UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DEPARTAMENT DE MATEMÀTICA APLICADA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dept. of Applied Mathematics

Physical principles and mathematical model of a three-qubit quantum computer

Master's Thesis

Master's Degree in Mathematical Research

AUTHOR: Orlando, Diego

Tutor: Garcia March, Miguel Angel

External cotutor: Falcó Montesinos, Antonio

ACADEMIC YEAR: 2023/2024

Università degli Studi di Ferrara

Máster Universitario en Investigación Matemática

Physical principles and mathematical model of a three-qubit quantum computer

Supervisor:
Prof. Miguel Ángel Garcia March

Student:
Diego Orlando

Co-supervisor:
Prof. Antonio Falcó Montesinos

Academic year 2023/2024

Acknowledgments

I would like to reserve a few words to thank the people who have supported me throughout the completion of this work.

I would like to express my sincere thanks to my supervisor, Professor Miguel Ángel Garcia March, for his faith in me and in my abilities. His patience and willingness to help, joint with his mastery of the field, allowed me to change my curiosity towards quantum computation in eagerness. For his kind and fruitful cooperation: Gracias Miguel Ángel!

I am also grateful to my co-supervisor, Professor Antonio Falcó Montesinos, for his enthusiasm and for his captivating knowledge, both of which he never failed to share. His guidance in operating the computer was impeccable, greatly contributing to my understanding of the machine. For his admirable and sincere attitude and his teachings: Gracias Antonio!

In no way I will be able to express here the degree to which I feel grateful towards my family. For their ever-lasting support and unconditioned love, for the genuine joy they find in my accomplishments and for the warmness I feel regardless of the distance that divides us: Grazie Famiglia!

I would also like to mention my friends. Whether colleagues or not, whether long-date or not, I thank you all for the moments we have shared to this day, and for the ones you will gift me in the future. To my friends from "Azienda agricola 2 compagnie", "Notiziona","Extranjeros" and all the others: Grazie Amici!

Abstracts

This section contains the abstract of this master's thesis in different languages. These are, in respective order: English, Spanish, Valencian and Italian.

Abstract

In this work, we will describe physically, model, and test a three-qubit quantum computer. More specifically, we will use SpinQ Triangulum, acquired by a project with co-Principal investigators from the Universitat Politècnica de València and Universidad CEU Cardenal Herrera.

After a brief mathematical introduction on the fundamentals of quantum computation, we shall present the quantum algorithms we will be running on the computer. These are Grover's algorithm for unstructured database search and the Harrow–Hassidim–Lloyd (HHL) algorithm for solving linear systems of equations.

Once this is done, we will look into the physics of the different proposals for quantum computation. The most researched ones will be discussed before focusing our attention on nuclear magnetic resonance (NMR) computing.

The last section of this work contains the results of the experiments on the quantum computer at our disposal. We will first explain the principles behind its functioning and then discuss the implementation on an NMR computer of Grover's and HHL algorithms. Additionally, we will also test for Bell's inequality, to confirm whether or not we have entanglement. In the end, we will perform the actual tests and hence evaluate the performance of the computer.

Resumen

En este trabajo, describiremos físicamente, modelaremos matemáticamente, y probaremos el funcionamiento práctico de un ordenador cuántico de tres qubits, SpinQ Triangulum, adquirido en un proyecto con investigadores principales de la Universitat Politècnica de Valéncia y la Universidad CEU Cardenal Herrera.

Tras una breve introducción matemática sobre los fundamentos de la computación cuántica, presentaremos los algoritmos cuánticos que ejecutaremos en la computadora. Estos son el algoritmo de Grover para la búsqueda en bases de

datos no estructuradas y el algoritmo Harrow-Hassidim-Lloyd (HHL) para resolver sistemas lineales de ecuaciones.

Una vez hecho esto, nos adentraremos en la física de las diferentes propuestas de computación cuántica. Se discutirán los más comunes antes de centrar nuestra atención en la computación usando resonancia magnética nuclear (NMR).

En la última sección de este trabajo, discutimos la física de SpinQ Triangulum y la implementación de los algoritmos que hemos visto. Como otra prueba, también vamos a probar si la desigualdad de Bell está o no verificada, para ver si hay entralazamiento. Finalmente, haremos pruebas prácticas y evaluaremos el comportamiento del ordenador.

Resum

En este treball, descriurem físicamente, modelarem matemáticamente, i probarems el funcionament práctico d'un ordenador cuántico de tres qubits, SpinQ Triangulum, adquirit en un projecte amb investigadors principals de la Universitat Politécnica de Valéncia i la Universitat CEU Cardenal Herrera. Després d'una breu introducción matemática sobre els fonaments de la computación cuántica, presentarem els algorismes cuánticos que executarem en la computadora. Estos són l'algorisme de Grover per a la búsqueda en bases de dades no estructurades i l'algorisme Harrow-Hassidim-Lloyd (HHL) per a resoldre sistemes lineals d'equacions. Una vegada fet això, ens endinsarem en la física de les diferents propostes de computaci ón cuántica. Es discutirán els más comuns abans de centrar nostra atención en la computación usant ressonància magnética nuclear (NMR). En la última sección d'este treball, discutim la física de SpinQ Triangulum i la implementación dels algorismes que hem vist. Com una altra prova, también provarem si la desigualtat de Bell está o no verificada, per a veure si hi ha entralazamiento. Finalment, farem proves prácticas i avaluarem el comportament de l'ordinador.

Riassunto

L'obbiettivo di questo lavoro è di descrivere dal punto di vista fisico e matematico un computer quantistico di 3 qubits , per poi verificare i risultati teorici tramite degli esperimenti. Nello specifico lo studio sarà basato su SpinQ Triangulum, un computer acquisito tramite un progetto di investigazione congiunto tra l'Universitat Politècnica de València e l'Universidad CEU Cardinal Herrera.

Dopo una breve introduzione matematica sulle fondamenta della computazione quantistica, presenteremo i due algoritmi che andremo poi a implementare sul computer. Questi sono: l'algoritmo di Grover per la ricerca in una base dati non ordinata e l'algoritmo Harrow–Hassidim–Lloyd (HHL) per la risoluzione di sistemi lineari.

Ci concentreremo poi sui principi fisici che caratterizzano le varie proposte per la realizzazione di un computer quantistico, introducendo le più accreditate, prima di dedicarci alla computazione tramite risonanza magnetica nucleare (NMR).

Nella parte finale di questo elaborato, illustreremo la struttura del computer a nostra disposizione e spiegheremo come possono essere implementati gli algoritmi di cui sopra. Come ulteriore test, verificheremo la violazione della disuguaglianza di Bell, ottenendo così dati sulla presenza di entanglement quantistico. I risultati ottenuti saranno infine discussi, analizzando quindi il comportamento del computer.

Keywords: Quantum computers; Nuclear magnetic resonance; Quantum algorithms; Entanglement; Bell's inequality;

Contents

1 Introduction **1**
- 1.1 Prior knowledge . . . 2
  - 1.1.1 Hilbert spaces . . . 2
  - 1.1.2 Quantum mechanics and quantum theory . . . 3
- 1.2 Structure of the work . . . 6

2 Fundamentals of quantum computation **7**
- 2.1 Quantum bits . . . 7
- 2.2 Quantum gates . . . 8
- 2.3 Quantum algorithms . . . 13
  - 2.3.1 Deutsch-Jozsa algorithm . . . 13
  - 2.3.2 Grover's algorithm . . . 15
  - 2.3.3 Quantum Fourier transform . . . 17
  - 2.3.4 Quantum phase estimation . . . 20
  - 2.3.5 HHL algorithm . . . 21

3 Quantum computers **27**
- 3.1 Physical realization . . . 27
- 3.2 Challenges . . . 28
- 3.3 Proposed models . . . 29
- 3.4 Nuclear Magnetic Resonance . . . 30

4 Experiments with NMR **35**
- 4.1 SpinQ Triangulum . . . 35
- 4.2 Grover's algorithm in NMR . . . 38
- 4.3 HHL algorithm in NMR . . . 44
- 4.4 Bell's inequality in NMR . . . 49
- 4.5 Discussion . . . 51

Conclusions and outlook **53**

Bibliography **55**

Chapter 1

Introduction

Over the last century, computers have become an essential component to our society. The technology they mount has indeed advanced and developed, but if we were to take a look to a generic modern day computer, its design would not be drastically different to one of its ancestors. How come then that the former can achieve complex feats in comparison with the latter?

The answer mainly lies in one of the biggest differences one can encounter between these two computers, that is their size. To be more specific, it is not the size of the computer itself that matters, whereas the size of the transistors, a key component of their motherboard. Transistors are small semiconductors which can control the electric signal. In modern day computers, they are the building blocks of the CPU. Roughly speaking, a significantly larger number of transistors translates into a greater computing power.

In 1965, Gordon Moore formulated a prediction on the crescent number of transistors per computer chip, stating that said number would double about every year. Ten years later, in 1975, Moore corrected himself acknowledging that a two year cycle was more precise. This statement, known as Moore's Law, has since guided the industry long-term planning and research. In this process, transistors have shrank down exponentially, at least till now. We are in fact reaching an halting point for Moore's Law [20], the size of transistors is already in the order of nanometers, lengths at which classical physical laws that govern our macroscopic world are not enough anymore to correctly predict the developments of complex systems, we have already entered the realm of quantum physics.

In a lecture entitled "There is plenty of room at the bottom: An Invitation to Enter a New Field of Physics", presented at the annual meeting of the American Physical Society in 1959, Richard Feynman [9] invited the audience to harness the power of infinitesimally small objects. While in that occasion he was not talking about quantum computing specifically, it very much applies to the context. Instead of regarding the quantum phenomena as problems or barriers preventing us to advance any further we should study them, because if we were able to understand and use this aspect to our advantage, then we would really have plenty of room for

improvement.

While the modern quantum theory started developing around 1920s, it would be only in the '80s that Paul Benioff and David Deutsch proposed the idea of a quantum Turing machine [6, 12]. The first quantum algorithms, such as the Deutsch's algorithm or the Bernstein-Vazirani algorithm, for black box problems, started emerging rapidly after. In the following years various breakthroughs were accomplished, highlighting how quantum computing could serve real world applications.

The first of which came from Peter Shor in 1994, with two algorithms capable of breaking commonly used encryption methods. Then in 1996, Lov Grover proposed an algorithm for the unstructured search problem. Finally, that same year, Seth Lloyd proved that quantum computers could simulate quantum systems exponentially faster then classical computers.

1.1 Prior knowledge

We will now recall some of the fundamental mathematical and physical concepts to which we will be referring throughout the thesis.

1.1.1 Hilbert spaces

Definition 1.1. Given a vector space V over a field $\mathbb{K}$, an **inner product** is a map

$$\langle \cdot, \cdot \rangle \colon V \times V \to \mathbb{K}$$

which satisfies the following properties:

- $\langle v, v \rangle \geq 0$, for every $u \in V$,
- $\langle u, v \rangle = \overline{\langle v, u \rangle}$, for every $u, v \in V$,
- $\langle u, av + bw \rangle = a\langle u, v \rangle + b\langle u, w \rangle$ for every $u, v, w \in V$ and every $a, b \in \mathbb{K}$.

When $\langle \cdot, \cdot \rangle$ is an inner product, we say that the pair $(V, \langle \cdot, \cdot \rangle)$ is an **inner product space**.

One ambiguity worth addressing is that in most fields of mathematics the inner product is taken to be linear in the first term and conjugate linear in the second one. On the other hand, in quantum mechanics it is common to define the linearity on the right-hand term to uniform with Dirac's notation.

Definition 1.2. Let V be a vector space over the field $\mathbb{K}$. A **norm** on V is a function

$$\|\cdot\| \colon V \to \mathbb{K},$$

such that

- $\|\lambda v\| = |\lambda|\|v\|$ for all $v \in V$,
- $\|u + v\| \leq \|u\| + \|v\|$ for all $u, v \in V$,
- $\|v\| = 0$ implies $v = 0$ for all $v \in V$.

When $\|\cdot\|$ is a norm, we say that the pair $(V, \|\cdot\|)$ is a **normed space**.

Norms are closely related to metrics, and so topology. A space can be associated with many norms, inducing different topologies. In the case of an inner product space, there is a particularly natural way of defining a norm.

Definition 1.3. Let $\langle\cdot,\cdot\rangle_V : V \times V \to \mathbb{K}$ be an inner product. We say that

$$\|v\|_V = \sqrt{\langle v, v\rangle_V} \quad \forall v \in V$$

is the **norm induced by the inner product**.

Definition 1.4. Let $(X, \|\cdot\|)$ be a normed space and $\{x_n\}_{n\in\mathbb{N}} \in X$ a sequence. We say that the sequence is a **Cauchy sequence** if

$$\forall \epsilon \in \mathbb{R}, \exists N \quad s.t. \quad \forall m, n > N \text{ we have that } \|x_n - x_m\| < \epsilon.$$

When every Cauchy sequence in $(X, \|\cdot\|)$ converges to a value in X, we say that the space $(X, \|\cdot\|)$ is **complete**.

Definition 1.5. Let $(H, \langle\cdot,\cdot\rangle_H)$ be an inner product space and $\|\cdot\|_H$ be the norm induced by the inner product. If the normed space $(H, \|\cdot\|_H)$ is complete, then we say that the pair $(H, \|\cdot\|_H)$ is an **Hilbert space**.

When $(H, \|\cdot\|_H)$ is an Hilbert space we will denote the pair only by writing $\mathbb{H}$.

1.1.2 Quantum mechanics and quantum theory

Quantum mechanics (QM) is a field of physics which describes the behaviour of infinitesimally small objects when they can be defined as a quantum system (which implies, loosely speaking, that its dynamics follows the quantum mechanical laws, and that there is a constant, the Planck constant $\hbar$, which permits, in some "nice" cases, to recover classical limit - classical mechanics - when it is made infinitely small). The following terminology is characteristic of quantum mechanics (and related to the postulates of QM).

- An **observable** is a measurable physical quantity.
- The **probability** of a result consists in the prediction of the relative frequency of its measurement.

