ for $N = 4$. Different colors, as in the Fig. 6.12, represent different pair interaction terms.

At this point we need to implement each pair interaction $u_{ij}(dt)$ as a sequence of one- and two-qubit gates taken from a library. The optimal way to do this is to consider the magic basis decomposition explained in Sec. 2.4.2 of Chapter 2. In fact, we can see that the pair interaction terms are of the form

$$u_{ij}(dt) = e^{-iJ_{ij}\sigma_i \cdot \sigma_j dt} = e^{-idtJ_{ij}(X_i \otimes X_j + Y_j \otimes Y_j + Z_i \otimes Z_j)}, \quad (6.3.4)$$

which is a special case of the gate $\mathcal{N}(\alpha, \beta, \gamma)$ in Eq. (2.4.9), where $\alpha = \beta = \gamma = -dtJ_{ij}$. Therefore, we can implement each term using the optimal CNOT-based quantum circuit in Eq. (2.4.29), i.e. for our parameters:

$$u_{ij}(dt) = \text{Circuit with } R_z(2\alpha - \frac{\pi}{2}), R_z(\frac{\pi}{2}), R_y(\frac{\pi}{2} - 2\alpha), R_y(2\alpha - \frac{\pi}{2}), R_z(-\frac{\pi}{2}), \text{ and CNOT gates.} \quad (6.3.5)$$

up to a global phase of $e^{i\pi/4}$. Note that the circuit in Eq. (6.3.5) is symmetric under the left-to-right exchange⁵, and also under qubit inversion⁶. The full optimal CNOT-based quantum circuit, evolving a system of $N = 4$ neutrinos by a single application of $U(dt)$, is shown in Fig. 6.14, where we have deleted and merged the maximum number of gates and omitted the rotation angles. Note that full connectivity allows to reduce the number of single qubit rotations, because the operation $w_{ij}(dt)$ in Eq. (6.2.11) is not in the useful form as in Eq. (6.3.4). The implementation of $w_{ij}(dt)$ requires, a priori, three CNOTs and 15 elementary rotations as proved for example in Ref. [18] (see circuit in Eq. (2.4.30)). Practically the authors of the paper in Ref. [67] used the *Qiskit* compiler, which implements the $w_{ij}(dt) \in \mathcal{U}(4)$ with a quantum circuit containing 8 general single-qubit rotations $\in \mathcal{SU}(2)$, represented by white boxes in the following quantum circuit

$$\text{Circuit with two qubits and four CNOT gates.} \quad (6.3.8)$$

⁵Quantum circuit symmetry under left-to-right inversion

$$u_{ij}(dt) = \text{Circuit with } R_z(-\frac{\pi}{2}), R_y(2\alpha - \frac{\pi}{2}), R_y(\frac{\pi}{2} - 2\alpha), R_z(\frac{\pi}{2}), \text{ and CNOT gates.} \quad (6.3.6)$$

⁶Quantum circuit symmetry under qubit inversion

$$u_{ij}(dt) = \text{Circuit with } R_z(\frac{\pi}{2}), R_y(\frac{\pi}{2} - 2\alpha), R_y(2\alpha - \frac{\pi}{2}), R_z(-\frac{\pi}{2}), \text{ and CNOT gates.} \quad (6.3.7)$$

Expressing each general rotation in terms of elementary rotations (i.e. using the zyz decomposition) it results in 24 elementary rotations instead of 15. A very good improvement of the swap network algorithm is to use the circuit in Eq. (2.4.28) derived in Chapter 2, which reduces the number of single qubit rotations to 4, coinciding with only 5 elementary rotations. Each pair interaction $u_{ij}(dt)$ followed by the SWAP gate can be in fact implemented by the circuit

$$(6.3.9)$$

This result has never been used in any other work and could provide a significant improvement in quantum simulation performed on quantum testbeds with linear topology.

In order to evolve the initial state up to a long time T , we can first split the evolution into a sequence of Trotter steps as in Eq. (6.2.13), which, in terms of quantum circuits, is as follows

$$(6.3.10)$$

and then implement each propagator $U(dt) \approx \tilde{U}(dt)$ in the same way as described above. First, we split the one-body and two-body parts, which reads

$$(6.3.11)$$

and then we use the commutativity between them to get

$$(6.3.12)$$

which means that we can exactly implement the one-body evolution for the whole evolution time T , since $U^{(1b)}(T) = (U^{(1b)}(dt))^k$ for $T = kdt$. So

$$(6.3.13)$$

Finally, we introduce the only error contribution in the implementation, i.e. the two-body evolution approximated as a product of pair interactions. The implementation of the multiple Trotter step according to the procedure just described is shown in Fig. 6.15.

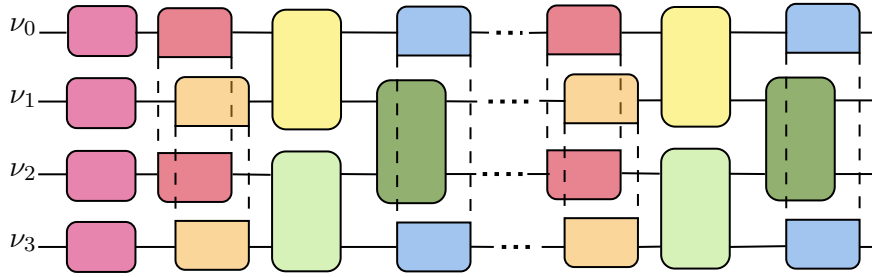


Figure 6.15: Quantum circuit representation of the multiple Trotter steps evolution exploiting the properties in Eqs. (6.3.11), (6.3.12) and (6.3.13), and the OO scheme for the pair decomposition.

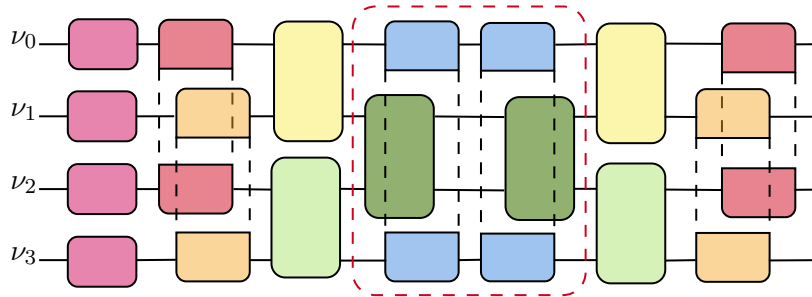


Figure 6.16: Quantum circuit representation of the multiple Trotter steps evolution exploiting the properties in Eqs. (6.3.11), (6.3.12) and (6.3.13) and using inverse order for the OO pair decomposition scheme for the case of $r = 2$ total steps. The four general two-qubit interactions $u_{03}(dt) \cdot u_{12}(dt) \cdot u_{03}(dt) \cdot u_{12}(dt)$ inside the red dashed square, can be merged and be performed together in a single layer, namely $u_{03}(2dt) \cdot u_{12}(2dt)$.

The total number of operations for the circuit can be further optimised by using an inverted order of interactions (which is also an equally optimal order) in alternate steps, as shown in Fig. 6.16 for two steps. This means that the last interactions $u_{12}(dt) \cdot u_{03}(dt)$ are performed at the beginning for steps indicated by even values of k . In this case, the total number of gates for the circuit results in a reduction of $N/2$ general two-qubit gates $u_{ij}(dt)$ for each step, since the last layer of the k -th step and the first layer of the $(k + 1)$ -th step can be combined and implemented in a single layer. This alternating scheme can also reduce the overall approximation error since it then becomes equivalent to a second order Trotter step with time step $2dt$ (see Sec. 6.6 for additional details).

6.4 Machine aware compilation

When performing a quantum simulation, it is always advisable to write and optimise the circuit using the physical gates actually implemented by the machine. This approach, as already mentioned, allows us to reduce the number of gates needed for the simulation and to keep under control the actual quantum circuit that the machine will execute.

We have tested the quantum algorithm presented in this chapter using the Quantinuum System Model (QSM) H1-2, powered by Honeywell, which is a trapped-ion device similar to the machine described in Ref. [82] and in Chapter 3. Each trapped-ion quantum machine, although based on the same type of technology, can use a different set of gates as long as it provides the universality feature. The compilation of a certain quantum circuit should always be carried out using the library provided in order to (1) have control over the transformation that is actually implemented by the machine and (2) ensure that the circuit is as short as possible

and therefore optimal. The basic concepts of the quantum gate compilation for a trapped-ion quantum machine can be found, for example, in Ref. [83] for a different library.

6.4.1 Quantinuum system model

Like other trapped-ion-based quantum devices, the QSM testbed offers the advantage of full connectivity between the qubits. The qubit system is defined by the stable orbitals of the valence electron of the $^{171}\text{Yb}^+$ atom, which are controlled by a laser beam. The two available machines (H1-1 and H1-2) allow a high fidelity gate implementation with a single-qubit gate error of $\varepsilon_q \sim 10^{-4}$ and a two-qubit gate error of $\varepsilon_{qq} \sim 10^{-3}$. Single and two-qubit gates are implemented in assignment zones, which rearrange the physical position of the ions. This means that any pair of qubits can be moved into the action zone and made to interact. This architecture thus creates an all-to-all connectivity hardware. Furthermore, because there are multiple interaction zones, multiple quantum operations can be performed in parallel. At the time we tested our algorithm, the machine had three interaction zones available (now the H1-1 machine has five interaction zones), meaning that a maximum of three pairs of qubits could be brought into contact in the same layer.

The native single-qubit gates used are

$$R'_z(\lambda) = \begin{pmatrix} e^{-i\lambda/2} & 0 \\ 0 & e^{i\lambda/2} \end{pmatrix}, \quad (6.4.1)$$

where we have defined the z rotation with R'_z to distinguish it from the z rotation defined in Eq. (2.1.5) of Chapter 2, which uses a different convention (the relation is $R_z(\alpha) = R'_z(-\alpha)$), and

$$U_q(\theta, \phi) = \begin{pmatrix} \cos \theta/2 & -ie^{-i\phi} \sin \theta/2 \\ -ie^{i\phi} \sin \theta/2 & \cos \theta/2 \end{pmatrix}, \quad (6.4.2)$$

which, depending on the value of θ and ϕ , can generate R_x and R_y rotations. In most trapped-ion-based quantum machines, such as the Quantinuum, the R_z rotations are implemented virtually. This means that the testbed only stores the phase and these operations take zero time and cause no error. The two-qubit gate is

$$\text{ZZ} = e^{(-i\frac{\pi}{4}Z \otimes Z)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (6.4.3)$$

up to a global phase of $e^{-i\pi/4}$, and its parametric version

$$R_{zz}(\theta) = e^{(-i\frac{\theta}{2}Z \otimes Z)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\theta} & 0 & 0 \\ 0 & 0 & e^{i\theta} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (6.4.4)$$

is also recently available. Actually, the QSM compiler accepts the R_x and R_y commands as well as the CNOT gate, but translating all the operations into the native language could help the user to further optimise the quantum circuit. This will be discussed in Sec. 6.5.4. Some properties of the above one- and two-qubit operations are listed below and used in the next section to translate and optimise our quantum algorithm for this particular quantum machine.

- (i) The Pauli matrices can be implemented with $X = U_q(\pi, 0)$, $Y = U_q(\pi, \pi/2)$, $Z = R'_z(\pi)$, and the Hadamard gate is $H = R'_z(\pi)U_q(\pi/2, -\pi/2)$.

(ii) The other non-native elementary rotations correspond to

$$R_y(\theta) = U_q\left(\theta, -\frac{\pi}{2}\right) = U_q\left(-\theta, \frac{\pi}{2}\right), \quad R_x(\theta) = U_q(\theta, \pi) = U_q(\theta, -\pi). \quad (6.4.5)$$

(iii) The adjoint of $U_q(\theta, \phi)$ can be obtained by changing only the first angle

$$U_q^\dagger(\theta, \phi) = U_q(-\theta, \phi). \quad (6.4.6)$$

This means that $U_q(\theta, \phi)U_q(-\theta, \phi) = \mathbb{I}$, but the same property does not hold for the second angle.

(iv) Two sequential $U_q(\theta, \phi)$ can be easily merged if $\phi_1 = \phi_2 := \phi$, namely

$$U_q(\theta_2, \phi) \cdot U_q(\theta_1, \phi) = U_q(\theta_1 + \theta_2, \phi). \quad (6.4.7)$$

(v) At the time of this research, only two possible values $\theta \in \{\pi, \pi/2\}$ for $U_q(\theta, \phi)$ were natively available on the device. This means that up to three physical gates were needed to compile a general unitary $\in \mathcal{SU}(2)$ as follows

$$U_q(\theta, \phi) = U_q\left(\frac{\pi}{2}, \phi + \frac{\pi}{2}\right) R'_z(\theta) U_q\left(\frac{\pi}{2}, \phi - \frac{\pi}{2}\right). \quad (6.4.8)$$

(vi) If the quantum circuit is to be assembled using the *Qiskit* package, the following relationships must be taken into account

$$U_q(\theta, \phi) = u\left(\theta, \phi - \frac{\pi}{2}, \frac{\pi}{2} - \phi\right) \quad (6.4.9)$$

where $u(\theta, \phi, \lambda) \in \mathcal{SU}(2)$ is the general single-qubit gate supported by *Qiskit*, and

$$ZZ = R_{zz}(\pi/2) \quad (6.4.10)$$

up to a global phase.

(vii) The R_z rotation commute with the two-qubit gate ZZ .

(viii) Note that, unlike the case of CNOT, the ZZ gate is not self-adjoint, i.e. two consecutive ZZ gates cannot be eliminated, but they give a separable gate, namely $ZZ \cdot ZZ = Z \otimes Z$ up to a global phase.

The cost of a submitted quantum circuit is measured in Honeywell Quantum Credits (HQC) and depends on the number of single-qubit rotations N_q , the number of two-qubit rotations N_{qq} , the number of state preparations and measurements N_m , and the number of shots M used to repeat the circuit through the relation

$$\text{HQC} = 5 + \frac{N_q + 10N_{qq} + 5N_m}{5000} M. \quad (6.4.11)$$

6.4.2 Optimal quantum circuit

The QSM compiler also works for non-native quantum gates such as CNOT, R_x and R_y operations, but a machine-aware translation of the circuit allows it to be further optimised by choosing the best translation of each operation to maximise the number of merges and cancellations. To

circuit using the *Open Qasm* language as a *job*. The *job* returns the results in the form of a dictionary in which the element called *results* contains a sub-item called *c* where the count of each output is expressed as a number associated to a string of n bits. One can convert the result given by the machine using a simple function that translates the dictionary into a histogram and then into a probability for each qubit to be measured in a given state $\in \{|0\rangle, |1\rangle\}$.

We perform two different simulations: first we approximate the full time evolution with a single Trotter step (see Eq. (6.3.1)) with different values of dt , as done in Ref. [67], and then we perform a multi-step simulation (as in Eq. (6.3.13)) with $dt = 4\mu^{-1}$. The results of the simulation are reported as inversion probabilities for each neutrinos to reproduce the evolution in Fig. 6.7 (blue line) for the first type of simulation and the evolution in Fig. 6.10 (blue line) for the second type of simulation. The results are reported with an error bar representing the statistical error calculated as in the next section.

6.5.1 Statistical error

For each simulation, the circuit is repeated $M = 200$ times to collect statistics on the measurement results. The results of the simulation are then analysed by calculating statistical confidence intervals using the Bayesian approach already used in Ref. [67]. Assuming that we measure M times a single qubit $|q\rangle$ and obtain m times the state $|1\rangle$, the probability distribution of obtaining m results against the M measurements is given by the binomial distribution of the probability p we sample:

$$\mathcal{P}_b(m|p) = \binom{M}{m} p^m (1-p)^{M-m}. \quad (6.5.1)$$

According to Bayes' theorem, the conditional probability of getting a probability p given the m measures is

$$\mathcal{P}(p|m) = \frac{\mathcal{P}(m|p)\mathcal{P}(p)}{\mathcal{P}(m)} = \frac{\mathcal{P}(m|p)\mathcal{P}(p)}{\int dq \mathcal{P}(m|q)\mathcal{P}(q)}. \quad (6.5.2)$$

This relation can be interpreted as the transformation of the *prior* distribution $\mathcal{P}(p)$ into the *posterior* distribution $\mathcal{P}(p|m)$ according to a *likelihood* distribution $\mathcal{P}(m|p)$. A distribution is called the *prior conjugate* of the likelihood function if it is transformed by Bayes' theorem into a distribution with the same algebraic form but with different parameters. Using the binomial distribution in Eq. (6.5.1) as the likelihood distribution, we have that the prior conjugate is the Beta distribution:

$$\mathcal{B}(\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1} (1-p)^{\beta-1}. \quad (6.5.3)$$

If we replace the Beta as the prior distribution and the binomial as the likelihood distribution in Eq. (6.5.2), we can see that the posterior distribution is still a Beta distribution with different parameters, namely

$$\begin{aligned} \mathcal{P}(p|m) &= \frac{\mathcal{P}_b(m|p)\mathcal{B}(\alpha, \beta)}{\int dq \mathcal{P}_b(m|q)\mathcal{B}(\alpha, \beta)} \\ &= \frac{p^m (1-p)^{M-m} p^{\alpha-1} (1-p)^{\beta-1}}{\int dq q^m (1-q)^{M-m} q^{\alpha-1} (1-q)^{\beta-1}} \\ &= \frac{p^{m+\alpha-1} (1-p)^{M-m+\beta-1}}{\int dq q^{m+\alpha-1} (1-q)^{M-m+\beta-1}} \\ &:= \frac{p^{\alpha'-1} (1-p)^{\beta'-1}}{\int dq q^{\alpha'-1} (1-q)^{\beta'-1}} \\ &= \mathcal{B}(\alpha', \beta'), \end{aligned} \quad (6.5.4)$$

where we have defined the parameters of the posterior distribution $\alpha' := \alpha + m$ and $\beta' := \beta + M - m$. We set the parameters of the prior distribution so that it is constant, i.e. $\alpha = \beta = 1$. After having performed M simulations of a circuit and obtained m times a certain result for a fixed qubit, we can estimate the confidence interval so that it contains the 68% or the 90% of the data generated by the posterior distribution $\mathcal{B}(m + 1, M - m + 1)$ by calculating the value p_{min} and p_{max} so that $\mathcal{P}(p_{min} < p < p_{max}) = 0.68$ or 0.90 (using the percentile). In this chapter we present the 68% and 90% confidence intervals of the results with error bars with and without caps respectively.

