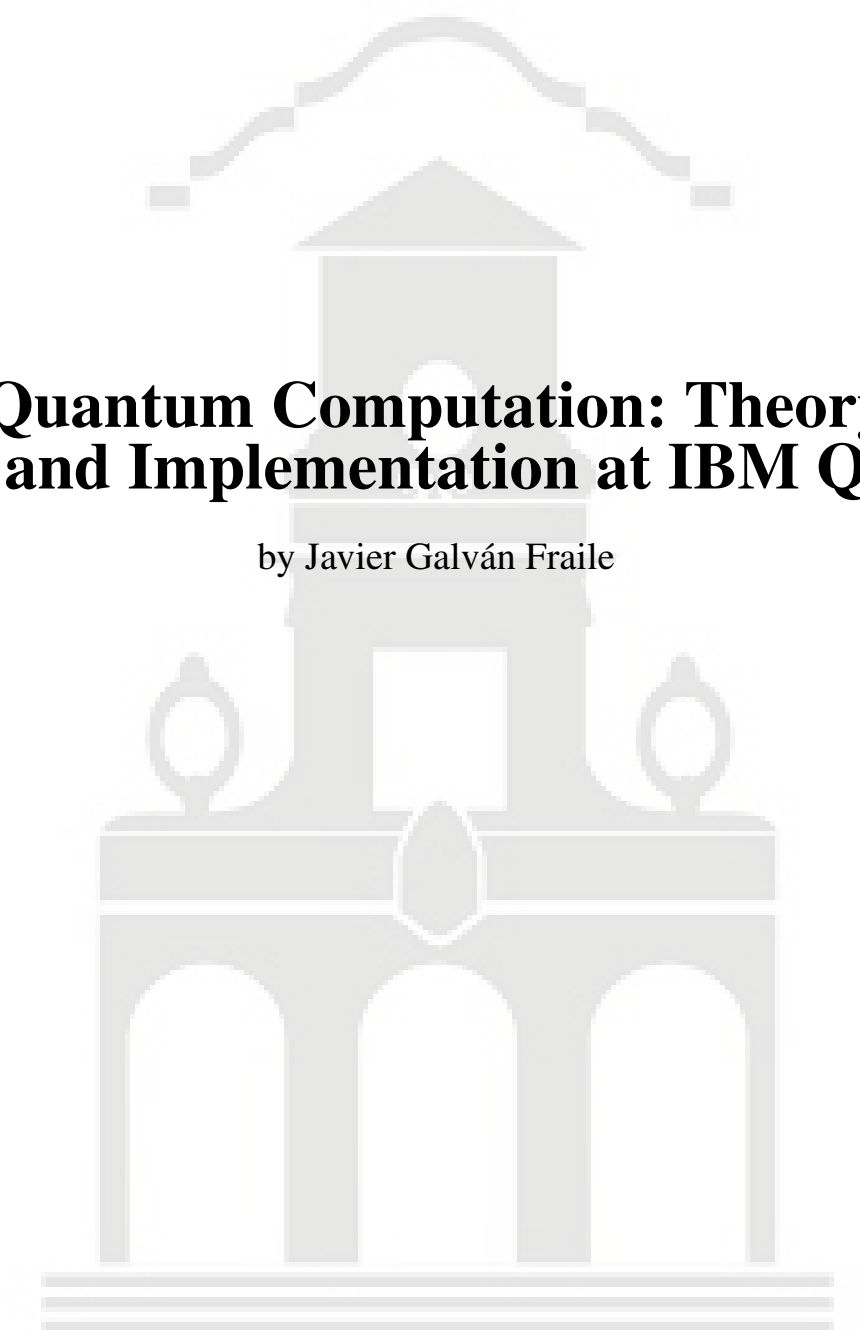


# Quantum Computation: Theory and Implementation at IBM Q

by Javier Galván Fraile

Universidad de La Laguna
July 2019


# Quantum Computation: Theory and Implementation at IBM Q

by
Javier Galván Fraile

**Supervised by:**
Dr. Juan José García Ripoll (CSIC)
Dr. Daniel Alonso Ramírez (ULL)

July 2019

"Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy."

— Richard P. Feynman, *Simulating Physics with Computers* [Fey82]

## Abstract

# Quantum Computation: Theory and Implementation at IBM Q

This end of degree project constitutes an introduction to Quantum Computation. It presents a combination of theoretical concepts, mainly based in the guidelines of "Quantum Computation and Quantum Information" of Michael A. Nielsen & Isaac L. Chuang [NC02], and the implementation of some of them at IBM's online quantum computers [ibm18]. The aim is therefore to realize a first approach to some basic concepts of Quantum Computation and Quantum Information and put them in practice. Particularly, after the introduction of qubits and essential ideas about entanglement and multiple qubit states, the 14-qubit quantum computer *IBM Q Melbourne* was employed to generate both Bell and GHZ states. After that, a quantum/classical hybrid algorithm known as *Variational Quantum Eigensolver (VQE)* [Com16] was introduced as a crucial tool for the next two targets of the project. The first of them consists on exhaustively analyzing and solving an optimization problem named *Exact cover problem* [Gal17]. The second one relates to find the ground state of a bidimensional Ising model and study the evolution of bipartite entanglement, as measured by the von Neumann entropy, in the approach of the system to its ground state.

**Keyword:** Quantum computation, Variational Quantum Eigensolver (VQE), Exact cover, Ising model, Entanglement entropy.

## Resumen

# Quantum Computation: Theory and Implementation at IBM Q

El presente trabajo de fin de grado constituye una introducción a la Computación Cuántica. En él se presenta una combinación de conceptos teóricos, basados principalmente en las directrices del libro "Quantum Computation and Quantum Information" of Michael A. Nielsen & Isaac L. Chuang [NC02], y la implemantación de algunos de ellos en los ordenadores cuánticos de IBM disponibles en la nube [ibm18]. El objetivo del proyecto es, por tanto, realizar una primera aproximación a algunos de los conceptos básicos de la Computación Cuántica e Información Cuántica y ponerlos en práctica. En particular, tras introducir el concepto de cúbit y las ideas esenciales sobre entrelazamiento y estados formados por multiples cúbits, el ordenador cuántico de 14 cúbits *IBM Q Melbourne* fue empleado para generar estados de Bell y GHZ. Tras esto, se introdujo un algoritmo híbrido clásico/cuántico conocido como *Variational Quantum Eigensolver (VQE)* [Com16], el cual es una herramienta fundamental a la hora de cumplimentar los siguientes dos objetivos del proyecto. El primero de ellos consiste en el análisis exhaustivo y la resolución de un problema de optimización conocido como *Exact cover problem* [Gal17]. El segundo se fundamenta en encontrar el estado fundamental de un modelo de Ising bidimensional y estudiar la evolución del entrelazamiento bipartito, medido mediante la entropía de von Neumann, en la evolución del sistema hacia el estado fundamental.

**Palabras clave:** Quantum computation, Variational Quantum Eigensolver (VQE), Exact cover, Ising model, Entanglement entropy.

# Preface

This work carried out by the undergraduate Javier Galván Fraile constitutes his Final Degree Project at the Physics degree of the University of La Laguna. It has mainly been developed at the Fundamental Physics Institute (CSIC, Madrid) under the framework of a JAE Intro 2018 scholarship and the mentoring of Dr. Juan José García Ripoll. This scholarship consist of 300 hours of research stay and was fulfilled between February and April. Also mentoring of Dr. Daniel Alonso Ramírez has been received while being in the island. During the stay at the center, an introduction to quantum computation and quantum information was undertaken, as well as a brief study of some applications of these technologies like solving optimization problems and improving the performance of Machine Learning techniques. For this purpose, the IBM's open-source framework for quantum computing, *Qiskit*, was employed and all the work developed in several Jupyter Notebooks is available at Github [GF19]. In the same way, the 55 hour course "Machine Learning" of the University of Standford taught online at Coursera [Ng16], was accomplished with success. Moreover, many bibliography was explored with particular interest in "Quantum Computation and Quantum Information" of Michael A. Nielsen & Isaac L. Chuang [NC02]. Therefore, the aim of this project is an educated introduction to quantum computation and quantum information theory.

# Acknowledgements

Quiero agradecer en primer lugar al Científico Titular Juan José García Ripoll, porque sin su dirección y apoyo no habría podido realizar este Trabajo de Fin de Grado, además sus grandes conocimientos en la Computación Cuántica y su capacidad para transmitirlos, me ha permitido conocer esta disciplina tan apasionante y haber pasado dos meses y medio de trabajo intenso tanto en el campo científico como en el personal.

También quiero agradecer al Catedrático Daniel Alonso Ramírez por haberme orientado y aconsejado desde su experiencia, despertando en mí el interés por la investigación. No sabe lo importante que ha sido para mí en estos años de dudas propias de alguien que está empezando, el tener una persona como referente y que siempre ha estado dispuesto a ayudarme.

No puedo dejar de nombrar al grupo del Instituto de Física Fundamental, en particular a Alejandro Valido Flores y Diego González Olivares pues me ayudaron mucho en la estancia que hice en dicho Instituto.

A mis compañeros de estudio, de alegrías y de fatiga Miguel, Raquel y Edu, gracias.

Por último, a mi madre, padre y hermano porque siempre han estado ahí cuando los he necesitado, gracias.

# Contents

**Preface** . . . . . **iv**

**Acknowledgements** . . . . . **v**

**1. Global overview and state of art** . . . . . **1**

**2. Introduction** . . . . . **6**
2.1 Quantum bits . . . . . 7
2.1.1 The Qubit . . . . . 7
2.1.2 Single qubit gates . . . . . 8
2.2 Multiple qubits . . . . . 10
2.2.1 Entanglement . . . . . 10
2.2.2 Bell and GHZ states . . . . . 11
2.2.3 Multiple qubit gates . . . . . 11
2.3 Quantum circuit . . . . . 12
2.4 Bell and GHZ states at IMB Q . . . . . 12

**3. Quantum algorithms** . . . . . **16**
3.1 Quantum vs classical algorithms . . . . . 17
3.2 Variational Quantum Eigensolver (VQE) . . . . . 17
3.2.1 Variational method . . . . . 17
3.2.2 The algorithm . . . . . 19
3.3 VQE in Qiskit . . . . . 20
3.3.1 Initial state . . . . . 20
3.3.2 Trial wave functions . . . . . 20
3.3.3 Classical optimizers . . . . . 22

**4. Quantum optimization** . . . . . **23**
4.1 Complexity theory . . . . . 24
4.1.1 P vs NP problem . . . . . 24
4.2 Exact cover problem . . . . . 25
4.3 Quantum optimization at IBM Q . . . . . 27

**5. Quantum information and Entropy** . . . . . **29**

5.1 Density operator . . . . . 30
5.2 Bipartite entanglement . . . . . 30
5.2.1 Entanglement of pure states . . . . . 30
5.2.2 Entanglement of mixed states . . . . . 32
5.3 Multipartite entanglement . . . . . 32
5.4 Entanglement entropy . . . . . 33
5.4.1 Bipartite entanglement entropy . . . . . 33
5.4.2 Multipartite entanglement entropy . . . . . 34
5.5 Application case: Ising model . . . . . 35

**6. Conclusions . . . . . 37**

**Bibliography . . . . . 38**

**Appendices . . . . . 41**
A Ising model . . . . . 41
B Schmidt decomposition . . . . . 43

# List of Figures


2.1 Bloch sphere representation for a qubit. . . . . 8
2.2 Bloch sphere representation for Pauli gates. . . . . 9
2.3 Bloch sphere representation for a Hadamard gate. . . . . 9
2.4 CNOT gate representation. . . . . 12
2.5 Bell and GHZ states at IBM Q . . . . . 14
2.5 Bell and GHZ states at IBM Q (cont) . . . . . 15

3.1 Scheme of VQE algorithm. . . . . 19
3.2 Quantum circuit for trial wave function preparation. . . . . 21

4.1 Euler diagram for P, NP, NP-Complete and NP-Hard problems. . 25
4.2 Most probable final state found to the exact cover problem via quantum statevector simulator. . . . . 28

5.1 Square lattice of dimensions $2 \times 2$. . . . . 35
5.2 Square Ising model optimum configurations. . . . . 35
5.3 Entropy evolution in the frustrated case. . . . . 36
5.4 Entropy evolution in the non-frustrated case . . . . . 36

# Chapter 1

# Global overview and state of art

**Abstract**

In this chapter, a brief introduction to the idea of quantum computation and quantum information is given within the view of Michael A. Nielsen & Isaac I. Chuang [NC02]. It is stated that the birth of quantum computation and quantum information can not be understood from the evolution of just one field, but certainly as a communion of several branches of science. In this sense, a quick review over the historical evolution of main related fields is presented just to place the reader in the context of the development of fundamental quantum computation and quantum information concepts. Also, current state of art is provided, as well as the main applications of the field.

**Resumen**

En este capítulo se proporciona una breve introducción al concepto de computación cuántica e información cuántica desde la visión de Michael A. Nielsen & Isaac I. Chuang [NC02]. Es un hecho que el nacimiento de la computación cuántica e información cuántica no puede ser entendido desde la evolución de un único campo, sino como la comunión de ideas de diferentes ramas de la ciencia. En este sentido, se presenta un breve repaso de la evolución histórica de los principales campos con el fin de situar al lector en el contexto del desarrollo de los principales conceptos de la computación cuántica y la información cuántica. Asimismo, se describen el estado del arte y las principales aplicaciones del campo.