- The **expectation value** is the prediction of the mean value of an observable in a sequence of measurements.

- A **state** is a statistical ensemble which possesses all the information about the system at a given moment.

The standard bra-ket notation introduced by Dirac is largely used in QM. This formalism is summarized in the following definition.

Definition 1.6. Let $\mathbb{H}$ be an Hilbert space an let $\mathbb{H}^*$ denote its dual. The vectors of the space are called **kets** and noted $|\psi\rangle$, while the elements of $\mathbb{H}^*$ are called **bras** and noted $\langle\varphi|$.

Under this notation, the inner product is usually written as

$$\langle\varphi|\psi\rangle\,,$$

while the tensor product, also called outer product is expressed as

$$|\psi\rangle\langle\varphi|\,.$$

For the sake of brevity, we now present four postulates to QM, without discussing them, or their implications with much precision, since this is not the scope of this work. In a similar way, throughout the thesis we will occasionally refer to some results as consequences of one or more of this axioms. The number and formulation of these postulates can vary depending on authors, we propose the following:

Postulate I. *Every isolated physical system is associated to a separable complex Hilbert space $\mathbb{H}$. A state of the system at instant t is described by a unit vector $|\psi(t)\rangle$ of the Hilbert space. The vector $|\psi(t)\rangle$ contains all the information about the system.*

Postulate II. *Every observable of a physical system is represented by a self-adjoint linear operator, acting on $\mathbb{H}$.*

Postulate III. *The only possible values of an observable that can be measured are its eigenvalues.*

Postulate IV. *The time evolution of a closed quantum system is described by the Schrodinger equation*

$$i\hbar\frac{d}{dt}\,|\psi(t)\rangle = \mathcal{H}(t)\,|\psi\rangle\,,$$

where $\mathcal{H}$ is the Hamiltonian of the system and $\hbar$ is Planck's constant divided by 2π.

Now, let $\mathbb{H}_1$ and $\mathbb{H}_2$ be two Hilbert spaces to two quantum systems. We define the composite system Hilbert space as the set of all linear combinations of all the possible tensor products of a vector of $\mathbb{H}_1$ with one of $\mathbb{H}_2$. That is, the composite system Hilbert space is

$$\mathbb{H}_1 \otimes \mathbb{H}_2 = \text{span}\{|\psi\rangle_1 \otimes |\varphi\rangle_2\},$$

with coefficient such that the resulting vector will be normalized. This definition is obviously redundant, but it will make our point without the need of fixing a basis on the starting Hilbert spaces. Looking at the vectors in $\mathbb{H}_1 \otimes \mathbb{H}_2$ we could in fact distinct two classes. One which contains all the vectors we can write as scalar product of two vectors in the initial space, while in the other we have all states that are not separable. Let

$$|s\rangle = \sum_{i,j} \alpha_{ij}(|\psi_i\rangle_1 \otimes |\varphi_j\rangle_2),$$

with $i, j \in \mathbb{N}$, be a state in the product Hilbert space. Then, if there exist

$$|\psi\rangle_1 = \sum_i \alpha_i |\psi\rangle_1 \text{ and } |\varphi\rangle_1 = \sum_j \alpha_j |\varphi\rangle_2 ,$$

such that we have

$$|s\rangle = |\psi\rangle \otimes |\varphi\rangle ,$$

we say that the state $|s\rangle$ is separable.

If a state happens to be non-separable states we say it is **entangled**.

A very well known set of entangled states are the four Bell states. Given $\mathbb{H}_1$ and $\mathbb{H}_2$ being each $\mathbb{C}^2$ (the simplest case - which we consider from here on), with respective bases $\{|0\rangle_1 , |1\rangle_1\}$ and $\{|0\rangle_2 , |1\rangle_2\}$, these are

$$\begin{aligned}
\left|\Phi^+\right\rangle &= \frac{1}{\sqrt{2}}(|0\rangle_1 \otimes |0\rangle_2 + |1\rangle_1 \otimes |1\rangle_2), \\
\left|\Phi^-\right\rangle &= \frac{1}{\sqrt{2}}(|0\rangle_1 \otimes |0\rangle_2 - |1\rangle_1 \otimes |1\rangle_2), \\
\left|\Psi^+\right\rangle &= \frac{1}{\sqrt{2}}(|0\rangle_1 \otimes |1\rangle_2 + |1\rangle_1 \otimes |0\rangle_2), \\
\left|\Psi^-\right\rangle &= \frac{1}{\sqrt{2}}(|0\rangle_1 \otimes |1\rangle_2 - |1\rangle_1 \otimes |0\rangle_2).
\end{aligned}$$

Entangled states play a key role in quantum mechanics. When two system are entangled, they cannot be represented independently anymore.

It is, in fact, been found that entangled states, such as Bell states, contradict the notion of local realism which characterizes the view of causality in classical physics. In this context, realism means that the result of a measurement is predetermined

and it is not affected by other measurements. Locality implies that in the case of spatially separated systems, the measurement of one does not affect the other. This idea of local realism is expressed mathematically through Bell's inequality. The CHSH form of Bell's inequality [19] states that the correlations resulting from local realistic theories must obey:

$$B(\alpha_1, \delta_1, \beta_2, \gamma_2) = |q(\delta_1, \gamma_2) - q(\alpha_1, \gamma_2)| + |q(\delta_1, \beta_2) - q(\alpha_1, \beta_2)| \leq 2,$$

where $\alpha_1, \delta_1, \beta_2, \gamma_2$ are particles detectors of a classical property, with only two possible outcomes, and q is a function indicating the quantum correlation between them.

Summarizing, in the presence of entangled states our experiment should violate Bell's inequality.

1.2 Structure of the work

The main body of this thesis is divided in three parts.

The first one, which is labelled Chapter 2, is about the mathematics of quantum computation. We first define the qubits, a quantum analogue of traditional bits, and the quantum gates, which transform qubits from one state to another. We then move on to study more complex structures, the algorithms. With the aim of understanding Grover's algorithm for unstructured search and a quantum algorithm for linear system of equations, we will also talk about some other related algorithm. These are Deutsch's algorithm, which is related to a particular black box problem, and two routines, one to perform a quantum version of the discrete Fourier transform and one to estimate the eigenvalue of a given eigenvector.

In Chapter 3 we will instead focus on the physical principles of quantum computing. We start by addressing the challenges that arise when building a quantum computer, summarized in DiVincenzo's criteria. Under this light, we introduce some of the most developed proposals that have been put forward to realise a quantum computer. More specifically, we talk about computation through superconductors, optical elements and ion traps before addressing the technicalities of nuclear magnetic resonance computing.

In the last chapter we introduce SpinQ Triangulum, the 3 qubits NMR quantum computer we will be using. We discuss its specifics and explain how it works, before re-adapting the quantum search and the linear systems algorithms to the 3 qubits case. These are then, together Bell's inequality, tested on the computer and the results are so discussed.

The very last section of the work are the conclusions, in which we include an outlook on further developments. In particular, we introduce the concept of non-Kolmogorovian probability and show a different mathematical formalization of quantum computing.

Chapter 2

Fundamentals of quantum computation

From the first postulate of quantum mechanics we know that the state vector posses all the information about the system. Quantum computation is about manipulating and processing this information, using particular techniques.

2.1 Quantum bits

As for their classical counterparts, quantum computers posses fundamental units in which the information is encoded.

Definition 2.1. A quantum bit, or **qubit**, is a quantum system described by a two dimensional complex Hilbert space $\mathbb{H}$. A **state** of the qubit is a unit vector $|\psi\rangle \in \mathbb{H}$, which represents the information stored in the qubit.

Being a vector, to express the state of a qubit we need to consider a basis to the Hilbert space. Depending on the context, one may choose to represent the state in a particular basis over another. Regardless of what the kets are chosen to be, we can label them in the following way.

Definition 2.2. Given $\mathbb{H}$ an n-dimensional complex Hilbert space, let $\mathcal{B} = \{|\alpha_x\rangle\}_x$, with $x = 0, \ldots, n-1$, be a base for $\mathbb{H}$. The **computational basis** is the basis obtained by labelling the elements of $\mathcal{B}$ by their index, defining the basis

$$\mathcal{B}_x = \{|0\rangle, \ldots, |n-1\rangle\},$$

where $|x\rangle = |\alpha_x\rangle$, with $x = 0, \ldots, n-1$. The elements of B_x are the **basis states**.

In practise, we will drop the x in $\mathcal{B}_x$ given the one-to-one correspondence with $\mathcal{B}$. One convention in use, stemming from classical computing, is to denote the basis states by their binary expression. Throughout this thesis we will use both notations, depending on the most practical for the use.

Example 2.3. Suppose we have 2 qubits system, which translates to a 4 dimensional complex Hilbert space $\mathbb{H}$. A vector $|\psi\rangle \in \mathbb{H}$ can be expressed as

$$|\psi\rangle = \sum_{x=0}^{3} \psi_x |x\rangle ,$$

or equivalently as

$$|\psi\rangle = \psi_0 |00\rangle + \psi_1 |01\rangle + \psi_2 |10\rangle + \psi_3 |11\rangle .$$

Recalling definition 2.1, the state of a single qubit is described by a two dimensional unit vector in the qubit's Hilbert space. We can express this vector in the basis states by

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle ,$$

with $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2+|\beta|^2= 1$. This means that the state of a qubit in a given basis is completely determined by four real valued coefficients. This is however redundant, using $|\alpha|^2+|\beta|^2= 1$, we can in fact express $|\psi\rangle$ as

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right) |0\rangle + e^{i\varphi} \sin\left(\frac{\theta}{2}\right) |1\rangle ,$$

where $0 \leq \theta \leq \pi$ and $0 \leq \varphi \leq 2\pi$.

Observe that θ and φ can be interpreted in spherical coordinates, the set of all states under this representation is called Bloch sphere.

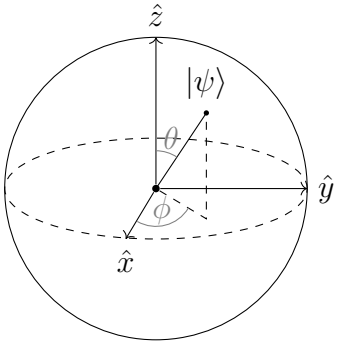


Figure 2.1: Bloch sphere containing a vector $|\psi\rangle = \cos\left(\frac{\theta}{2}\right) |0\rangle + e^{i\varphi} \sin\left(\frac{\theta}{2}\right) |1\rangle$.

As a convention, the basis state vectors $|0\rangle$ and $|1\rangle$ are mapped respectively to the north and south pole of the Bloch sphere.

2.2 Quantum gates

To modify a qubit's state we need some class of operators on the Hilbert space.

Definition 2.4. A **quantum gate** of n qubits is a unitary operator

$$U : \mathbb{H}^{\otimes n} \to \mathbb{H}^{\otimes n}.$$

The requirement for quantum gates to be unitary comes from the time evolution postulate, this reassures that the vectors remain of norm 1.

Given the Hilbert space $\mathbb{H}$, the quantum gate $U : \mathbb{H} \to \mathbb{H}$ is usually called single qubit gate.

Unary gates

We will now present the most common single qubit gates and discuss their use. The simplest gate one can imagine is the **identity gate** , which leaves the state unchanged, that is

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

A particular set of gates, useful to realise many operations, is that of the **Pauli matrices** X, Y and Z. These are defined as it follows:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

The Pauli matrices represent rotations by π radians about the x, y and z axis respectively. They are, just like the identity, involutory. That is

$$I^2 = X^2 = Y^2 = Z^2 = I.$$

Occasionally, we will change this uppercase notation stemming from classical computing, handful in circuit representation, for the algebraic notation σ^x, σ^y, σ^z, less ambiguous in formulas.

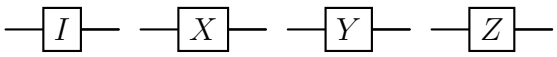


Figure 2.2: Usual circuit notation of identity and the Pauli gates.

Remark 1. The Pauli matrices and the identity matrix form a basis for the vector space of 2×2 Hermitian matrices. In particular, every 2×2 Hermitian matrix can be written as a unique linear combination of $\{I, X, Y, Z\}$. △

Another set of important gates is the one of **phase shift gates** P_φ. These leave the state $|0\rangle$ unchanged while applying a shift of φ to the phase of $|1\rangle$. Their matrix representation is

$$P_\varphi = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix}.$$

Phase shift gates are rotations along the z axis. In particular, we have

$$P(\pi) = Z, \quad S := P(\frac{\pi}{2}) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{bmatrix} = \sqrt{Z}, \quad T := P(\frac{\pi}{4}) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix} = \sqrt[4]{Z}.$$

We can also rotate about the three axis $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$ and $\hat{\mathbf{z}}$, using the gates

$$R_x(\theta) = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix},$$

$$R_y(\theta) = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix},$$

and

$$R_z(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2}. \end{bmatrix}.$$

To indicate a negative angle rotation we will write $R_{-x}(\theta)$ instead of $R_x(-\theta)$.

One last fundamental single qubit gate is the **Hadamard gate** H. Its action on the basis states creates a superposition of them. The matrix representation of an Hadarmard gate is

$$H = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Proposition 2.5. *The NOT gate X can be decomposed as HZH.*

Proof.

$$HZH = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 & 2 \\ 2 & 0 \end{pmatrix} = X$$

□

We can see that $H|0\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $H|1\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$, these two states are usually noted $|+\rangle$ and $|-\rangle$ respectively. One interesting property of these superposition states is that they possess an equal distribution of the two basis states. This suggests that the Hadamard gate can be used to prepare equally distributed superposition states.

As we shall see, being able to produce such states is particularly handy in quantum computing. This is why, When working with n qubits, we aim to build a gate capable of producing the same effects.