6.5.2 Single Trotter step evolution

The single-step results are obtained both for a system with $N = 4$ neutrinos, equivalent to the one studied in Ref. [67] on a superconducting device and in Ref. [69] on a quantum annealer, and for a larger system with $N = 8$. A more recent paper (Ref. [84]) has applied the same algorithm on the same quantum hardware for a system of $N = 12$ and obtained excellent results for the case of single Trotter evolution. This suggests the possibility of increasing the number of neutrinos to be simulated.

The initial flavour states for $N = 4$ and $N = 8$ are chosen to contain a mixture of both ν_e and ν_x flavours as in Eq. (6.1.7). In particular, we set

$$|\varphi(0)\rangle^{(4)} = |0011\rangle, \quad |\varphi(0)\rangle^{(8)} = |00001111\rangle, \quad (6.5.5)$$

respectively, and initialise the quantum register with single qubit gates $U_q(\pi, 0) = X$. We then apply the Trotter step according to the diagram in Fig. 6.12 (and its generalisation to the case $N = 8$), using the gate decomposition in Eq. (6.4.13) for each $\binom{N}{2}$ two-qubit operation. Once the initial state is set for all simulations, the propagation of a single Trotter step is performed by applying the circuit with the parameter $\alpha = -J_{ij}dt$, for each pair and using different values of dt . This requires a number of ZZ gates equal to 18 for $N = 4$ and 84 for $N = 8$.

Since the trapped-ion device has more than four qubits at our disposal, in order to minimise the execution cost of the $N = 4$ experiments (see the constant contribution to the cost in Eq. (6.4.11)), we always use two sets of four qubits simultaneously, each of which performs the circuit corresponding to two different values of dt_1 and dt_2 , where $dt_2 = dt_1 + 4\mu^{-1}$. The results presented below are therefore all from experiments with eight qubits, as follows

$$\begin{array}{c}
 |0\rangle \\
 |0\rangle \\
 |1\rangle \\
 |1\rangle
 \end{array}
 \begin{array}{|c}
 \hline
 U(dt_1) \\
 \hline
 \end{array}
 \begin{array}{c}
 |0\rangle \\
 |0\rangle \\
 |1\rangle \\
 |1\rangle
 \end{array}
 \quad \text{and} \quad
 \begin{array}{c}
 |0\rangle \\
 |0\rangle \\
 |0\rangle \\
 |0\rangle \\
 |1\rangle \\
 |1\rangle \\
 |1\rangle \\
 |1\rangle
 \end{array}
 \begin{array}{|c}
 \hline
 U(dt) \\
 \hline
 \end{array}
 \begin{array}{c}
 |0\rangle \\
 |0\rangle \\
 |0\rangle \\
 |0\rangle \\
 |1\rangle \\
 |1\rangle \\
 |1\rangle \\
 |1\rangle
 \end{array}, \quad (6.5.6)$$

for $N = 4$ and $N = 8$ respectively. In the case of $N = 4$, the two-qubit gate depth is equal to 9, as can be easily seen in Fig. 6.12, where we have three ZZ gates for each of the three layers. In the case of $N = 8$ the quantum circuit, similar to the one shown in Fig. 6.13, contains 7 layers, corresponding to a two-qubit gate depth of $7 \cdot 3 = 21$. Using this algorithm, the gate depth for the circuit implementing the single Trotter-step evolution of a system of N neutrinos is $3(N - 1)$.

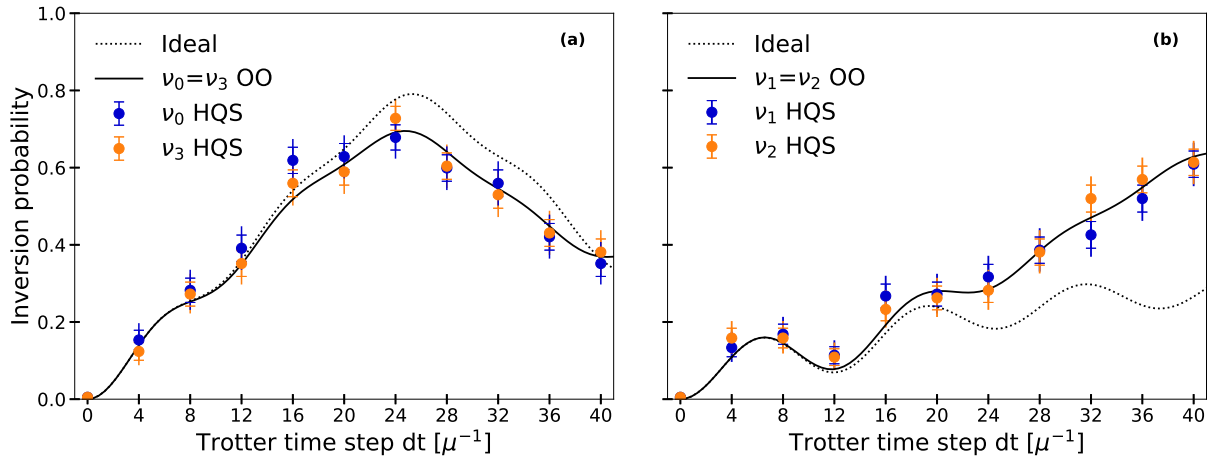


Figure 6.17: Single Trotter step evolution for the inversion probability starting from the initial state $|\varphi(0)\rangle^{(4)} = |0011\rangle$. Panel (a) is for neutrinos ν_0 and ν_3 , and panel (b) is for ν_1 and ν_2 . The black dotted line represents the ideal results using the exact propagator, while the black solid line represents the ideal result using a single Trotter step for $dt \in [0, 40] \mu^{-1}$. The results obtained from the experiments on QSM H1-2 are represented by data points and error bars with and without caps corresponding to 68% and 90% confidence intervals, respectively, and for time steps $dt = 4k \mu^{-1}$ where $k \in \{1, \dots, 10\}$.

Figure 6.17 shows the results for the inversion probability obtained for the simulation of a system of $N = 4$ neutrinos, while Fig. 6.18 shows those for a system of $N = 8$ neutrinos. In both figures, the pairs of neutrinos related by the exchange symmetry of Eq. (6.1.4) are shown in the same panel. In the limit of negligible machine errors, the implementation of the propagator proposed in this work guarantees perfect symmetry under particle exchange in the case of $N = 4$, and the results obtained on the real device almost always respect this symmetry within a confidence interval of 68% and always in the interval of 90%. For $N = 8$ neutrinos our implementation of the propagator respects the symmetry of particle exchange up to $dt \approx 24 \mu^{-1}$, as can be seen from the theoretical results shown as solid lines in Fig. 6.18. The real data respect the same symmetry within the 90% confidence interval. The results obtained for the evolution of a single step are very promising and much more compatible with the theoretical ones than those obtained in Ref. [67] and reported also here in Fig. 6.19. To further compare the results obtained in this work using a trapped-ion device and all-to-all connectivity with the previous results obtained using the swap network on the IBMQ Vigo superconducting device [67], in Table 6.1 we present the values for χ^2 of the inversion probabilities of each neutrino for the $N = 4$ simulation. The quality of the results is assessed by measuring the distance of the calculated results $P_i(t)$ from the theoretical prediction of the inversion probability $P_i^{(th)}(t)$ of

	ν_0	ν_1	ν_2	ν_3
OO + QSM H1-2 (bare)	0.36	0.35	0.27	0.14
SN + IBMQ Vigo (bare)	424.31	527.64	545.28	502.60
SN + IBMQ Vigo (mit)	71.35	73.64	126.38	142.72
SN + IBMQ Vigo (bare*)	10.36	12.88	13.31	12.27

Table 6.1: Values of χ^2 for each neutrino calculated on the results obtained from the propagation of a single Trotter step. The results denoted IBMQ Vigo are taken from Ref. [67] (see Fig. 6.19), while the QSM H1-2 ones are from the present work.

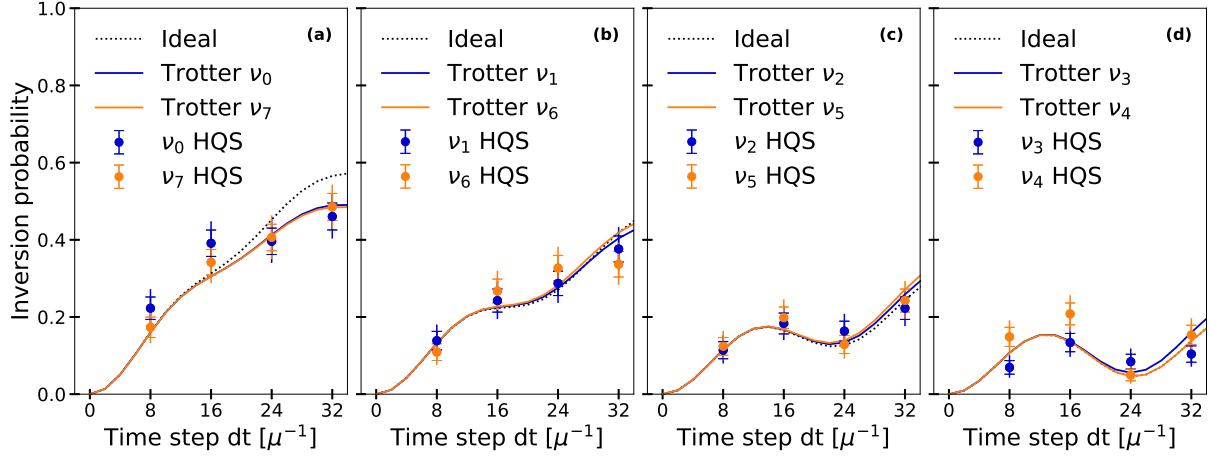


Figure 6.18: Single Trotter step evolution for the inversion probability starting from the initial state $|\varphi(0)\rangle^{(8)} = |00001111\rangle$ of a system with $N = 8$ neutrinos. The results are calculated using a single Trotter step evolution for $dt \in \{8, 16, 24, 32\} \mu^{-1}$. Panel (a) shows the result for the pair (ν_0, ν_7) , panel (b) for (ν_1, ν_6) , panel (c) for (ν_2, ν_5) and panel (d) for (ν_3, ν_4) . Curves and data points follow the same convention as in Fig. 6.17.

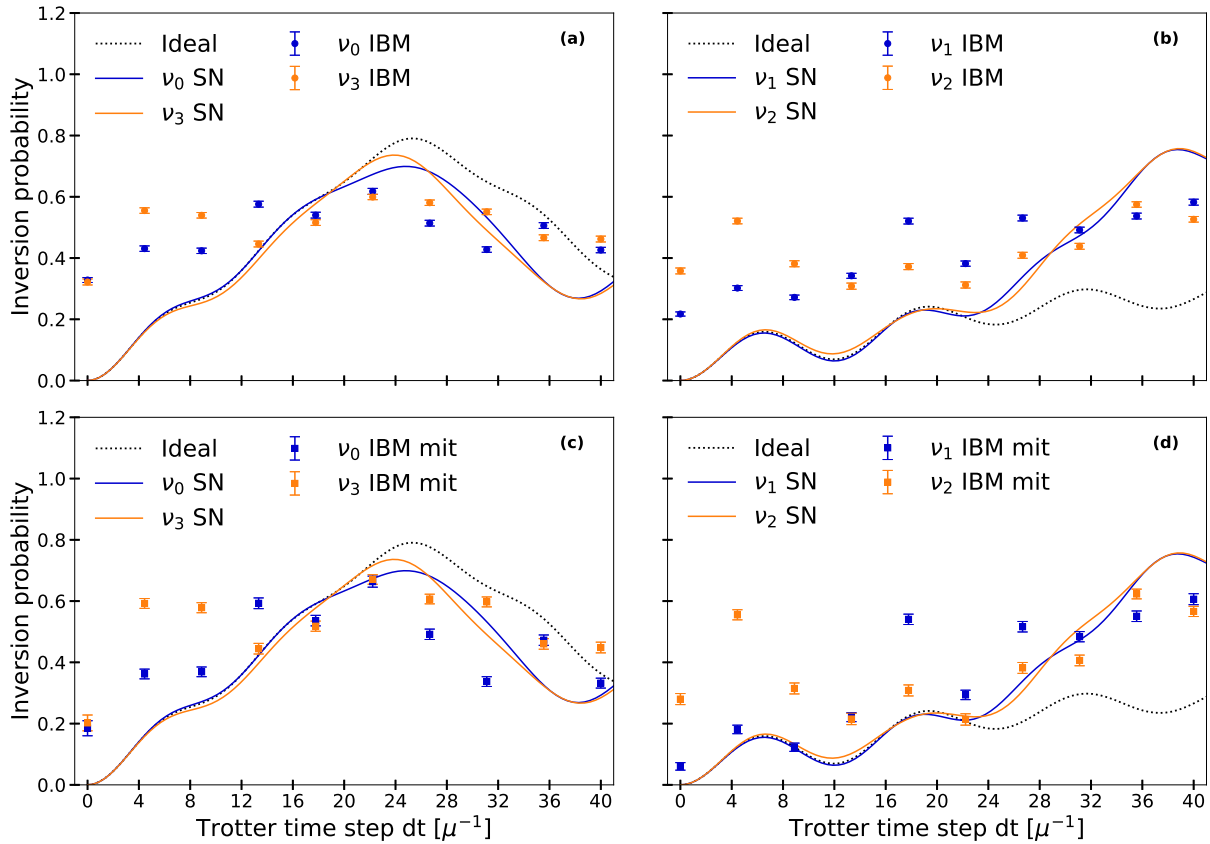


Figure 6.19: Results for the inversion probability for the $N = 4$ system obtained in previous work (Ref. [67]) using the swap network scheme and the IBMQ Vigo quantum machine.

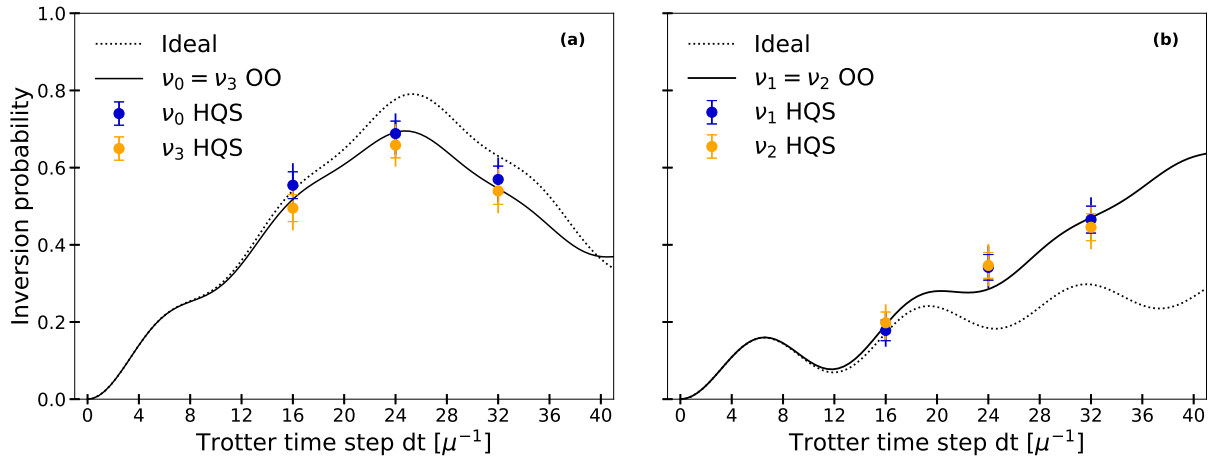


Figure 6.20: Single Trotter step evolution for the system of $N = 4$ neutrinos for time steps $dt \in \{16, 24, 32\} \mu^{-1}$, using three parallel simulations of four neutrinos each.

the neutrino ν_i at time t using the following function:

$$\chi_i^2 = \frac{1}{10} \sum_{k=1}^{10} \frac{\left(P_i(kdt) - P_i^{(th)}(kdt) \right)^2}{\delta P_i(kdt)^2}, \quad (6.5.7)$$

where $\delta P_i(t)$ is the estimated variance (taken to coincide with the 68% confidence interval) and 10 is the number of simulated points, for each neutrino, used to calculate χ^2 . The third row of table 6.1 shows results from the superconducting IBMQ device after error mitigation, for our current simulation we did not attempt to mitigate errors and report the bare results directly. As can be seen from these results, there is a significant increase in fidelity with the new results presented here. This is an effect of both the higher gate fidelity provided by the trapped-ion device and the reduction in one-qubit rotations provided by the different Trotter decomposition adopted here. The number of general single-qubit operations $\in \mathcal{SU}(2)$ in a single step for $N = 4$ is indeed reduced from 40 with the decomposition adopted in Ref. [67] to 36 in the present work. More importantly, the 40 rotations adopted in the previous work have arbitrary angles due to the combination of the pair propagator and the SWAP gate. Three elementary rotations are always required to implement them. Instead, by exploiting the full connectivity and the optimal decomposition in Fig. 6.14, about 2/3 of the unitaries consist of rotations of angles that are multiples of $\pi/2$. This helps to reduce the effect of coherent errors in the final results. Finally, the results of the IBMQ simulations are obtained using a much larger statistical sample (8192 circuit repetitions instead of 200). In order to compare the new results more directly with those obtained there, in the last line of Table 6.1 we also report the estimated χ^2 we would have expected to see if we had reduced the statistics of the bare IBMQ results (obtained by multiplying by 200/8192). The same procedure cannot be applied consistently to the mitigated results, since such an estimate is also affected by systematic errors (for more details see Refs. [67, 85, 86]). However, the strong effect of increased gate fidelity is still evident. The use of error mitigation techniques could further improve the quality of the results. These techniques, including but not limited to symmetry protection (as elucidated by Tran et al. in Ref. [87]), virtual distillation (as introduced by Huggins et al. in Ref. [88]), and symmetry verification (as articulated by Bonet et al. in Ref. [89]), hold the promise of substantially ameliorating the noise-induced perturbations that affect the simulation results.