Quantum computation and quantum information is defined as the study of information processing tasks that can be accomplished using quantum mechanical systems [NC02]. To gain some insight into the fundamental ideas of quantum computation and quantum information and how did they arise, is not possible just to focus on one science field, but in the collaborative environment established between diverse branches such as computer science, quantum mechanics, information theory and cryptography. A brief overview of the historical evolution of them will allow us to get an overview of the need of quantum computation and quantum information.

In the early nineteenth century, during the Industrial Revolution, the baptized as the first programmable computing device was designed by Charles Babbage, inspired in the work of J. M. Jacquard. This British mathematician and engineer devised the Difference Engine (1812), which calculated tables of operations applying the difference method. This machine is recognised as the first prototype of current calculators. He also designed the Analytical Engine (1816-1871) which was programmable to realize desired calculations. These machines worked with information stored in punched cards and returned results with the same format, constituting one of the first approaches to bits concept[1] and assembly language. Years later, another notoriously advanced mechanical machine was designed by Herman Hollerith to tabulate data for the 1890 census. Hollerith's company gained popularity and years later it merged with a competitor to found the corporation lately named as International Business Machines (IBM) [Ade48]. All these preliminary efforts of developing the first programmable machine concluded with the publication of *On computable numbers, with an application to the Entscheidungsproblem* by a brilliant mathematician named Alan Turing [Tur37]. In this paper Turing proved that it can not exist an algorithm able to solve the decision problem (*German: Entscheidungsproblem*), posed by David Hilbert and Wilhelm Ackermann in 1928 as:

> *Given a statement of a first-order logic defined by a finite set of axioms, is there an algorithm that can decide whether the statement is true or false?*

In this publication he introduced the concept of *Turing machine*, an abstract notion of what we know as programmable computer. The Turing machine mathematically models a machine that operates on an infinite memory tape which is divided into discrete cells. In each cell may be written either a symbol or a white space. The machine has a tape head in a specific internal state which is positioned over a cell. Then, it reads the symbol, writes a new

[1] Although punched cards were developed by J. M. Jacquard in 1804, the modern concept of bit as a logarithmic measure of information was introduced by Claude E. Shannon [Sha48].

one, changes its state (or not) and moves one place to the right or left. Consequently, programming a Turing Machine consist in giving the machine the set of instructions associated to each possible situation. This machine completely captures what it means to perform a task in the algorithmic sense. That is, if an algorithm can be ran on any hardware device, then there is an equivalent algorithm for a Universal Turing Machine that performs exactly the same task. This assertion is known as *Church-Turing thesis* and it establishes an equivalence between the physical concept of what kind of algorithms can be performed on a physical device with the concept of Universal Turing Machine. The results of Turing's work are considered the origin of computer science from a theoretical perspective. Shortly afterwards, John von Neumann developed a computer architecture fully as capable as Universal Turing Machine [VN93]. Subsequent works came up with a strong Church-Turing thesis:

> *Any algorithmic process can be simulated efficiently using a probabilistic Turing machine.*

Note that efficiently means that the algorithm runs in polynomial time in the size of the problem solved. Motivated by this assertion, in 1985 David Deutsch looked for a physical theory to provide a foundation for the Church-Turing as solid as the physical theory. As a result, Deutsch tried to define a computational device capable of simulating efficiently an arbitrary physical system. This device must be ruled by the principles of quantum mechanics as law of physics are ultimately quantum mechanical. Even though it have not been proved yet that Deutsch's notion of a Universal Quantum Computer is sufficient to efficiently simulate any physical system, it opens the question of whether a quantum computer can solve problems which have no efficient solution in a probabilistic Turing machine. Further investigations culminate in 1994, when Peter Shor proved that the problem of finding the prime factors of an integer and the *discrete logarithm* problem could be solved efficiently on a quantum computer. This led to the quantum Church-Turing thesis:

> *A quantum Turing machine can efficiently simulate any realistic model of computation.*

being the quantum Turing machine a quantum analog of the Turing machine proposed by Deutsch.

Since the development of the transistor in 1947, computer hardware has grown in power at the same rate as the number of transistors in an integrated circuit, doubling its power every two years (the well known *Moore's law*). Nevertheless, this astonishing pace will take to the end in the following years as engineers can not control electron's flow (due to quantum tunneling) at the

nanometric scale that transistors have reached (*silicon limit*). One way to face this challenge is to move to a different computer paradigm provided by the theory of quantum computation. Moreover, quantum computers offer a remarkable speed advantage over classical computers.

Alongside with development of the twentieth century computer science, a new conception of the world jumped onto the stage. Wilhelm Wien and Ludwig Boltzmann studies on the black-body radiation, the photoelectric effect explanation by Albert Einstein, the atomic theory by Niels Bohr and Ernest Rutherford; and other quantum phenomena led to a new interpretation of the tiny and cold physics systems. In the mid-twenties, the explanation of all this phenomena resulted in the creation of quantum mechanics, which can be defined as a mathematical framework for the construction of physical theories. One of the main goals of quantum computation and quantum information is to gain some transparency over the counter-intuitive nature of quantum mechanics. In this sense, some success have been achieved, for example, with the *no-cloning theorem* (1982) which states that it is not possible to clone an unknown quantum state. Notice that, if cloning is achievable, it would be possible for signals to travel faster than light. Besides that, many advances have been done in obtaining complete control over single quantum systems. These advances are essential in order to exploit the power of quantum mechanics in quantum computation and quantum information applications [Car00].

At the same time computer science and quantum mechanics were rising, another revolution was taking place in the field of communication. In 1948 Claude Shannon defined mathematically the concept of information and proved two fundamentals theorems of information theory: the *noiseless channel coding theorem* and the *noisy channel coding theorem*, which gives an upper limit to the correction provided by error-correcting codes. The irruption of quantum mechanics led to creation of the quantum information theory, and in 1995 Ben Schumacher came up with the formal definition of a 'quantum bit' or 'qubit' as a tangible physical quantity [Sch95]. Besides, a theory of quantum error-correction has been developed and it allows quantum computers to compute effectively in the presence of noise.

Finally, cryptography is the practice and study of techniques for secure communication in the presence of third parties [ASM12]. Its main challenge is how to safely distribute the keys that allow to decrypt the messages sent. In this sense, quantum mechanics can be used to ensure security between emitter and receiver due to the basic principle that observation disturbs the system. Consequently, if a third party tries to "listen" to the signal containing the key the emitter and receiver will notice, throwing away the "contamined" bits and starting over.

Between the various disciplines in which quantum computation can suppose a great advance, Machine Learning (ML) stands out . This branch of computational algorithms focus in designing algorithms that emulate human kind of thinking in the way that they learn from the "surroundings" [ENM15]. To perform a specific task these algorithms learn from a given *dataset*, which means that they look for patterns, and then predict over a *test set*. Several types of ML algorithms are known since the '90s[2] such as *multivariate regression* and *neural networks*. However, they demand great computational power so it has not been until the evolution of modern computers when they have become a great commotion and fruitful field of research. Their applications cover a range that goes from computer vision to finances and medicine.

After all the aforesaid, it is clear that quantum computation emerges as a worthwhile field of study. The current idea of quantum computation is due to Paul Benioff, who in 1981 suggested to work with quanta instead of electric voltages. Later works proved the viability of designing quantum computers and many companies ventured in the quantum computing race. Among these companies, IBM is the one which has obtained more fruitful results. In 2017 IBM presented the model of a 49 qubit quantum computer whose simulation power overcome today's existing supercomputers [PGN+17]. Moreover, IBM has launched a project named as *IBM Q* which is pioneer in providing quantum computing service available in the cloud for everyone, including a 20 qubit quantum computer [Fer18]. Nevertheless, quantum computation still have to face some obstacles such as quantum decoherence, which forces quantum computers to work with really low temperatures ($\sim \mathrm{mK}$) to keep reasonable coherence times. Also minimizing the interference between qubits in small quantum chips is an open problem.

[2] We all remember how an IBM computer, named Deep Blue, beat world chess champion Garry Kasparov in 1997 by using machine learning techniques [Pan97].

# Chapter 2

# Introduction

**Abstract**

In this chapter, the main building blocks of quantum computation and quantum information are studied. First of all, the qubit is defined and its principal properties are outlined, as well as the fundamental single qubit quantum gates. Once single qubits are understood, attention will be focused in multiple qubit system and its main gates and features, with special interest in *entanglement*. Finally, the concepts of quantum computer, quantum simulator and quantum circuits are introduced. With these fundamentals, quantum circuits for creating both Bell and GHZ states are designed and executed in a quantum simulator and in a 14-qubit IBM quantum computer named *IBM Q Melbourne* [ibm18]. Finally, the results are analyzed and the principal error sources are described.

**Resumen**

En el presente capítulo se estudian los pilares sobre los que se asientan la computación cuántica y la información cuántica. En primer lugar se define el concepto de cúbit así como sus propiedades y las principales puertas lógicas cuánticas de un cúbit. Una vez se han comprendido los cúbits, la atención se centrará en sistemas de varios cúbits, analizando sus características y puertas lógicas esenciales, con especial interés en la noción de *entrelazamiento*. Finalmente, se introducen los conceptos de ordenador cuántico, simulador cuántico y circuito cuántico. Con estos cimientos se procede a diseñar circuitos cuánticos capaces de generar estados de Bell y GHZ. Estos circuitos son ejecutados en un simulador cuántico y en un ordenador cuántico de 14 cúbits de IBM denominado *IBM Q Melbourne* [ibm18]. Los resultados son analizados así como las principales fuentes de error.

## 2.1 Quantum bits

### 2.1.1 The Qubit

The main advantage of quantum computation over classical computation lies in the power of qubits over bits. But, what is a qubit? A qubit is basically, a quantum system consisting of two levels, which we will label as $|0\rangle$ and $|1\rangle$. According to this, many different types of physical systems can be used as qubits: atomic orbitals, photon polarization, spin of 1/2 spin fermions, etc.[1] Nevertheless, in this project we will treat qubits as mathematical objects without concerning about their physical implementation [Vaz12].

As we already know, the classical bit of information have two possible states - either $0$ or $1$ - just as the qubit - either $|0\rangle$ or $|1\rangle$. However, the qubit can be in many more states, specifically in a superposition of both $|0\rangle$ and $|1\rangle$. Therefore, it can be represented by a two-dimensional vector space over the complex numbers, $\mathbb{C}^2$, where the widely used basis is the computational (or standard) basis: $\{|0\rangle, |1\rangle\}$. In Dirac notation, the state of a qubit is given by:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \qquad \alpha, \beta \in \mathbb{C}. \tag{2.1}$$

with $|\alpha|^2 + |\beta|^2 = 1$. Making a measurement on the qubit usin the standard basis yields $0$ with probability $|\alpha|^2$ or $1$ with probability $|\beta|^2$, according to Born rule, and give us the classical bit of information. Besides, the normalization condition just mentioned provide us a geometric representation of the qubit state. Notice that the qubit state can also be written as:

$$|\psi\rangle = \cos(\theta/2)|0\rangle + e^{i\varphi}\sin(\theta/2)|1\rangle \tag{2.2}$$

where $0 \leq \varphi < 2\pi$ and $0 \leq \theta \leq \pi$. Equation (2.2) characterizes the state of the qubit within a global phase factor which turns out to have no observable effect. The parameters $\theta$ and $\varphi$ define a one-to-one correspondence between qubit states ($\mathbb{C}^2$) and the points on the surface of a unit sphere ($\mathbb{R}^3$). This sphere, known as *Bloch sphere* (see Figure 2.1), provides a useful visualization of the state of a single qubit, but has no generalization for the multiple qubits case.

[1] IBM Q quantum computers uses a physical type of qubit called *superconducting transmon qubit*, which is made from superconducting materials such as niobium and aluminium, patterned on a silicon substrate [IR17].

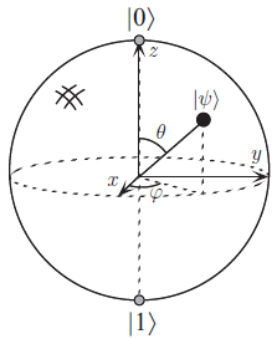



Figure 2.1: Bloch sphere representation for a qubit [NC02].