Let us start with $n = 2$, the basis states are $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. We are looking for a gate U such that

$$U|00\rangle = \frac{1}{2}\Big(|00\rangle + |01\rangle + |10\rangle + |11\rangle\Big). \tag{2.1}$$

It is easily checked that Eq. (2.1) is equivalent to $|+\rangle \otimes |+\rangle$. So we have

$$U|00\rangle = (H|0\rangle) \otimes (H|0\rangle) = (H \otimes H)|00\rangle . \tag{2.2}$$

According to the Kronecker product convention, we will denote $(H \otimes H)$ as $H^{\otimes 2}$. The matrix for the gate $H^{\otimes 2}$ has the following form

$$H^{\otimes 2} = \frac{1}{\sqrt{2}} \begin{bmatrix} H & H \\ H & -H \end{bmatrix} .$$

For the case of n qubits we lay the following definition

Definition 2.6. Let n be the number of qubits, the **Hadamard transfrom** of degree n is the operator $H^{\otimes n} = H \otimes H^{\otimes n-1}$, where H is the Hadamard gate operator. Its matrix representation is:

$$H^{\otimes n} = \frac{1}{\sqrt{2}} \begin{bmatrix} H^{\otimes n-1} & H^{\otimes n-1} \\ H^{\otimes n-1} & H^{\otimes n-1} \end{bmatrix} .$$

Proposition 2.7. $H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$

Proof. The statement can be proved by induction over n. Take Eqs. (2.1) and (2.2) as the first step, now suppose the claim true for the n-th case. We want to prove that

$$H^{\otimes n+1}|0\rangle^{\otimes n+1} = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^{n+1}-1} |x\rangle .$$

From the definition of Hadamard transform we know that $H^{\otimes n+1} = H \otimes H^{\otimes n}$.

$$H^{\otimes n+1}|0\rangle^{\otimes n+1} = (H \otimes H^{\otimes n})|0\rangle^{\otimes n+1} = (H|0\rangle) \otimes (H^{\otimes n}|0\rangle^{\otimes n}).$$

Substituting, using the inductive statement, we find

$$H^{\otimes n+1}|\psi\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle ,$$

which is the same as

$$H^{\otimes n+1}|\psi\rangle = \frac{1}{\sqrt{2^{n+1}}} \left[\Big(|0\rangle \otimes \sum_{x=0}^{2^n-1} |x\rangle \Big) + \Big(|1\rangle \otimes \sum_{x=0}^{2^n-1} |x\rangle \Big) \right]. \tag{2.3}$$

Now, the two summations on the right side of the equation contain the 2^n basis states of n qubits. If we were to multiply these the two sums for $|0\rangle$ and $|1\rangle$, we would obtain a summation of all the basis states of $n+1$ qubits. In other words, we have

$$\frac{1}{\sqrt{2^{n+1}}} \left[\Big(|0\rangle \otimes \sum_{x=0}^{2^n-1} |x\rangle \Big) + \Big(|1\rangle \otimes \sum_{x=0}^{2^n-1} |x\rangle \Big) \right] = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^{n+1}-1} |x\rangle .$$

This, combined with (2.3), concludes the proof. □

Controlled gates

In the class of multi qubit gates, some are capable of interconnecting the qubits to a deeper level. It is the case of controlled gates, in which the inputs can be divided into control and target qubits. Control qubits do not undergo any transformation, their state is observed to decide on whether or not apply an operation on the target.

It is possible to build a controlled gate starting from virtually any other gate, adding a conditionality to the application of a certain gate. Some of the most common examples of this are the **controlled-Z** gate

$$CZ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix},$$

and the controlled NOT, or **CNOT**.

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

The CNOT controls the state of the first qubit, if the state is measured to be $|1\rangle$, the phase of the second qubit is flipped. Conversely, if the control reads $|0\rangle$, the target is left unaffected.

Remark 2. Given Proposition 2.5, it is easy to see that

$$(I \otimes H)CZ(I \otimes H) = CNOT$$

△

Controlled gates can also have multiple control qubits, as it is the case for the Toffoli gate, which is a controlled-CNOT.

$$TOFFOLI = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

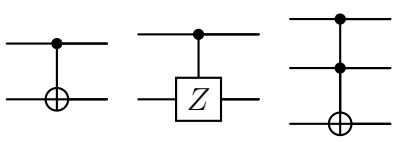


Figure 2.3: Binary CNOT, controlled-Z and Toffoli gate in circuit representation.

2.3 Quantum algorithms

The purpose of this section is to introduce the two algorithms we will be testing on the computer: Grover's and HHL. In addition to these, we will also study the first ever quantum algorithm, the Deutsch-Jozsa, as well as two famous routines used in the HHL, the quantum Fourier transform and quantum phase estimation.

2.3.1 Deutsch-Jozsa algorithm

The Deutsch-Jozsa algorithm [7] was the first quantum algorithm ever proposed offering an exponential speedup over a deterministic classical one. The algorithm is devised to solve the following black box problem:

Deutsch's problem. *Let $N \in \mathbb{N}$ be a natural number and U_f an oracle for a function $f\colon \mathbb{Z}_{2N} \to \mathbb{Z}_2$. Find a true statement in the following list*

A) the sequence $\{f(0) \dots f(2N-1)\}$ does not contain only the same value.

B) the sequence $\{f(0) \dots f(2N-1)\}$ does not contain exactly N zeros.

A function only satisfying B) is constant. Similarly, we will use the term balanced to distinguish a function only realizing A).

Based on the formulation of the problem, one could observe the following.

Remark 3. For every $f\colon \mathbb{Z}_{2N} \to \mathbb{Z}_2$, at least one of the two affirmations is true. Suppose f constant, then the sequence $f(0) \dots f(2N-1)$ is composed of either all zeros or all ones, depending on the constant value f assumes in Z_2. This means that when f is constant the claim B) is clearly true. Conversely, suppose f non constant, then trivially f satisfies the condition A). △

In the case of f not being neither constant nor balanced, whichever of the two claims the algorithm suggests will result in a correct answer. The problem is so reduced to : *Given that f is either constant or balanced, decide whether f is constant or balanced.*

A classical algorithm has not many ways to approach this problem, if not trying values until it can safely decide on the output. Clearly for a deterministic algorithm picking a random number $n = 0 \dots 2N-1$ to plug through f gives the same amount of information of selecting them in crescent order, starting from 0. It is easy to see that the best case scenario consist in only evaluating $f(0)$ and $f(1)$, finding two different values. When this happens we can safely say that the function satisfies condition A). On the other hand, the worst case scenario is that of the subsequences $f(0) \dots f(N-1)$ and $f(N) \dots f(2N-1)$ both being constant. This situation does not allow the deterministic algorithm to decide whether or not f is constant or balance until the evaluation of $f(N)$: if the value $f(N)$ is the same as that of the subsequence of the first half of the domain, then f cannot be balanced. Hence in the worst case scenario $N+1$ queries are needed.

If we were to design a quantum algorithm, we could exploit superposition states to complete the task in only one query to the oracle.

Given the same problem, we initialize the qubits to the state

$$|\psi_0\rangle = |0\rangle^{\otimes n}.$$

We then create superposition using Hadarmard transform on the query register and an Hadamard gate on the answer register

$$|\psi_1\rangle = H^{\otimes n}\,|\psi_0\rangle = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle\,,$$

and evaluate the function f using the gate U_f

$$|\psi_2\rangle = U_f\,|\psi_1\rangle = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}(-1)^{f(x)}\,|x\rangle\,.$$

Finally another Hadamard transform is applied to the query register

$$|\psi_3\rangle = H^{\otimes n}\,|\psi_2\rangle = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}(-1)^{f(x)}\Big[\frac{1}{\sqrt{2^n}}\sum_{y=0}^{2^n-1}(-1)^{x\cdot y}\,|y\rangle\,\Big],$$

which is the same as writing

$$|\psi_3\rangle = \sum_{y=0}^{2^n-1}\Big[\frac{1}{2^2}\sum_{x=0}^{2^n-1}(-1)^{f(x)}(-1)^{x\cdot y}\Big]\,|y\rangle\,.$$

Hence, the probability of measuring a state y is

$$\Big|\frac{1}{2^n}\sum_{x=0}^{2^n-1}(-1)^{f(x)}(-1)^{x\cdot y}\Big|^2.$$

In particular, when $y = 0$

$$\Big|\frac{1}{2^n}\sum_{x=0}^{2^n-1}(-1)^{f(x)}\Big|^2.$$

We can see that the probability of measuring 0 is 1 if the function is constant and 0 if the function is balanced.

The steps we just described can be summarized in the following scheme.

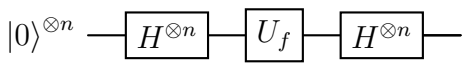


Figure 2.4: Overview of the quantum circuit of the refined Deutsch-Jozsa algorithm.

The original solution involved a different implementation of the oracle function, requiring the use of an ancilla qubit. For the sake of comparing the two, here is the quantum circuit of the classical solution.

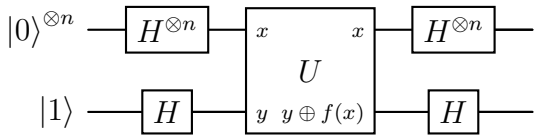


Figure 2.5: Overview of the quantum circuit of the classical Deutsch-Jozsa algorithm.

2.3.2 Grover's algorithm

The quantum search algorithm, devised in 1996 by Lov Grover [10], is one of the most famous quantum algorithms. It consists in an random search which, by repeated iterations, approaches the target element. As for the previous section, let us start by formulating the problem.

Grover's problem. *Let $N \in \mathbb{N}$ be a natural number and $f\colon \mathbb{Z}_N \to \mathbb{Z}_2$ a function such that is zero valued for all the elements in $\mathbb{Z}_N$ except for a special marked element α. Given an oracle U_f for the function f, identify α with high probability.*

Grover's solution finds the marked element in $\mathcal{O}(\sqrt{N})$ evaluations. The algorithm works by applying a sequence of gates, known as Grover's iteration, in order to produce a state with an high probability of $|\alpha\rangle$ being measured. Grover's iteration consists in a 2 step cycle:

1. The oracle gate U_f is applied to the qubits,
2. The probability of measuring the basis state corresponding to α is amplified.

The way the second step works is by applying a gate $U_\psi = 2\,|\psi_1\rangle\langle\psi_1| - I$ called diffusion operator. This gate acts by reflecting all the basis states with respect to the previous state of the system $|\psi_1\rangle$.

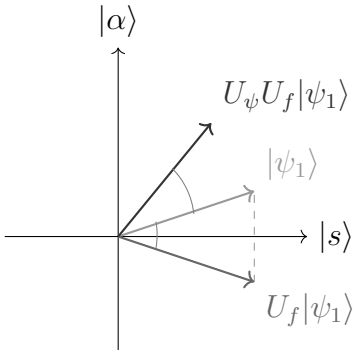


Figure 2.6: Geometrical representation of Grover's iteration.

We start by considering n qubits initialized to the state $|0\rangle$

$$|\psi_0\rangle = |0\rangle^{\otimes n}.$$

As for the Deutsch-Jozsa algorithm, we can create superposition through an Hadamard transform

$$|\psi_1\rangle = H^{\otimes n}|\psi_0\rangle = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle.$$

We can also express $|\psi_1\rangle$ as:

$$|\psi_1\rangle = \sqrt{\frac{N-1}{N}}|s\rangle + \sqrt{\frac{1}{N}}|\alpha\rangle,$$

where $|\alpha\rangle$ is the marked element and $|s\rangle$ is the product state of the remaining elements. In this base we can appreciate the Grover iteration geometrically. Let

$$\frac{\theta}{2} = \arcsin\left(\frac{1}{\sqrt{N}}\right)$$

be the angle between $|\psi_1\rangle$ and $|s\rangle$. Note that $\arcsin(x) \approx x$ for a small argument. It follows that

$$\theta = 2\arcsin\left(\frac{1}{\sqrt{N}}\right) \approx \frac{2}{\sqrt{N}}. \tag{2.4}$$

We now have

$$|\psi_1\rangle = \cos\frac{\theta}{2}|s\rangle - \sin\frac{\theta}{2}|\alpha\rangle.$$

The first step of querying the oracle can be seen as a reflection about $|s\rangle$:

$$U_f|\psi_1\rangle = \cos\frac{\theta}{2}|s\rangle - \sin\frac{\theta}{2}|\alpha\rangle.$$

In this same context, the diffusion operator is a reflection about $|\psi_1\rangle$. To avoid intricate calculations we can observe that this reflection is a counterclockwise rotation of θ, which is described by the matrix

$$\begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix}.$$

This means that the effect of the diffusion operator on $U_f|\psi_1\rangle$ is

$$|\psi_2\rangle = \begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix}\begin{pmatrix}\cos\frac{\theta}{2}\\ -\sin\frac{\theta}{2}\end{pmatrix} = \cos\frac{3\theta}{2}|s\rangle + \sin\frac{3\theta}{2}|\alpha\rangle.$$

If we compare $|\psi_2\rangle$ to $|\psi_1\rangle$ we can see how the probability of measuring $|\alpha\rangle$ increased. In particular, it is not hard to see that after k cycles the state of the system will be

$$|\psi_k\rangle = \cos\left(\frac{2k+1}{2}\theta\right)|s\rangle + \sin\left(\frac{2k+1}{2}\theta\right)|\alpha\rangle.$$

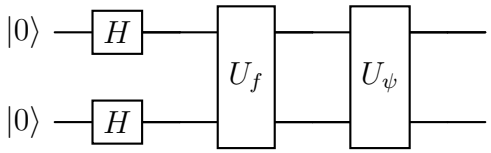


Figure 2.7: Representation of the 2-qubits Grover's algorithm in a general case.

These steps can be easily summarized in the circuit visualization of the algorithm. We aim to find k such that the probability of measuring α is sufficiently high. This means that we want $\frac{2k+1}{2}\theta$ as close to 1 as possible. We can see that

$$\frac{2k+1}{2}\theta = \frac{\pi}{2},$$

implies that the whole part of the k is

$$k = \left\lfloor \frac{\pi}{2\theta} - \frac{1}{2} \right\rfloor.$$

Using the estimate for theta in 2.4, we can rewrite the above as

$$k = \left\lfloor \frac{\pi}{4}\sqrt{N} \right\rfloor,$$

given that $[x - \frac{1}{2}] = \lfloor x \rfloor$ for every $x \in \mathbb{R}$.