Because we chose to use 8 qubits to run two simulations in parallel for $N = 4$, the result errors could in principle be affected by cross-talk between the two simulations. However, we have

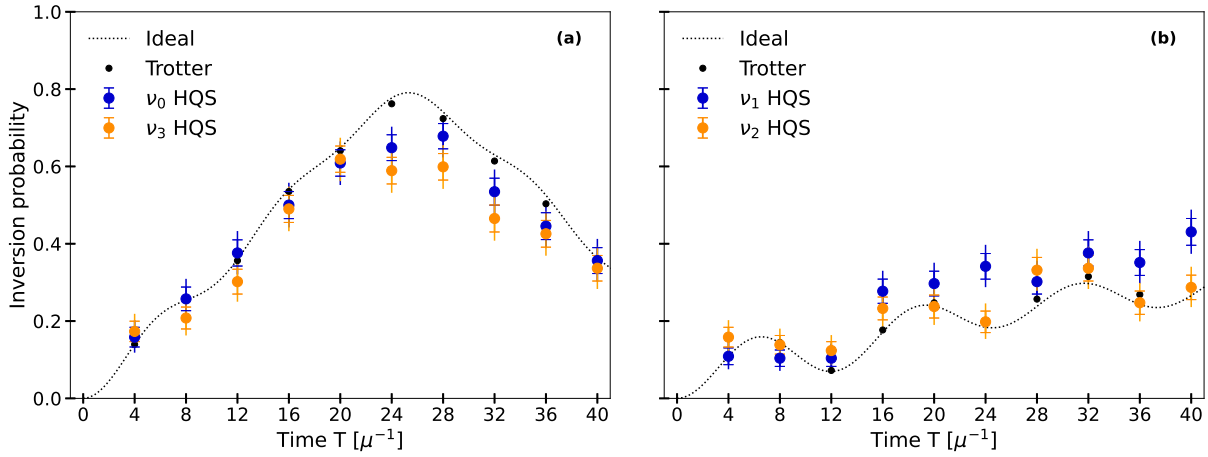


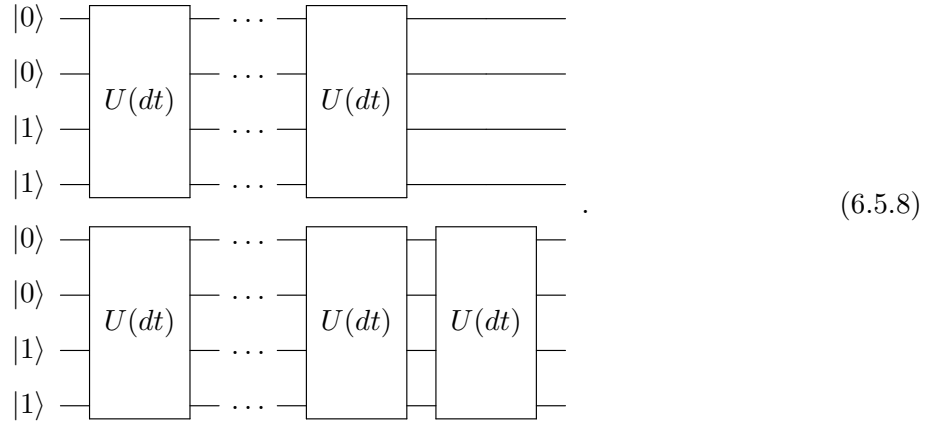
Figure 6.21: Real-time evolution of a system with $N = 4$ neutrinos for time step $dt = 4 \mu^{-1}$ and for a total time of $T = kdt$ with $k \in \{1, \dots, 10\}$ using the QSM H1-2 trapped-ion device. Panel (a) for neutrinos ν_0 and ν_3 and panel (b) for ν_1 and ν_2 . Dashed black lines are the ideal evolutions, which, using this small time step, is almost the same as the Trotter approximated propagation (black points).

analysed the results of three parallel runs over 12 qubits using time steps $dt = (16, 24, 32) \mu^{-1}$ and the results are shown in Fig. 6.20. Since we found them to be compatible with the results shown in Fig. 6.17, this suggests that cross-talk effects are minimal. Furthermore, the quantum machine has three interaction zones, which means that at most three pairs of qubits can be made to interact simultaneously. Since in this last simulation we ran the same circuit (with different angles in the individual qubit rotations) on three quartets of qubits, the moment of interaction of the pairs via the ZZ gate should presumably occur at the same time. It can therefore be assumed that the excess pairs are made to interact at a later time during compilation, which could cause the so-called *idle error*. However, since the data are compatible with the compilation of the $4 + 4$ circuit, we can conclude that also this error did not significantly affect the simulation.

6.5.3 Multiple Trotter step evolution

As mentioned in the previous sections, in order to achieve long simulation times while keeping the error under control, the standard approach is to divide the full interval into time steps which are then approximated by a short time step. The real-time evolution of the system of $N = 4$ neutrinos, initially in the state $|\varphi(0)\rangle^{(4)} = |0011\rangle$, is achieved by the sequential application of the trotterised evolution, as shown in the scheme in Fig. 6.15, a number of times $k \in \{1, \dots, 10\}$ to obtain the final state $|\varphi(kdt)\rangle = \tilde{U}^{(2b)}(dt)^k U^{(1b)}(dt)^k |\varphi(0)\rangle$ with $dt = 4 \mu^{-1}$. The results are plotted in Fig. 6.21. In this case, too, in order to make the best use of the whole machine, we ran the simulations in pairs, allowing the first four neutrinos to evolve up to a time $T = kdt$ and

the last four up to $T = (k + 1)dt$, as follows:



(6.5.8)

This means that the circuit simulating the first two times $T = dt$ and $T = 2dt$ contains 18 ZZ gates applied to the first four qubits and 36 to the last four. The Table 6.2 summarises the number of ZZ gates and single qubit gates needed to simulate the system up to a given evolution time $T = kdt$. For each step we need $\binom{N}{2}3 = 18$ two-qubit gates, while exploiting cancellations between neighbouring pair propagators, the number of general single-qubit rotations required is $32k + 4$. In fact, the first step involves 36 rotations $\in SU(2)$, two of which are applied to neutrinos ν_0 and ν_1 at the end of the circuit. When we add the second step, the one-body part goes to the beginning and adds nothing, so 34 gates are added. But the second step still has two gates applied to ν_0 and ν_1 at the beginning of the circuit, which can then be merged. So the final count is

$$\# SU(2) = 36 + (34 - 2)(k - 1) = 32k + 4. \quad (6.5.9)$$

Considering the two blocks of 4+4 qubits isolated from each other, the data for $T = 5dt$ is obtained with a circuit of 90 gates applied on 4 qubits so the deep is compatible with the circuits used to simulate each single Trotter step of the system of $N = 8$ qubits, presented in Fig. 6.18, for which 84 ZZ gates are used. Their comparison underlines the compatibility of the error and therefore suggests again that the cross-talk contribution is negligible. Regarding the idle error, we can make another observation. In this quantum machine, by default, measurements are taken at the end of the circuit (contrary to what is called *Mid-circuit measurement*). Our simulations include two independent circuits applied to two separate packets of qubits, where the first ends before the second. This means that the first 4 qubits are subject to idle errors which, in the case of trapped ions, are mainly related to dephasing. One way to improve the simulation could be to use the dynamical decoupling approach [90], which consists in applying a sequence of single-qubit gates to the observing qubits, which have the overall effect of an identity. However, if the idle error had significantly affected the results, we should observe some degradation in the results related to the odd value of k .

# of steps k	1	2	3	4	5	6	7	8	9	10
# of ZZ gates	18	36	54	72	90	108	126	144	162	180
# of $SU(2)$ gates	36	68	100	132	164	196	228	260	292	324

Table 6.2: Number of single-qubit gates $\in SU(2)$ and two-qubit ZZ gates to evolve the system of $N = 4$ neutrinos to a fixed time $T = kdt$ to produce the results in Fig. 6.21.

6.5.4 Machine-aware vs automatic compilation

As already mentioned, the QSM compiler does not apply quantum gates as R_x , R_y and CNOT, but allows them to be read and translated from the *Qasm* language. What the compiler does is to decompose such operations using native gates, for example using the circuit in Eq. (6.4.12) for the CNOT case. However, it should be clear at this point that an important aspect of quantum circuit optimisation is precisely the translation of the quantum algorithm into the native library. This guarantees to have control over the quantum circuit that will actually be run, and increases the optimisation of the decomposition. To appreciate how important the machine-aware feature of the compiler is, we tested two different quantum circuits for the 4-neutrino system on the noisy Honeywell Quantum Emulator (HQE), which theoretically implements the noise of the machine H1-1. The first implementation consists in using the *Qiskit* compiler, which allows to decompose $u_{ij}(dt) \in SU(4)$ using three CNOTs and 8 single-qubit rotations, as shown in Eq. (6.3.8). This approach is the same used in previous work (Ref. [67]) to implement $w_{ij}(dt) = \text{SWAP} \cdot u_{ij}(dt)$. The *Qasm* code is then given to the HQE, which automatically translates the quantum circuit into its own library. The second approach is instead the optimal quantum circuit, as used to obtain the real evolution results in Fig. 6.21, and consists in optimising the quantum circuit using the native gate set of the quantum machine. Thus, in theory, the machine does not need to perform any translation or optimisation and can directly apply the algorithm written in *Qasm*. The Figure 6.22 shows the results obtained by the emulator using the two approaches just described. The top panels show results using the first approach and the bottom panels the second. We observe a clear reduction in errors using the machine-aware optimisation approach. To further compare the results we reported in Table 6.3 the χ^2 , for each neutrino, relative to the results of the two decomposition methods.

6.6 Error scaling and gate cost

In this section we analyse the scaling of the Trotter error and the gate cost of the circuit required to simulate the collective neutrino oscillation as a function of the number N of neutrinos. To compare the different decomposition techniques, we look at the scaling corresponding to a description of a system in which we fix the neutrino density $n_\nu = N/V$ and increase the dimension of the system. Further details and full calculations of the results presented here can be found in the Appendix A (Sec. A.1 for the first order and Sec. A.2 for the second order).

6.6.1 First order Trotter error for the interaction Hamiltonian

The two-body part of the neutrino system Hamiltonian describes the $\nu-\nu$ scattering and consists of a sum of $\Gamma = \binom{N}{2} = N(N-1)/2$ terms

$$H^{(2b)} = \sum_{i<j}^N h_{ij} := \sum_{K=1}^{\Gamma} h_K, \quad (6.6.1)$$

	ν_0	ν_1	ν_2	ν_3
CNOT-based circuit	0.57	0.32	0.12	0.49
ZZ-based circuit	0.11	0.32	0.55	0.11

Table 6.3: Values of χ^2 for each neutrino calculated on the results obtained from the multiple-Trotter step propagation on the HQE shown in Fig. 6.22. First line for the CNOT-based automatically optimised quantum circuit and second line for the machine-aware compilation.

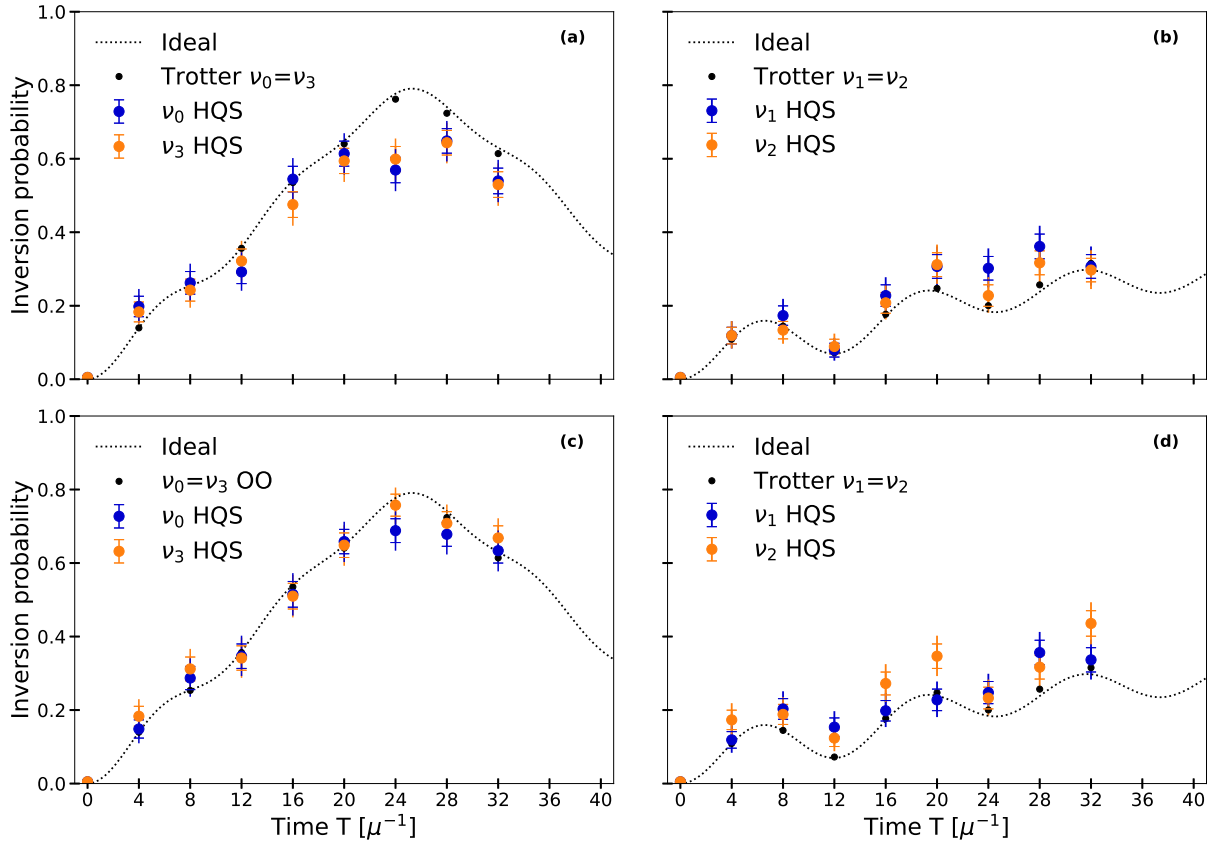


Figure 6.22: Real-time evolution of a system with $N = 4$ neutrinos for time step $dt = 4 \mu^{-1}$ and for total time $T = kdt$ with $k \in \{1, \dots, 8\}$ using the noisy QSE. Panels (a) and (b) show results using a CNOT-based quantum circuit automatically optimised by the Honeywell compiler (neutrinos ν_0 and ν_3 in panel (a) and ν_1 and ν_2 in panel (b)), and panels (c) and (d) show the results obtained by running on the emulator the optimal machine-aware optimised quantum circuit (neutrinos ν_0 and ν_3 in panel (c) and ν_1 and ν_2 in panel (d)).

where $h_{ij} = \mu J_{ij} \boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_j / N$ and we count the neutrinos from 1 to N instead of from 0 to $N - 1$. Using the first-order Trotter decomposition, we can implement the propagator (see Eq. (6.2.8)) using a standard first-order product formula

$$U^{(2b)}(dt) \approx \mathcal{L}_1(dt) := \prod_{i < j}^N e^{-ih_{ij}dt} = \prod_{K=1}^{\Gamma} e^{-ih_K dt}. \quad (6.6.2)$$

Using the result from Proposition 9 of Ref. [91], we bound the first-order Trotter error as

$$\varepsilon_1(dt) = \left\| \mathcal{L}_1(dt) - U^{(2b)}(dt) \right\| \leq \frac{dt^2}{2} \sum_{K=1}^{\Gamma} \left\| \sum_{L=K+1}^{\Gamma} [h_K, h_L] \right\|, \quad (6.6.3)$$

where in our case K and L correspond to pair indices $K = (i, j)$ and $L = (k, l)$. The sum inside the norm in the expression above can be expressed explicitly using the double indices and separating the sum for $L > K$ into two contributions: those for which the first index of K is equal to the first index of L and those for which the first index of L is greater than the first index of K . Finally, considering that a commutator $[h_{ij}, h_{kl}]$ is different from zero if at least one index in (k, l) matches an index in (i, j) , we obtain that the Trotter error bound for a small time step dt is

$$\varepsilon_1(dt) \leq \frac{dt^2}{2} \sum_{i < j}^N \left\| \sum_{k=i+1}^{j-1} [h_{ij}, h_{kj}] + \sum_{l=j+1}^N ([h_{ij}, h_{il}] + [h_{ij}, h_{jl}]) \right\|. \quad (6.6.4)$$