### 2.1.2 Single qubit gates

In the same way classical computers perform operations over bits by logic gates, quantum computers change qubit states by the use of *quantum gates*. If we restrict ourselves to the case of one qubit system, quantum gates can be described by two by two matrices. Nevertheless, not every two by two matrix is valid to represent an evolution of the quantum system, this transformation must be *unitary*, and therefore *reversible*:

$$|\psi'\rangle = U|\psi\rangle, \qquad U^\dagger U = \mathbb{1} \tag{2.3}$$

Since a unitary transformation represents a rotation of the Hilbert space, the state vector length does no change and the new state remain in the Bloch sphere. Therefore, the most general quantum gate must be able to take a qubit from $|0\rangle$ to the general state mentioned before (2.2). Thus, the matrix representing this general quantum gate would be:

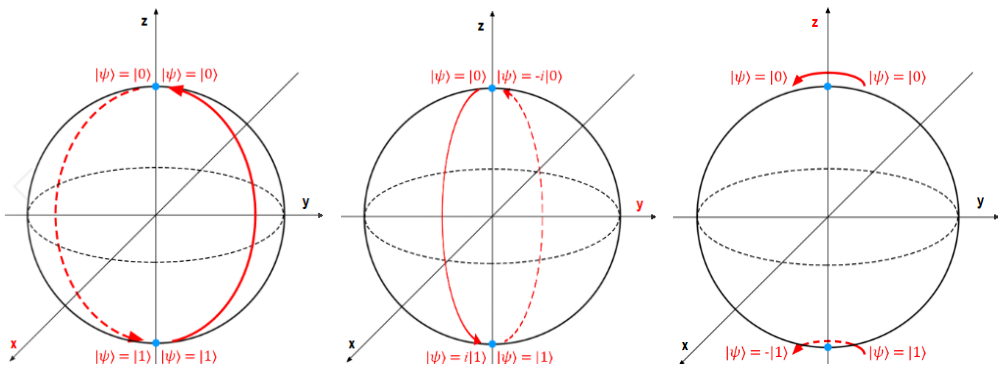
$$U_3(\theta,\varphi,\lambda) = \begin{pmatrix} \cos(\theta/2) & -e^{i\lambda}\sin(\theta/2) \\ e^{i\varphi}\sin(\theta/2) & e^{i(\lambda+\varphi)}\cos(\theta/2) \end{pmatrix} \tag{2.4}$$

From the general gate (2.4) every single qubit gate can be obtained. Particularly, the following ones:

- **Measurement.** This operation converts a single qubit state into a probabilistic classical bit.
- **Paulis gates (X, Y, Z).** These are the simplest quantum gates and their mainly action is to perform a half rotation of the Bloch sphere around the x, y and z axes respectively (see Figure 2.2).

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{2.5}$$

Notice that gate X is basically the classical NOT gate (or bit-flip): $X|0\rangle = |1\rangle$ and $X|1\rangle = |0\rangle$; just as the Y gate, which also adds a phase shift. Meanwhile, Z gate leaves $|0\rangle$ unchanged and flips the sign of $|1\rangle$.




(a) X gate. (b) Y gate. (c) Z gate.

Figure 2.2: Bloch sphere representation for Pauli gates [Ram18].

- **Hadamard gate (H).** This is the main quantum gate in order to generate superposition states. Its matrix representation is given by:

$$H = U_3(\pi/2, 0, \pi) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{2.6}$$

Notice that this gate turns a $|0\rangle$ state into $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2})$, and turns a $|1\rangle$ state into $|-\rangle = (|0\rangle - |1\rangle)/\sqrt{2})$. In the Bloch sphere picture it can be viewed as a rotation of the sphere about the y axis by $90°$, followed by a rotation about the x axis by $180°$(see Figure 2.3).

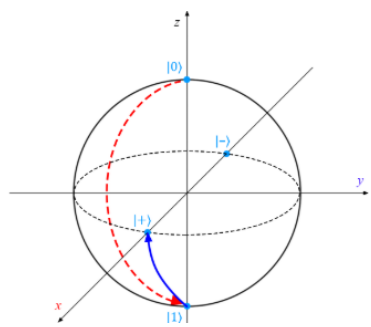



Figure 2.3: Bloch sphere representation for a Hadamard gate [Ram18].

## 2.2 Multiple qubits

Let us consider now a quantum system formed by $n$ qubits. The vector space associated to this system is the tensor product of the $n$ single qubit vector spaces and, consequently, has dimension $2^n$, denoted as $\mathbb{C}^{2^n}$. Therefore, the computational basis of the system, given by $\{|a_0\rangle \otimes |a_1\rangle \otimes ...|a_{n-1}\rangle : a_i \in \{0,1\}\}$, presents an exponential growth in its dimension with the number of qubits. Consider as an example a classical computer with n-bits. It presents $2^n$ possible configurations but, at any point of time, the computer state is in one and only one configuration. In a single operation the classical computer takes a $n$-bit number, manipulates it, and outputs a $n$-bit number. Moreover, a quantum computer also takes a $n$-bit number as input and outputs a $n$-bit number but, due to superposition, the intermediate state requires $2^n$ complex numbers to be described, providing an astonishing computational power[2].

### 2.2.1 Entanglement

Just as in the case of single qubit systems, multiple qubit systems present the superposition property. Furthermore, this multiple qubit superposition states can exhibit another quantum property: entanglement. Consider two quantum systems $A$ and $B$, with Hilbert spaces $\mathscr{H}_{\mathscr{A}}$ and $\mathscr{H}_{\mathscr{B}}$, respectively. If the state $|\psi\rangle_{AB}$ of the composite system can be written in the form:

$$|\psi\rangle_{AB} = |\phi\rangle_A \otimes |\phi\rangle_B \tag{2.7}$$

where $|\phi\rangle_i \in \mathscr{H}_i$ is the state of system $i$, then the state is called *product (or separable) state*[3] . If the state is not separable, it is known as an *entangled state*. In an entangled state, the whole system is in a definite state, even though its parts are not. Observing one of two entangled particles makes it behave randomly, but tells the observer exactly how the other particle would act if a similar observation was made on it. It must be pointed out that entanglement implies a correlation between individually random behaviours of the two particles, and thus it can not be used to send a message, as it is sometimes wrongly thought.

[2]Even though superposition is computationally stronger than classical computation, it is actually weaker than massive parallelism, which means having an army of $2^N$ classical computer working on the problem at once.

[3]Within the physics community, the qubits of a multi-qubit systems are typically ordered with the first qubit on the left-most side of the tensor product and the last qubit on the right-most side. This enables easy conversion from bitstrings to integers after measurements are performed.

### 2.2.2 Bell and GHZ states

Among all the possible entangled states for a two qubit system, the *Bell states* are the simplest with maximum entanglement. They constitute a basis of the four-dimensional Hilbert space of two qubits. They are given in term of the computational basis by:

$$\begin{aligned} |\Phi^+\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \qquad |\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \\ |\Psi^+\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \qquad |\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle). \end{aligned} \tag{2.8}$$

The maximum entanglement property of these states lead us to an indispensable tool of quantum computation and quantum information. To illustrate this, consider a two qubit system in the Bell state $|\Phi^+\rangle$. If we measure the first qubit in the computational basis we would obtain a perfectly random outcome: either 0 or 1, both with probability 1/2. However, if then we measure the second qubit, we would obtain the same outcome as for the first qubit, showing a perfect correlation between both qubits. One interesting property of Bell state $|\Phi^+\rangle$ is that the particular correlation between the measurement outcomes on the two qubits holds true in every basis. The understanding and implementation of Bell states turns out to be essential in the analysis of quantum communication (*superdense coding*) and quantum teleportation [Vaz12][NC02].

If we consider quantum systems with more than two qubits we may come up with *GHZ states* (or *"Schrödinger cat states"*) which constitute a "generalization" of the Bell states. They were first studied in 1989 by Greenberger, Horne, and Zeilinger [GHZ89], who gave them its name. Basically, these states constitute a quantum superposition of all the qubits being in state $|0\rangle$ with all of them being in state $|1\rangle$:

$$|GHZ\rangle = \frac{|0\rangle^{\otimes n} + |1\rangle^{\otimes n}}{\sqrt{2}} \tag{2.9}$$

### 2.2.3 Multiple qubit gates

In the same way as classical computation uses multiple bit logical gates, quantum computation also have multiple qubit quantum gates. They are unitary transformations specified by a $2^m \times 2^m$ matrix, where $m$ is the number of qubits involved in the operation. For our purpose, we will consider only two different multiple qubit quantum gates:

- **Controlled Not (CNOT).** This gate acts on two qubits: the control qubit and the target qubit. Its *modus operandi* is the following: the con-

trol qubit does not change, meanwhile the target qubit flips if and only if the control qubit is 1.

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

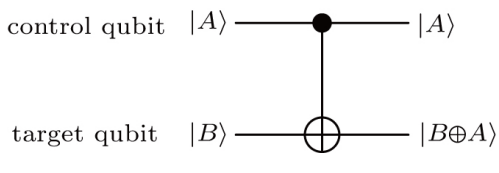



Figure 2.4: CNOT gate representation [LZH19].

- **Toffoli (CCNOT).** Also known as "controlled-controlled-not" gate, it acts on three qubits: two control qubits and a target qubit. It operates in the following way: if and only if the two control qubits are 1 then the target qubit is flipped and control qubits remain unchanged; in any other case all the qubits will remain unaltered.

These two quantum gates are universal in the sense that any quantum circuit can be built up from those gates.

## 2.3 Quantum circuit

A quantum computer is, essentially, a device able to manipulate delicate quantum states in a controlled fashion, in the same way an ordinary computer manipulates its bits. They are used to implement quantum circuits, one of the building blocks of quantum information theory, which are composed by three main elements: quantum gates, wires (not necessarily physical ones) and an input state. There are some features in classical circuits that are not allowed in quantum circuits such as loops (quantum circuits are *acyclic*), FANIN (joining of wires) and FANOUT (division of wires). Moreover, there have also been designed quantum simulators for theoretical purposes. They are standard computers operating in the same way as a quantum computer would do. However, simulators have memory requirements and runtime that are exponential in the number of qubits due to the fact that they track the quantum state of the system, what makes them really inefficient. In the next section, some quantum circuits which generate Bell states and GHZ states are shown (see Figure 2.5).

## 2.4 Bell and GHZ states at IMB Q

As it was said in Chapter 1 of this document, IBM is one of the pioneer companies in developing quantum computation technologies. In this sense, they

have launched IBM Q, where IBM offers both quantum simulators and real quantum devices to the public via quantum cloud services that can be accessed through the IBM Q Experience or Qiskit.

In this project we will make use of Qiskit [AAB+19], an open-source quantum programming framework based on Python language. Our main goal is to design quantum circuits to generate the different Bell states and the GHZ state: $|GHZ\rangle = (|000\rangle + |111\rangle)/\sqrt{2}$. Once we have designed those quantum circuits we will run them in both quantum simulator and quantum computer and then analyze the results. This is only a brief description of the implementation is made. For further details check [GF19].

So, first of all, we have to construct quantum circuits to create the different Bell and GHZ states. These circuits turn out to be really simple as they are formed by the combination of three main quantum gates: X, Hadamard and CNOT. The initial state of the qubits system presents all the qubits in the state $|0\rangle$. Focusing on the case of the state $|\Phi^+\rangle$, at first instance a Hadamard gate must be used to generate superposition in the first qubit. Then, a CNOT gate is placed to create entanglement between the two qubits and finally, both qubits are measured. Similar circuits are designed for the rest of states introducing in some of them X gates or another CNOT gate in the case of the GHZ state (see Figure 2.5). Once the circuits have been constructed, they are run in a quantum simulator and in a real quantum device, particularly in a 14-qubit quantum computer called *IBM Q Melbourne* [ibm18]. In each experiment, 1024 shots were taken and then the probability of each final state was calculated and represented in an histogram. The results along with the quantum circuits for each Bell and GHZ state are shown in Figure 2.5.

It can be seen that the quantum simulator does not offer the theoretical result of 1/2 probability for each state in every case. This is due to the finite number of shots taken in the simulation and the law of large numbers. Moreover, quantum device *IBM Q Melbourne* presents substantial errors and their nature can be really complex. In Section 2.2.1 the importance of entanglement in quantum computing was pointed out. However, entanglement can also occur between environment and quantum computer, making quantum effects disappear. This issue becomes more important if we take into account that the quantum computer must be coupled to the external world so the user can run programs on it and read the output from those programs. This requirement sets a limit on how long can the system maintain its quantum behaviour, known as *coherence time*. Consequently, the complex nature of real errors that happen on a quantum device makes characterizing them very complicated, and creating accurate error models is an active area of quantum computing research [Cho14].