2.3.3 Quantum Fourier transform

The quantum Fourier transform (QFT) is a very common subroutine in many quantum algorithms [22]. The fundamental idea of QFT essentially expressing the vector in a different basis, the Fourier basis, in the following way:

$$|x\rangle \to \frac{1}{\sqrt{2^n}}(|0\rangle + e^{2\pi i x/2}|1\rangle) \otimes (|0\rangle + e^{2\pi i x}|1\rangle/2^2) \otimes \cdots \otimes (|0\rangle + e^{2\pi i x/2^n}|1\rangle), \quad (2.5)$$

where $x \in \{|0\rangle, \ldots |N-1\rangle\}$ is a basis state.

Notice how the most significant qubit is always either in the state $|+\rangle$ or $|-\rangle$.

The action of the QFT on the a basis state can be visualized using Bloch spheres. Suppose we have a two qubits system in the state $|01\rangle$, that is:

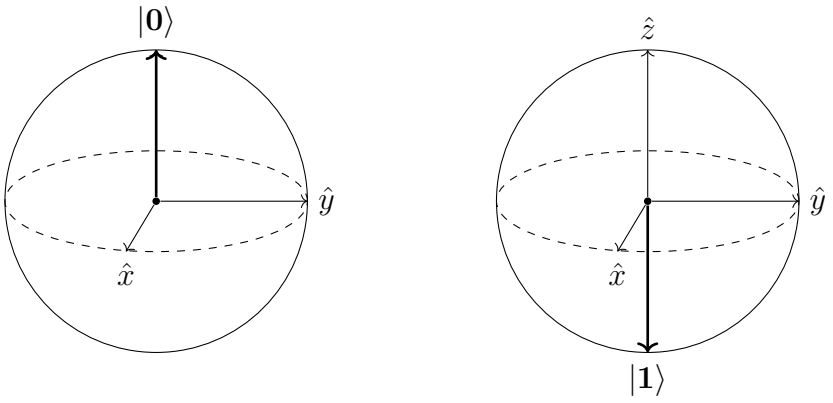


Figure 2.8: Bloch sphere representation of the state $|01\rangle$.

The system, after undergoing a quantum fourier transform, is left in the state:

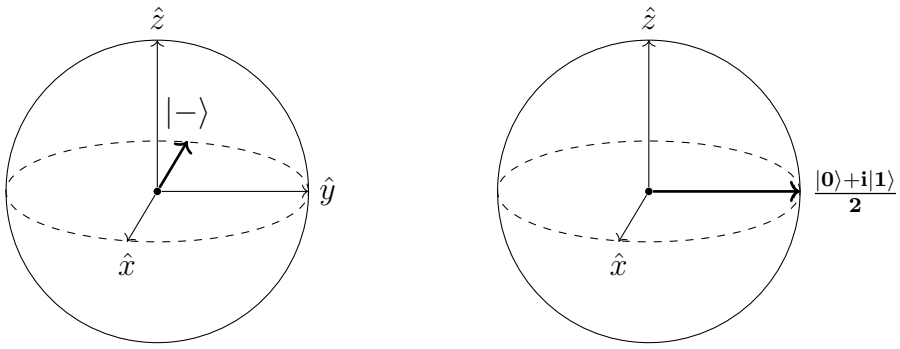


Figure 2.9: Bloch sphere representation of QTF $|01\rangle$.

We will now provide a formal definition of QFT and its inverse.

Definition 2.8. Let $\{|0\rangle, \ldots |N-1\rangle\}$ be the computational basis. For every basis state $|x\rangle$ the **quantum fourier transform** (QFT) is the linear operator

$$\text{QFT}\colon\ |x\rangle \to \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i x k/N} |k\rangle\,.$$

Similarly, we define the **inverse quantum Fourier transform** (IQFT) as

$$\text{IQFT}\colon\ |k\rangle \to \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{-2\pi i x k/N} |x\rangle\,.$$

It is possible to derive the form used in 2.5 from this definition. Now, suppose we aim to transform a n qubit state $|x\rangle = |x_0 \ldots x_{n-1}\rangle$,

$$|x_0 \ldots x_{n-1}\rangle \to \frac{(|0\rangle + e^{2\pi i 0.x_{n-1}} |1\rangle)(|0\rangle + e^{2\pi i 0.x_{n-2}x_{n-1}} |1\rangle) \ldots (|0\rangle + e^{2\pi i 0.x_0 \ldots x_{n-1}} |1\rangle)}{2^{\frac{n}{2}}}$$

The quantum code for the QFT, represented in below in figure 2.10, uses Hadamard gates, SWAP gates and phase shift gates. This phase shift gates used are usually noted R_k, their expression is

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{bmatrix}.$$

When the Hadamard gate is applied to the first bit $|x_0\rangle$ we get the state

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0.x_0} |1\rangle) |x_1 \dots x_{n-1}\rangle .$$

Going through the R_2 gate, the state becomes

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0.x_0 x_1} |1\rangle) |x_1 \dots x_{n-1}\rangle ,$$

then the application of $R_3, \dots, R_n$ produces

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0.x_0 \dots x_{n-1}} |1\rangle) |x_1 \dots x_{n-1}\rangle .$$

Repeating the process on every other bit we obtain the state

$$|\psi^*\rangle = \frac{1}{\sqrt{2^n}}(|0\rangle + e^{2\pi i 0.x_0 \dots x_{n-1}} |1\rangle) \dots (|0\rangle + e^{2\pi i 0.x_{n-2} x_{n-1}} |1\rangle)(|0\rangle + e^{2\pi i 0.x_{n-1}} |1\rangle).$$

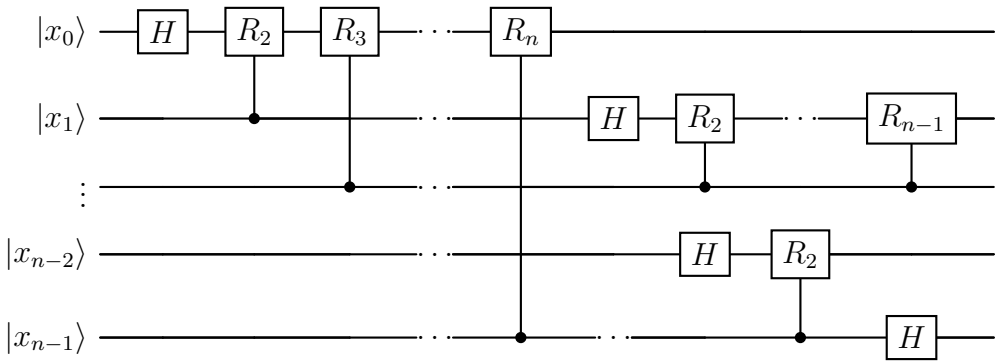


Figure 2.10: Circuit of the QFT up to the state $|\psi^*\rangle$

In the end, a SWAP operation is applied to get the desired output from the algorithm, producing

$$|\psi\rangle = \frac{1}{\sqrt{2^n}}(|0\rangle + e^{2\pi i 0.x_{n-1}} |1\rangle)(|0\rangle + e^{2\pi i 0.x_{n-2} x_{n-1}} |1\rangle) \dots (|0\rangle + e^{2\pi i 0.x_0 \dots x_{n-1}} |1\rangle).$$

Since the QFT is unitary, performing the inverse quantum Fourier transform is equivalent to applying the operations described above backwards.

2.3.4 Quantum phase estimation

The quantum phase estimation (QPE) is a procedure used to estimate the phase of an eigenvector of a unitary operator. It is key to many quantum algorithms.

QPE consists in two main parts, in the first one it uses oracles to encode the eigenvalue in a Fourier basis and then it applies the IQFT in order to produce a state that, once measured, gives the desired result with high probability. Formally speaking, this algorithm is a solution to the following problem.

QPE problem. *Let U be a unitary operator in $H^{\otimes m}$ and $|v\rangle$ one of his eigenvectors. This is*

$$U|v\rangle = e^{2\pi i\lambda}|v\rangle,$$

with $\lambda \in \mathbb{R}$. Find a good value for λ within an approximation ϵ.

The QPE works with two different registers, one is the eigenvector, which contains the necessary qubits to store the value of v. The other register consists of n qubits initialized to $|0\rangle$. The number n will depend on the precision we require and how many digits of the eigenvector we aim to obtain.

The first step of the algorithm is to set the first register in superposition

$$|\psi_1\rangle = \Big(\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle\Big)|v\rangle$$

After the controlled sequence, showed in figure 2.11, the global state is

$$|\psi_2\rangle = \Big(\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}e^{2\pi i\lambda x}|x\rangle\Big)|v\rangle.$$

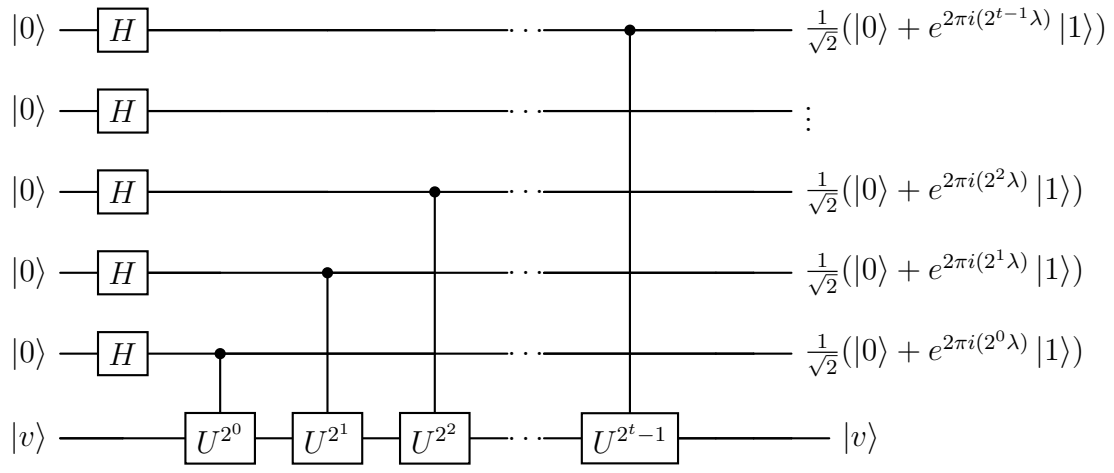


Figure 2.11: Circuit representation of the first part of the QPE

Now, by applying the IQFT we get to the final state of the system, which is

$$|\psi_3\rangle = |\lambda\rangle|v\rangle.$$

2.3.5 HHL algorithm

Given the known effectiveness of linear systems at describing a very wide range of relations across many areas of science, its no wonder that the interests and implications in faster methods to solving them are always big.

In the case of a sparse and positive semi-definite problem, the fastest known classical algorithm to date is the Conjugate Gradient method, which produces a solution in $\mathcal{O}(N)$ time [23].

In 2009, a quantum algorithm was designed by Harrow, Hassidim and Lloyd [11] to approach the same problem. The Harrow, Hassidim and Lloyd algorithm (HHL) consists in a quantum code capable of estimating the result of a scalar measurement on the solution vector of a given linear system. In the case of a sparse system with a low condition number, the HHL has a runtime of $\mathcal{O}(\log(N))$.

Let the following be a linear system of equations

$$A\left|x\right\rangle = \left|b\right\rangle,$$

with $A \in \mathbb{C}^{N\times N}$ assumed to be Hermitian. The fundamental idea behind the HHL algorithm is to evaluate the solution as

$$\left|x\right\rangle = A^{-1}\left|b\right\rangle = \sum_{j=0}^{2^n-1} b_j \lambda_j^{-1}\left|u_j\right\rangle, \tag{2.6}$$

where $\left|u_j\right\rangle$ and λ^{-1} are the eigenvectors and eigenvalues of A and b_j are the coordinates of $\left|b\right\rangle$ in the eigenbasis. One important requirement for the vectors $\left|b\right\rangle$ and $\left|x\right\rangle$ to be represent able on a quantum computer is to be normalized, this is

$$\sum_{j=0}^{2^n-1} \left|b_j\right|^2 = 1, \qquad \sum_{j=0}^{2^n-1} \left|b_j\lambda_j^{-1}\right|^2 = 1. \tag{2.7}$$

The code for the HHL algorithm can be divided in four major steps

1. Preparing the states

2. Performing QPE

3. Rotating the ancilla bit

4. Disentangling the registers

In the following sections we will delve a little bit deeper in each step, explaining the workings of the HHL and highlighting important keypoints.

Preparing the states

Let n be the number of qubits needed to represent $|b\rangle$ and n_c the number of qubits used to represent the phase of the eigenvalues of A. Additionally let $N = 2^n$ and $N_c = 2^{n_c}$. In the first step $n + n_c + 1$ qubits are set to the state $|0\rangle$. These qubits are divided in three different registers:

$$|\psi_0\rangle = |0\dots0\rangle_b |0\dots0\rangle_c |0\rangle_a .$$

The b-register is the one where we will store fist the value of $|b\rangle$ and then the solution. The c-register, also referred to as clock register, is the one on which we will perform the QPE and hence store the eigenvalues of A. As for the last qubit, it will be used as an ancilla bit to carry over information. The first step of the algorithm consists in rotating the b-register, to encode the coefficients of the vector $|b\rangle$. Hence, we have

$$|\psi_1\rangle = |b\rangle_b |0\dots0\rangle_c |0\rangle_a .$$

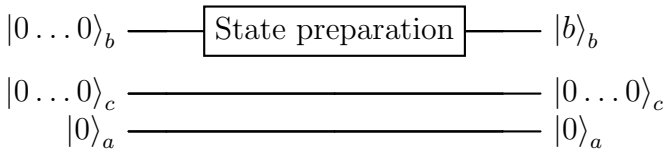


Figure 2.12: Caption

Performing QPE

The aim of this step is to estimate the phase of the eigenvalues of the unitary rotation matrix $U = e^{iAt}$. We first create the superposition on the c-register through an Hadamard transform

$$|\psi_2\rangle = |b\rangle_b (\frac{1}{\sqrt{2^{n_c}}} \sum_{j=0}^{2^{n_c}-1} |j\rangle)_c |0\rangle_a .$$

To simplify the following calculations, we will assume $|b\rangle$ to be an eigenvector of U, at the end of this section we will then return to the general case.