The commutators between different two-body Hamiltonians can be calculated in a straightforward way as follows

$$[h_{ij}, h_{ik}] = \frac{\mu^2}{N^2} J_{ij} J_{ik} [\boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_j, \boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_k] = \frac{\mu^2}{N^2} 2i J_{ij} J_{ik} \boldsymbol{\sigma}_i \cdot (\boldsymbol{\sigma}_j \wedge \boldsymbol{\sigma}_k), \quad (6.6.5)$$

where we have used $\mathbf{a} \wedge \mathbf{b}$ to denote the standard cross product in three dimensions. Substituting this equation into Eq. (6.6.4) we have

$$\varepsilon_1(dt) \leq dt^2 \frac{\mu^2}{N^2} \sum_{i < j}^N \left\| J_{ij} \left(\sum_{l=j+1}^N (J_{il} - J_{jl}) \boldsymbol{\sigma}_i \cdot (\boldsymbol{\sigma}_j \wedge \boldsymbol{\sigma}_l) + \sum_{k=i+1}^{j-1} J_{kj} \boldsymbol{\sigma}_j \cdot (\boldsymbol{\sigma}_i \wedge \boldsymbol{\sigma}_k) \right) \right\|, \quad (6.6.6)$$

where we used the cyclic permutation equivalence of the cross product. Then, considering that the coupling matrix J_{ij} is positive and using the bound $\|\boldsymbol{\sigma}_i \cdot (\boldsymbol{\sigma}_j \wedge \boldsymbol{\sigma}_k)\|_{\infty} \leq 4$ for all $i, j, k \in \{1, \dots, N\}$ we get

$$\varepsilon_1(dt) \leq 4dt^2 \frac{\mu^2}{N^2} \sum_{i < j}^N J_{ij} \left\| \sum_{l=j+1}^N (J_{il} - J_{jl}) + \sum_{k=i+1}^{j-1} J_{kj} \right\|. \quad (6.6.7)$$

For a specific choice of angular distributions the sums can be computed straightforwardly. However, in order to obtain a general bound on the error we can introduce

$$\Theta := \max_{i,j} (1 - \cos(\theta_{ij})) \quad (6.6.8)$$

and obtain the upperbound

$$\varepsilon_1(dt) \leq 12dt^2 \frac{\mu^2 \Theta^2}{N^2} \binom{N}{3} = \mathcal{O}(dt^2 \mu^2 N). \quad (6.6.9)$$

Fixing a total evolution time T and evolving the system for $r_1 = T/dt$ steps, the total error can be bounded by

$$\varepsilon_1(T) \leq r_1 \varepsilon_1(dt) \leq r_1 12 \frac{T^2 \mu^2 \Theta^2}{r_1^2 N^2} \binom{N}{3} = 12 \frac{T^2 \mu^2 \Theta^2}{r_1 N^2} \binom{N}{3}. \quad (6.6.10)$$

where we used the union bound to obtain $\varepsilon_1(T) \leq r \varepsilon_1(dt)$. One also finds that the total number of steps $r_1(T, \epsilon)$ required to evolve to a time T while keeping the total error below ϵ scales linearly with the system size, namely

$$r_1(T, \epsilon) \leq 12 \frac{T^2 \mu^2 \Theta^2}{\epsilon N^2} \binom{N}{3} = \mathcal{O}\left(\frac{T^2 \mu^2}{\epsilon} N\right). \quad (6.6.11)$$

We can then bound the gate cost $\mathcal{C}_1(T, \epsilon)$ of a quantum circuit implementing the full evolution in terms of the number of general two-qubit gates $\in \mathcal{SU}(4)$ needed to implement all the steps. For N neutrinos, we have $\binom{N}{2} = N(N-1)/2$ interaction terms in each Trotter step, so the gate cost using a first-order product formula scales as

$$\mathcal{C}_1(T, \epsilon) \leq \binom{N}{2} r_1(T, \epsilon) = 12 \frac{T^2 \mu^2 \Theta^2}{\epsilon N^2} \binom{N}{3} \binom{N}{2} = \mathcal{O}\left(\frac{T^2 \mu^2}{\epsilon} N^3\right). \quad (6.6.12)$$

Since all transformations $\in \mathcal{SU}(4)$ can be decomposed with at most three entanglement gates [17], as we proved for the native gate set available on the QSM H1-2 system in Eq. (6.4.13) and proved in Sec. 2.4.2, the total number of two-qubit gates is given by $3\mathcal{C}_1(T, \epsilon)$.

6.6.2 Second order Trotter error for the interaction Hamiltonian

It is possible to get a more accurate approximation of $U^{(2b)}(dt)$ using a second order Trotter-Suzuki formula, which can be expressed as

$$U^{(2b)}(dt) \approx \mathcal{L}_2(dt) = \mathcal{L}_1\left(\frac{dt}{2}\right) \mathcal{L}_1^\dagger\left(-\frac{dt}{2}\right), \quad (6.6.13)$$

Using the result of Proposition 10 from Ref. [91], the second order Trotter error can be bounded by:

$$\varepsilon_2(dt) \leq \frac{dt^3}{12} \sum_K^\Gamma \left\| \sum_{L>K}^\Gamma \sum_{M>K}^\Gamma [h_L, [h_M, h_K]] \right\| + \frac{dt^3}{24} \sum_K^\Gamma \left\| \sum_{L>K}^\Gamma [h_K, [h_K, h_L]] \right\|. \quad (6.6.14)$$

To bound the second term, we use a similar procedure to the one adopted in the first order case by expanding the sums and keeping contributions $[h_{ij}, [h_{ij}, h_{kl}]]$ with one of the (k, l) indices matching one of the (i, j) indices. Translating the sums into a double index sums ($K = (i, j)$ and $L = (k, l)$) the second term becomes:

$$\begin{aligned} \frac{dt^3}{24} \sum_K^\Gamma \left\| \sum_{L>K}^\Gamma [h_K, [h_K, h_L]] \right\| = \\ \frac{dt^3}{24} \sum_{i<j}^N \left\| \sum_{l=j+1}^N [h_{ij}, [h_{ij}, h_{il}]] + \sum_{k=i+1}^{j-1} [h_{ij}, [h_{ij}, h_{kj}]] + \sum_{l=j+1}^N [h_{ij}, [h_{ij}, h_{jl}]] \right\|. \end{aligned} \quad (6.6.15)$$

The bound can then be found by using the expression for the nested commutator

$$[h_{ij}, [h_{ij}, h_{ik}]] = -4J_{ij}^2 J_{ik} \boldsymbol{\sigma}_i \cdot (\boldsymbol{\sigma}_j \wedge (\boldsymbol{\sigma}_j \wedge \boldsymbol{\sigma}_k)), \quad (6.6.16)$$

together with the bound $\|\sigma_i \cdot (\sigma_j \wedge (\sigma_j \wedge \sigma_k))\| \leq 8$. The result reads

$$\frac{dt^3}{24} \sum_K^\Gamma \left\| \sum_{L>K}^\Gamma [h_K, [h_K, h_L]] \right\| \leq 4dt^3 \frac{\mu^3 \Theta^3}{N^3} \binom{N}{3}. \quad (6.6.17)$$

We now turn our attention to the first contribution in Eq. (6.6.14). Using the double indices $K = (i, j)$, $L = (k, l)$ and $M = (p, q)$ we have

$$\begin{aligned} \frac{dt^3}{12} \sum_K^\Gamma \left\| \sum_{L>K}^\Gamma \sum_{M>K}^\Gamma [h_L, [h_M, h_K]] \right\| = \\ \frac{dt^3}{12} \sum_{i<j}^N \left\| \left(\sum_{l=j+1}^N \delta_{ki} + \sum_{k=i+1}^N \sum_{l=k+1}^N \right) \left(\sum_{q=j+1}^N \delta_{pi} + \sum_{p=i+1}^N \sum_{q=p+1}^N \right) [h_{kl}, [h_{pq}, h_{ij}]] \right\| \end{aligned} \quad (6.6.18)$$

which can be bound as the sum of the different norms. After long and boring calculations, which can be found in Appendix A.2, we found that the first contribution gives

$$\sum_{i<j}^N \sum_{l=j+1}^N \sum_{q=j+1}^N \|[h_{il}, [h_{iq}, h_{ij}]]\| \leq 32 \frac{\mu^3}{N^3} \Theta^3 \left(2 \binom{N}{4} + \binom{N}{3} \right), \quad (6.6.19)$$

the second gives

$$\sum_{i<j}^N \sum_{l=j+1}^N \sum_{p=i+1}^N \sum_{q=p+1}^N \|[h_{il}, [h_{pq}, h_{ij}]]\| \leq 32 \frac{\mu^3}{N^3} \Theta^3 \left(3 \binom{N}{4} + \binom{N}{3} \right), \quad (6.6.20)$$

the third gives

$$\sum_{i<j}^N \sum_{k=i+1}^N \sum_{l=k+1}^N \sum_{q=j+1}^N \|[h_{kl}, [h_{iq}, h_{ij}]]\| \leq 32 \frac{\mu^3}{N^3} \Theta^3 \left(6 \binom{N}{4} + \binom{N}{3} \right), \quad (6.6.21)$$

and the last one gives

$$\sum_{i<j}^N \sum_{k=i+1}^N \sum_{l=k+1}^N \sum_{p=i+1}^N \sum_{q=p+1}^N \|[h_{kl}, [h_{pq}, h_{ij}]]\| \leq 32 \frac{\mu^3}{N^3} \Theta^3 \left(10 \binom{N}{4} + 3 \binom{N}{3} \right). \quad (6.6.22)$$

The final bound we found for the first contribution in Eq. (6.6.14) is then

$$\begin{aligned} \frac{dt^3}{12} \sum_K^\Gamma \left\| \sum_{L>K}^\Gamma \sum_{M>K}^\Gamma [h_L, [h_M, h_K]] \right\| &\leq \frac{dt^3}{12} 32 \frac{\mu^3}{N^3} \Theta^3 \left(21 \binom{N}{4} + 6 \binom{N}{3} \right) \\ &\leq \frac{1}{9} dt^3 \mu^3 \Theta^3 \frac{(N-1)(N-2)}{N^2} (21N - 39). \end{aligned} \quad (6.6.23)$$

We conclude that the second order Trotter error can be bound, also in this case, by a linear scaling with N . In particular,

$$\varepsilon_2(dt) \leq dt^3 \frac{\mu^3 \Theta^3}{N^3} \left(20 \binom{N}{3} + 56 \binom{N}{4} \right) = \mathcal{O}(dt^3 \mu^3 N). \quad (6.6.24)$$

Evolving up to T using r_2 steps, the total error can be bounded by

$$\varepsilon_2(T) \leq r_2 \varepsilon_2(dt) = \frac{T^3}{r_2^2} \frac{\mu^3 \Theta^3}{N^3} \left(20 \binom{N}{3} + 56 \binom{N}{4} \right) = \mathcal{O}\left(\frac{T^3 \mu^3 N}{r_2^2}\right), \quad (6.6.25)$$

and the necessary number of steps to have a target error ϵ is

$$r_2(T, \epsilon) \leq \frac{(T\mu\Theta)^{3/2}}{\sqrt{\epsilon}N^{3/2}} \sqrt{20\binom{N}{3} + 56\binom{N}{4}} = \mathcal{O}\left(\frac{T^{3/2}\mu^{3/2}\sqrt{N}}{\sqrt{\epsilon}}\right). \quad (6.6.26)$$

Finally, the complexity in terms of two-qubit gates is

$$\mathcal{C}_2(T, \epsilon) \leq \left(2\binom{N}{2} - \frac{N}{2}\right)r_2(T, \epsilon) = \mathcal{O}\left(\frac{(T\mu)^{3/2}}{\sqrt{\epsilon}}N^{5/2}\right). \quad (6.6.27)$$

The scaling with the number of neutrinos N has improved considerably within this scheme. The use of even higher order formulas may allow one to achieve a near optimal scaling $\mathcal{C}(N^{2+\delta})$ for $\delta \ll 1$, but with possibly much larger constant prefactors.

6.6.3 Qubitisation scaling

Finally, we would like to comment on the prospect of using more modern approaches to simulate the time evolution operator, such as qubitisation [25, 92]. This scheme also approximates the propagator over small time intervals, but unlike Trotter-Suzuki based approaches, it is able to achieve optimal scaling in both T and the error ϵ for the number of steps

$$r_Q = \mathcal{O}\left(T\alpha_H + \log\left(\frac{1}{\epsilon}\right)\right), \quad (6.6.28)$$

where α_H is a suitable norm of the Hamiltonian operator, which for the two-body neutrino Hamiltonian is given by

$$\begin{aligned} \alpha_H &= \frac{\mu}{N} \left\| \sum_{i<j}^N (1 - \cos(\theta_{ij})) \boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_j \right\| \\ &= 3\frac{\mu}{N} \sum_{i<j}^N (1 - \cos(\theta_{ij})) \\ &= \mathcal{O}(\mu N). \end{aligned} \quad (6.6.29)$$

For a general angular distribution, the gate cost for each of the steps scales with the number of different coefficients, and thus the gate cost for the algorithm scales as

$$\mathcal{C}_Q(T, \epsilon) = \mathcal{O}\left(T\mu N^3 + N^2 \log\left(\frac{1}{\epsilon}\right)\right). \quad (6.6.30)$$

For a fixed evolution time T and target error ϵ , the second-order Trotter-Suzuki scheme then scales better than a qubitisation-based approach. This is not a special property of the neutrino system and has been observed in other applications. Rather, it is related to the fact that qubitisation does not exploit the commutation properties of the terms that form the Hamiltonian.

Figure 6.23 shows the scaling of the 2-qubit operations $\mathcal{C}(T, \epsilon)$ required to evolve a system of N neutrinos up to a total time $T = 40\mu^{-1}$ as a function of the number of particles N and for different methods of propagator decomposition. The error limit has been set to $\epsilon = 0.15$ and we use a strength of $J_{ij} = 1/N$ for all particles that coincide with neutrinos propagating in the orthogonal direction $\theta_{ij} = \pi/2$. We also summarise the results for the bounds on both the number of time steps and the circuit (gate) complexity of a simulation with N neutrinos, maximum time T and error tolerance ϵ in Table 6.4. As can be seen, the qubitisation approach wins for arbitrarily small error threshold ϵ and arbitrarily long evolution time T , but for fixed times and errors there exists a value of N for which the second order Trotter decomposition is advantageous.

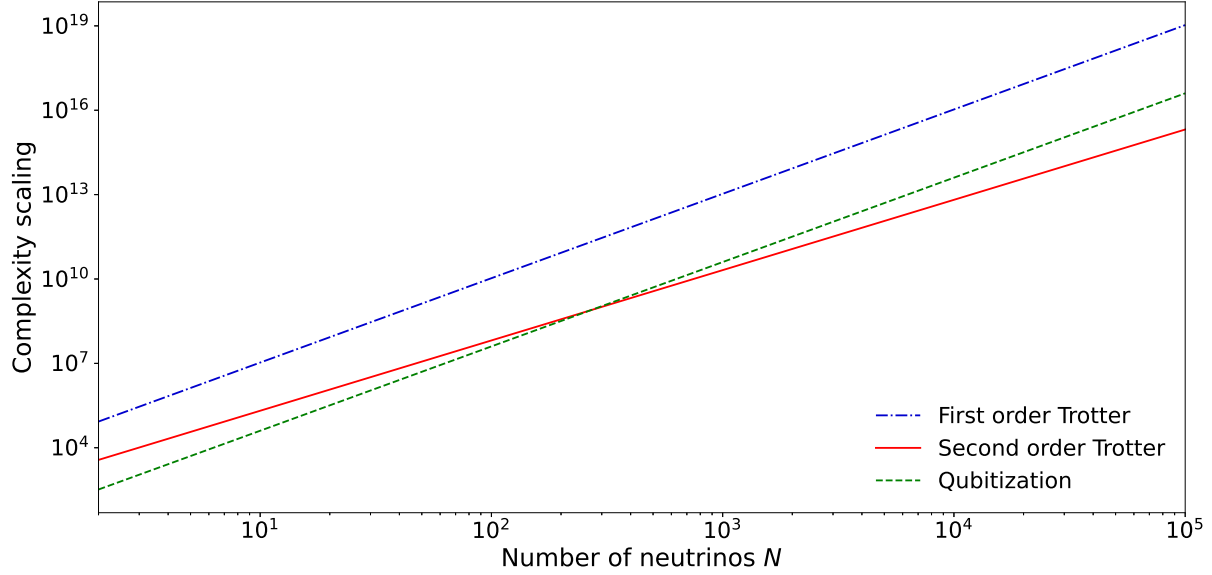


Figure 6.23: Number of two-qubit universal operations needed to evolve a system of N neutrinos for a time $T = 40 \mu^{-1}$ guaranteeing an error $\epsilon \leq 0.15$. Dashed blue line represents the asymptotic scaling of the complexity $\mathcal{C}_1(T, \epsilon)$ using the first order product formula (Eq. (6.6.12)) and the solid red one is $\mathcal{C}_2(T, \epsilon)$ for the second order product formula (Eq. (6.6.27)). Finally, the dashed green line is for qubitization complexity $\mathcal{C}_Q(T, \epsilon)$ in Eq. (6.6.30).

Decomposition type	Single-step error	Number of steps	Circuit complexity
First order Trotter	$\mathcal{O}(dt^2 \mu^2 N)$	$\mathcal{O}\left(\frac{T^2 \mu^2}{\epsilon} N\right)$	$\mathcal{O}\left(\frac{T^2 \mu^2}{\epsilon} N^3\right)$
Second order Trotter	$\mathcal{O}(dt^3 \mu^3 N)$	$\mathcal{O}\left(\frac{T^{3/2} \mu^{3/2}}{\sqrt{\epsilon}} \sqrt{N}\right)$	$\mathcal{O}\left(\frac{T^{3/2} \mu^{3/2}}{\sqrt{\epsilon}} N^{5/2}\right)$
Qubitization	-	$\mathcal{O}\left(T \mu N + \log\left(\frac{1}{\epsilon}\right)\right)$	$\mathcal{O}\left(T \mu N^3 + \log\left(\frac{1}{\epsilon}\right) N^2\right)$

Table 6.4: Asymptotic scaling of the error, needed steps, and number of two-qubit operations to evolve a system until T , keeping the error below ϵ as a function of the number of particles N and for different methods of propagator decomposition.