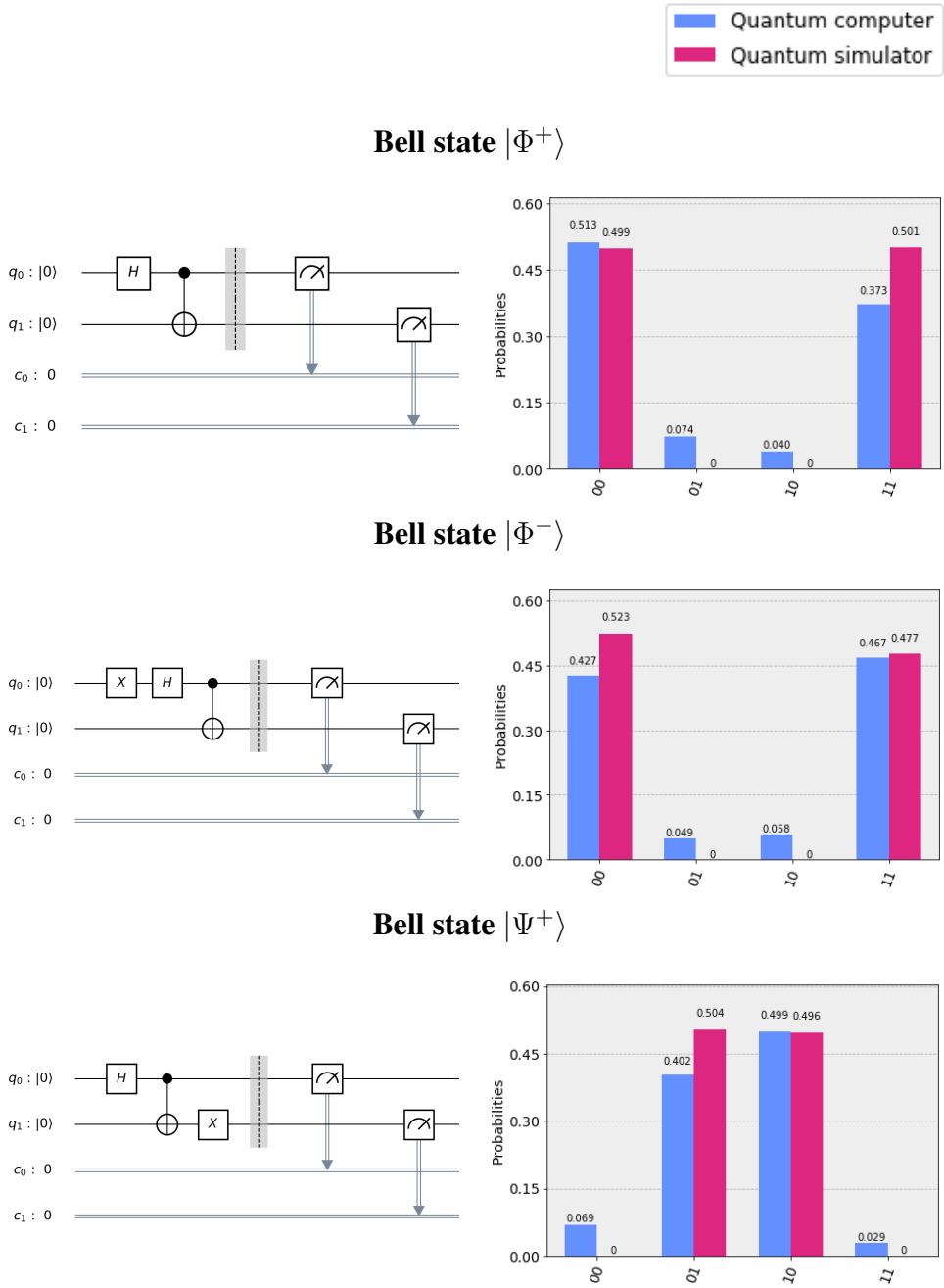



Figure 2.5: Quantum circuits (left) and output histograms for quantum simulator and 14-qubit quantum computer *IBM Q Melbourne* (right) for every Bell and GHZ states. For each experiment 1024 shots were made.

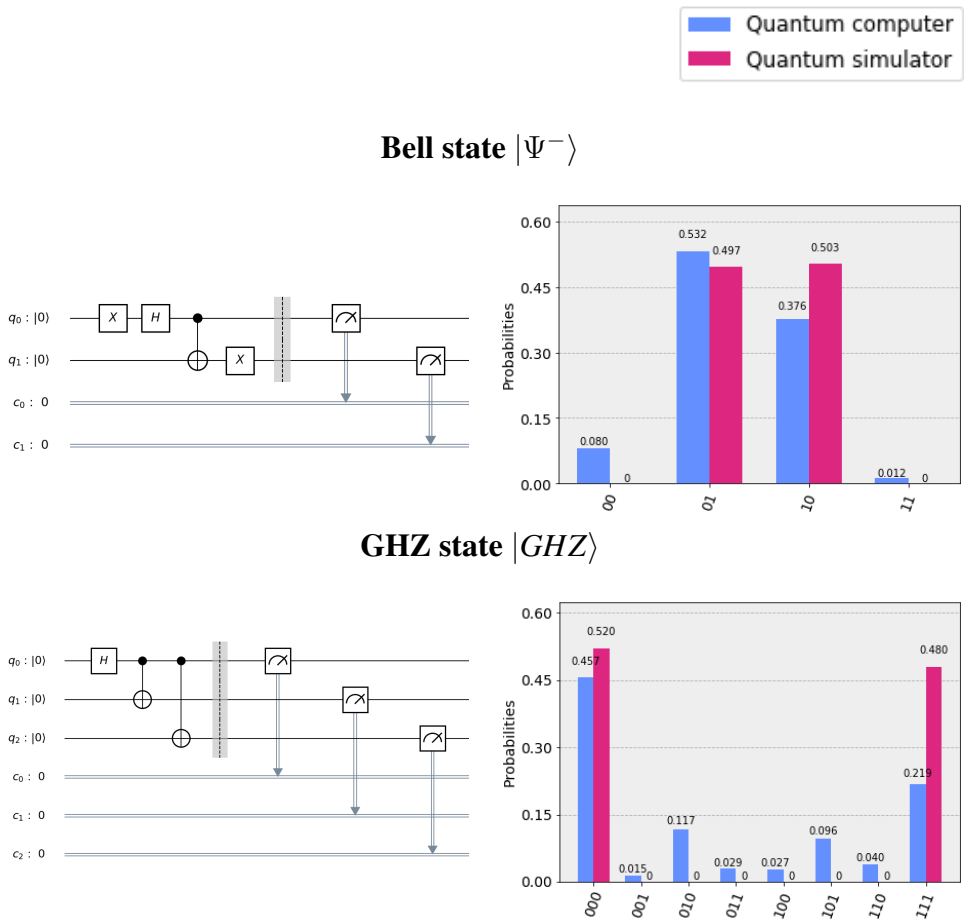



Figure 2.5: Quantum circuits (left) and output histograms for quantum simulator and 14-qubit quantum computer *IBM Q Melbourne* (right) for every Bell and GHZ states. For each experiment 1024 shots were made. (cont)

# Chapter 3

# Quantum algorithms

**Abstract**

In this chapter, we will introduce quantum algorithms and compare them to the classical ones, showing their main features and pointing out some quantum algorithms which had overcome the classical ones [DJ92]. After this, our attention will be focused in a particular quantum/classical hybrid algorithm really useful in finding optimal combinations: the Variational Quantum Eigensolver (VQE) [Com16]. This algorithm is based on the variational method, so a quick review of it will also be done [CTDLD06]. Finally, the algorithm will be sketched and its main characteristics analyzed.

**Resumen**

En el presente capítulo introduciremos los algoritmos cuánticos y, tras compararlos con los clásicos y destacar sus principales características, señalaremos algunos que han conseguido superar a los clásicos [DJ92]. Tras esto, pondremos nuestra atención en un algoritmo híbrido (cuántico-clásico) realmente útil a la hora de encontrar combinaciones óptimas: el Variational Quantum Eigensolver (VQE) [Com16]. Este algoritmo se basa en el método variacional por lo que realizaremos un breve repaso al mismo [CTDLD06]. Finalmente, esquematizaremos los diferentes pasos del algoritmo y estudiaremos sus principales propiedades.

## 3.1 Quantum vs classical algorithms

In the same sense that Newtonian mechanics are "recovered" from quantum mechanics when taking the classical limit, quantum circuits can reproduce classical circuits. This arises from the fact that every classical circuit has an equivalent circuit containing only NAND gates[1] which, despite being irreversible, can be replaced by an equivalent quantum circuit only making use of the reversible *Toffoli gate*. Thus, quantum computation can, at least, emulate classical computation algorithms. As it was said in Section 2.2, quantum superposition gives quantum computation the ability of evaluating a function on several values of the system at once (*quantum parallelism*). However, the skill to *extract* information about more than one output of the function is required. One of the first quantum algorithms that overcome classical computation was *Deutsch–Jozsa algorithm* [DJ92]. There exist three types of quantum algorithms which provide an advantage over known classical algorithms: quantum simulation algorithms, quantum search algorithms and quantum Fourier transform, where Deutsch–Jozsa algorithm belongs [NC02].

## 3.2 Variational Quantum Eigensolver (VQE)

Among the different quantum search algorithms one of particular interest is the Variational Quantum Eigensolver (VQE), a quantum/classical hybrid algorithm useful at finding optimal combinations. It minimizes cost functions by applying the variational method. In this chapter, we will focus in understanding this algorithm to apply it in subsequent problems.

### 3.2.1 Variational method

Consider an arbitrary physical system whose Hamiltonian, $H$, is time independent. In order to simplify the treatment of the problem assume that $H$ presents a discrete and non-degenerate spectrum:

$$H|\varphi_n\rangle = E_n|\varphi_n\rangle, \qquad n = 0,1,2,... \tag{3.1}$$

According to the *spectral theorem for Hermitian operators*, we have that the eigenstates of the Hamiltonian satisfy the orthonormality condition:

$$\langle\varphi_i|\varphi_j\rangle = \delta_{i,j} \tag{3.2}$$

where $\delta_{i,j}$ is the Kronecker delta.

[1] NAND gate is a logic gate whose output is 0 if and only if all its inputs are 1.

Although the Hamiltonian might be known, this does not yield for the eigenstates, $\{|\varphi_n\rangle\}$, and eigenvalues, $\{E_n\}$, of the system which must be determined from the eigenvalue equation (3.1). However, diagonalizing $H$ is not always computationally accessible so the variational method results crucial to estimate them.

Let us consider that the discrete spectrum of the Hamiltonian is labelled in such a way that $n = 0$ is the ground state (greatest lower bound), $n = 1$ is the first excited state, and so on. Then, for an arbitrary state $|\psi\rangle$ of the Hilbert space of the system, the expectation value of H satisfies:

$$\langle H \rangle = \frac{\langle \psi | H | \psi \rangle}{\langle \psi | \psi \rangle} \geq E_0 \tag{3.3}$$

Towards finding the state $|\psi\rangle$ that satisfies the equality we could vary over all possible states of the Hilbert space of the system and, once we have found it, this would be the ground state of the system with energy $E_0$. However, varying over the entire Hilbert space is too expensive, so a subspace, $\mathscr{F}$, of the entire Hilbert space is chosen and parametrized by some real differentiable parameters $\theta_i$ ($i = 1, 2, ..., N_\theta$). This subspace is called *ansatz* and its election is crucial in finding an accurate approximation[2]. Now we wish to minimize the mean value of the energy for a *trial wave function* that belongs to the *ansatz* subspace:

$$\varepsilon(\theta_i) = \langle \psi(\theta_i) | H | \psi(\theta_i) \rangle, \qquad |\psi(\theta_i)\rangle \in \mathscr{F} \tag{3.4}$$

under the the normalization constraint:

$$\langle \psi(\theta_i) | \psi(\theta_i) \rangle = 1 \tag{3.5}$$

Finding the global minimum might be difficult as setting the partial derivatives of $\varepsilon(\theta_i)$ over all $\theta_i$ equal to zero could yield to local minimum. Several methods can be applied to find the global minimum such as *Ritz method* or *Hartree-Fock method*. An important inconvenient of the variational method is that as $\varepsilon$ tends toward $E_0$ in the minimization proceeding, there is no guarantee that the trial wave function will tend to the ground state of the system, so not always the global minimum is achieved [CTDLD06].

[2] A good selection of the *ansatz* implies that there is some overlap between the *ansatz* and the ground state.

### 3.2.2 The algorithm

VQE is a hybrid algorithm that uses the variational method and interleaves quantum and classical computations in order to find the eigenvalues of a large matrix. Particularly, we will employ this algorithm to the case of the Hamiltonian $H$ of a given system [Com16].

In this hybrid algorithm a quantum subroutine is run inside of a classical optimization loop. The quantum subroutine has two fundamental steps:

1. Prepare the quantum state $|\psi(\vec{\theta})\rangle$, where $\vec{\theta}$ are the set of parameters that parametrize the *ansatz* Hilbert subspace.

2. Measure the expectation value $\langle\psi(\vec{\theta})|H|\psi(\vec{\theta})\rangle$. This measure is usually performed on these states based on a Pauli operator decomposition of $H$, as it will be seen in Section 5.5.

From Section 3.2.1 we know that the variational principle ensures that this expectation value is always greater than the smallest eigenvalue of $H$. This bound allows us to use classical computation to run an optimization loop to find this eigenvalue with the following steps:

1. Use a classical non-linear optimizer to minimize the expectation value by varying *ansatz* parameters $\vec{\theta}$.

2. Iterate until convergence.

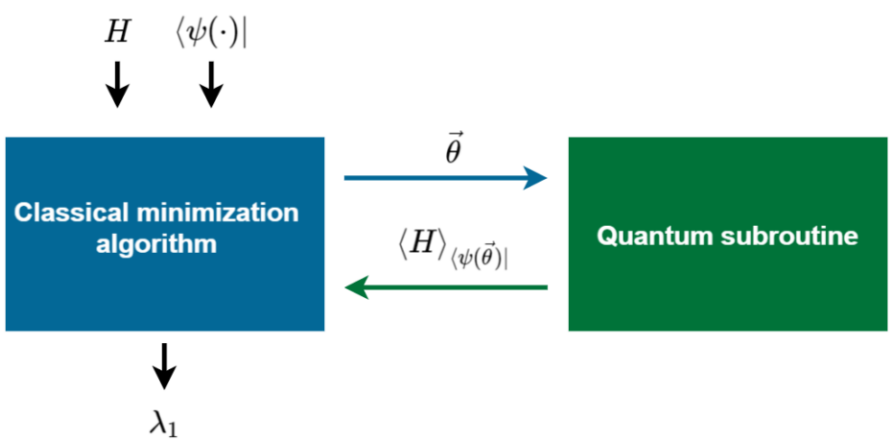



Figure 3.1: Scheme of VQE algorithm [Kop19].

## 3.3 VQE in Qiskit

To implement this hybrid algorithm in our simulations we make use of Qiskit Aqua library [AAB+19]. Qiskit's instance for VQE requires defining three algorithmic subcomponents:

- An initial state from Aqua's Initial States library in order to define the input state for the trial function.
- A trial function from Aqua's Variational Forms library.
- A classical optimizer from Aqua's Optimizers library.