We now apply the C-U gate to the b-register with control on the c-register. As we have seen in the QPE algorithm, this has the effect of applying U^{2^r} to the b-register only in the case of the r-th clock qubit being $|1\rangle$. Since the c-register is currently in a superposition state, we can express this conditionality simply by adding a prefactor to each clock qubit in the following way: suppose that

$$U |b\rangle = e^{2\pi i\varphi} |b\rangle , \tag{2.8}$$

then we have

$$|\psi_3\rangle = |b\rangle_b \left(\frac{1}{\sqrt{2^{n_c}}} (|0\rangle + e^{2\pi i \varphi 2^{n_c-1}} |1\rangle) \otimes \cdots \otimes (|0\rangle + e^{2\pi i \varphi 2^0} |1\rangle) \right)_c |0\rangle_a$$
$$= |b\rangle_b \left(\frac{1}{\sqrt{2^{n_c}}} \sum_{j=0}^{2^{n_c}-1} e^{2\pi i \varphi j} |j\rangle \right)_c |0\rangle_a .$$

We now apply the IQFT to the c-register

$$|\psi_4\rangle = \frac{1}{2^{n_c}} |b\rangle_b \left(\sum_{y=0}^{2^{n_c}-1} \sum_{j=0}^{2^{n_c}-1} e^{2\pi i j(\varphi - y/N_c)} |y\rangle \right)_c |0\rangle_a .$$

We can observe that, since when $(\varphi - y/N_c) = 0$ we have

$$\frac{1}{2^{n_c}} \sum_{j=0}^{2^{n_c}-1} e^{2\pi i j 0} = 1,$$

all the other states where $(\varphi - y/N_c) \neq 0$ must have 0 amplitude. By ignoring these states we can rewrite $|\psi_4\rangle$ as

$$|\psi_4\rangle = |b\rangle_b |N_c \varphi\rangle_c |0\rangle_a . \tag{2.9}$$

Given that $|b\rangle$ is an eigenvector of U, we also have that $U|b\rangle = e^{i\lambda_j t} |u_j\rangle$ for some j. Equating this to 2.8 we get

$$\varphi = \frac{\lambda_j t}{2\pi} \text{ for some } j.$$

Substituting in 2.9 we get

$$|\psi_4\rangle = |u_j\rangle_b |N_c \lambda_j t/2\pi\rangle_c |0\rangle_a .$$

To go back to the general case of $|b\rangle$ not being an eigenvector, it is sufficient to change the b-register to the expression of $|b\rangle$ in the eigenbasis. Additionally, we also choose t so that the expression $N_c \lambda_j t/2\pi$ takes integer value. In the end, we have

$$|\psi_4\rangle = \sum_{j=0}^{2^n-1} b_j |u_j\rangle_b |\rho_j\rangle_c |0\rangle_a ,$$

where $\rho_j = N_c \lambda_j t/2\pi$ is now an integer.

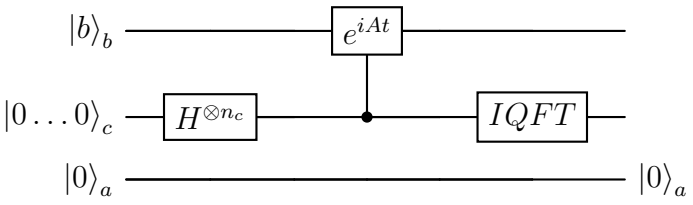


Figure 2.13: Caption

Rotating the ancilla bit

We rotate the ancilla bit to the state

$$|\psi_5\rangle = \sum_{j=0}^{2^n-1} b_j \, |u_j\rangle_b \, |\rho_j\rangle_c \left(\sqrt{1 - \frac{C^2}{\rho_j^2}} \, |0\rangle + \frac{C}{\rho_j} \, |1\rangle \right)_a ,$$

where $C \in \mathbb{R}$ is user defined constant. Now we measure the ancilla qubit, if the result is $|0\rangle_a$ the computation rolls back to the first step. We can see how the value of C affects the probability of measuring $|1\rangle_a$, for this reason we set C to be as big as possible. Once we successfully measure $|1\rangle_a$, we are left with the state

$$\begin{aligned} |\psi_6\rangle &= \frac{1}{\sqrt{\sum_{k=0}^{2^n-1} |\frac{b_j C}{\rho_j}|^2}} \sum_{j=0}^{2^n-1} b_j \, |u_j\rangle_b \, |\rho_j\rangle_c \, \frac{C}{\rho_j} \, |1\rangle_a \\ &= \frac{C}{\sqrt{\sum_{k=0}^{2^n-1} |\frac{b_j C}{\rho_j}|^2}} \sum_{j=0}^{2^n-1} b_j \, |u_j\rangle_b \, |\rho_j\rangle_c \, \frac{1}{\rho_j} \, |1\rangle_a , \end{aligned}$$

which by simplifying C, recalling 2.7 and that $\rho_j = N_c \lambda_j t / 2\pi$, leaves

$$\begin{aligned} |\psi_6\rangle &= \frac{N_c t}{2\pi} \sum_{j=0}^{2^n-1} b_j \, |u_j\rangle_b \, |\rho_j\rangle_c \, \frac{1}{\rho_j} \, |1\rangle_a \\ &= \sum_{j=0}^{2^n-1} b_j \lambda_j^{-1} \, |u_j\rangle_b \, |\rho_j\rangle_c \, |1\rangle_a \end{aligned}$$

Note that $\sum_{j=0}^{2^n-1} b_j \lambda_j^{-1} \, |u_j\rangle$ is our solution but we cannot substitute already since the b-register is entangled with the c-register.

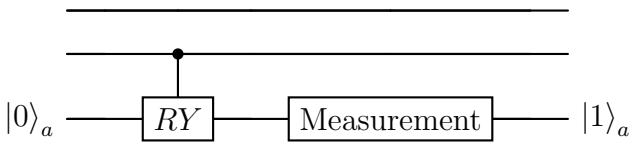


Figure 2.14: Caption

Disentangling the registers

First, we apply the QFT to the c-register

$$|\psi_7\rangle = \sum_{j=0}^{2^n-1} b_j \lambda_j^{-1} \, |u_j\rangle_b \left(\frac{1}{\sqrt{2^{n_c}}} \sum_{y=0}^{2^{n_c}-1} e^{2\pi i y \rho_j / N_c} \, |y\rangle \right)_c |1\rangle_a .$$

Then a controlled inverse rotation $C\text{-}U^{-1}$ is applied to the b-register. With a similar argument to before, it is not hard to see that this is equivalent to multiplying the c-register by a factor of $e^{-i\lambda_j ty}$:

$$|\psi_8\rangle = \sum_{j=0}^{2^n-1} b_j \lambda_j^{-1} |u_j\rangle_b \left(\frac{1}{\sqrt{2^{n_c}}} \sum_{y=0}^{2^{n_c}-1} e^{-i\lambda_j ty} e^{2\pi i y\rho_j/N_c} |y\rangle \right)_c |1\rangle_a .$$

Since $2\pi i y\rho_j/N_c - i\lambda_j ty = 0$, the exponentials cancel out, leaving

$$|\psi_8\rangle = \sum_{j=0}^{2^n-1} b_j \lambda_j^{-1} |u_j\rangle_b \left(\frac{1}{\sqrt{2^{n_c}}} \sum_{y=0}^{2^{n_c}-1} |y\rangle \right)_c |1\rangle_a$$

We can now apply an Hadamard transform to the c-register to get

$$|\psi_9\rangle = \sum_{j=0}^{2^n-1} b_j \lambda_j^{-1} |u_j\rangle_b |0\rangle_c |1\rangle_a .$$

The final state of the system, substituting the first term for 2.6, is

$$|\psi_9\rangle = |x\rangle_b |0\rangle_c |1\rangle_a .$$

We have successfully encode the solution in the b-register.

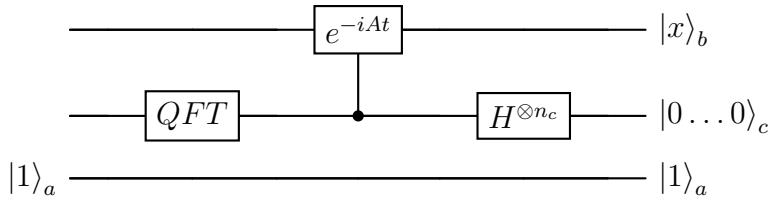


Figure 2.15: Caption

Summary

Once this steps are completed, we are left with a state which has the solution $|x\rangle$ encoded in the basis. It is important to note that we cannot extract the whole solution vector, but only a measurement over it. Nonetheless, the HHL algorithm has seen great use both on his own and as a subroutine to more complex algorithms, in the case of studies where the interest are particular features of the solution, rather than the solution itself. The algorithm presents interesting perspective of application in fields such as quantum chemistry [2] and machine learning [17].

Chapter 3

Quantum computers

Intuitively, a quantum computer is a calculator which takes advantage of quantum phenomena and achieves results using the logic of quantum computation. From a mathematical point of view a quantum computer performing an algorithm is an operator acting on the system state vector. In this chapter we will focus on the mechanical perspective of constructing a machine capable of initializing, manipulating and using quantum states to fulfill a task.

3.1 Physical realization

Building a quantum computer is a complex matter, nonetheless we will present the basic ideas. Theoretical physicist David DiVincenzo [8] proposed a list of necessary requirements for building a quantum computer.

These can be summarized in the following properties:

Scalability Clearly a quantum computer must have a collection of elements capable of embodying the qubits. As we have previously seen, a qubit is a 2-level system. A physical example of this could be a photon with its polarization states.

Qubits should be well characterized, that is, their physical parameters are known, most importantly the Hamiltonian, given that the energy eigenstates are usually taken to be $|0\rangle$ and $|1\rangle$.

Another fundamental requirement to make good qubits is for them to have long quantum coherence times. These indicate how long a system maintains coherent with the laws of quantum mechanics before decaying to a classical state. Coherence times essentially indicate the stability of the qubit, longer times allow for better manipulation.

It is also important that the proposed computer can be scaled, to an arbitrary number of qubits, without an exponential cost of resources.

Correctability The machine must be capable of initializing the qubits to a simple state, such as $|000\ldots\rangle$ ideally. This requirement is needed to know the values of the registers prior to the start of computation. The initialization process also

has to be relatively fast, given the need for a large number of low entropy states, e.g. $|000\ldots\rangle$, in the error corrections procedures.

Universal logic gates A quantum computer needs to be equipped with a universal set of quantum gates, this is, a set capable of reproducing the effect of any possible transformation through a combination of its elements. While there exists single universals quantum gates, such as the BARENCO gate [1], their implementation is rather complex. For this reason a much more manageable set of gates is usually implemented, consisting of some basic single qubit gates (e.g. the Hadarmard gate H and the $\frac{\pi}{4}$ phase shift gate T), and a non-trivial two qubit gate, such as the CNOT. We call a two qubit gate "non-trivial" if at least one of the outcomes depends on the initial states of both qubits. This is important in order to produce entanglement.

Measurement capability In quantum computers measurement can be a very complicated topic and can be performed in vary different ways. Nonetheless it is indeed a fundamental component, without which the calculations performed would be useless. Let

$$\rho = \begin{pmatrix} p & \alpha \\ \alpha^* & 1-p \end{pmatrix}$$

be a qubit's density matrix. We say that a measurement is ideal, or that has 100% quantum efficiency, when reads out "0" with probability p and "1" with probability $1-p$, without dependence on α or any other parameter of the system. Even though real measurements never achieve 100% efficiency, many techniques have been developed as a solution to this problem. In some cases, simply by remeasuring the qubit one can obtain with a higher precision. This effectively enables us to work with measurements regardless of their efficiency, assuming that their repeated application does not disturb the quantum computer.

Along with these criteria, which regard the computation, there are two more features of a quantum computer that DiVincenzo highlights. These regard are about the transmission of information in a computer and do not impact the computational requirements.

3.2 Challenges

The central challenge in actually building quantum computers is maintaining the simultaneous abilities to control quantum systems, to measure them, and to preserve their strong isolation from uncontrolled parts of their environment. DiVincenzo's criteria can in fact appear to be almost self-contradictory. This is because, in order to achieve rapid measurement, the same specific hardware must be turned strongly 'on' for error correction and readout, but must be turned strongly "off" to preserve the coherence. Generally, neither the "on" state nor the "off" state is as difficult to implement as the ability to switch between the two. This problem does not suggest that the criteria are wrong or incomplete, it instead raises doubts on the possibili-

ties of quantum computers to be scalable. When engineering a quantum computer, these conflicts are often aided by techniques for quantum communication. This allows small quantum computers to be connected to make a larger one, moving the specialized measurement hardware far from sensitive quantum memories.

3.3 Proposed models

In this section, we introduce the technologies researchers are currently using to meet DiVincenzo's criteria[15].

Superconductors

The most famous proposal for quantum computing is probably that of a superconducting quantum computer (SQC).

Resistive power loss in traditional electric circuits causes a quick decoherence. This issue can be fixed with superconductors, where electrons bind into a particular structure, known as Cooper pair.

Cooper pairing is a quantum effect taking place at low temperature. It is due to an increase of the positive charge density in proximity of an electron, which then causes it to pair up with another electron. Once formed, Cooper pairs then condense into a state with no resistance current and a well defined phase. The potential of these condensed pairs can be changed by modifying macroscopic quantities such as inductances and capacitances of the circuit. There are three types of superconducting qubits, charge, flux and phase qubit, all derived from condensed Cooper pairs at different potentials.

In the last years most of the research has focused on this technology, given its promising perspectives. To this day, many industries have invested in SQC, with IBM being recognized as the leader in the field.

Photons

Photonic quantum computing (PQC) uses photons as qubits and linear optics elements (e.g. mirrors) to process them.