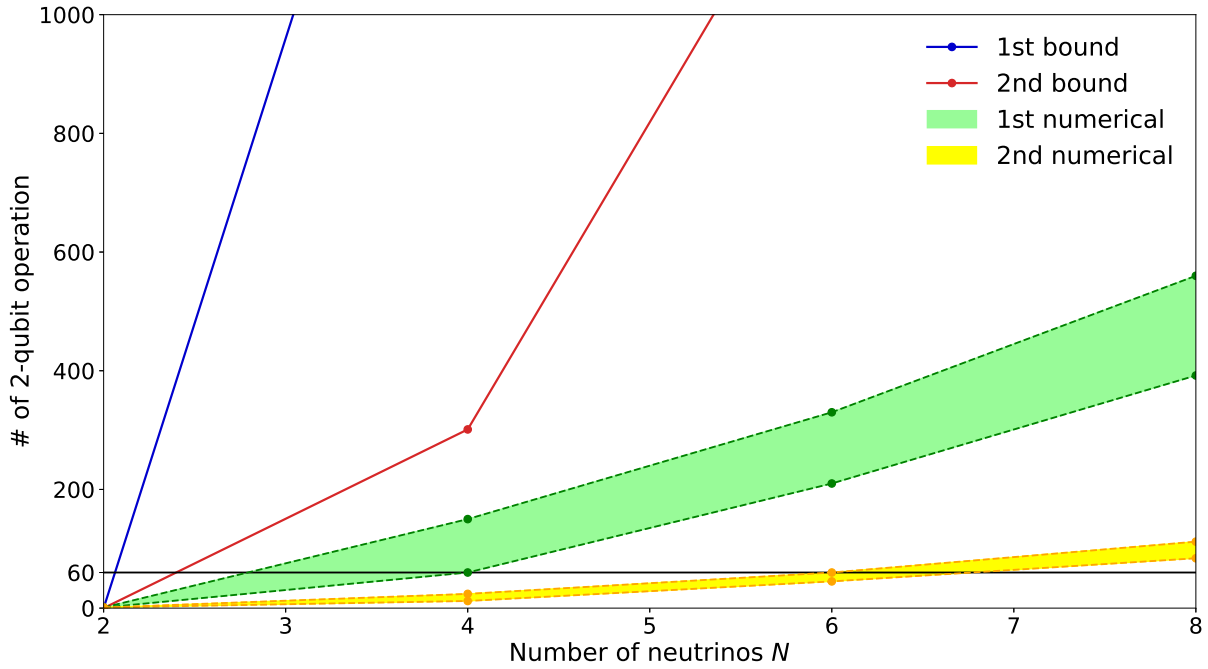


Figure 6.24: Number of two-qubit operations $\mathcal{C}_i(T, \epsilon)$, as a function of N , needed to evolve the system of total time $T = 40 \mu^{-1}$ with an error $\varepsilon_i(T) \leq \epsilon = 0.15$ for order decomposition $i \in \{1, 2\}$. The solid blue line corresponds to the first order bound and the green band to the real scaling achievable using different pair orders. The solid red line instead refers to the second order bound and the yellow band is the corresponding real achievable complexity. Using the optimal decomposition, for the system of $N = 4$ neutrinos, the number of two-qubit universal operations required is $10N(N-1)/2 = 60$ and corresponds to 180 CNOT or ZZ gates, which we actually use to obtain the last point in Fig. 6.21.

6.6.4 Real numerical cost

The actual implementation cost is several orders of magnitude lower than predicted by the theoretical bounds. Moreover, as shown in Section 6.2.2, the cost can be further reduced by using a good decomposition of the propagator, which guarantees a smaller error and therefore allows us to use a larger value of the time step dt . For a total evolution time of $T = 40 \mu^{-1}$ and a target error of $\epsilon = 0.15$, we show in Fig. 6.24 the theoretical limits of the number of gates $\mathcal{C}_1(T, \epsilon)$ and $\mathcal{C}_2(T, \epsilon)$ for the first and second order decompositions as solid blue and red lines, respectively. We also numerically determine the actual range of gate counts required for this simulation as we vary the order in the Trotter decomposition, which we show in Fig. 6.24 as the shaded green and yellow bands for the first and second formulas, respectively. The real complexity is computed using a linear accumulation error, i.e. we search for dt such that $r = T/dt$ guarantees

$$r \left\| e^{-iH^{(2b)}T/r} - \tilde{U}^{(2b)}(T/r) \right\| \leq \epsilon, \quad (6.6.31)$$

where $\tilde{U}^{2b}(T/r) = \mathcal{L}_1(T/r)$ or $\tilde{U}^{2b}(T/r) = \mathcal{L}_2(T/r)$ as defined in Eqs. (6.6.2) and (6.6.13) respectively. Using the exact error accumulation instead of the linear accumulation boundary,

$$\left\| e^{-iH^{(2b)}T} - \tilde{U}^{(2b)}(T/r)^r \right\| \leq \epsilon, \quad (6.6.32)$$

the number of steps required may be smaller.

Conclusions

This thesis deals with the topic of digital quantum computing applied to the simulation of many-body systems, with a particular application to collective flavour oscillation. After an introduction to the general concept of quantum computing, the second chapter introduced the concept of quantum gate decomposition and its formulation in terms of quantum circuits. The design of quantum algorithms in general, and especially in the era of noisy quantum computers, reduces to the definition of the gate decomposition of the unitary transformation to be implemented. An overview of existing methods has been given. However, none of them can be considered efficient, since exponential scaling is intrinsic to the group of unitary transformations that define the evolution of a qubit system. To circumvent this problem two possible solutions exist. The first is to push for substantial improvements in the hardware. This should increase the decoherence time of the qubits and reduce the error introduced by the quantum gates, so that the simulation error becomes small enough to be negligible even with exponential scaling of the complexity. The second, which is more realistic and also applicable today, is to adapt the decomposition procedure to the particular problem we are dealing with. An optimal universal compiler that does not require exponential complexity does not yet exist. Therefore, a computer scientist should be familiar with the various existing decomposition methods, taking into account all their drawbacks and potentials. This will firstly ensure that one can consciously choose the most appropriate method for a given problem and, more importantly, lay the foundation for the search for further, increasingly sophisticated and efficient decomposition methods. While it is necessary to be aware of the different methods of decomposition, the development of quantum algorithms cannot be separated from the knowledge of the different types of existing hardware and, consequently, the different approaches that need to be taken to use them. This argument applies to both a software perspective (each machine has its own library and computational language) and a hardware perspective (each machine has its own topology and natural interactions). Each quantum computer can be deeply different from the others, and each hardware manifests its own advantages and challenges. The hope of this work was also to convey the importance of machine-aware compilation and how this affects the cost of a circuit and the quality of the results obtained. The aim of the chapter on a trapped-ion machine was to present the physics behind the construction of a quantum computer in a comprehensive and didactic way.

The key message of the chapter on the tensor notation of quantum computing was that a given problem can be approached from different perspectives, each with different potentials and advantages. This is true for both the quantum gate decomposition problem and the quantum computing formulation. Each representation hides and reveals different features that may be important from a mathematical, computational and physical point of view. The complexity of a problem can hardly be tamed, but it can most likely be shifted and reformulated in a different way, perhaps allowing it to be tackled more efficiently. The aim of this part of the thesis was certainly not to propose a better way to study and teach quantum computation, but rather to display the richness of describing and observing the same theory from different perspectives. The standard matrix approach is of course still very powerful and, above all, directly applicable

without the need for much effort in representation; the explicit tensor approach simply wants to allow looking at the same problem in an alternative way, proposing advantages and disadvantages of both.

The thesis concluded with a more applied and purely physical part, as opposed to the computer science nature of the first half. Quantum computing has incredible potential to describe complex physical systems. This is because a classical and at the same time rigorous approach to the dynamics of quantum systems inherently requires more effort than is generally available. Instead, the best approach we can think of to describe a quantum many-body system is to mimic it with a quantum system itself. The key requirement is that the system used to represent the original problem is controllable. This application is called Hamiltonian simulation and was an idea of Feynman. Although it is not the only application of quantum computing, it is one of the most promising areas related to it due to its intrinsic exponential speed-up. The last part of this thesis focuses on the simulation of neutrinos which, when emitted at high density, undergo coherent flavour oscillations, also called *collective oscillations*. Such dynamics are particularly important in the field of astrophysics, where supernova explosions, massive star mergers and the physics of the early universe represent interesting and not fully understood phenomena. Using appropriate approximations, we can write the Hamiltonian describing the flavour part of these neutrinos as an isospin Hamiltonian with an all-to-all interaction between the particles. Solving this dynamics without introducing mean-field approximations is almost impossible for a sufficiently large number of neutrinos, precisely because of the exponential growth of the Hilbert space to which the system belongs. However, given the exponential speed-up inherent in Hamiltonian simulations, the use of a system of qubits is feasible and natural. The quantum algorithm designed to simulate the dynamics of the collective flavour oscillation mechanism is presented in detail. Its formulation, written in terms of a quantum circuit, has been obtained and optimised to run on a trapped-ion-based machine. The unexpectedly accurate results, obtained without any error mitigation techniques, and the gentle complexity scaling guarantee that future applications will be even more accurate and physically useful. To define a quantum algorithm as efficient, it is necessary to analyse its complexity scaling. Therefore, the last part of this chapter is the core of the discussion as it proves the usefulness and scalability of the proposed algorithm. The algorithm offers substantial room for further refinement and improvement. Given the quality of the results, it would be possible to increase both the number of neutrinos in the system and the duration of its evolution. The incorporation of various error mitigation strategies could further improve the quality of the simulations. Other approaches can be explored to develop and enhance the collective neutrino oscillation algorithm. One promising avenue could involve alternative encoding maps, strategically designed to reduce the size of the quantum register by exploiting the symmetries inherent in the Hamiltonian, such as flavour conservation. In order to describe more realistic and phenomenologically rich neutrino systems, it will be important to extend the algorithms presented here to simulate neutrinos with different energies. This can be modelled with a particle-dependent external field. Finally, a relevant direction for future investigations concerns the description of collective oscillations in the presence of electrons, thereby incorporating the matter component into the Hamiltonian framework. This extension will serve to address a crucial aspect of physical reality and pave the road for a more comprehensive understanding of neutrino behaviour in extreme environments.

The transdisciplinary nature of this work, crossing from a more computational part to a more physical part, is consistent with the fact that the research effort should be projected in two directions. The first is the purely computational part and aims to develop and improve the hardware of quantum machines and, in parallel, to study, develop and optimise the compilation of unitary transformations into sequences of logic gates. The second direction is the development of efficient quantum algorithms that can solve realistic physical problems in different domains

without requiring approximations that would destroy the quantum nature of the dynamics. On the other hand, there is a risk of introducing different but still strong approximations in an attempt to simplify the mapping of a system into the qubit world. This would make quantum computation efficient for today's noisy machines, but not useful from a physical point of view. Therefore, the two efforts should be balanced and eventually converge. The convergence of the two research directions will make it possible to simulate large and highly entangled systems without introducing uncontrolled approximations and without simplifying the description by forcing the mapping onto the Pauli algebra. When the two directions work hand in hand towards a common goal, the fields that can be explored will be incredibly broad.

Acknowledgements

This research used resources of the Oak Ridge Leadership Computing Facility, which is a U.S. Department of Energy Office of Science User Facility supported under Contract No. DE-AC05-00OR22725. This work was prepared in part by LLNL under Contract No. DE-AC52-07NA27344 with support from the Laboratory Directed Research and Development Grant No. 19-DR-005.

Bibliography

- [1] Richard P. Feynman. “Simulating physics with computers”. In: *International Journal of Theoretical Physics* 21 (6 1982), pp. 467–488. DOI: [10.1007/BF02650179](https://doi.org/10.1007/BF02650179).
- [2] David Deutsch. “Quantum theory, the Church–Turing principle and the universal quantum computer”. In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 400.1818 (1985), pp. 97–117. DOI: [10.1098/rspa.1985.0070](https://doi.org/10.1098/rspa.1985.0070).
- [3] Christopher M Dawson and Michael A Nielsen. “The solovay-kitaev algorithm”. In: *arXiv preprint quant-ph/0505030* (2005). DOI: [10.48550/arXiv.quant-ph/0505030](https://doi.org/10.48550/arXiv.quant-ph/0505030).
- [4] Michael A Nielsen and Isaac L Chuang. “Quantum computation and quantum information”. In: *Phys. Today* 54.2 (2001), p. 60. DOI: [10.1017/CB09780511976667](https://doi.org/10.1017/CB09780511976667).
- [5] Daniel Gottesman. “Stabilizer codes and quantum error correction”. In: *CalTech Ph.D. Thesis* (May 1997). DOI: [10.48550/arXiv.quant-ph/9705052](https://doi.org/10.48550/arXiv.quant-ph/9705052). arXiv: [quant-ph/9705052](https://arxiv.org/abs/quant-ph/9705052).
- [6] Daniel Gottesman. “The Heisenberg representation of quantum computers”. In: *22nd International Colloquium on Group Theoretical Methods in Physics*. July 1998. DOI: [10.48550/arXiv.quant-ph/9807006](https://doi.org/10.48550/arXiv.quant-ph/9807006). arXiv: [quant-ph/9807006](https://arxiv.org/abs/quant-ph/9807006).
- [7] Guifré Vidal. “Efficient Classical Simulation of Slightly Entangled Quantum Computations”. In: *Phys. Rev. Lett.* 91 (14 Oct. 2003), p. 147902. DOI: [10.1103/PhysRevLett.91.147902](https://doi.org/10.1103/PhysRevLett.91.147902).
- [8] Norbert Schuch et al. “Entropy Scaling and Simulability by Matrix Product States”. In: *Phys. Rev. Lett.* 100 (3 Jan. 2008), p. 030504. DOI: [10.1103/PhysRevLett.100.030504](https://doi.org/10.1103/PhysRevLett.100.030504).
- [9] Seth Lloyd. “Universal Quantum Simulators”. In: *Science* 273.5278 (1996), pp. 1073–1078. ISSN: 0036-8075. DOI: [10.1126/science.273.5278.1073](https://doi.org/10.1126/science.273.5278.1073).
- [10] Natalie Klco, Alessandro Roggero, and Martin J Savage. “Standard model physics and the digital quantum revolution: thoughts about the interface”. In: *Reports on Progress in Physics* 85.6 (Mar. 2022), p. 064301. DOI: [10.1088/1361-6633/ac58a4](https://doi.org/10.1088/1361-6633/ac58a4).
- [11] Juan Carlos Garcia-Escartin and Pedro Chamorro-Posada. “Equivalent quantum circuits”. In: *arXiv preprint arXiv:1110.2998* (2011). DOI: [10.48550/arXiv.quant-ph/0505030](https://doi.org/10.48550/arXiv.quant-ph/0505030).
- [12] Adriano Barenco et al. “Elementary gates for quantum computation”. In: *Physical review A* 52.5 (1995), p. 3457. DOI: [10.1103/PhysRevA.52.3457](https://doi.org/10.1103/PhysRevA.52.3457).
- [13] Chi-Kwong Li, Rebecca Roberts, and Xiaoyan Yin. “Decomposition of unitary matrices and quantum gates”. In: *International Journal of Quantum Information* 11.01 (2013), p. 1350015. DOI: [10.1142/S0219749913500159](https://doi.org/10.1142/S0219749913500159).
- [14] Martin Gardner. *Knotted doughnuts and other mathematical entertainments*. American Mathematical Soc., 2020.

- [15] Juha J Vartiainen, Mikko Möttönen, and Martti M Salomaa. “Efficient decomposition of quantum gates”. In: *Physical review letters* 92.17 (2004), p. 177902. URL: [10.1103/PhysRevLett.92.177902](https://doi.org/10.1103/PhysRevLett.92.177902).
- [16] Vivek V Shende, Igor L Markov, and Stephen S Bullock. “Minimal universal two-qubit controlled-NOT-based circuits”. In: *Physical Review A* 69.6 (2004), p. 062321. DOI: [10.1103/PhysRevA.69.062321](https://doi.org/10.1103/PhysRevA.69.062321).
- [17] Farrokh Vatan and Colin Williams. “Optimal quantum circuits for general two-qubit gates”. In: *Physical Review A* 69.3 (2004), p. 032315. DOI: [10.1103/PhysRevA.69.032315](https://doi.org/10.1103/PhysRevA.69.032315).
- [18] Guifre Vidal and Christopher M Dawson. “Universal quantum circuit for two-qubit transformations with three controlled-NOT gates”. In: *Physical Review A* 69.1 (2004), p. 010301. DOI: [10.1103/PhysRevA.69.010301](https://doi.org/10.1103/PhysRevA.69.010301).
- [19] David Elieser Deutsch. “Quantum computational networks”. In: *Proceedings of the royal society of London. A. mathematical and physical sciences* 425.1868 (1989), pp. 73–90. DOI: [10.1098/rspa.1989.0099](https://doi.org/10.1098/rspa.1989.0099).
- [20] Vivek V Shende, Stephen S Bullock, and Igor L Markov. “Synthesis of quantum logic circuits”. In: *Proceedings of the 2005 Asia and South Pacific Design Automation Conference*. 2005, pp. 272–275. DOI: [10.1145/1120725.1120847](https://doi.org/10.1145/1120725.1120847).
- [21] Anna M Krol et al. “Efficient decomposition of unitary matrices in quantum circuit compilers”. In: *Applied Sciences* 12.2 (2022), p. 759. DOI: [10.3390/app12020759](https://doi.org/10.3390/app12020759).
- [22] Robert R Tucci. “A rudimentary quantum compiler (2cnd ed.)” In: *arXiv preprint quant-ph/9902062* (1999). DOI: [10.48550/arXiv.quant-ph/9902062](https://doi.org/10.48550/arXiv.quant-ph/9902062).
- [23] Mikko Möttönen et al. “Quantum circuits for general multiqubit gates”. In: *Physical review letters* 93.13 (2004), p. 130502. DOI: [10.1103/PhysRevLett.93.130502](https://doi.org/10.1103/PhysRevLett.93.130502).
- [24] Guang Hao Low and Isaac L Chuang. “Optimal Hamiltonian simulation by quantum signal processing”. In: *Physical review letters* 118.1 (2017), p. 010501. DOI: [10.1103/PhysRevLett.118.010501](https://doi.org/10.1103/PhysRevLett.118.010501).
- [25] Guang Hao Low and Isaac L Chuang. “Hamiltonian simulation by qubitization”. In: *Quantum* 3 (2019), p. 163. DOI: [10.22331/q-2019-07-12-163](https://doi.org/10.22331/q-2019-07-12-163).
- [26] Jonathan Welch et al. “Efficient quantum circuits for diagonal unitaries without ancillas”. In: *New Journal of Physics* 16.3 (2014), p. 033040. DOI: [10.1088/1367-2630/16/3/033040](https://doi.org/10.1088/1367-2630/16/3/033040).
- [27] Ivan Kassal et al. “Polynomial-time quantum algorithm for the simulation of chemical dynamics”. In: *Proceedings of the National Academy of Sciences* 105.48 (2008), pp. 18681–18686. DOI: [10.1073/pnas.0808245105](https://doi.org/10.1073/pnas.0808245105).
- [28] Nicolas PD Sawaya et al. “Resource-efficient digital quantum simulation of d-level systems for photonic, vibrational, and spin-s Hamiltonians”. In: *npj Quantum Information* 6.1 (2020), p. 49. DOI: [10.1038/s41534-020-0278-0](https://doi.org/10.1038/s41534-020-0278-0).
- [29] John Preskill. “Quantum computing in the NISQ era and beyond”. In: *Quantum* 2 (2018), p. 79. DOI: [10.22331/q-2018-08-06-79](https://doi.org/10.22331/q-2018-08-06-79).
- [30] David P DiVincenzo. “The physical implementation of quantum computation”. In: *Fortschritte der Physik: Progress of Physics* 48.9-11 (2000), pp. 771–783.
- [31] Hartmut Häffner, Christian F Roos, and Rainer Blatt. “Quantum computing with trapped ions”. In: *Physics reports* 469.4 (2008), pp. 155–203. DOI: [10.1016/j.physrep.2008.09.003](https://doi.org/10.1016/j.physrep.2008.09.003).