We will give a brief description of each of these topics in the following sections, with special attention in those functions used in further simulations.

### 3.3.1 Initial state

An initial state in Aqua is an object that specifies a starting state for the trial state circuit. This allows the user to define a state, and then provide a circuit that can take the starting point of all zero qubits to the defined state. Among the diverse initial states that Aqua offer, for simplicity we will restrict ourselves to the initial state *Zero*, which consists of having all the qubits at state $|0\rangle$ at the entrance of the trial wave function circuit, as it can be seen in Figure 3.2.

### 3.3.2 Trial wave functions

Practically, the quantum subroutine of VQE amounts to preparing a state based off of a set of parameters $\vec{\theta}$ and performing a series of measurements. These parameters usually refer to rotation angles so we briefly introduce the *rotation operators*.

#### Rotation operators

One of the most important properties of Pauli matrices is that, when they are exponentiated, they give rise to the *rotation operators*, which rotate the state of the system in the Bloch sphere about the $\hat{x}$, $\hat{y}$ and $\hat{z}$ axes, by an angle $\theta$:

$$\begin{aligned} R_x(\theta) &\equiv e^{-i\frac{\theta}{2}X} \\ R_y(\theta) &\equiv e^{-i\frac{\theta}{2}Y} \\ R_z(\theta) &\equiv e^{-i\frac{\theta}{2}Z}. \end{aligned} \tag{3.6}$$

The most common type of trial wave functions are composed by several layers where, in each of them, both rotation and entanglement gates are considered (see Figure 3.2). The entangler circuit ($U_{ENT}$) that creates entanglement between qubits, is usually achieved with the use of *CNOT* and *Toffoli* gates. It could be *linear* (next-neighbor coupling) or *full* (all-to-all coupling). Also a different entangler map can be designed, but the most used are the previous ones. These entanglers are interleaved with single-qubit rotations, which are implemented as the product of $j$ rotation operators, each of them rotating an angle $\theta_j$:

$$U^{q,i}(\vec{\theta}_k) = \prod_j R_j^{q,i}(\theta_{k,j}^{q,i}) \tag{3.7}$$

where $\vec{\theta}_k$ represents the set of all the Euler angles (optimizer parameters) at iteration $k$, $q$ identifies the qubit and $i = 0, 1, ..., d$ refers to the depth position in the circuit. With this, the N-qubit trial state, considering that the initial state is $|00...0\rangle$, is obtained by applying $d$ entanglers $U_{ENT}$ that alternate with N single qubit rotations:

$$|\psi(\vec{\theta}_0)\rangle = \prod_{q=1}^{N}[U^{q,d}(\vec{\theta}_0)] \times U_{ENT} \times \prod_{q=1}^{N}[U^{q,d-1}(\vec{\theta}_0)] \times ... \prod_{q=1}^{N}[U^{q,0}(\vec{\theta}_0)]|00...0\rangle \tag{3.8}$$

When none of qubits are unentangled to other qubits the number of optimizer parameters is given by $j \times N \times (d+1)$, where $N$ is the total number of qubits, $j$ the number of rotation operators for each rotation step and qubit and $d$ is the depth of the circuit.

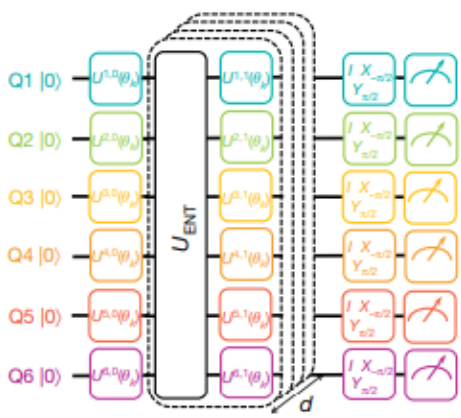



Figure 3.2: Quantum circuit for trial function preparation and energy estimation. For each iteration $k$, the circuit is composed of a sequence of interleaved single-qubit rotations $U^{q,i}(\vec{\theta}_k)$ and entangling unitary operations $U_{ENT}$. A final set of post-rotations ($I$, $X_{-\pi/2}$ or $Y_{\pi/2}$) is used to measure the expectation values of the individual Pauli terms in the Hamiltonian and to estimate the energy of the trial state [KMT+17].

From all the trial wave functions (also known as *variational forms*) that Qiskit Aqua offer, we will focus in two of them: *Ry* and *RyRz*.

- **Ry**. This trial wave function consists in layers of $y$ rotations, $U^{q,i}(\vec{\theta}_k) = R_y^{q,i}(\vec{\theta}_k)$, with entanglements. Therefore, if there are no unentangled qubits, the number of optimizer parameters is $N \times (d+1)$.
- **RyRz**. This trial wave function consists in layers of $y$ plus $z$ rotations, $U^{q,i}(\vec{\theta}_k) = R_y^{q,i}(\vec{\theta}_k) R_z^{q,i}(\vec{\theta}_k)$, with entanglements. Therefore, if there are no unentangled qubits, the number of optimizer parameters turn out to be $2 \times N \times (d+1)$.

One important thing that will be studied in subsequent sections is that many Hamiltonians can be written in terms of Pauli matrices and identity matrix. For that reason, at the end of the quantum circuit for the trial function of Figure 3.2, there is a set of rotations to measure the expectation values of the individual Pauli matrices terms of the Hamiltonian.

### 3.3.3 Classical optimizers

To conclude this chapter, let us remind that as it was previously stated, outside the quantum subroutine there must be a classical optimization loop. Therefore, a classical optimizer is needed and basically, there are two classes of them attending to where the optimal value is looked for: *local* or *global*. Here we will focus in two local optimizers which look for an optimal value within the neighboring set of a candidate solution.

- **Sequential Least Squares Programming (SLSQP).** This is a *gradient-based* algorithm and, thus, attempt to compute the absolute minimum of the function through its gradient. This algorithm applies SQP method and is commonly used for nonlinearly constrained, gradient-based optimization, supporting both equality and inequality constraints.
- **Constrained Optimization by Linear Approximation (COBYLA).** This is a numerical optimization method for constrained problems where the derivative of the objective function is not known. It works by iteratively approximating the actual constrained optimization problem with linear programming methods.

# Chapter 4

# Quantum optimization


**Abstract**

Nowadays, it is still unknown how powerful is quantum computation and what kind of problems it can solve that classical computation can not. To get a deeper understanding of the type of problems which quantum computation is expected to solve, we will introduce the field of *computational complexity theory* and its most important open problem: *P* vs *NP* [NC02]. Afterwards, an optimization NP problem called *Exact cover problem* will be presented and exhaustively analyzed. This study will show how physical models, such as those for ferromagnetism, appear in the most unexpected situations [Gal17]. Finally, a academical example of exact cover problem will be solved with both, classical and quantum computation, showing that quantum computation has much to offer.

**Resumen**

A día de hoy aún se desconoce cuán poderosa es la computación cuántica y qué problemas, irresolubles desde la computación clásica, puede abordar. Para adquirir un mayor entendimiento acerca de los problemos que se espera que la computación cuántica nos permita resolver introduciremos la *Teoría de la complejidad computacional* y su problema abierto más importante: *P vs NP* [NC02]. Tras esto, se presentará el *Exact cover problem* y será exhaustivamente analizado. Este estudio mostrará cómo modelos físicos, como el ferromagnetismo, aparecen en las situaciones más inesperadas [Gal17]. Finalmente se resolverá un ejemplo académico del *exact cover problem* mediante computación clásica y cuántica, poniendo de manifiesto que la computación cuántica tiene mucho que ofrecer.

## 4.1 Complexity theory

Before entering into further details, we will define a *decision problem* as a problem whose answer is "Yes" or "No". Another important concept is *polynomial time*: we say that an algorithm has polynomial time if for some $k > 0$, its running time on inputs of size $n$ is $\mathcal{O}(n^k)$. Within this framework, we define *complexity class* as a collection of computational problems with some common features according to the computational resources needed to be solved [NC02]. Some of the most important complexity classes are the following, which are sketched at Figure 4.1:

- **P:** set of all decision problems that can be solved in polynomial time.
- **NP:** set of all decision problems whose solutions can be verified in polynomial time. Hence, all P problems are contained in NP.
- **NP-Complete:** set of all problems $X$ in NP for which it is possible to reduce any other NP problem $Y$ to $X$ in polynomial time. Then, NP-complete is a subset of the "toughest" problems in NP.
- **NP-Hard:** set of problems $X$, such that for each NP-hard problem $x$ exist an NP-complete problem $y$ that is reducible to $x$ in polynomial time. Note that NP-hard problems do not have to be in NP.
- **PSPACE:** set of problems that can be solved using resources which are few in spatial size and unlimited in time.

### 4.1.1 P vs NP problem

The P vs NP problem constitutes one of the biggest questions in Computational Theory, being also a Millennium Problem. It asks whether every problem whose solution can be verified in polynomial time can also be solved in polynomial time ($P = NP$). Therefore, if every NP-complete problem was included in P then the above statement will be satisfied (see Figure 4.1. If $P \neq NP$, then no NP-complete problem can be efficiently (in *polynomial time*) solved in a classical computer. Quantum computers have achieved to solve quickly some NP problems -like factoring-, but it is unknown yet whether quantum computers can solve all problems in NP and, consequently, prove $P \neq NP$.

Some examples of NP-complete problems are *exact cover*, *max-cut* and *3-SAT*. The *exact cover* will be studied in the following section with both classical and quantum computer.

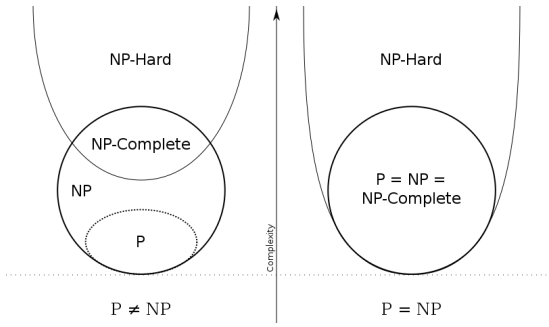



Figure 4.1: Euler diagram for P, NP, NP-Complete and NP-Hard set of problems [Esf07].

## 4.2 Exact cover problem

In computer science, the exact cover problem is a decision problem to determine whether an exact cover exists. It belongs to the NP-complete set and is a kind of constraint satisfaction problem. The most well-known exact cover problem is the traditional Sudoku [Gal17].

**Definition 4.1.** *Given a finite nonempty set $U = \{u_1, ..., u_n\}$, called universe, and a family $F = \{S_1, ..., S_m\}$ of $m \geq 1$ nonempty subsets of $U$. Then the question is whether there is an **exact cover**, which is a subfamily $C \subseteq F$ of subsets in $F$ such that the sets in $C$ are disjoint and their union is equal to $U$.*

The exact cover problem can be viewed as an optimization problem, which means that the objective is to find some optimal solution by minimizing or maximizing a certain cost function (or Hamiltonian of the system), $H$. We have said that exact cover was a decision problem, thus we can reformulate it by incorporating a budget cost $B$ and instead of asking whether the cost function $H$ has a minimum or a maximum $w$, we ask if there is a solution $w$ such that $H(w) \leq B$ in the case of a minimum solution, or $H(w) \geq B$ in the case of a maximum solution. For the exact cover problem, the objective is to find a subfamily where each element appears only once and that the union of its subsets form the universe. Therefore, we define the state of the system (or subfamily) as:

$$|\psi\rangle = |\alpha_{m-1}\rangle \otimes |\alpha_{m-2}\rangle \otimes ... \otimes |\alpha_0\rangle \tag{4.1}$$

where $m$ is the total number of subsets of the family $F$ and $\alpha_i$ indicates whether subset $i$ is contained in the subfamily ($\alpha_i = 1$) or not ($\alpha_i = 0$). So now we look for a Hamiltonian that presents its ground state for the exact cover set:

$$H = \sum_{e \in U} \left(1 - \sum_{i=0}^{m} \xi(e,i) X_i\right)^2 \tag{4.2}$$

where the first sum is extended over all the elements of the universe, the second sum extends over all the subsets of family $F$ and $\xi(e,i)$ is a function

that satisfies:

$$\xi(e,i) = \begin{cases} 1, & e \in \textit{subset } i \\ 0, & e \notin \textit{subset } i \end{cases} \tag{4.3}$$

Finally, $X_i$ is an operator which takes eigenvalues:

$$X_i|\alpha_i\rangle = \alpha_i|\alpha_i\rangle, \qquad \alpha_i \in \{0,1\} \tag{4.4}$$

It is clear that this Hamiltonian's ground state, with energy $E = 0$, represents the solution of the exact cover problem, which could be degenerated if several exact covers are possible. Notice that operator $X_i$ can also be written as[1]:

$$X_i = |1\rangle\langle 1| = \frac{1}{2}(\mathbb{1} + \sigma_i^z), \qquad \sigma_i^z = |1\rangle\langle 1| - |0\rangle\langle 0|. \tag{4.5}$$