Photons appear to be fitting candidates for quantum computing. Their polarization states realise good qubits, given that they are relatively free of decoherence. Photon interference can be observed and produced easily, thus sparking the idea for this proposal. It was, however, believed necessary to introduce non-linear elements in the framework in order to perform efficient quantum computation. In 2001 Knill, Laflamme and Milburn proved that a universal quantum computer can be constructed using only single-photons sources, detectors and a linear optical system. This particular model is known as KLM scheme[14].

Xanadu is one of the most important companies working on PQC, partnering with Google, Nvidia and IBM.

Ion traps

Ions are obtained when atoms stand at a particular energy level. A ion trap suspends these particles using electromagnetic fields. Given a trapped atom, its energy levels can form very reliable qubits. Qubits of this sort can be measured with very high efficiency through a state-dependent optical fluorescence detector. Single qubit rotations can be implemented using magnetic dipole transitions. As for the CNOT gate, its first implementation was proposed by Ignacio Cirac and Peter Zoller in 1995[5].

Trapped-ion quantum computing represents a possibility for a large scale quantum computer. IonQ is currently the leading company researching and developing this technology.

While many other models have been proposed to this day, they all either align with the ones above or come with too many flaws to make for a good candidate without any major breakthroughs, currently. For the peculiarities it presents and also because it is the computer we will test in this work, one last particular model is worth our interest, the NMR quantum computer.

3.4 Nuclear Magnetic Resonance

Atomic nuclei lying in a strong magnetic field can either absorb or emit electromagnetic signals with resonant frequencies. This behaviour is today known as nuclear magnetic resonance (NMR) and was first observed in 1946 by Bloch and Purcell. NMR technology has since then been largely developed, even prior to its quantum computing involvement, due to its utility in analyzing chemical compounds. It is only in the 1998 that Chuang [4], Laflamme [18] and other scientists started proposing the manipulation of nuclear spins through NMR as a mean for quantum information processing, laying the basic ideas for a nuclear magnetic resonance quantum computer (NMRQC).

Over the following years, interest and research in NMR quantum computing flourished. In 2001, Shor's algorithm was implemented on a 7-qubit NMR computer [21], leading to realizations of different algorithms, such as Deutsh-Jozsa or HHL, in the following years.

Even though these accomplishments, NMR computing does not come without its drawbacks. First of all, it is well established that the most popular scheme, the liquid state bulk ensemble NMR, has poor scaling capabilities. Secondly, and crucially, it has been shown that experiments with a small number of qubits do not produce proper quantum entanglement [3]. This probably due to the use of pseudo-pure states to account for the difficulty of reaching an effective $|0\rangle^{\otimes n}$ state in an ensemble computer.

While NMR quantum computing contributed significantly to the early development of quantum information science, the limitations in scalability and in the

ability to maintain entanglement made it less viable for the development of large-scale, practical quantum computers. This caused this technology to be almost abandoned in the early 2000's. Nowadays, the NMRQC maintains its role as a research tool and a test ground for quantum computation.

In order to better understand why this is the case, let us first focus on what precisely is an NMR computer and how does it work.

Principles of NMR

We will now give a quick summary on the fundamentals of NMR spectroscopy, to highlight its key concepts, a more complete explain of this phenomenon can be found in [16], which we used for this description. Essentially, NMR analysis aims to exploit magnetic properties of atomic nuclei to investigate the molecular structure of samples.

The first of which is spin, a fundamental property in quantum mechanics that all subatomic particles posses. While there are formal, mathematical definitions of spin, this is not the right context to indulge on such formulations, for the sake of keeping things concise. We will for now stand for spin being a degree of freedom of subatomic particles, representing a sort of intrinsic angular momentum. Spin in nuclei generates a tiny magnetic field associated with the atom's nucleus. Once the atom is placed in an external magnetic field, its nucleus precesses around the direction of the field. The frequency of precession is characteristic to each nucleus and is called Larmor frequency. Larmor frequency depends on the strength of the outer field and the internal properties of the nucleus. When in this precession state, radio-frequency (RF) pulses are applied to the sample, causing the nuclei to transition between spin states. This causes with an absorption of energy at certain resonance. Once the pulse is turned off, the nuclei relax to their original positions emitting the absorbed energy as electromagnetic signals. Nuclear spins give raise to a particular phenomenon, called J coupling, in which nuclear magnetic fields interact with each other, causing the NMR signal to split into multiplets. The magnitude and pattern of this splittings along the sample depends on a particular constant J, called coupling constant, related to the chemical environment around the interacting nuclei.

Thus, by analyzing the resulting spectra of resonant frequencies in a given sample, NMR spectroscopy can give valuable information about its chemical composition and atomic interactions. This is widely used in nuclear magnetic resonance spectroscopy, to study the structure of organic molecules in solution, molecular physics, crystals, non-crystalline materials, and in advanced medical imaging techniques, such as in magnetic resonance imaging.

Qubits

As we have already seen, atoms in magnetic fields present interesting properties, the one we are interested in is related to the spin of their nucleus. A spin-1/2 system is a two level quantum system, with basis states called spin up and spin down, which can be identified with $|0\rangle$ and $|1\rangle$ respectively. Only particular atoms nuclei realise spin-1/2 systems when subjected to a strong magnetic field. These include ^{1}H, ^{13}C, ^{15}N, ^{19}F and ^{31}P, which all make for good qubits.

In Table 3.1 we can appreciate a partial list of the molecules used in NMR quantum computing.

Table 3.1: List of molecules commonly used as samples in NMR computing

Molecule	**Chemical Formula**	**Nuclei**	**Number of Qubits**
Chloroform	$CHCl_3$	^{1}H,^{13}C	2
Methanol	CH_3OH	^{1}H,^{13}C	2
Iodotrifluoroethene	C_2F_3I	^{19}F	3
Alanine	$C_3H_7N0_2$	^{1}H,^{13}C	4
Glycine fluoride	C_2H_4ClNO	^{1}H,^{15}N,^{19}F	5
Inosine	$C_{10}H_{12}N_4O_5$	^{1}H	6
Crotonic acid	$C_4H_60_2$	^{1}H,^{13}C	7

We can classify the samples in two groups. If the nuclei considered to form the qubits are all of the same species of atom, we say the sample is homonuclear. When instead the qubits are realised with different kinds of atoms we call the sample heteronuclear. While having chemically distinct nuclei is certainly an advantage when it comes to distinguish among them, it is also true that a fully heteronuclear NMR computer cannot be scalable. This is due to the fact that there is only a small number of spin-1/2 nuclei, hence limiting the number of qubits of an heteronuclear computer.

One of the biggest differences from other proposed models is that a NMRQC does not consider a single isolated system but rather an ensemble of them. This is due to the signal emitted from a single molecule being too weak to be detected. Hence each qubit is actually represented by a large number of nuclei, not distinguished by their spacial position but rather by the localization of the sample.

Initialization

The ability to transform one state to another using gates would be rather useless if the starting state is unknown. For this reason, a quantum computer has to be capable of initializing the qubits to a well defined state.

The most simple choice appears to be setting the qubits to the $|0\rangle$. This operation is called CLEAR, its implementation on an NMR computer turns out to be

a difficult task. This is due to the distribution of all states being almost equal at room temperature, hence trying to relax the system at its energetic ground state would not produce the effect of CLEAR. Nonetheless, given that NMR computer are based on ensembles, an alternative solution to the initialization problem exists.

The key idea is in fact to produce a particular state, which is called pseudo-pure state (PPS), whose behaviour indistinguishable from a pure ground state.

Gates

In a liquid state NMRQC gates are usually implemented using pulses to manipulate the value of qubits. Essentially, single qubit gates can be realized through RF pulses, while the J-coupling provides a natural spin-spin interaction to recreate the correlation effect of controlled two qubit gates.

One qubit gates correspond to rotations. In other quantum computing proposals these rotations can be applied selectively to each individual qubit, using the position of the physical objects implementing them. This is of course not feasible in NMR computing, given the presence of an ensemble of molecules, all having different spatial localization. Using resonant frequencies it is however possible to only target the nuclei located in a specific position in the molecule, hence affecting the desired qubit.

The realisation of two qubit gates exploits the coupling network existing in the molecules that make up the ensemble. It is important to note that J-coupling only occurs within nuclei of the same molecule, without affecting the others in the sample.

Readout

The measurement procedure in NMR starts by detecting the signal emitted by the molecules. This is achieved by applying an RF pulse, called readout pulse, of similar frequency to the Larmor frequency of the nuclei, causing them to oscillate. When precessing about the $\hat{z}$-axis, nuclei also induce a voltage in the RF coil. This signal is known as free induction decay (FID). The FID due to the NMR sample is in the form

$$S(t)e^{-it/T_2^*},$$

where $S(t)$ is the expression of the voltage and T_2^* is the decoherence time of the signal. When measured, the FID is a time-dependent sinusoidal signal, which does not give much information about the system. Luckily, using a Fourier transform we can extrapolate a frequency spectrum. Given that an NMRQC is an ensemble computer, the spectrum represents an average of the qubits' measurement in each molecule. This allows us to read it to determine the final state of the system in basis states.

Chapter 4

Experiments with NMR

In this chapter we will present the 3-qubit quantum computer at our disposal, then implement and test the quantum algorithms we have seen, as well as a Bell inequality verification.

4.1 SpinQ Triangulum

SpinQ Triangulum is a 3-qubit homonuclear liquid state NMR quantum computer from SpinQ Technologies `https://www.spinquanta.com/products-solutions/Triangulum`. It comes with a software interface on Windows called SPINQUASAR, though which the user can operate the machine (see sucint description in Fig. 4.1).

Figure 4.1: Quantum computing interface on the SPINQUASAR software: Starting from the top right we have the task manager and at its left the quantum code editor. On the bottom left we have a visualization of the molecule in the sample, with the fluorine nuclei in green, an finally in the bottom right the readout interface, where the results of the measurements are displayed.

The software is equipped with a library of pre-made algorithms, composed of

the classical examples, and the possibility to select the number of qubits we want to use.

Specifics

The computer is equipped with a Quantum Processing Unit (QPU) containing a sample of iodofluoroethylene (C_2F_3I) molecules (see Fig 4.2). The molecules are placed in the center of parallel permanent NdFeB magnets to activate the magnetic properties of the three ^{19}F spin 1/2 nuclei. The Larmor frequencies of the three fluorine nuclei are slightly different from each other, all standing in the range of 40MHz at 1 T of field intensity. As for the decoherence times, T_1 and T_2 are approximately 7s and 0,2s each.

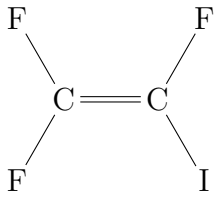


Figure 4.2: Molecular representation of C_2F_3I

Having homonuclear nuclei with similar Larmor frequencies, simple RF pulses are not enough to correctly manipulate the nuclei. For this reason SpinQ Triangulum utilizes composite pulses, in particular GRAPE pulses (see [13] for details), which guarantee an accurate control over the qubits. The motherboard is in fact equipped with a Field Programmable Gate Array (FPGA) capable of producing arbitrary pulses, with an accuracy of 1/65536 and $2\pi/65536$ for the magnitude and phase respectively.

The spin Hamiltonian for the system is:

$$\mathcal{H}_0 = 2\pi(\nu_1\sigma_z^1 + \nu_2\sigma_z^2 + \nu_3\sigma_z^3) + \sum_{k=x,y,z} 2\pi(J_{12}\sigma_k^1\sigma_k^2 + J_{13}\sigma_k^1\sigma_k^3 + J_{23}\sigma_k^2\sigma_k^3),$$

where ν_i is the chemical shift on the i-th nucleus, σ^i are the Pauli matrices on the $i-th$ nucleus and J_{ij} is the coupling coefficient between the i-th and j-th nuclei (we remind there are three ^{19}F spin 1/2 nuclei in each molecule).

Pseudo-pure state

As we have discussed in the previous chapter, NMR computer cannot produce an initial pure state by letting the system cool down. For this reason the 3 qubits are prepared to be in a PPS of $|000\rangle$, is to say in a state of the form:

$$\psi_{pps} = \frac{1-\eta}{2^3} I^{\otimes 8} + \eta\,|000\rangle\langle 000|\,.$$

This is achieved using a permutation matrix that is applied to the state until the system is in a desired state. In practice, the following operator

$$U_{permute} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

is applied until the system reaches a condition where all the basis states are equally present with the exception of the state $|000\rangle$ occurring slightly more often. That will be our initial state.

The gate above, like the vast majority of the gates on Triangulum is realized through GRAPE pulses.