- [32] Rainer Blatt and Christian F Roos. “Quantum simulations with trapped ions”. In: *Nature Physics* 8.4 (2012), pp. 277–284. DOI: [10.1038/nphys2252](https://doi.org/10.1038/nphys2252).
- [33] Philipp Schindler et al. “A quantum information processor with trapped ions”. In: *New Journal of Physics* 15.12 (2013), p. 123012. DOI: [10.1088/1367-2630/15/12/123012](https://doi.org/10.1088/1367-2630/15/12/123012).
- [34] Juan I Cirac and Peter Zoller. “Quantum computations with cold trapped ions”. In: *Physical review letters* 74.20 (1995), p. 4091. DOI: [10.1103/PhysRevLett.74.4091](https://doi.org/10.1103/PhysRevLett.74.4091).
- [35] Chris Monroe et al. “Demonstration of a fundamental quantum logic gate”. In: *Physical review letters* 75.25 (1995), p. 4714. DOI: [10.1103/PhysRevLett.75.4714](https://doi.org/10.1103/PhysRevLett.75.4714).
- [36] Gilbert Strang. *Linear algebra and its applications*. 2012.
- [37] Joseph M Landsberg. “Tensors: geometry and applications”. In: *Representation theory* 381.402 (2012), p. 3.
- [38] Nick Vannieuwenhoven, Raf Vandebril, and Karl Meerbergen. “A new truncation strategy for the higher-order singular value decomposition”. In: *SIAM Journal on Scientific Computing* 34.2 (2012), A1027–A1052. DOI: [10.1137/110836067](https://doi.org/10.1137/110836067).
- [39] Charles H Bennett et al. “Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels”. In: *Physical review letters* 70.13 (1993), p. 1895. DOI: [10.1103/PhysRevLett.70.1895](https://doi.org/10.1103/PhysRevLett.70.1895).
- [40] William K Wootters and Wojciech H Zurek. “A single quantum cannot be cloned”. In: *Nature* 299.5886 (1982), pp. 802–803. DOI: [10.1038/299802a0](https://doi.org/10.1038/299802a0).
- [41] S Balakrishnan and R Sankaranarayanan. “Measures of operator entanglement of two-qubit gates”. In: *Physical Review A* 83.6 (2011), p. 062320. DOI: [10.1103/PhysRevA.83.062320](https://doi.org/10.1103/PhysRevA.83.062320).
- [42] James Pantaleone. “Neutrino oscillations at high densities”. In: *Physics Letters B* 287.1 (1992), pp. 128–132. ISSN: 0370-2693. DOI: [10.1016/0370-2693\(92\)91887-F](https://doi.org/10.1016/0370-2693(92)91887-F). URL: <http://www.sciencedirect.com/science/article/pii/037026939291887F>.
- [43] James Pantaleone. “Dirac neutrinos in dense matter”. In: *Phys. Rev. D* 46 (2 June 1992), pp. 510–523. DOI: [10.1103/PhysRevD.46.510](https://doi.org/10.1103/PhysRevD.46.510).
- [44] Yong-Zhong Qian and George M Fuller. “Neutrino-neutrino scattering and matter-enhanced neutrino flavor transformation in supernovae”. In: *Physical Review D* 51.4 (1995), p. 1479. DOI: [10.1103/PhysRevD.51.1479](https://doi.org/10.1103/PhysRevD.51.1479).
- [45] Stuart Samuel. “Bimodal coherence in dense self-interacting neutrino gases”. In: *Phys. Rev. D* 53 (10 May 1996), pp. 5382–5393. DOI: [10.1103/PhysRevD.53.5382](https://doi.org/10.1103/PhysRevD.53.5382).
- [46] Sergio Pastor and Georg Raffelt. “Flavor Oscillations in the Supernova Hot Bubble Region: Nonlinear Effects of Neutrino Background”. In: *Phys. Rev. Lett.* 89 (19 Oct. 2002), p. 191101. DOI: [10.1103/PhysRevLett.89.191101](https://doi.org/10.1103/PhysRevLett.89.191101).
- [47] A B Balantekin and H Yüksel. “Neutrino mixing and nucleosynthesis in core-collapse supernovae”. In: *New Journal of Physics* 7 (Feb. 2005), pp. 51–51. DOI: [10.1088/1367-2630/7/1/051](https://doi.org/10.1088/1367-2630/7/1/051).
- [48] Meng-Ru Wu and Irene Tamborra. “Fast neutrino conversions: Ubiquitous in compact binary merger remnants”. In: *Phys. Rev. D* 95 (10 May 2017), p. 103007. DOI: [10.1103/PhysRevD.95.103007](https://doi.org/10.1103/PhysRevD.95.103007).
- [49] Yong-Zhong Qian and George M. Fuller. “Matter-enhanced antineutrino flavor transformation and supernova nucleosynthesis”. In: *Phys. Rev. D* 52 (2 July 1995), pp. 656–660. DOI: <https://doi.org/10.1103/PhysRevD.52.656>.

- [50] Yong-Zhong Qian et al. “Connection between flavor-mixing of cosmologically significant neutrinos and heavy element nucleosynthesis in supernovae”. In: *Phys. Rev. Lett.* 71 (13 Sept. 1993), pp. 1965–1968. DOI: [10.1103/PhysRevLett.71.1965](https://doi.org/10.1103/PhysRevLett.71.1965).
- [51] Hans-Thomas Janka. “Explosion Mechanisms of Core-Collapse Supernovae”. In: *Annual Review of Nuclear and Particle Science* 62.1 (2012), pp. 407–451. DOI: [10.1146/annurev-nucl-102711-094901](https://doi.org/10.1146/annurev-nucl-102711-094901).
- [52] James Pantaleone. “Neutrino oscillations at high densities”. In: *Physics Letters B* 287.1-3 (1992), pp. 128–132. DOI: [10.1016/0370-2693\(92\)91887-F](https://doi.org/10.1016/0370-2693(92)91887-F).
- [53] Huaiyu Duan et al. “Coherent Development of Neutrino Flavor in the Supernova Environment”. In: *Phys. Rev. Lett.* 97 (24 Dec. 2006), p. 241101. DOI: [10.1103/PhysRevLett.97.241101](https://doi.org/10.1103/PhysRevLett.97.241101).
- [54] Huaiyu Duan et al. “Simulation of coherent nonlinear neutrino flavor transformation in the supernova environment: Correlated neutrino trajectories”. In: *Phys. Rev. D* 74 (10 Nov. 2006), p. 105014. DOI: [10.1103/PhysRevD.74.105014](https://doi.org/10.1103/PhysRevD.74.105014).
- [55] Sovan Chakraborty et al. “Collective neutrino flavor conversion: Recent developments”. In: *Nuclear Physics B* 908 (2016), pp. 366–381. DOI: [10.1016/j.nuclphysb.2016.02.012](https://doi.org/10.1016/j.nuclphysb.2016.02.012).
- [56] Michael J. Cervia et al. “Entanglement and collective flavor oscillations in a dense neutrino gas”. In: *Phys. Rev. D* 100 (8 Oct. 2019), p. 083001. DOI: [10.1103/PhysRevD.100.083001](https://doi.org/10.1103/PhysRevD.100.083001).
- [57] Ermal Rrapaj. “Exact solution of multiangle quantum many-body collective neutrino-flavor oscillations”. In: *Phys. Rev. C* 101 (6 June 2020), p. 065805. DOI: [10.1103/PhysRevC.101.065805](https://doi.org/10.1103/PhysRevC.101.065805).
- [58] Amol V. Patwardhan, Michael J. Cervia, and A. B. Balantekin. “Spectral splits and entanglement entropy in collective neutrino oscillations”. In: *Phys. Rev. D* 104 (12 Dec. 2021), p. 123035. DOI: [10.1103/PhysRevD.104.123035](https://doi.org/10.1103/PhysRevD.104.123035).
- [59] Savas Birol et al. “Neutrino spectral split in the exact many-body formalism”. In: *Phys. Rev. D* 98 (8 Oct. 2018), p. 083002. DOI: [10.1103/PhysRevD.98.083002](https://doi.org/10.1103/PhysRevD.98.083002).
- [60] Joshua D. Martin et al. “Classical and quantum evolution in a simple coherent neutrino problem”. In: *Phys. Rev. D* 105 (8 Apr. 2022), p. 083020. DOI: [10.1103/PhysRevD.105.083020](https://doi.org/10.1103/PhysRevD.105.083020).
- [61] Zewei Xiong. “Many-body effects of collective neutrino oscillations”. In: *Phys. Rev. D* 105 (10 May 2022), p. 103002. DOI: [10.1103/PhysRevD.105.103002](https://doi.org/10.1103/PhysRevD.105.103002).
- [62] Alessandro Roggero, Ermal Rrapaj, and Zewei Xiong. “Entanglement and correlations in fast collective neutrino flavor oscillations”. In: *Phys. Rev. D* 106 (4 Aug. 2022), p. 043022. DOI: [10.1103/PhysRevD.106.043022](https://doi.org/10.1103/PhysRevD.106.043022).
- [63] Alessandro Roggero. “Entanglement and many-body effects in collective neutrino oscillations”. In: *Phys. Rev. D* 104 (10 Nov. 2021), p. 103016. DOI: [10.1103/PhysRevD.104.103016](https://doi.org/10.1103/PhysRevD.104.103016).
- [64] Alessandro Roggero. “Dynamical phase transitions in models of collective neutrino oscillations”. In: *Phys. Rev. D* 104 (12 Dec. 2021), p. 123023. DOI: [10.1103/PhysRevD.104.123023](https://doi.org/10.1103/PhysRevD.104.123023).
- [65] Michael J Cervia et al. “Collective neutrino oscillations with tensor networks using a time-dependent variational principle”. In: *Physical Review D* 105.12 (2022), p. 123025. DOI: [10.1103/PhysRevD.105.123025](https://doi.org/10.1103/PhysRevD.105.123025).

- [66] Michael J. Cervia et al. “Collective neutrino oscillations with tensor networks using a time-dependent variational principle”. In: *Phys. Rev. D* 105 (12 June 2022), p. 123025. DOI: [10.1103/PhysRevD.105.123025](https://doi.org/10.1103/PhysRevD.105.123025).
- [67] Benjamin Hall et al. “Simulation of collective neutrino oscillations on a quantum computer”. In: *Physical Review D* 104.6 (2021), p. 063009. DOI: [10.1103/PhysRevD.104.063009](https://doi.org/10.1103/PhysRevD.104.063009).
- [68] Kübra Yeter-Aydeniz et al. “Collective neutrino oscillations on a quantum computer”. In: *Quantum Information Processing* 21.3 (2022), p. 84. DOI: [10.1007/s11128-021-03348-x](https://doi.org/10.1007/s11128-021-03348-x).
- [69] Marc Illa and Martin J. Savage. “Basic elements for simulations of standard-model physics with quantum annealers: Multigrid and clock states”. In: *Phys. Rev. A* 106 (5 Nov. 2022), p. 052605. DOI: [10.1103/PhysRevA.106.052605](https://doi.org/10.1103/PhysRevA.106.052605).
- [70] A Yu Smirnov. “The MSW effect and matter effects in neutrino oscillations”. In: *Physica Scripta* 2005.T121 (2005), p. 57. DOI: [10.1088/0031-8949/2005/T121/008](https://doi.org/10.1088/0031-8949/2005/T121/008).
- [71] H-Th Janka et al. “Theory of core-collapse supernovae”. In: *Physics Reports* 442.1-6 (2007), pp. 38–74.
- [72] Huaiyu Duan, George M. Fuller, and Yong-Zhong Qian. “Collective Neutrino Oscillations”. In: *Annual Review of Nuclear and Particle Science* 60.1 (2010), pp. 569–594. DOI: [10.1146/annurev.nucl.012809.104524](https://doi.org/10.1146/annurev.nucl.012809.104524).
- [73] AB Balantekin and GM Fuller. “Constraints on neutrino mixing”. In: *Physics Letters B* 471.2-3 (1999), pp. 195–201. DOI: [10.1016/S0370-2693\(98\)00137-4](https://doi.org/10.1016/S0370-2693(98)00137-4).
- [74] Pooja Siwach, Anna M Suliga, and A Baha Balantekin. “Entanglement in three-flavor collective neutrino oscillations”. In: *Physical Review D* 107.2 (2023), p. 023019. DOI: [10.1103/PhysRevD.107.023019](https://doi.org/10.1103/PhysRevD.107.023019).
- [75] Ziro Maki, Masami Nakagawa, and Shoichi Sakata. “Remarks on the unified model of elementary particles”. In: *Progress of Theoretical Physics* 28.5 (1962), pp. 870–880.
- [76] Y Pehlivan et al. “Invariants of collective neutrino oscillations”. In: *Physical Review D* 84.6 (2011), p. 065008. DOI: [10.1103/PhysRevD.84.065008](https://doi.org/10.1103/PhysRevD.84.065008).
- [77] Tobias Stirner, Günter Sigl, and Georg Raffelt. “Liouville term for neutrinos: Flavor structure and wave interpretation”. In: *Journal of Cosmology and Astroparticle Physics* 2018.05 (2018), p. 016. DOI: [10.1088/1475-7516/2018/05/016](https://doi.org/10.1088/1475-7516/2018/05/016).
- [78] Valentina Amitrano et al. “Trapped-ion quantum simulation of collective neutrino oscillations”. In: *Physical Review D* 107.2 (2023), p. 023007. DOI: [10.1103/PhysRevD.107.023007](https://doi.org/10.1103/PhysRevD.107.023007).
- [79] Tobias Stirner, Günter Sigl, and Georg Raffelt. “Liouville term for neutrinos: flavor structure and wave interpretation”. In: *Journal of Cosmology and Astroparticle Physics* 2018.05 (May 2018), p. 016. DOI: [10.1088/1475-7516/2018/05/016](https://doi.org/10.1088/1475-7516/2018/05/016).
- [80] Andrew M. Childs, Aaron Ostrander, and Yuan Su. “Faster quantum simulation by randomization”. In: *Quantum* 3 (Sept. 2019), p. 182. ISSN: 2521-327X. DOI: [10.22331/q-2019-09-02-182](https://doi.org/10.22331/q-2019-09-02-182).
- [81] Chi-Fang Chen et al. “Concentration for Random Product Formulas”. In: *PRX Quantum* 2 (4 Oct. 2021), p. 040305. DOI: [10.1103/PRXQuantum.2.040305](https://doi.org/10.1103/PRXQuantum.2.040305).
- [82] J. M. Pino et al. “Demonstration of the trapped-ion quantum CCD computer architecture”. In: *Nature* 592.7853 (Apr. 2021), pp. 209–213. DOI: [10.1038/s41586-021-03318-4](https://doi.org/10.1038/s41586-021-03318-4).