The Hamiltonian of the system can be rewritten as a sum of individual Hamiltonians acting of each element:

$$H = \sum_e H_e, \qquad H_e = \left(1 - \sum_i^{N_e} X_i\right)^2, \tag{4.6}$$

where now the sum present in individual Hamiltonians $H_e$ extends only over the subsets that include element $e$, and $N_e$ is the number of subsets which include element $e$ and is obviously a constant of the problem. Inserting then equation (4.5) into the individual Hamiltonian $H_e$ and expanding the expression we have:

$$\begin{aligned} H_e &= \left(1 - \frac{1}{2}\sum_i^{N_e}(1+\sigma_i^z)\right)^2 = \frac{1}{4}\left((2-N_e) - \sum_i^{N_e}\sigma_i^z\right)^2 = \\ &= \frac{1}{4}\left((N_e-2)^2 + 2(N_e-2)\sum_i^{N_e}\sigma_i^z + \sum_{i,j}^{N_e}\sigma_i^z\sigma_j^z\right). \end{aligned} \tag{4.7}$$

Observing the form of the global Hamiltonian (4.6) we conclude that for having the ground state with energy $E = 0$ each individual term $H_e$ must be in its own ground state with energy $E_e = 0$. This means that each element must be in one and only one subset, as it states the exact cover problem. Rewriting Hamiltonian $H_e$ we get:

$$H_e = \varepsilon_e + \frac{1}{2}(N_e - 2)\sum_i^{N_e}\sigma_i^z + \frac{1}{4}\sum_{i,j}^{N_e}\sigma_i^z\sigma_j^z \tag{4.8}$$

[1] In the notation we are following $\{X,Y,Z\}$ are used when we are talking about quantum gates and $\{\sigma_x,\sigma_y,\sigma_z\}$ when developing the Hamiltonian due to the analogy that will be established with ferromagnetism and the literature's notation commonly used in this field.

being $\varepsilon_e$ a constant shift unique for each element. Comparing (4.8) with the general Hamiltonian of an Ising model (see Equation (A.4) in Appendix A) we see that the single Hamiltonian $H_e$ presents the form of an Ising model with:

$$\begin{cases} \varepsilon_e = \frac{1}{4}(N_e - 2)^2 \\ J = -\frac{1}{4} < 0 \\ H' = \mu_0 H = -\frac{1}{2}(N_e - 2) \begin{cases} < 0, & \text{if } N_e < 2 \\ = 0, & \text{if } N_e = 2 \\ > 0, & \text{if } N_e > 2 \end{cases} \end{cases} \tag{4.9}$$

Notice that having $J < 0$ implies that there is an antiferromagnetic interaction, with adjacent subsets tending to have opposite value of $\alpha$; this is, if one subset contents element $e$ the next one does not. Otherwise, lower values of the second term of $H_e$ will be achieved when less subsets contain element $e$, which means that subsets tend to present the state $|0\rangle$, achieving its minimum value at $N_e = 0$. Finally, the energy shift $\varepsilon_e$ increases quadratically with $N_e$, presenting a minimum at $N_e = 2$. In the ground state of $H_e$, a commitment between the three contributions is achieved showing a lower bound energy $E_e = 0$ at $N_e = 1$.

## 4.3 Quantum optimization at IBM Q

Once the exact cover problem has been understood we will present a simple problem of finding the exact cover for the following family of subsets:

$$\{\{1,3\},\{2,3,6\},\{1,5\},\{2,3,4\},\{5,6\},\{2,4\}\}$$

The problem will be solved by three different algorithms:

1. **Brute-force method.** This method does not require a cost function and it basically consists in exhaustively try all the binary assignments.

2. **Classical optimization.** The classical algorithm used, *Exact Eigensolver*, computes up to the first $k$ eigenvalues of a complex-valued square matrix of dimension $n \times n$, with $k \leq n$.

3. **Optimization with the Variational Quantum Eigensolver (VQE).** We will run this algorithm in an quantum statevector simulator[2].

[2]Due to the communication bottlenecks when the quantum device and classical computers are not physically co-located, it could not be run on real quantum processors of the IBM Q Cloud.

This problem, despite its simplicity, will allow us to program an optimization problem based in a cost function and run it in both classical and quantum ways[3]. The results obtained with the two classical algorithms satisfied the expected solution of the problem:

$$\text{Solution found: } |1,0,0,0,1,1\rangle$$

which means that an exact cover is given by: $\{\{1,3\},\{5,6\},\{2,4\}\}$, what is certainly true. One important fact that was checked is the slowness of the brute force method compared with the classical optimization, with an approximate difference of time of one order of magnitude. This is not a surprise as brute force is always the worst way. Besides, the VQE was programmed with the initial state *Zero*, a full entangler map, two layers of depth with RyRz rotations in the trial function circuit and COBYLA optimizer. The result obtained with the quantum statevector simulator also was the expected as it can be seen in Figure 4.2, which evidence that quantum computers can solve optimization problems.

```
Solution is [1. 0. 0. 0. 1. 1.]
Time (s): 68.23676228523254
Energy: 1.888090119805952e-07
Number of evaluations: 1000
```

Figure 4.2: Most probable final state found to the exact cover problem via quantum statevector simulator. The time of computation in seconds, the energy of the final state in arbitrary units and the number of evaluations are also shown.

[3]To check the code take a look to [GF19].

# Chapter 5

# Quantum information and Entropy

**Abstract**

As it was discussed in Section 2.2.1, entanglement is one of the most powerful resources of quantum computation. In this chapter, we will discuss the basic notions of bipartite entanglement [GT09] and introduce the Schmidt decomposition [Wil17], which is one of the most important theorems for understanding pure bipartite states. As many experiments nowadays aim at the generation of multiparticle entangled states [PLNT12], we will develop some basic notions about the theory of multipartite entanglement as well as introducing the entropy as a measure of entanglement [GK16]. Finally, the evolution of entropy in the resolution of an optimization problem in a quantum simulator will be studied. Particularly, the problem studied consist of finding the ground state of a two-dimensional square-lattice Ising model.

**Resumen**

Como se señaló en la Sección 2.2.1, el entrelazamiento es uno de los recursos más potentes de la computación cuántica. En este capítulo discutiremos las nociones básicas del entrelazamiento bipartito [GT09] e introduciremos la descomposición de Schmidt [Wil17], uno de los teoremas más importantes a la hora de entender los estados puros bipartitos. Hoy en día muchos experimentos se centran en la generación de estados multipartícula entrelazados [PLNT12], por ello desarrollaremos algunos conceptos básicos sobre la teoría de entrelazamiento multipartito además de introducir la entropía como unidad de medida del entrelazamiento [GK16]. Finalmente, se analizará la evolución de la entropía en la resolución de un problema de optimización empleando un simulador cuántico. En particular, el problema estudiado consistirá en determinar el estado fundamental de un modelo bidimensional de Ising de malla cuadrada.

## 5.1 Density operator

Our main goal is to quantify the degree of entanglement in a given multipartite quantum state. Before that, some important concepts of quantum mechanics must be introduced. The *density operator* is an alternative representation of the state of a quantum system which is really useful when the quantum system is an statistical ensemble of several quantum systems. It is defined as an Hermitian operator, $\rho$, with $Tr(\rho) = 1$ and strictly positive ($\rho \geq 0$):

$$\rho = \sum_j p_j |\psi_j\rangle\langle\psi_j| \tag{5.1}$$

for some ensemble $\{p_j, |\psi_j\rangle\}$. In this case, the density operator (5.1) represents a *mixed state* with probability $p_j$ that the system is in the state $|\psi_j\rangle$. Otherwise, the density operator $\rho$ would describe a *pure state* if and only if:

$$p_j = 1 \wedge p_i = 0,\ \forall i \neq j \qquad or \qquad \rho = \rho^2 \tag{5.2}$$

In further analysis we will put our attention on pure states because quantum simulators work within this framework. Moreover, working with real quantum computers will require a treatment in the mixed states context [CTDLD06].

## 5.2 Bipartite entanglement

The experiments run at quantum computers work with multiple qubits, generating multiparticle entangled states. To understand this kind of systems we will first study the bipartite states. Consider two quantum systems $A$ and $B$, owned by Alice and Bob, that are described by states in Hilbert spaces $\mathscr{H}_A$ of dimension $d_A$ and $\mathscr{H}_B$ with dimension $d_B$, respectively. The composite systems is described by state vectors that belong to $\mathscr{H}_A \otimes \mathscr{H}_B$.

### 5.2.1 Entanglement of pure states

In Section 2.2.1 we defined the *product state* as a state vector of the composite system $\mathscr{H}_A \otimes \mathscr{H}_B$ that presents the following shape:

$$|\psi\rangle_{AB} = |\phi\rangle_A \otimes |\phi\rangle_B. \tag{5.3}$$

If we restrict ourselves to the density operator framework, the product state of the composite system will be:

$$\rho_{AB} = \rho_A \otimes \rho_B \tag{5.4}$$

which can be derived directly from (5.3). If the system is described by either of the representations (5.3) and (5.4), then we say that subsystems $A$ and

*B* are *independent*. In this sense, a product state can be prepared in a local independent way by Alice and Bob. Otherwise, if those representations does not yield for the composite system state vector, we say that the subsystems are *entangled* [GT09]. A crucial tool for analyzing bipartite pure states in quantum information theory and quantifying the level of entanglement of the system is the Schmidt decomposition, presented in Lemma 5.1.

**Lemma 5.1** (Schmidt decomposition.)**.** *Suppose $|\psi\rangle_{AB}$ is a pure state of the composite system, AB, given by the general form:*

$$|\psi\rangle_{AB} = \sum_{i,j=1}^{d_A,d_B} c_{ij}|\phi_i\rangle_A \otimes |\phi_j\rangle_B \quad \in \quad \mathscr{H}_A \otimes \mathscr{H}_B \tag{5.5}$$

*with $C = [c_{ij}]$ a complex $d_A \times d_B$ matrix. Then there exist orthonormal states $|i\rangle_A$ for system A, and orthonormal states $|i\rangle_B$ of system B such that the system state can be expressed as:*

$$|\psi\rangle_{AB} = \sum_{i=1}^{d} \lambda_i |i\rangle_A |i\rangle_B, \tag{5.6}$$

*where the amplitudes $\lambda_i$, known as **Schmidt coefficients**, are the singular values of matrix C. They are real, strictly positive and normalized, so that $\sum_i \lambda_i^2 = 1$. The **Schmidt rank** (or **Schmidt number**), d, of a bipartite state is equal to the number of non-zero Schmidt coefficients $\lambda_i$ in the Schmidt decomposition of the state. From this we see that a state $|\psi\rangle_{AB}$ is a product state if and only if it has Schmidt rank 1. The Schmidt rank, d, satisfies the following inequality:*

$$d \leq min\{d_A, d_B\} \tag{5.7}$$

*and it can be proved that it is preserved under unitary transformations on system A or system B. This reflects that local quantum operations and classical communication (LOCC) can not increase the level of entanglement (Check proof at Appendix B.) [Wil17].*

**Corollary 5.2.** *Given the composite system AB, the reduced density operators $\rho_A$ and $\rho_B$ can be described in terms of the Schmidt coefficients and the basis $\{|i\rangle_A\}$ and $\{|i\rangle_B\}$:*

$$\rho_A = \mathrm{Tr_B}(\rho_{AB}) = \sum_{i=1}^{d} \lambda_i^2 (|i\rangle_A \langle i|) = \mathrm{Tr_A}(\rho_{AB}) = \rho_B \tag{5.8}$$

*Therefore, if $|\psi\rangle_{AB}$ has a Schmidt rank of 1, the reduced matrices $\rho_A$ and $\rho_B$ have only one non-zero eigenvalue and are pure states. Otherwise, if Schmidt rank is greater than 1, the reduced matrices have multiple non-zero eigenvalues and are mixed states (Check proof at Appendix B.) [Wil17].*

### 5.2.2 Entanglement of mixed states

**Definition 5.3.** A mixed state $\rho_{AB}$ is *separable* if and only if it can be represented as a convex combination of the product of projectors on local states as stated in the following expression:

$$\rho_{AB} = \sum_{i=1}^{K} p_i |e\rangle_i\langle e| \otimes |f\rangle_i\langle f| \tag{5.9}$$

Otherwise, the mixed state is said to be entangled [BL19].

It should be noticed that (5.9) is the most general state that Alice and Bob can prepare by local quantum operations and classical communication (LOCC). Therefore, entangled states can not be prepared locally by two parties, physical systems must be brought together to interact, which means that a non-local unitary operator must necessarily act in the physical system to produce an entangled state. Besides, the "level of entaglement" is bounded and cannot be infinitely increased by non-local transformations [GT09].