Measurement

SpinQ Triangulum measures the diagonal elements of the system's density matrix. This means that only the probabilities of the 8 standard basis vector for 3 qubits are measured. The NMR spectrometer collects the expected value $\langle \sigma_x + i\sigma_y \rangle$ of a qubit, where σ_x and σ_y are the Pauli matrices. Then the peaks in the resulting spectrum are measured, and through the solution of a linear system we can obtain the values $\langle \sigma_z^1 \rangle$, $\langle \sigma_z^2 \rangle$, $\langle \sigma_z^3 \rangle$, $\langle \sigma_z^1 \sigma_z^2 \rangle$, $\langle \sigma_z^1 \sigma_z^3 \rangle$, $\langle \sigma_z^2 \sigma_z^3 \rangle$, and $\langle \sigma_z^1 \sigma_z^2 \sigma_z^3 \rangle$. We can write the each ρ_{ii} as a linear combination of these expectations in the following way:

$$\begin{aligned}
\rho_{11} &= \frac{1}{8}(1 + \langle \sigma_z^1 \rangle + \langle \sigma_z^2 \rangle + \langle \sigma_z^3 \rangle + \langle \sigma_z^1 \sigma_z^2 \rangle + \langle \sigma_z^1 \sigma_z^3 \rangle + \langle \sigma_z^2 \sigma_z^3 \rangle + \langle \sigma_z^1 \sigma_z^2 \sigma_z^3 \rangle), \\
\rho_{22} &= \frac{1}{8}(1 + \langle \sigma_z^1 \rangle + \langle \sigma_z^2 \rangle - \langle \sigma_z^3 \rangle + \langle \sigma_z^1 \sigma_z^2 \rangle - \langle \sigma_z^1 \sigma_z^3 \rangle - \langle \sigma_z^2 \sigma_z^3 \rangle - \langle \sigma_z^1 \sigma_z^2 \sigma_z^3 \rangle), \\
\rho_{33} &= \frac{1}{8}(1 + \langle \sigma_z^1 \rangle - \langle \sigma_z^2 \rangle + \langle \sigma_z^3 \rangle - \langle \sigma_z^1 \sigma_z^2 \rangle + \langle \sigma_z^1 \sigma_z^3 \rangle - \langle \sigma_z^2 \sigma_z^3 \rangle - \langle \sigma_z^1 \sigma_z^2 \sigma_z^3 \rangle), \\
\rho_{44} &= \frac{1}{8}(1 + \langle \sigma_z^1 \rangle - \langle \sigma_z^2 \rangle - \langle \sigma_z^3 \rangle - \langle \sigma_z^1 \sigma_z^2 \rangle - \langle \sigma_z^1 \sigma_z^3 \rangle + \langle \sigma_z^2 \sigma_z^3 \rangle + \langle \sigma_z^1 \sigma_z^2 \sigma_z^3 \rangle), \\
\rho_{55} &= \frac{1}{8}(1 - \langle \sigma_z^1 \rangle + \langle \sigma_z^2 \rangle + \langle \sigma_z^3 \rangle - \langle \sigma_z^1 \sigma_z^2 \rangle - \langle \sigma_z^1 \sigma_z^3 \rangle + \langle \sigma_z^2 \sigma_z^3 \rangle - \langle \sigma_z^1 \sigma_z^2 \sigma_z^3 \rangle), \\
\rho_{66} &= \frac{1}{8}(1 - \langle \sigma_z^1 \rangle + \langle \sigma_z^2 \rangle - \langle \sigma_z^3 \rangle - \langle \sigma_z^1 \sigma_z^2 \rangle + \langle \sigma_z^1 \sigma_z^3 \rangle - \langle \sigma_z^2 \sigma_z^3 \rangle + \langle \sigma_z^1 \sigma_z^2 \sigma_z^3 \rangle), \\
\rho_{77} &= \frac{1}{8}(1 - \langle \sigma_z^1 \rangle - \langle \sigma_z^2 \rangle + \langle \sigma_z^3 \rangle + \langle \sigma_z^1 \sigma_z^2 \rangle - \langle \sigma_z^1 \sigma_z^3 \rangle - \langle \sigma_z^2 \sigma_z^3 \rangle + \langle \sigma_z^1 \sigma_z^2 \sigma_z^3 \rangle), \\
\rho_{88} &= \frac{1}{8}(1 - \langle \sigma_z^1 \rangle - \langle \sigma_z^2 \rangle - \langle \sigma_z^3 \rangle + \langle \sigma_z^1 \sigma_z^2 \rangle + \langle \sigma_z^1 \sigma_z^3 \rangle + \langle \sigma_z^2 \sigma_z^3 \rangle - \langle \sigma_z^1 \sigma_z^2 \sigma_z^3 \rangle).
\end{aligned}$$

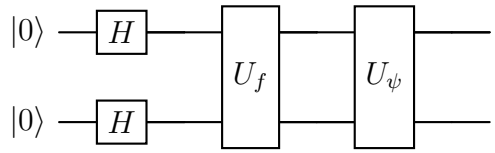


Figure 4.3: Representation of the 2-qubits Grover's algorithm in a general case.

By doing so, we have reconstructed the diagonal of the density matrix and obtained the probabilities associated with each basis state.

4.2 Grover's algorithm in NMR

In this section we will build and run Grover's algorithm on the NMR computer.

Several test will be conducted, using both the 2-qubits and 3-qubits version of the code and choosing various basis elements as the marked one, in order to highlight any possible discrepancies in the performance.

In the following we will present and discuss the quantum code for the 2-qubits version since the same reasoning can be easily applied to the 3-qubits case, as we shall see. As we have previously described in Section 2.3.2, the generic circuit for the 2-qubits Grover's algorithm is represented in Fig. 4.3.

Using 2.3.2 we can see that the number of iterations required for a 2-qubit algorithm is only 1,

$$k = \left\lfloor \frac{\pi}{4}\sqrt{4} \right\rfloor = 1$$

Let us start with the case of the market element being $|11\rangle$. We need to build a quantum oracle such that

$$U_f\Big(\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)\Big) = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle - |11\rangle).$$

It is easily seen that this corresponds to the following operator matrix

$$U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & Z \end{bmatrix},$$

which is exactly the controlled-Z gate.

Hence the quantum oracle for the state $|11\rangle$ is the controlled-Z gate. We now have to find the expression for the diffusion operator. Using the fact that

$$U_\psi = 2\,|\psi\rangle\langle\psi| - I,$$

where ψ is the superposition state obtained through Hadamard gates, we can write

$$U_\psi = \frac{1}{2}\begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix}.$$

This matrix can be obtained as result of the following product

$$U_\psi = H^{\otimes 2} \cdot Z^{\otimes 2} \cdot CZ \cdot H^{\otimes 2},$$

which is represented in figure 4.4.

Remark 4. While the oracle changes depending on the marked element, the diffusion operator remains the same. Observe that

$$\frac{1}{2}\begin{pmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{pmatrix}\begin{pmatrix} 1/2 \\ 1/2 \\ -1/2 \\ 1/2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}.$$

Similarly we have

$$\frac{1}{2}\begin{pmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{pmatrix}\begin{pmatrix} -1/2 \\ 1/2 \\ 1/2 \\ 1/2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

and the same for other basis states. In fact we have that

$$U_\psi\Big(\frac{1}{2}\sum_{x=0}^{3}(-1)^f |x\rangle\Big) = -|\alpha\rangle,$$

with $|\alpha\rangle$ marked element and f oracle function. △

We now have all the necessary to build the 2-qubits algorithm for the search of the state $|11\rangle$.

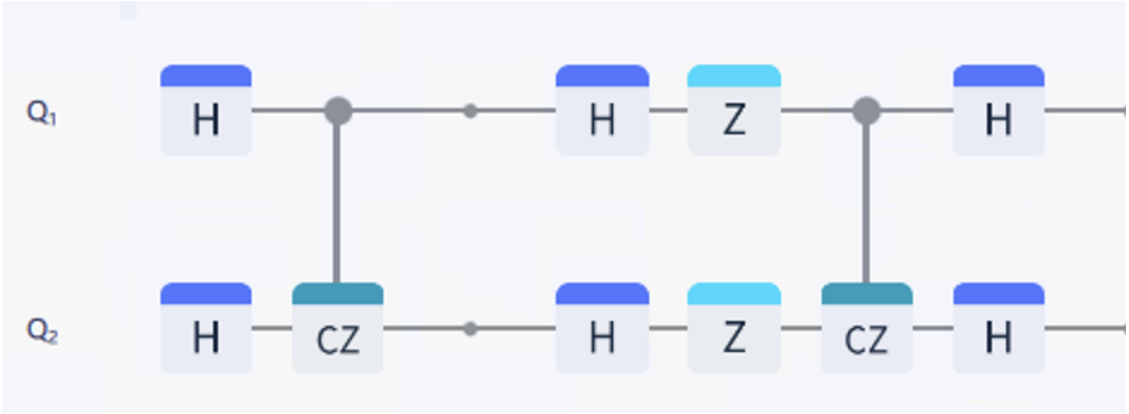


Figure 4.4: Circuit representation of the 2-qubits Grover's algorithm for the state $|11\rangle$

If we were to mark another element instead, it is sufficient to adjust the oracle function using NOT gates to get the desired output. As an example, suppose we are looking for the element $|10\rangle$, then our oracle should produce the output

$$\psi_2 = \frac{1}{2}(|00\rangle + |01\rangle - |10\rangle + |11\rangle).$$

Notice how, if we apply a NOT to the second qubit, we have

$$(I \otimes X)\psi_2 = \frac{1}{2}(|01\rangle + |00\rangle - |11\rangle + |10\rangle),$$

which is the output of the controlled-Z gate. This suggests that after the application of the CZ we can use NOT gates to obtain the state 4.2. Using this method we can obtain the circuit for the other cases as they are shown in the figures 4.5 and 4.6.

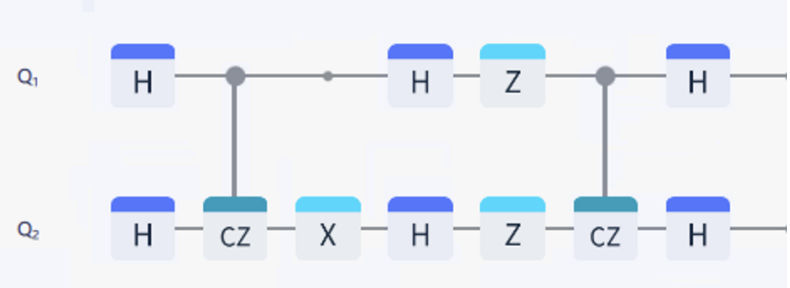


Figure 4.5: Circuit representation of the 2-qubits Grover's algorithm for the state $|10\rangle$

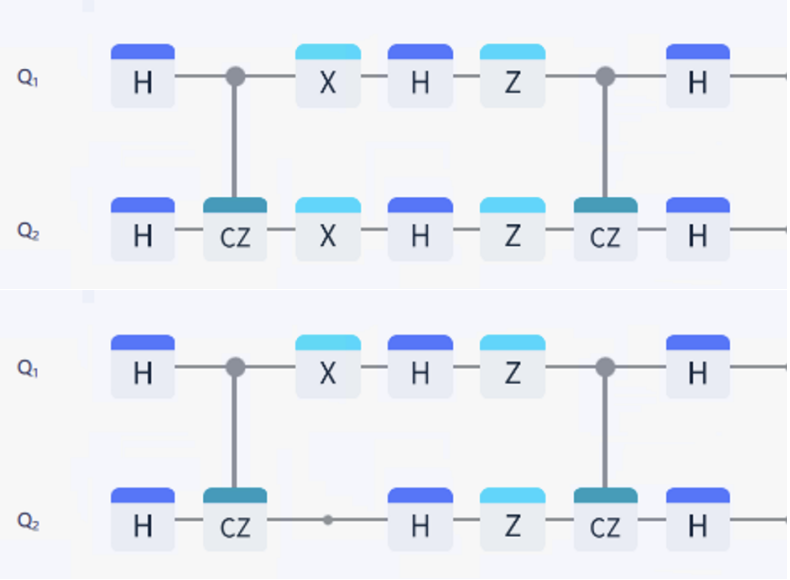


Figure 4.6: Circuit representations of the 2-qubits Grover's algorithm for the states $|00\rangle$, above, and $|01\rangle$, below

As for the 3-qubits case, the ideas remain the same. The only changes are the use of the CCZ, instead of the controlled-Z and the form of the diffusion operator, which for 3-qubits is expressed as

$$U_\psi = H^{\otimes 3} \cdot X^{\otimes 3} \cdot CCZ \cdot X^{\otimes 3} \cdot H^{\otimes 3}.$$

Another fundamental difference is that this time the number of iterations k is

$$k = \left\lfloor \frac{\pi}{4}\sqrt{8} \right\rfloor = 2$$

meaning that we need to add another iteration of the Grover's step. Given these changes, we propose two circuits, one for the search of the state $|111\rangle$ and the other for $|100\rangle$ to better understand the scale-up – see Fig. 4.7.