- [83] Dmitri Maslov. “Basic circuit compilation techniques for an ion-trap quantum machine”. In: *New Journal of Physics* 19.2 (2017), p. 023035. DOI: [10.1088/1367-2630/aa5e47](https://doi.org/10.1088/1367-2630/aa5e47).
- [84] Marc Illa and Martin J Savage. “Multi-neutrino entanglement and correlations in dense neutrino systems”. In: *Physical Review Letters* 130.22 (2023), p. 221003. DOI: [10.1103/PhysRevLett.130.221003](https://doi.org/10.1103/PhysRevLett.130.221003).
- [85] Alessandro Roggero et al. “Preparation of excited states for nuclear dynamics on a quantum computer”. In: *Phys. Rev. C* 102 (6 Dec. 2020), p. 064624. DOI: [10.1103/PhysRevC.102.064624](https://doi.org/10.1103/PhysRevC.102.064624).
- [86] Alessandro Roggero et al. “Quantum computing for neutrino-nucleus scattering”. In: *Phys. Rev. D* 101 (7 Apr. 2020), p. 074038. DOI: [10.1103/PhysRevD.101.074038](https://doi.org/10.1103/PhysRevD.101.074038).
- [87] Minh C Tran et al. “Faster digital quantum simulation by symmetry protection”. In: *PRX Quantum* 2.1 (2021), p. 010323. DOI: [10.1103/PRXQuantum.2.010323](https://doi.org/10.1103/PRXQuantum.2.010323).
- [88] William J Huggins et al. “Virtual distillation for quantum error mitigation”. In: *Physical Review X* 11.4 (2021), p. 041036. DOI: [10.1103/PhysRevX.11.041036](https://doi.org/10.1103/PhysRevX.11.041036).
- [89] Xavi Bonet-Monroig et al. “Low-cost error mitigation by symmetry verification”. In: *Physical Review A* 98.6 (2018), p. 062339. DOI: [10.1103/PhysRevA.98.062339](https://doi.org/10.1103/PhysRevA.98.062339).
- [90] Poulami Das et al. “ADAPT: Mitigating Idling Errors in Qubits via Adaptive Dynamical Decoupling”. In: *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*. 2021, pp. 950–962. DOI: [10.1145/3466752.3480059](https://doi.org/10.1145/3466752.3480059).
- [91] Andrew M Childs et al. “Theory of trotter error with commutator scaling”. In: *Physical Review X* 11.1 (2021), p. 011020. DOI: [10.1103/PhysRevX.11.011020](https://doi.org/10.1103/PhysRevX.11.011020).
- [92] Guang Hao Low and Isaac L. Chuang. “Optimal Hamiltonian Simulation by Quantum Signal Processing”. In: *Phys. Rev. Lett.* 118 (1 Jan. 2017), p. 010501. DOI: [10.1103/PhysRevLett.118.010501](https://doi.org/10.1103/PhysRevLett.118.010501).

Appendices

Appendix A

Full Trotter error calculations

A.1 Explicit calculations for the first order Trotter error

The two-body part of the neutrino system Hamiltonian describes the ν - ν scattering and consists of a sum of $\Gamma = \binom{N}{2} = N(N-1)/2$ terms

$$H^{(2b)} = \sum_{i < j}^N h_{ij} := \sum_{K=1}^{\Gamma} h_K, \quad (\text{A.1.1})$$

where $h_{ij} = \mu J_{ij} \boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_j / N$ and we count the neutrinos from 1 to N instead of from 0 to $N-1$. Using the first-order Trotter decomposition, we can implement the propagator (see Eq. (6.2.8) in the main text) using a standard first-order product formula

$$U^{(2b)}(dt) \approx \mathcal{L}_1(dt) := \prod_{i < j}^N e^{-ih_{ij}dt} = \prod_{K=1}^{\Gamma} e^{-ih_K dt}. \quad (\text{A.1.2})$$

and from Proposition 9 of Ref. [91] we bound the first-order Trotter error as

$$\varepsilon_1(dt) = \left\| \mathcal{L}_1(dt) - U^{(2b)}(dt) \right\| \leq \frac{dt^2}{2} \sum_{K=1}^{\Gamma} \left\| \sum_{L=K+1}^{\Gamma} [h_K, h_L] \right\|, \quad (\text{A.1.3})$$

where in our case K and L correspond to the pair indices $K = (i, j)$ and $L = (k, l)$. The sum within the norm in the above expression can be expressed explicitly using the double indices as

$$\begin{aligned} \sum_{L=K+1}^{\Gamma} [h_K, h_L] &= \delta_{ik} \sum_{l=j+1}^N [h_{ij}, h_{kl}] + \sum_{k=i+1}^N \sum_{l=k+1}^N [h_{ij}, h_{kl}] \\ &= \sum_{l=j+1}^N [h_{ij}, h_{il}] + \sum_{k=i+1}^N \sum_{l=k+1}^N [h_{ij}, h_{kl}], \end{aligned} \quad (\text{A.1.4})$$

where we have split the sum for $L > K$ into two contributions: those for which the first index of K is equal to the first index of L and those for which the first index of L is greater than the first index of K . The last contribution can be simplified by splitting the sum around the index j and realising that for the commutator $[h_{ij}, h_{kl}]$ to be different from zero, at least one index in

(k, l) must match an index in (i, j) . So

$$\begin{aligned}
\sum_{k=i+1}^N \sum_{l=k+1}^N [h_{ij}, h_{kl}] &= \left(\sum_{k=i+1}^{j-1} + \delta_{kj} + \sum_{k=j+1}^N \right) \sum_{l=k+1}^N [h_{ij}, h_{kl}] \\
&= \sum_{k=i+1}^{j-1} \sum_{l=k+1}^N [h_{ij}, h_{kl}] + \sum_{l=j+1}^N [h_{ij}, h_{jl}] + \sum_{k=j+1}^N \sum_{l=k+1}^N [h_{ij}, h_{kl}] \\
&= \sum_{k=i+1}^{j-1} \sum_{l=k+1}^N [h_{ij}, h_{kl}] + \sum_{l=j+1}^N [h_{ij}, h_{jl}] \\
&= \sum_{k=i+1}^{j-1} [h_{ij}, h_{kj}] + \sum_{l=j+1}^N [h_{ij}, h_{jl}].
\end{aligned} \tag{A.1.5}$$

Note that the last term in the second step is zero because there is no case where the Hamiltonian pair has a common index (indeed $l > k > j > i$), and the first term in the third step is nonzero only when $l = j$. The Trotter error over a small time step dt is thus

$$\varepsilon_1(dt) \leq \frac{dt^2}{2} \sum_{i < j} \left\| \sum_{k=i+1}^{j-1} [h_{ij}, h_{kj}] + \sum_{l=j+1}^N ([h_{ij}, h_{il}] + [h_{ij}, h_{jl}]) \right\|. \tag{A.1.6}$$

The commutators between different two-body Hamiltonians can be easily calculated as

$$[h_{ij}, h_{ik}] = \frac{\mu^2}{N^2} J_{ij} J_{ik} [\boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_j, \boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_k] = \frac{\mu^2}{N^2} 2i J_{ij} J_{ik} \boldsymbol{\sigma}_i \cdot (\boldsymbol{\sigma}_j \wedge \boldsymbol{\sigma}_k), \tag{A.1.7}$$

where we used $\mathbf{a} \wedge \mathbf{b}$ to denote the standard cross product in three dimensions. So we have

$$\begin{aligned}
\varepsilon_1(dt) &\leq \frac{dt^2}{2} \sum_{i < j} \left\| \sum_{l=j+1}^N ([h_{ij}, h_{il}] + [h_{ij}, h_{jl}]) + \sum_{k=i+1}^{j-1} [h_{ij}, h_{kj}] \right\| \\
&\leq dt^2 \frac{\mu^2}{N^2} \sum_{i < j} \left\| \sum_{l=j+1}^N J_{ij} i (J_{il} \boldsymbol{\sigma}_i \cdot (\boldsymbol{\sigma}_j \wedge \boldsymbol{\sigma}_l) + J_{jl} \boldsymbol{\sigma}_j \cdot (\boldsymbol{\sigma}_i \wedge \boldsymbol{\sigma}_l)) + \sum_{k=i+1}^{j-1} J_{ij} J_{kj} i \boldsymbol{\sigma}_j \cdot (\boldsymbol{\sigma}_i \wedge \boldsymbol{\sigma}_k) \right\| \\
&\leq dt^2 \frac{\mu^2}{N^2} \sum_{i < j} \left\| \sum_{l=j+1}^N (J_{ij} J_{il} - J_{ij} J_{jl}) \boldsymbol{\sigma}_i \cdot (\boldsymbol{\sigma}_j \wedge \boldsymbol{\sigma}_l) + \sum_{k=i+1}^{j-1} J_{ij} J_{kj} \boldsymbol{\sigma}_j \cdot (\boldsymbol{\sigma}_i \wedge \boldsymbol{\sigma}_k) \right\| \\
&\leq dt^2 \frac{\mu^2}{N^2} \sum_{i < j} \left\| J_{ij} \left(\sum_{l=j+1}^N (J_{il} - J_{jl}) \boldsymbol{\sigma}_i \cdot (\boldsymbol{\sigma}_j \wedge \boldsymbol{\sigma}_l) + \sum_{k=i+1}^{j-1} J_{kj} \boldsymbol{\sigma}_j \cdot (\boldsymbol{\sigma}_i \wedge \boldsymbol{\sigma}_k) \right) \right\|,
\end{aligned} \tag{A.1.8}$$

where we used the cyclic permutation equivalence of the cross product. Then considering that the coupling matrix J_{ij} is positive and using the bound $\|\boldsymbol{\sigma}_i \cdot (\boldsymbol{\sigma}_j \wedge \boldsymbol{\sigma}_k)\|_\infty \leq 4$ for all $i, j, k \in [1, N]$ one arrives at

$$\varepsilon_1(dt) \leq 4dt^2 \frac{\mu^2}{N^2} \sum_{i < j} J_{ij} \left\| \sum_{l=j+1}^N (J_{il} - J_{jl}) + \sum_{k=i+1}^{j-1} J_{kj} \right\|. \tag{A.1.9}$$

The number of terms in the sums is:

$$\sum_{i < j} \left(2 \sum_{l=j+1}^N + \sum_{k=i+1}^{j-1} \right) = 2 \sum_{i < j < l} + \sum_{i < k < j} = 3 \binom{N}{3}, \tag{A.1.10}$$

where

$$\binom{N}{k} = \frac{N!}{k!(N-k)!} \quad (\text{A.1.11})$$

is the binomial coefficient and corresponds to the number of possible combinations of 3 elements taken from N elements. For a particular choice of angular distributions, the sums can be computed straightforwardly. However, to obtain a general bound on the error, we can introduce

$$\Theta := \max_{i,j} (1 - \cos(\theta_{ij})) \quad (\text{A.1.12})$$

and obtain the upperbound

$$\varepsilon_1(dt) \leq 12dt^2 \frac{\mu^2 \Theta^2}{N^2} \binom{N}{3} = \mathcal{O}(dt^2 \mu^2 N). \quad (\text{A.1.13})$$

Given a total evolution time T and evolving the system for $r = T/dt$ steps, the total error can be bounded by

$$\varepsilon_1(T) \leq r\varepsilon_1(dt) \leq r12 \frac{T^2 \mu^2 \Theta^2}{r^2 N^2} \binom{N}{3} = 12 \frac{T^2 \mu^2 \Theta^2}{r N^2} \binom{N}{3}, \quad (\text{A.1.14})$$

where we used the union bound to obtain $\varepsilon_1(T) \leq r\varepsilon_1(dt)$. We also find that the total number of steps $r_1(T, \epsilon)$ required to evolve for a final time T while keeping the total error below ϵ scales linearly with system size, namely

$$r_1(T, \epsilon) \leq 12 \frac{T^2 \mu^2 \Theta^2}{\epsilon N^2} \binom{N}{3} = \mathcal{O}\left(\frac{T^2 \mu^2}{\epsilon} N\right). \quad (\text{A.1.15})$$

We can then bound the gate cost $\mathcal{C}_1(T, \epsilon)$ of a quantum circuit by implementing the full evolution in terms of the number of general two-qubit gates $\in \mathcal{SU}(4)$ needed to implement all the steps. For N neutrinos, we have $N(N-1)/2$ interaction terms in each Trotter step, so the gate cost, using a first-order product formula, scales with

$$\begin{aligned} \mathcal{C}_1(T, \epsilon) &\leq \binom{N}{2} r_1(T, \epsilon) = 12 \frac{T^2 \mu^2 \Theta^2}{\epsilon N^2} \binom{N}{3} \binom{N}{2} \\ &= \frac{T^2 \mu^2 \Theta^2}{\epsilon N^2} 12 \frac{N(N-1)(N-2)}{3!} \frac{N(N-1)}{2!} \\ &= \frac{T^2 \mu^2 \Theta^2}{\epsilon N^2} 12 \frac{N^2(N-1)^2(N-2)}{12} \\ &= \frac{T^2 \mu^2 \Theta^2}{\epsilon} (N-1)^2(N-2) = \mathcal{O}\left(\frac{T^2 \mu^2}{\epsilon} N^3\right). \end{aligned} \quad (\text{A.1.16})$$

Since any two-qubit transformation $\in \mathcal{SU}(4)$ can be decomposed with at most three entangling gates [17], as we did for the native gate set available on the QSM H1-2 system, the total number of two-qubit gates is given by $3\mathcal{C}_1(T, \epsilon)$.

A.2 Second order Trotter error for the interaction Hamiltonian

It is possible to get a more accurate approximation of $U^{(2b)}(dt)$ using a second order Trotter-Suzuki formula, which can be expressed as

$$\begin{aligned} U^{(2b)}(dt) &\approx \mathcal{L}_2(dt) = \prod_{L=\Gamma}^1 e^{-i\frac{dt}{2}h_L} \prod_{K=1}^{\Gamma} e^{-i\frac{dt}{2}h_K} \\ &= \mathcal{L}_1\left(\frac{dt}{2}\right) \mathcal{L}_1^\dagger\left(-\frac{dt}{2}\right), \end{aligned} \quad (\text{A.2.1})$$

where we have used the multi-index notation $K = (i, j)$, $L = (k, l)$ and $\Gamma = N(N - 1)/2$ as before. Using the result of Proposition 10 from Ref. [91], the second order Trotter error can be bounded by

$$\varepsilon_2(dt) \leq \frac{dt^3}{12} \sum_K^\Gamma \left\| \sum_{L>K}^\Gamma \sum_{M>K}^\Gamma [h_L, [h_M, h_K]] \right\| + \frac{dt^3}{24} \sum_K^\Gamma \left\| \sum_{L>K}^\Gamma [h_K, [h_K, h_L]] \right\|. \quad (\text{A.2.2})$$

To bound the second term we use a similar procedure to the first order case by expanding the sums and keeping contributions $[h_{ij}, [h_{ij}, h_{kl}]]$ with one of the (k, l) indices matching one of the (i, j) indices. Translating the sums into a double index sum ($K = (i, j)$ and $L = (k, l)$), the second term becomes

$$\begin{aligned} & \frac{dt^3}{24} \sum_K^\Gamma \left\| \sum_{L>K}^\Gamma [h_K, [h_K, h_L]] \right\| = \\ & = \frac{dt^3}{24} \sum_{i<j}^N \left\| \left(\sum_{l=j+1}^N \delta_{ki} + \sum_{k=i+1}^N \sum_{l=k+1}^N \right) [h_{ij}, [h_{ij}, h_{kl}]] \right\| \\ & = \frac{dt^3}{24} \sum_{i<j}^N \left\| \sum_{l=j+1}^N [h_{ij}, [h_{ij}, h_{il}]] + \left(\sum_{k=i+1}^{j-1} + \delta_{kj} + \sum_{k=j+1}^N \right) \sum_{l=k+1}^N [h_{ij}, [h_{ij}, h_{kl}]] \right\| \quad (\text{A.2.3}) \\ & = \frac{dt^3}{24} \sum_{i<j}^N \left\| \sum_{l=j+1}^N [h_{ij}, [h_{ij}, h_{il}]] + \sum_{k=i+1}^{j-1} \sum_{l=k+1}^N [h_{ij}, [h_{ij}, h_{kl}]] + \sum_{l=j+1}^N [h_{ij}, [h_{ij}, h_{jl}]] \right\| \\ & = \frac{dt^3}{24} \sum_{i<j}^N \left\| \sum_{l=j+1}^N [h_{ij}, [h_{ij}, h_{il}]] + \sum_{k=i+1}^{j-1} [h_{ij}, [h_{ij}, h_{kj}]] + \sum_{l=j+1}^N [h_{ij}, [h_{ij}, h_{jl}]] \right\|. \end{aligned}$$

We now use the following expression for the nested commutator

$$\begin{aligned} [h_{ij}, [h_{ij}, h_{ik}]] &= \frac{\mu^3}{N^3} J_{ij}^2 J_{ik} [\boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_j, [\boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_j, \boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_k]] \\ &= \frac{\mu^3}{N^3} 2i J_{ij}^2 J_{ik} [\boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_j, \boldsymbol{\sigma}_i \cdot (\boldsymbol{\sigma}_j \wedge \boldsymbol{\sigma}_k)] \quad (\text{A.2.4}) \\ &= -4 \frac{\mu^3}{N^3} J_{ij}^2 J_{ik} \boldsymbol{\sigma}_i \cdot (\boldsymbol{\sigma}_j \wedge (\boldsymbol{\sigma}_j \wedge \boldsymbol{\sigma}_k)), \end{aligned}$$

together with the bound $\|\boldsymbol{\sigma}_i \cdot (\boldsymbol{\sigma}_j \wedge (\boldsymbol{\sigma}_j \wedge \boldsymbol{\sigma}_k))\| \leq 8$. Considering that the above sums contain $3\binom{N}{3}$ terms, the result is

$$\begin{aligned} \frac{dt^3}{24} \sum_K^\Gamma \left\| \sum_{L>K}^\Gamma [h_K, [h_K, h_L]] \right\| &\leq \frac{dt^3}{24} 4 \cdot 8 \frac{\mu^3 \Theta^3}{N^3} 3\binom{N}{3} = 4dt^3 \frac{\mu^3 \Theta^3}{N^3} \binom{N}{3} \\ &= \frac{2(N-1)(N-2)}{3N^2} dt^3 \mu^3 \Theta^3. \quad (\text{A.2.5}) \end{aligned}$$