## 5.3 Multipartite entanglement

Let us consider a pure N-partite system. We say that the state of the system is *fully separable* if it is a product state of all the parties state vectors:

$$|\psi\rangle = \bigotimes_{i=1}^{N} |\phi_i\rangle \tag{5.10}$$

If this is not the case, there exist some entanglement. This entanglement does not have to be N-partite entanglement. In this sense, we say that a pure state is *m-separable* if the N-partites state vector can be split into $m$ parts as:

$$|\psi\rangle = \bigotimes_{i=1}^{m} |\varphi_i\rangle, \qquad 1 < m < N \tag{5.11}$$

So let us consider a system constituted of $N$ qubits. The general wave function of the system will be:

$$|\psi\rangle = \sum_{s_1,\ldots,s_N \in \{0,1\}} c_{s_1,\ldots,s_N} |s_1,\ldots,s_N\rangle \tag{5.12}$$

where the sum extends over all possible indices combinations. In the general case, a total of $2^N$ coefficients are necessary to fully describe the state of the system. Suppose now the case where the general wave function can be written as the product of the states of two subsystems:

$$|\psi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle \tag{5.13}$$

where $|\phi_1\rangle$ is the state of the first $k$ qubits $\{q_1,...,q_k\}$ and $|\phi_2\rangle$ the state of the $N-k$ following qubits $\{q_{k+1},...,q_N\}$. In this case, the total amount of coefficients necessary to characterize the system is: $2^k+2^{N-k}$, which is exponentially less than in the general case. The number of coefficients necessary to characterize the system gives us an idea of the entanglement level. With this in mind, we introduce the following Lemma 5.4:

**Lemma 5.4.** *The Schmidt decomposition applies not only to bipartite systems but to any number of systems where we can make a bipartite cut. For example, given a state $|\phi\rangle_{ABCDE}$ on systems ABCDE, we could make a bipartition AB-CDE and write a Schmidt decomposition for the state[1] as follows:*

$$|\phi\rangle_{ABCDE} = \sum_y \beta_y |y\rangle_{AB} |y\rangle_{CDE} \tag{5.14}$$

*where $\{|y\rangle_{AB}\}$ is an orthonormal basis for the joint system AB and $\{|y\rangle_{CDE}\}$ is an orthonormal basis for the joint system CDE.*

## 5.4 Entanglement entropy

In order to quantify the entanglement of a system, the *von Neumann entropy* is frequently used. For a system in a state $\rho$ it is given by[2]:

$$S = -\mathrm{Tr}[\rho \log_2 \rho]. \tag{5.15}$$

### 5.4.1 Bipartite entanglement entropy

Let us now consider a bipartition of a system of $N$ qubits into two parts, $A$ and $B$, with $k$ and $N-k$ qubits, respectively. To quantify the entanglement of the state of the composite system we require the von Neumann entanglement entropy $S$, defined as the von Neumann entropy of either of the reduced subsystems:

$$S(A) = S(\rho_A) = -\mathrm{Tr}_A[\rho_A \log_2 \rho_A], \tag{5.16}$$

where $\rho_A = \mathrm{Tr}_B \rho$. In terms of their Schmidt decomposition[3]:

$$S(A) = -\sum_{i=1}^{d} \lambda_i \log_2 \lambda_i = S(B) \tag{5.17}$$

[1] The Schmidt decomposition does not apply for partitions bigger than two [NC02].

[2] The common definition of von Neumann entropy presents a natural logarithm, however, the binary nature of qubits makes more fruitful work in base-2 logarithm and in "bits" unit.

[3] From Corollary 5.2 we know that both reduced density matrices $\rho_A$ and $\rho_B$ present the same eigenvalues according to their Schmidt decomposition.

where $d$ is the Schmidt rank. As $S(A) = S(B)$, it makes sense talking about entropy as a measure of the entanglement of the state in the way we define it. From (5.17) we also see that entropy values goes from 0 (*product state*) to $\log_2(min\{d_A, d_B\})$ in the maximally entangled states[4].

### 5.4.2 Multipartite entanglement entropy

As we stated in Section 5.3, the classification of entangled states in pure multipartite systems ($m > 2$ subsystems) is notably richer than the bipartite case. Several methods have been developed for measuring entanglement in these systems: *Tangle*, *Schmidt measure*, *measures based on normal forms*, etc. [HHHH09]. However, we will focus in a generalization of the entropy of entanglement to the multipartite case.

Consider an $N$ qubit system given by the generic state function (5.12). Let $U = \{1, 2, ..., N\}$ and $\Pi$ the set of all bipartitions of $U$. For example, consider a bipartition $\pi = (\pi_1, \pi_2 = \overline{\pi_1}) \in \Pi$, where: $\pi_1 \subseteq U$, $\pi_1 \neq \emptyset$ and $\pi_1 \cup \pi_2 = U$. The total number of different bipartitions composing $\Pi$ is equal to $2^{N-1} - 1$. Then, for a given bipartition $\pi = (\pi_1, \pi_2)$:

$$\begin{aligned} \pi_1 &= \{i_1, i_2, ..., i_k\}, \quad i_1 < i_2 < i_k \\ \pi_2 &= \{j_1, j_2, ..., j_{N-k}\}, \quad j_1 < j_2 < j_{N-k} \end{aligned} \tag{5.18}$$

where letters $i$ and $j$ represent the different qubits. Therefore, we have the corresponding matrix:

$$M^{\pi}(|\psi\rangle) = [c_{\pi_1, \pi_2}] \tag{5.19}$$

which corresponds to the bipartite division $\pi$ of the $N$-qubit state into two parts given by the sequences $\pi_1$ and $\pi_2$. We now apply the Schmidt decomposition by getting the singular value decomposition of the matrix $M^{\pi}(|\psi\rangle)$. Then, with the knowledge of the Schmidt coefficients, we can quantify the entanglement of the given bipartition $\pi$ by calculating the von Neumann entropy: $S(\pi, |\psi\rangle)$, given by (5.17). Extending this procedure to all the bipartitions of $\Pi$ we compute the total entropy of the system as:

$$S(|\psi\rangle) = \sum_{\pi \in \Pi} S(\pi, |\psi\rangle). \tag{5.20}$$

This definition of the entropy of a given state of the system is useful as it obeys some natural conditions that should be fulfilled for this kind of quantity; such as the local unitary invariance and monotonicity with respect to local manipulations [GK16].

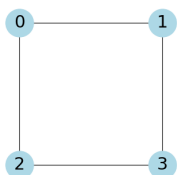
[4] This result can be obtained by finding the maximum of entropy defined as (5.17) under the restriction $\sum_i \lambda_i^2 = 1$ by using Lagrange multipliers.

## 5.5 Application case: Ising model

Consider an optimization problem where the objective is to minimize a cost function to determine the optimal combination, as we did in Section 4.3. Our main goal now is to analyze the entanglement present in the $N$-qubit state in each minimization step of an optimization problem. For that purpose, we will consider an $m \times m$ square lattice Ising model with Hamiltonian (A.1):

$$H(\sigma) = -\mu_0 \sum_j h_j \sigma_j - \sum_{<i,j>} J_{ij} \sigma_i \sigma_j \tag{5.21}$$

where the sums extend over all the $N = m^2$ particles of the lattice and $< i, j >$ indicates that the sum extends over nearest neighbours. We will consider that the parameters of the Hamiltonian take the following values: $\mu_0 h_j = 2$, $J_{ij} = -1$; and we will not concern much about their units and just take arbitrary units. As the internal interaction in the lattice only takes place between the nearest neighbors, we require to define the adjacency matrix $A$, whose elements indicate whether there is an interaction between the spins located at each pair of vertices. Let us focus in the case of $m = 2$ just for illustration:




$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

Figure 5.1: Bidimensional square lattice and its adjacency matrix.

Just as we did with the Exact cover problem, we will associate to each lattice vertex a qubit with value 1 if its spin is up ($\sigma_j = 1$) and 0 if it is down ($\sigma_j = -1$). To find the ground state energy combination we will employ a quantum simulator and the VQE algorithm. Particularly, we will use the *SLSQP* optimizer, an initial state *Zero* and a trial wave function *Ry* with 3 layers of depth and linear entanglement. This lead us to the solution:

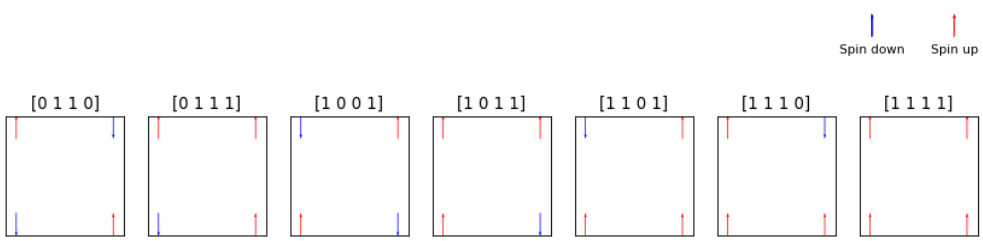



Figure 5.2: Square Ising model optimum configurations. Energy: $E \approx -4$.

As we said before, we want to quantify the level of entanglement of each state in the minimization process. To achieve that we could employ expression (5.20), however, finding all the bipartitions of a given state is an NP-problem and implies a high computational cost as the number of qubits grow. Therefore, we consider an educated approximation where, for a given state, we only consider the combinations: $\{\{s_1\},\{s_2,s_3,s_4\}\},\{\{s_1,s_2\},\{s_3,s_4\}\}$, $\{\{s_1,s_2,s_3\},\{s_4\}\}$, being $s_i$ qubit $i$, and as an approximate upper bound to entropy we take $2^{N-1}-1$ times the highest entropy among the Schmidt bipartite decomposition's of the cited combinations. Then we repeat this process to the successive states achieved in the minimization, resulting in Figure 5.3.

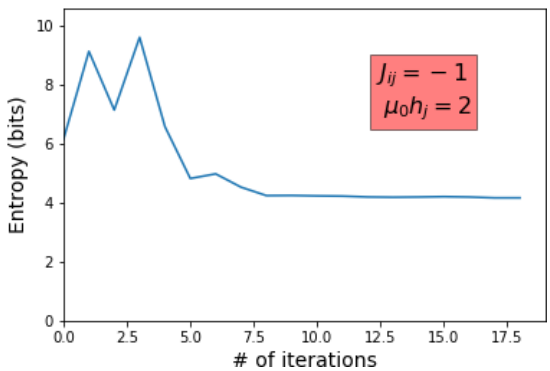



Figure 5.3: Entropy evolution in the successive iterations of the minimization process with low magnetic field (*frustrated case*).

We see that the existence of degeneracy in the ground state, also known as *frustration*, causes a non-zero final entropy. Finally, let us consider the case of an stronger magnetic field (see Figure 5.4).

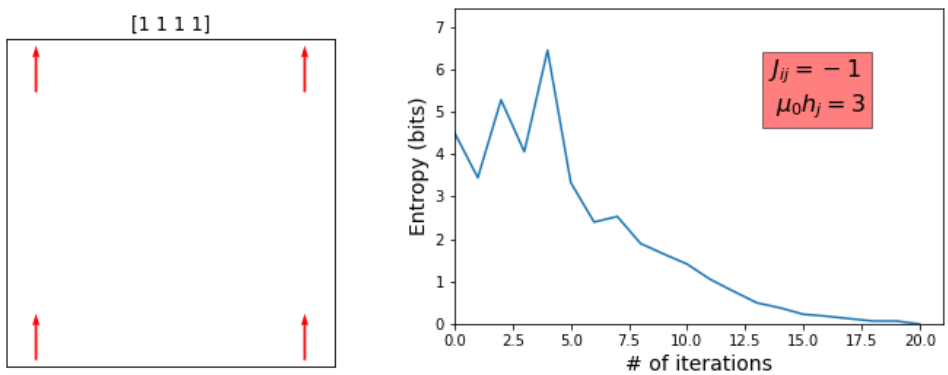



Figure 5.4: Configuration of the ground state and entropy evolution in the successive iterations of the minimization process with high magnetic field (*non-frustrated case*).

In the same way the degeneracy is reduced the same happens with the entropy, achieving a zero entropy in the case of a full product state vector.

# Chapter 6

# Conclusions

To sum up, we have started this project giving a global inside of quantum computation and quantum information origin and its evolution until the present. This overview has allowed us to notice the relevance that future advances in this field could mean towards science, society and technology. With this in mind, we went over the fundamental concepts that settle the basis for quantum computation and quantum information. This review concluded with the designing of quantum circuits capable of generating Bell and GHZ states, and their implementation in a 14-qubit quantum computer. Then, basic notations about quantum algorithms were presented with special attention in the Variational Quantum Eigensolver (VQE). Afterwards, the main advantages of quantum computation for solving optimization problems were illustrated with the resolution of the Exact cover problem, which turned out to be ruled by the Ising model. Finally, it was studied the level of entanglement of the successive states in the resolution of an optimization problem at a quantum simulator. The continuation of this project may focus on analyzing with more detail the relation between entropy and the path of minimization at optimization problems solved in quantum simulators.

Quantum computation has taught us to think physically about computation, and we have discovered that this vision yields outstanding capabilities for information processing and communication. In this sense, quantum computation also shows that many interesting aspects of Nature arise when considering large and complex systems. In accordance with this idea, further steps might be taken in applying quantum algorithms in fields which require the computational advantage that quantum computation can offer. Among these fields, Quantum Machine Learning stands out as a groundbreaking research area, halfway between quantum mechanics and computer science. For all the aforesaid, I expect to take further steps in this field under the mentoring of prominent researchers like Professor Juan José García Ripoll.

# Bibliography


[AAB+19] G Aleksandrowicz, T Alexander, P Barkoutsos, L Bello, Y Ben-Haim, D Bucher, et al. Qiskit: An Open-source Framework for Quantum Computing. https://github.com/Qiskit/qiskit, 2019.

[Ade48] Onifade Adebowale. History of the computer. 1948.

[Aro13] Daniel Arovas. Lecture notes on thermodynamics and statistical mechanics (a work in progress), 2013.

[ASM12] Mohiuddin Ahmed, T.M. Shahriar Sazzad, and Md Elias Mollah. Cryptography and state-of-the-art techniques. *IJCSI International Journal of Computer Science Issues*, 9(2-3), 2012.

[BL19] Dagmar Bruss and Gerd Leuchs. *Quantum Information: From Foundations to Quantum Technology Applications*. John Wiley & Sons, 2019.

[Car00] Cathryn Carson. The origins of the quantum theory. *Beam Line*, 30(2):6–19, 2000.

[Cho14] Jerry Chow. Dealing with errors in quantum computing. https://www.ibm.com/blogs/research/2014/06/dealing-with-errors-in-quantum-computing/, Jun 2014.

[Com16] Rigetti Quantum Computing. Variational-quantum-eigensolver (vqe). https://grove-docs.readthedocs.io/en/latest/vqe.html, 2016.

[CTDLD06] Claude Cohen-Tannoudji, Bernard Diu, Frank Laloe, and Bernard Dui. *Quantum Mechanics (2 vol. set)*. Wiley-Interscience, 2006.

[DJ92] David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439(1907):553–558, 1992.

[ENM15] Issam El Naqa and Martin J. Murphy. What is machine learning? In *Machine Learning in Radiation Oncology*, pages 3–11. Springer, 2015.

[Esf07] Behnam Esfahbod. File:p np np-complete np-hard.svg. https://commons.wikimedia.org/wiki/File:P_np_np-complete_np-hard.svg, Nov 2007.

[Fer18] Joseph Jon Fernandez. Quantum computing: The state of the art. https://medium.com/quantum1net/quantum-computing-the-state-of-the-art-979c30d1ccaf, Jan 2018.

[Fey82] Richard P. Feynman. Simulating physics with computers. *International journal of theoretical physics*, 21(6):467–488, 1982.

[Gal17] Jean Henri Gallier. Introduction to The Theory of Computation. chapter 10. Some NP-Complete Problems, cis 511 course. university of pennsylvania. https://www.seas.upenn.edu/~cis511/notes/cis511-sl15.pdf, May 2017.

[GF19] Javier Galván Fraile. Quantum computation: Theory and implementation at ibm q (TFG). https://github.com/JavierGalvan9/TFG-Javier_Galvan_Fraile, 2019.

[GHZ89] Daniel M. Greenberger, Michael A. Horne, and Anton Zeilinger. Going beyond bell's theorem. In *Bell's theorem, quantum theory and conceptions of the universe*, pages 69–72. Springer, 1989.

[GK16] Roman Gieralak and Wojciech Kozik. Morphology of two-partite entanglement in some, highly entangled, small qubit systems. *Studia Informatica*, 37(2):45–56, 2016.

[GT09] Otfried Gühne and Géza Tóth. Entanglement detection. *Physics Reports*, 474(1-6):1–75, 2009.

[HHHH09] Ryszard Horodecki, Paweł Horodecki, Michał Horodecki, and Karol Horodecki. Quantum entanglement. *Reviews of modern physics*, 81(2):865, 2009.

[ibm18] Quantum devices simulators. https://www.research.ibm.com/ibm-q/technology/devices/, Jun 2018.

[IR17] IBM QX team IBM Research. User guide. https://quantumexperience.ng.bluemix.net/proxy/tutorial/full-user-guide/introduction.html, 2017.

[KMT+17] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M. Chow, and Jay M. Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549(7671):242, 2017.

[Kop19] Dawid Kopczyk. Quantum algorithms: Variational-quantum-eigensolver. http://dkopczyk.quantee.co.uk/vqe/, May 2019.

[LZH19] Gao-Feng Luo, Ri-Gui Zhou, and Wen-Wen Hu. Novel quantum secret image-sharing scheme. *Chinese Physics B*, 28(4):040302, 2019.

[NC02] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

[Ng16] Andrew Ng. Machine Learning. Coursera. Stanford University, 2016.

[Pan97] Bruce Pandolfini. *Kasparov and Deep Blue: The historic chess match between man and machine*. Simon and Schuster, 1997.

[PGN+17] Edwin Pednault, John A Gunnels, Giacomo Nannicini, Lior Horesh, Thomas Magerlein, Edgar Solomonik, and Robert Wisnieff. Breaking the 49-qubit barrier in the simulation of quantum circuits. https://arxiv.org/abs/1710.05867, 2017.

[PLNT12] ZH Peng, Yu-xi Liu, Y Nakamura, and JS Tsai. Fast generation of multiparticle entangled state for flux qubits in a circle array of transmission line resonators with tunable coupling. *Physical Review B*, 85(2):024537, 2012.

[Ram18] Anita Ramanan. Quantum gates and circuits: The crash course. https://blogs.msdn.microsoft.com/uk_faculty_connection/2018/02/26/quantum-gates-and-circuits-the-crash-course/, Feb 2018.

[Sch95] Benjamin Schumacher. Quantum coding. *Physical Review A*, 51(4):2738, 1995.

[Sha48] Claude Elwood Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

[Tur37] Alan Mathison Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London mathematical society*, 2(1):230–265, 1937.

[Vaz12] Umesh Vazirani. Qubits, quantum mechanics, and computers. Course Chem/CS/Phys191. University of California. Berkeley . http://www-inst.eecs.berkeley.edu/~cs191/sp12/notes/chap1&2.pdf, 2012.

[VN93] John Von Neumann. First draft of a report on the EDVAC. *IEEE Annals of the History of Computing*, 15(4):27–75, 1993.

[Wil17] Mark M. Wilde. Quantum information theory. Course PHYS 7895. Louisiana State University. http://www.markwilde.com/teaching/2017-spring-qit/, 2017.

# Appendices

## A Ising model

### General model

Consider a set $\Lambda$ of lattice sites, each with a set of adjacent sites forming a d-dimensional lattice. For each lattice site $k \in \Lambda$, there is a discrete variable $\sigma_k$ such that $\sigma_k \in \{-1, +1\}$, representing the site's spin. A spin configuration, $\sigma = (\sigma_k)_{k\in\Lambda}$, is an assigment of spin value to each lattice site [Aro13].

For any two adjacent sites $i, j \in \Lambda$, there is an interaction $J_{ij}$. Also a site $j \in \Lambda$ has an external magnetic field $h_j$ interacting with it. The energy of a configuration $\sigma$ is given by the Hamiltonian:

$$H(\sigma) = -\mu_0 \sum_j h_j \sigma_j - \sum_{<i,j>} J_{ij}\sigma_i\sigma_j \tag{A.1}$$

where $\mu_0$ is the magnetic moment, the first sum is over pairs of adjacent spins (every pair counted once) and the notation $< i, j >$ indicates that sites $i$ and $j$ are nearest neighbours.

The probability of finding the system with configuration $\sigma$ is given by:

$$P(\sigma) = \frac{e^{-\beta H(\sigma)}}{Z_\beta}, \qquad Z_\beta = \sum_\sigma e^{-\beta H(\sigma)} = \mathrm{Tr} e^{-\beta H(\sigma)}, \qquad \beta = \frac{1}{k_B T} \tag{A.2}$$

where this probability is maximum in the case of being $\sigma$ the ground state of the system.

### Qualitative discussion

Ising models can be classified according to the sign of the interaction. If for all pairs $i, j$:

$$\begin{cases} J_{ij} > 0 \longrightarrow \textit{Spins parallel } (\textit{Ferromagnetic interaction}) \\ J_{ij} < 0 \longrightarrow \textit{Spins antiparallel } (\textit{Antiferromagnetic interaction}) \\ J_{ij} = 0 \longrightarrow \textit{Spins not interacting} \end{cases} \tag{A.3}$$

When no magnetic field affects the spins ($h_j = 0$), the Ising model is symmetric under switching the value of the spin in all the lattice sites. A non-zero field, therefore, breaks the symmetry. For our purposes, we will consider that the interaction strength is the same for all pairs: $J = J_{ij}$ and also the field will be the same over each spin: $H = h_j$, so from (A.1) we have:

$$H(\sigma) = -\mu_0 H \sum_j \sigma_j - J \sum_{<i,j>} \sigma_i \sigma_j \tag{A.4}$$

# B Schmidt decomposition

***Proof: Lemma 5.1.*** Consider systems A and B with dimensions $d_A$ and $d_B$, respectively, where $d_A \geq d_B$. Let $\{|\phi\rangle_A\}$ and $\{|\phi\rangle_B\}$ be any fixed orthonormal basis for systems A and B. Then the global state $|\psi_{AB}\rangle$ can be written as:

$$|\psi\rangle_{AB} = \sum_{j=1}^{d_A} \sum_{k=1}^{d_B} c_{jk} |\phi_j\rangle_A |\phi_k\rangle_B \tag{B.1}$$

Let us consider now the matrix $C$ formed by the complex coefficients $c_{jk}$ which is an $d_A \times d_B$ matrix:

$$[C_{jk}] = c_{jk} \tag{B.2}$$

Since every matrix has a singular value decomposition (SVD) for $C$ we can write:

$$C = U \Lambda V^{\dagger} \tag{B.3}$$

where $U$ is a $d_A \times d_A$ unitary matrix, $V$ is a $d_B \times d_B$ unitary matrix and $\Sigma$ is a $d_A \times d_B$ matrix with $d$ real, strictly positive coefficients $\lambda_i$ along the diagonal and zeros elsewhere. If we denote the matrix elements of $U$ as $u_{ji}$ and those of $V$ as $v_{ik}$, the above matrix equation (B.3) is equivalent to the following set of equations:

$$c_{jk} = \sum_{i=1}^{d} u_{ji} \lambda_i v_{ik} \tag{B.4}$$

Making this substitution into the expression for the state (B.1) we get:

$$|\psi\rangle_{AB} = \sum_{j=1}^{d_A} \sum_{k=1}^{d_B} \left( \sum_{i=1}^{d} u_{ji} \lambda_i v_{ik} \right) |\phi_j\rangle_A |\phi_k\rangle_B \tag{B.5}$$

Readjusting the terms we find that:

$$|\psi\rangle_{AB} = \sum_{i=1}^{d} \lambda_i \left( \sum_{j=1}^{d_A} u_{ji} |\phi_j\rangle_A \right) \otimes \left( \sum_{k=1}^{d_B} v_{ik} |\phi_k\rangle_B \right) = \sum_{i=1}^{d} \lambda_i |i\rangle_A |i\rangle_B \tag{B.6}$$

where we defined the orthonormal basis on systems A and B as:

$$\begin{aligned} |i\rangle_A &\equiv \sum_{j=1}^{d_A} u_{ji} |\phi_j\rangle_A \\ |i\rangle_B &\equiv \sum_{k=1}^{d_B} v_{ik} |\phi_k\rangle_B \end{aligned} \tag{B.7}$$

It is easy to see that setting the normalization condition over the state $|\psi\rangle_{AB}$ requires that the Schmidt coefficients satisfy the relation:

$$\sum_{i=1}^{d} \lambda_i^2 = 1 \tag{B.8}$$

Let us proof now that the sets $\{|i\rangle_A\}$ and $\{|i\rangle_B\}$ are orthonormal basis of systems A and B, respectively. Starting with system A we remember that $dim(\mathscr{H}_A) = d_A$ and that $\{|\phi_j\rangle_A\}$ constitutes an orthonormal basis of system A. From (B.7) we see that, since $i$ varies from 1 to $d_A$, $dim(\{|i\rangle_A\}) = d_A$. Let us see that this set satisfies the orthonormal relation. For this purpose we will ignore notation related to the system in the bra and kets:

$$\begin{aligned}\langle l|i\rangle = \left(\sum_{k=1}^{d_A} u_{lk}^* \langle k|\right)\left(\sum_{j=1}^{d_A} u_{ji}|j\rangle\right) = \sum_{k=1}^{d_A}\sum_{j=1}^{d_A} u_{lk}^* u_{ji}\langle k|j\rangle = \\ = \sum_{k=1}^{d_A}\sum_{j=1}^{d_A} u_{lk}^* u_{ji}\delta_{kj} = \sum_{k=1}^{d_A} u_{lk}^* u_{ki} = \delta_{li}\end{aligned} \tag{B.9}$$

where the unitary matrix condition has been used in the last step. Therefore, the set $\{|i\rangle_A\}$ cosntitute a basis of system A. For system B the same steps must be followed achieving the same result. □

***Proof: Corollary 5.2.*** The density operator of the composite system is given by:

$$\rho_{AB} = |\psi\rangle_{AB}\langle\psi| \tag{B.10}$$

where $|\psi\rangle_{AB}$ can be written in its Schmidt decomposition (B.6) resulting in the density operator:

$$\rho_{AB} = \sum_{i,j=1}^{d} \lambda_i\lambda_j(|i\rangle_A\langle j|)\otimes(|i\rangle_B\langle j|) \tag{B.11}$$

Then, the reduced density operator for system A is:

$$\begin{aligned}\rho_A = \mathrm{Tr_B}(\rho_{AB}) = \sum_{k=1}^{d} {}_B\langle k|\left(\sum_{i,j=1}^{d} \lambda_i\lambda_j(|i\rangle_A\langle j|)\otimes(|i\rangle_B\langle j|)\right)|k\rangle_B = \\ = \sum_{k=1}^{d}\sum_{i,j=1}^{d} \lambda_i\lambda_j\delta_{ik}\delta_{jk}(|i\rangle_A\langle j|) = \sum_{k=1}^{d}\lambda_k^2(|k\rangle_A\langle k|)\end{aligned} \tag{B.12}$$

where the orthonormalization relation of states $\{|i\rangle_A\}$ and $\{|i\rangle_B\}$ has been used. Proceeding in the same way for the reduced density operator of system B the following result is obtained:

$$\rho_B = \mathrm{Tr_A}(\rho_{AB}) = \sum_{k=1}^{d}\lambda_k^2(|k\rangle_B\langle k|) \tag{B.13}$$

This shows that both reduced density operators present the same eigenvalues. □