We now turn our attention to the first contribution in Eq. (A.2.2). Using the double indices $K = (i, j)$, $L = (k, l)$ and $M = (p, q)$ we have

$$\begin{aligned} & \frac{dt^3}{12} \sum_K^\Gamma \left\| \sum_{L>K}^\Gamma \sum_{M>K}^\Gamma [h_L, [h_M, h_K]] \right\| = \\ & = \frac{dt^3}{12} \sum_{i<j}^N \left\| \left(\sum_{l=j+1}^N \delta_{ki} + \sum_{k=i+1}^N \sum_{l=k+1}^N \right) \left(\sum_{q=j+1}^N \delta_{pi} + \sum_{p=i+1}^N \sum_{q=p+1}^N \right) [h_{kl}, [h_{pq}, h_{ij}]] \right\|, \quad (\text{A.2.6}) \end{aligned}$$

which can be bound as the sum of the different norms. The first term in Eq. (A.2.6) means to contribute as

$$\begin{aligned}
& \sum_{i < j} \sum_{l=j+1}^N \sum_{q=j+1}^N \|[h_{il}, [h_{iq}, h_{ij}]]\| = \\
& = \sum_{i < j}^N \left(\sum_{l=j+1}^N \delta_{lq} \|[h_{il}, [h_{iq}, h_{ij}]]\| + \sum_{l=j+1}^q \sum_{q=l+1}^N \|[h_{il}, [h_{iq}, h_{ij}]]\| + \sum_{q=j+1}^l \sum_{l=q+1}^N \|[h_{il}, [h_{iq}, h_{ij}]]\| \right) \\
& = \sum_{i < j < l}^N \|[h_{il}, [h_{il}, h_{ij}]]\| + \sum_{i < j < l < q}^N \|[h_{il}, [h_{iq}, h_{ij}]]\| + \sum_{i < j < q < l}^N \|[h_{il}, [h_{iq}, h_{ij}]]\|,
\end{aligned} \tag{A.2.7}$$

where the first norm contains a sum of combinations of 3 indices out of N and gives a contribution of $\binom{N}{3}$. The second one contains 4 indices and gives $\binom{N}{4}$ and the same for the last one which contains $\binom{N}{4}$ elements. So the result can be bounded by

$$\begin{aligned}
\frac{dt^3}{12} \sum_{i < j}^N \left\| \sum_{l=j+1}^N \sum_{q=j+1}^N \|[h_{il}, [h_{iq}, h_{ij}]]\| \right\| & \leq \frac{dt^3}{12} 32 \frac{\mu^3}{N^3} \Theta^3 \left(2 \binom{N}{4} + \binom{N}{3} \right) \\
& = \frac{2(N-1)^2(N-2)}{9N^2} dt^3 \mu^3 \Theta^3.
\end{aligned} \tag{A.2.8}$$

The second contribution in Eq. (A.2.6) is

$$\begin{aligned}
& \sum_{i < j}^N \sum_{l=j+1}^N \sum_{p=i+1}^N \sum_{q=p+1}^N \|[h_{il}, [h_{pq}, h_{ij}]]\| = \\
& = \sum_{i < j}^N \sum_{l=j+1}^N \left(\sum_{p=i+1}^{j-1} + \delta_{pj} + \sum_{p=j+1}^N \right) \sum_{q=p+1}^N \|[h_{il}, [h_{pq}, h_{ij}]]\| \\
& = \sum_{i < j}^N \left(\sum_{l=j+1}^N \sum_{p=i+1}^{j-1} \sum_{q=p+1}^N \|[h_{il}, [h_{pq}, h_{ij}]]\| + \sum_{l=j+1}^N \sum_{q=j+1}^N \|[h_{il}, [h_{jq}, h_{ij}]]\| + \right. \\
& \quad \left. + \sum_{l=j+1}^N \sum_{p=j+1}^N \sum_{q=p+1}^N \|[h_{il}, [h_{pq}, h_{ij}]]\| \right) \\
& = \sum_{i < j}^N \left(\sum_{l=j+1}^N \sum_{p=i+1}^{j-1} \|[h_{il}, [h_{pj}, h_{ij}]]\| + \sum_{l=j+1}^N \sum_{q=j+1}^N \|[h_{il}, [h_{jq}, h_{ij}]]\| \right) \\
& = \sum_{i < p < j < l}^N \|[h_{il}, [h_{jq}, h_{ij}]]\| + \sum_{i < j < q, l}^N \|[h_{il}, [h_{jq}, h_{ij}]]\|.
\end{aligned} \tag{A.2.9}$$

The first term gives $\binom{N}{4}$ and the second includes $i < j < q < l$ and $i < j < l < q$ giving $2\binom{N}{4}$ and also $i < j < l = q$ which adds $\binom{N}{3}$. So the above contribution can be bound by

$$\begin{aligned}
\frac{dt^3}{12} \sum_{i < j}^N \left\| \sum_{l=j+1}^N \sum_{p=i+1}^N \sum_{q=p+1}^N \|[h_{il}, [h_{pq}, h_{ij}]]\| \right\| & \leq \frac{dt^3}{12} 32 \frac{\mu^3}{N^3} \Theta^3 \left(3 \binom{N}{4} + \binom{N}{3} \right) \\
& = \frac{1(N-1)(N-2)}{9N^2} (3N-5) dt^3 \mu^3 \Theta^3.
\end{aligned} \tag{A.2.10}$$

The third contribution in Eq. (A.2.6) gives

$$\begin{aligned}
& \sum_{i < j}^N \sum_{k=i+1}^N \sum_{l=k+1}^N \sum_{q=j+1}^N \|[h_{kl}, [h_{iq}, h_{ij}]]\| = \\
&= \sum_{i < j}^N \left(\sum_{k=i+1}^{j-1} + \delta_{kj} + \sum_{k=j+1}^N \right) \sum_{l=k+1}^N \sum_{q=j+1}^N \|[h_{kl}, [h_{iq}, h_{ij}]]\| \\
&= \sum_{i < j}^N \left(\sum_{k=i+1}^{j-1} \sum_{l=k+1}^N \sum_{q=j+1}^N \|[h_{kl}, [h_{iq}, h_{ij}]]\| + \sum_{l=j+1}^N \sum_{q=j+1}^N \|[h_{jl}, [h_{iq}, h_{ij}]]\| + \right. \\
&\quad \left. + \sum_{k=j+1}^N \sum_{l=k+1}^N \sum_{q=j+1}^N \|[h_{kl}, [h_{iq}, h_{ij}]]\| \right) \\
&= \sum_{i < j}^N \left(\sum_{k=i+1}^{j-1} \sum_{q=j+1}^N \|[h_{kj}, [h_{iq}, h_{ij}]]\| + \sum_{k=i+1}^{j-1} \sum_{q=j+1}^N \|[h_{kq}, [h_{iq}, h_{ij}]]\| + \right. \\
&\quad + \sum_{l=j+1}^N \sum_{q=j+1}^N \|[h_{jl}, [h_{iq}, h_{ij}]]\| + \sum_{k=j+1}^N \sum_{l=k+1}^N \|[h_{kl}, [h_{ik}, h_{ij}]]\| + \\
&\quad \left. + \sum_{k=j+1}^N \sum_{l=k+1}^N \|[h_{kl}, [h_{il}, h_{ij}]]\| \right) \\
&= \sum_{i < j}^N \left(\sum_{k=i+1}^{j-1} \sum_{q=j+1}^N (\|[h_{kj}, [h_{iq}, h_{ij}]]\| + \|[h_{kq}, [h_{iq}, h_{ij}]]\|) + \sum_{l=j+1}^N \sum_{q=j+1}^N \|[h_{jl}, [h_{iq}, h_{ij}]]\| \right. \\
&\quad \left. + \sum_{k=j+1}^N \sum_{l=k+1}^N (\|[h_{kl}, [h_{ik}, h_{ij}]]\| + \|[h_{kl}, [h_{il}, h_{ij}]]\|) \right) \\
&= \left(2 \sum_{i < k < j < q}^N + \sum_{i < j < l, q}^N + 2 \sum_{i < j < k < l}^N \right) \|[h_{kl}, [h_{il}, h_{ij}]]\|.
\end{aligned} \tag{A.2.11}$$

The first sum gives $2\binom{N}{4}$, the second $\binom{N}{3}$, which corresponds to $i < j < l = q$ and $2\binom{N}{4}$ to $i < j < l < q$ and $i < j < q < l$. The last one will give us $2\binom{N}{4}$. So we can rewrite it as

$$\begin{aligned}
\frac{dt^3}{12} \sum_{i < j}^N \left\| \sum_{k=i+1}^N \sum_{l=k+1}^N \sum_{q=j+1}^N \|[h_{kl}, [h_{iq}, h_{ij}]]\| \right\| &\leq \frac{dt^3}{12} 32 \frac{\mu^3}{N^3} \Theta^3 \left(6 \binom{N}{4} + \binom{N}{3} \right) \\
&= \frac{1}{9} \frac{(N-1)(N-2)}{N^2} (6N-14) dt^3 \mu^3 \Theta^3. \tag{A.2.12}
\end{aligned}$$

Finally the last contribution in Eq. (A.2.6) is

$$\begin{aligned}
 & \sum_{i < j}^N \sum_{k=i+1}^N \sum_{l=k+1}^N \sum_{p=i+1}^N \sum_{q=p+1}^N \|[h_{kl}, [h_{pq}, h_{ij}]]\| = \\
 & = \sum_{i < j}^N \sum_{k=i+1}^N \sum_{l=k+1}^N \left(\sum_{p=i+1}^{j-1} + \delta_{pj} + \sum_{p=j+1}^N \right) \sum_{q=p+1}^N \|[h_{kl}, [h_{pq}, h_{ij}]]\| \\
 & = \sum_{i < j}^N \left(\sum_{k=i+1}^N \sum_{l=k+1}^N \sum_{p=i+1}^{j-1} \sum_{q=p+1}^N \|[h_{kl}, [h_{pq}, h_{ij}]]\| + \right. \\
 & \quad \left. + \sum_{k=i+1}^N \sum_{l=k+1}^N \sum_{q=j+1}^N \|[h_{kl}, [h_{jq}, h_{ij}]]\| + \sum_{k=i+1}^N \sum_{l=k+1}^N \sum_{p=j+1}^N \sum_{q=p+1}^N \|[h_{kl}, [h_{pq}, h_{ij}]]\| \right) \\
 & = \sum_{i < j}^N \left(\sum_{k=i+1}^N \sum_{l=k+1}^N \sum_{p=i+1}^{j-1} \|[h_{kl}, [h_{pj}, h_{ij}]]\| + \sum_{k=i+1}^N \sum_{l=k+1}^N \sum_{q=j+1}^N \|[h_{kl}, [h_{jq}, h_{ij}]]\| \right) \\
 & = \sum_{i < j}^N \left[\left(\sum_{k=i+1}^{j-1} + \delta_{kj} + \sum_{k=j+1}^N \right) \sum_{l=k+1}^N \sum_{p=i+1}^{j-1} \|[h_{kl}, [h_{pj}, h_{ij}]]\| + \right. \\
 & \quad \left. + \left(\sum_{k=i+1}^{j-1} + \delta_{kj} + \sum_{k=j+1}^N \right) \sum_{l=k+1}^N \sum_{q=j+1}^N \|[h_{kl}, [h_{jq}, h_{ij}]]\| \right] \\
 & = \sum_{i < j}^N \left[\left(\sum_{k=i+1}^{j-1} (\delta_{kp} + \delta_{lj}) + \delta_{kj} \right) \sum_{l=k+1}^N \sum_{p=i+1}^{j-1} \|[h_{kl}, [h_{pj}, h_{ij}]]\| + \right. \\
 & \quad \left. + \left(\sum_{k=i+1}^{j-1} \delta_{lj} + \delta_{kj} + \sum_{k=k+1}^N (\delta_{qk} + \delta_{ql}) \right) \sum_{l=k+1}^N \sum_{q=j+1}^N \|[h_{kl}, [h_{jq}, h_{ij}]]\| \right] \\
 & = \sum_{i < j}^N \left(\sum_{k=i+1}^{j-1} \sum_{l=k+1}^N \|[h_{kl}, [h_{kj}, h_{ij}]]\| + \sum_{k=i+1}^{j-1} \sum_{p=i+1}^{j-1} \|[h_{kj}, [h_{pj}, h_{ij}]]\| + \sum_{l=j+1}^N \sum_{p=i+1}^{j-1} \|[h_{jl}, [h_{pj}, h_{ij}]]\| + \right. \\
 & \quad \left. + \sum_{k=i+1}^{j-1} \sum_{q=j+1}^N \|[h_{kj}, [h_{jq}, h_{ij}]]\| + \sum_{l=j+1}^N \sum_{q=j+1}^N \|[h_{jl}, [h_{jq}, h_{ij}]]\| + \right. \\
 & \quad \left. + \sum_{k=j+1}^N \sum_{l=k+1}^N (\|[h_{kl}, [h_{jk}, h_{ij}]]\| + \|[h_{kl}, [h_{jl}, h_{ij}]]\|) \right). \tag{A.2.13}
 \end{aligned}$$

The first term includes the sum over $i < k < j < l$ and over $i < k < l < j$ and $i < k < j = l$ giving $2\binom{N}{4} + \binom{N}{3}$. The second allows $i < k < p < j$, $i < p < k < j$ and $i < k = p < j$ so it contains a number of terms of $2\binom{N}{4} + \binom{N}{3}$. The third allows only $i < p < j < l$ which gives $\binom{N}{4}$. The fourth includes $i < k < j < q$ giving $\binom{N}{4}$, the fifth has $i < j < l < q$, $i < j < q < l$ and $i < j < q = l$ giving $2\binom{N}{4} + \binom{N}{3}$, and finally the last term has $i < j < k < l$ giving $2\binom{N}{4}$. So we

bound it as

$$\begin{aligned} \frac{dt^3}{12} \sum_{i < j} \left\| \sum_{k=i+1}^N \sum_{l=k+1}^N \sum_{p=i+1}^N \sum_{q=p+1}^N \|[h_{kl}, [h_{pq}, h_{ij}]]\| \right\| &\leq \frac{dt^3}{12} 32 \frac{\mu^3}{N^3} \Theta^3 \left(10 \binom{N}{4} + 3 \binom{N}{3} \right) \\ &= \frac{2(N-1)(N-2)}{9} \frac{(5N-9) dt^3 \mu^3 \Theta^3}{N^2}. \end{aligned} \quad (\text{A.2.14})$$

The sum of all four contributions (Eqs. (A.2.8), (A.2.10), (A.2.12) and (A.2.14)) transforms Eq. (A.2.6) into

$$\begin{aligned} \frac{dt^3}{12} \sum_K^\Gamma \left\| \sum_{L>K}^\Gamma \sum_{M>K}^\Gamma [h_L, [h_M, h_K]] \right\| &\leq \frac{dt^3}{12} 32 \frac{\mu^3}{N^3} \Theta^3 \left(21 \binom{N}{4} + 6 \binom{N}{3} \right) \\ &= \frac{1}{9} dt^3 \mu^3 \Theta^3 \frac{(N-1)(N-2)}{N^2} (21N-39). \end{aligned} \quad (\text{A.2.15})$$

Finally, by summing Eq. (A.2.5) and Eq. (A.2.15), we obtain that the second order Trotter error in Eq. (A.2.2) scales linearly with N . In particular,

$$\begin{aligned} \varepsilon_2(dt) &\leq 4dt^3 \frac{\mu^3 \Theta^3}{N^3} \binom{N}{3} + \frac{8}{3} dt^3 \frac{\mu^3 \Theta^3}{N^3} \left(21 \binom{N}{4} + 6 \binom{N}{3} \right) \\ &\leq dt^3 \frac{\mu^3 \Theta^3}{N^3} \left(20 \binom{N}{3} + 56 \binom{N}{4} \right) \\ &\leq dt^3 \mu^3 \Theta^3 \frac{(N-1)(N-2)}{N^2} \frac{(7N-11)}{3} = \mathcal{O}(dt^3 \mu^3 N). \end{aligned} \quad (\text{A.2.16})$$

Developing the system up to T in r_2 steps, the total error satisfies

$$\varepsilon_2(T) \leq r_2 \varepsilon_2(dt) = \frac{T^3}{r_2^2} \frac{\mu^3 \Theta^3}{N^3} \left(20 \binom{N}{3} + 56 \binom{N}{4} \right) = \mathcal{O}\left(\frac{T^3 \mu^3 N}{r_2^2}\right), \quad (\text{A.2.17})$$

and the necessary steps to have a target error below ϵ are

$$r_2(T, \epsilon) \leq \frac{(T\mu\Theta)^{3/2}}{\sqrt{\epsilon} N^{3/2}} \sqrt{20 \binom{N}{3} + 56 \binom{N}{4}} = \mathcal{O}\left(\frac{T^{3/2} \mu^{3/2} \sqrt{N}}{\sqrt{\epsilon}}\right). \quad (\text{A.2.18})$$

The number of two-qubit general operations $\in \mathcal{SU}(4)$ needed to implement the interaction Hamiltonian for each second order Trotter formula is

$$2 \binom{N}{2} - \frac{N}{2} = \frac{N(2N-3)}{2}, \quad (\text{A.2.19})$$

because it is given by two steps with $dt/2$ in reverse order, minus the layer in the middle, as shown in Fig. 6.16 of the main text. The complexity is thus

$$\begin{aligned} \mathcal{C}_2(T, \epsilon) &\leq \left(2 \binom{N}{2} - \frac{N}{2} \right) r_2(T, \epsilon) = \frac{(T\mu\Theta)^{3/2}}{\sqrt{\epsilon}} \frac{1}{N^{3/2}} \sqrt{20 \binom{N}{3} + 56 \binom{N}{4}} \frac{N(2N-3)}{2} \\ &= \frac{(T\mu\Theta)^{3/2}}{\sqrt{\epsilon}} \frac{1}{N^{3/2}} \sqrt{N(N-1)(N-2)} \frac{(7N-11)}{3} \frac{N(2N-3)}{2} \\ &= \frac{(T\mu\Theta)^{3/2}}{\sqrt{\epsilon}} \sqrt{(N-1)(N-2)} \frac{(7N-11)}{3} \frac{(2N-3)}{2} \\ &= \mathcal{O}\left(\frac{(T\mu)^{3/2}}{\sqrt{\epsilon}} N^{5/2}\right). \end{aligned} \quad (\text{A.2.20})$$

The scaling with the number of neutrinos N has improved considerably within this scheme. The use of even higher order formulas may allow one to achieve a near optimal scaling $\mathcal{C}(N^{2+\delta})$ for $\delta \ll 1$, but with possibly much larger constant prefactors.