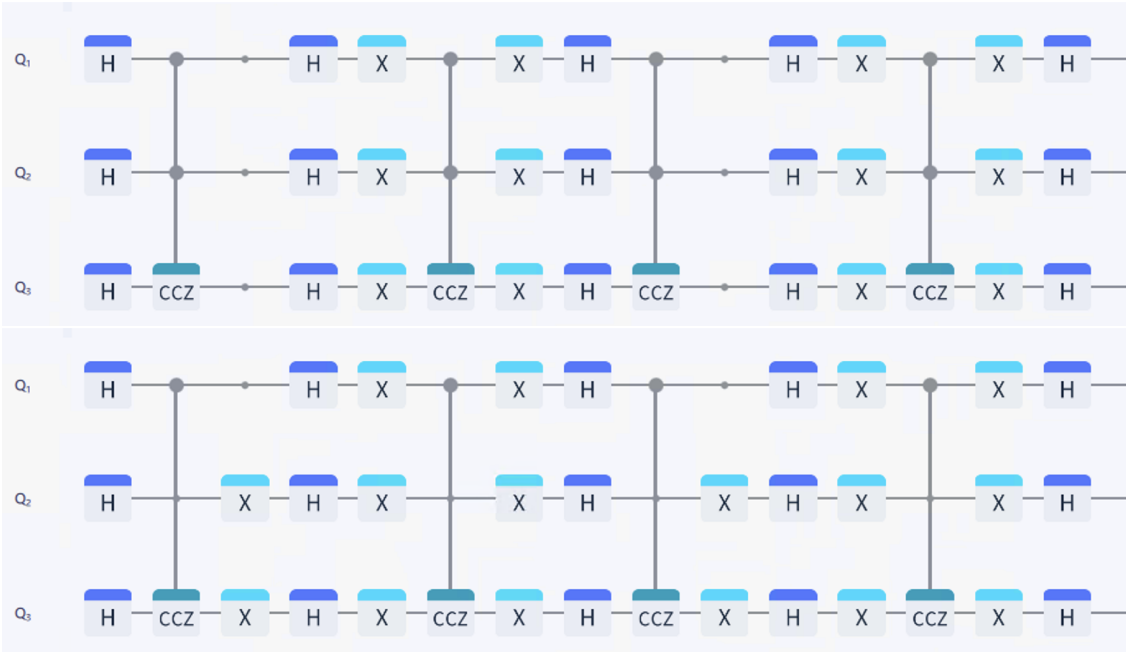
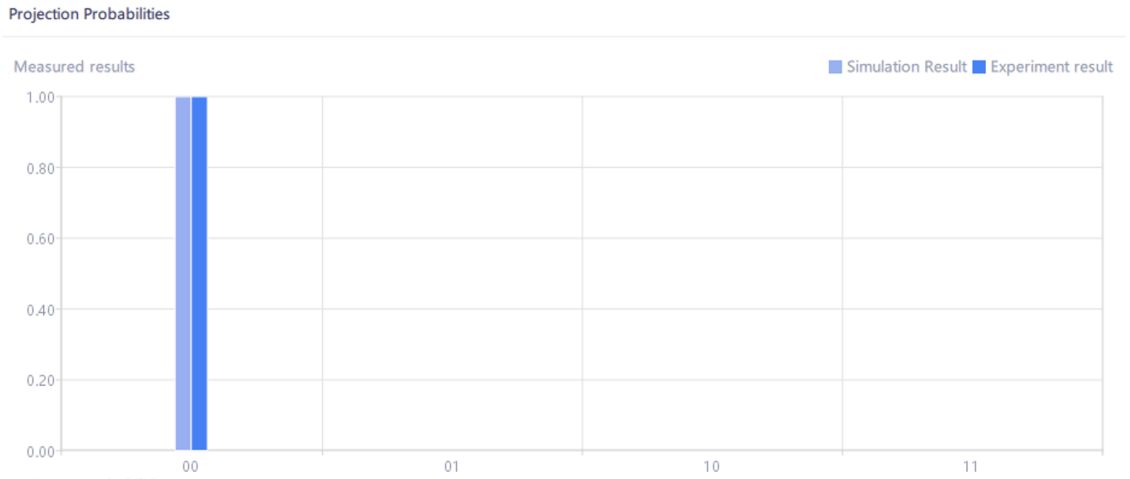
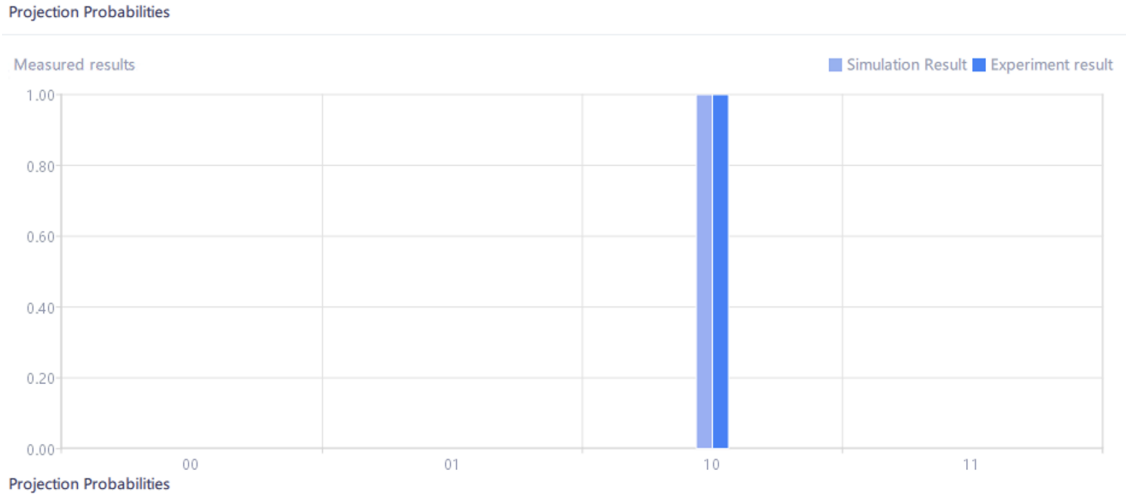


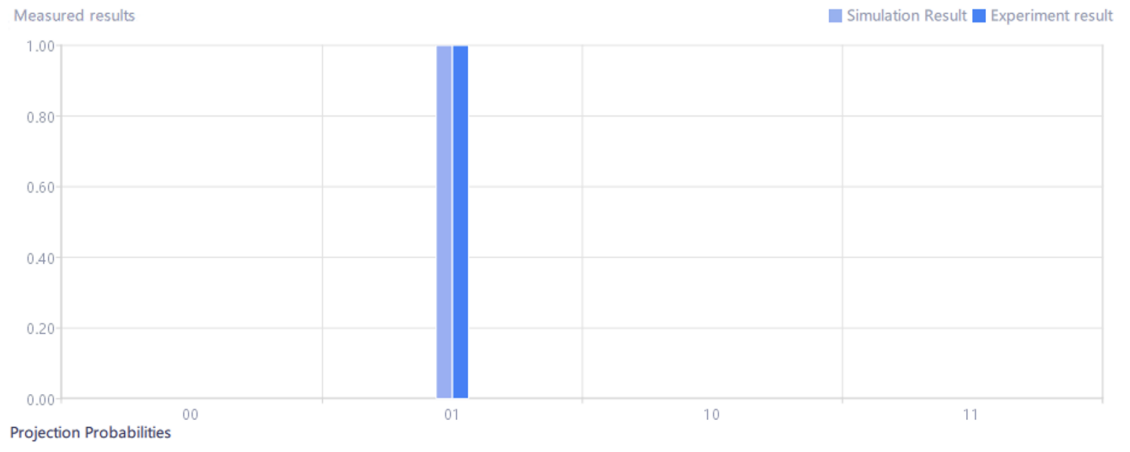
Figure 4.7: Circuit representations of the 3-qubits Grover's algorithm for the states $|111\rangle$, above, and $|100\rangle$, below

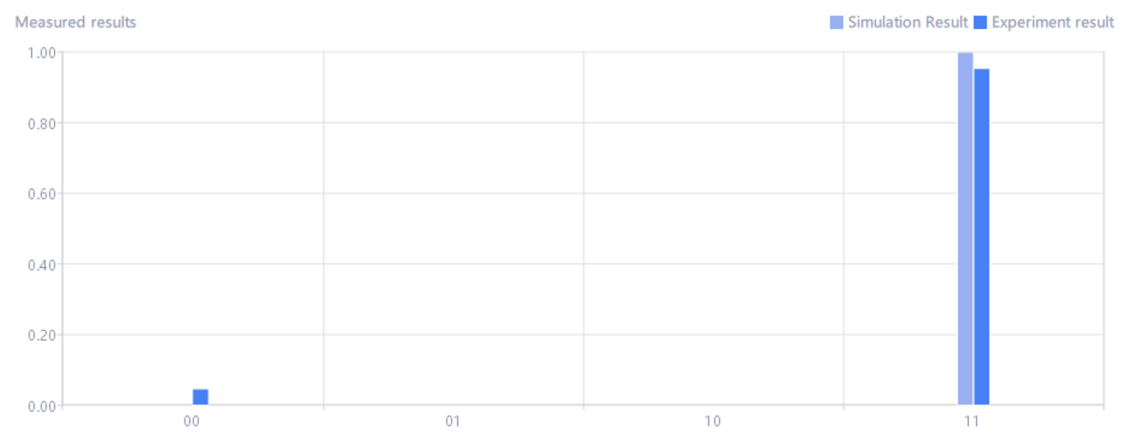
Results

Five test per basis state were performed both for the 2- and 3-qubits algorithms. As for the 2-qubits cases, the results were almost identical with the expected values from the theory. This can be appreciated in Fig. **??**.


Projection Probabilities
Measured results
Simulation Result
Experiment result
1.00
0.80
0.60
0.40
0.20
0.00
00
01
10
11


Projection Probabilities
Measured results
Simulation Result
Experiment result
1.00
0.80
0.60
0.40
0.20
0.00
00
01
10
11


Projection Probabilities
Measured results
Simulation Result
Experiment result
1.00
0.80
0.60
0.40
0.20
0.00
00
01
10
11


Measured results
Simulation Result
Experiment result
1.00
0.80
0.60
0.40
0.20
0.00
00
01
10
11

Whereas for the 3-qubits algorithm, the result were not so clean. Taking as an example the $|111\rangle$ case we can see in Fig. 4.9 how, on 5 tests, there has been one occurrence of an unacceptable result, two occurrences of almost good results and two more of essentially perfect results.

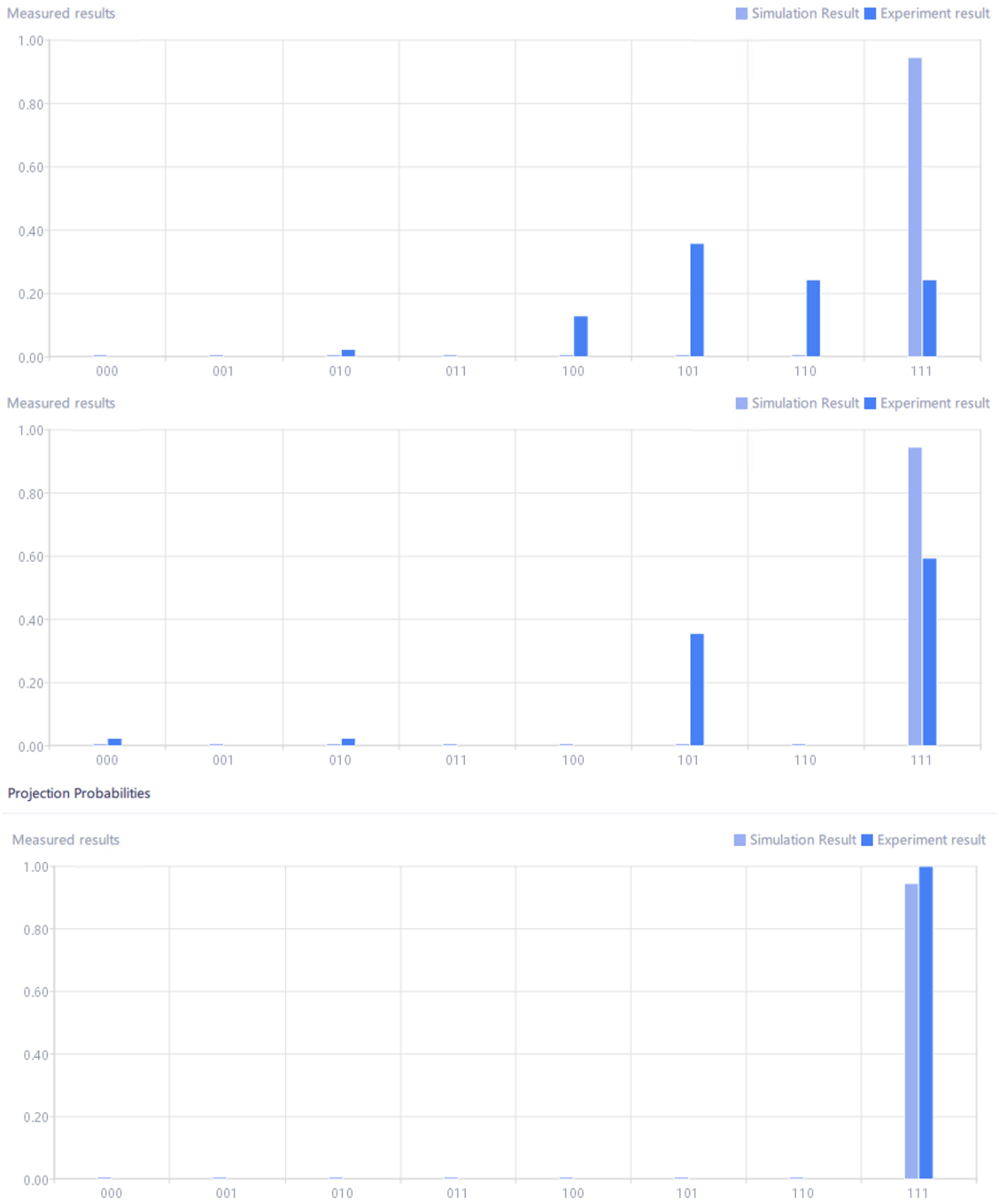


Figure 4.9: Sample of the graphics from the tests on the 3-qubits algorithm for the search of $|111\rangle$, showing the unacceptable, the good and the great results obtained.

The same pattern was observed, without virtually any differences, for the search of the other states.

4.3 HHL algorithm in NMR

In this section we will build and run the HHL algorithm on the NMR computer.

Given that the number of qubits at our disposal is 3, there are not many options on how to divide the registers. We will have one qubit for the constant vector $|b\rangle$, one for the estimate of the eigenvalues and the last one for the ancilla bit.

We will conduct two different tests, one with a system which has a low condition number, and another with a system with an higher condition number, in order to understand how the algorithm performs on diverse problems.

We now discuss the first of this scenarios. Let

$$A_1 = \begin{pmatrix} \frac{6-\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} \\ -\frac{\sqrt{2}}{4} & \frac{6+\sqrt{2}}{4} \end{pmatrix} \text{ and } |b\rangle = \begin{pmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{pmatrix}$$

be respectively the matrix and constant of a linear system of equations. We can see that the condition number of the matrix A is

$$\kappa(A_1) = \left\|A_1^{-1}\right\| \|A_1\| \approx 2,$$

hence the use algorithm should be successful. To solve this problem using HHL the first thing we will need to do is to encode $|b\rangle$ in the respective register and 1 in the ancilla bit, this can be done applying a rotation on the first qubit and a NOT on the last one.

$$\psi_1 = (R_y^1\left(\frac{\pi}{2}\right) \otimes I \otimes X)\psi_0 = |b\rangle\,|0\rangle\,|1\rangle\,. \tag{4.1}$$

In order to determine that $R_y(\frac{\pi}{2})\,|0\rangle = |b\rangle$ it is enough to remember that

$$R_y(\theta) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}.$$

When $\theta = \frac{\pi}{2}$ is not hard to see that

$$R_y(\frac{\pi}{2})\,|0\rangle = \begin{pmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{pmatrix}.$$

In the following we will use the superscript notation, just like in 4.1, to indicate on which qubit are acting the gates. We will then need to implement the controlled-$e^{iAt_0/2}$ gate C-U and the controlled rotation of the ancilla bit $RY(\theta)$. To realise the controlled gate for the QPE we will need to diagonalize A. Once the matrix A is decomposed as

$$A_1 = U_d \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda 2 \end{pmatrix} U_d^\dagger,$$

we can encode the gate C-U as

$$C\text{-}U = U_d \cdot CZ \cdot U_d^\dagger. \tag{4.2}$$

Using this decomposition, in QPE and IQPE we should have the structure shown in Fig. 4.10.

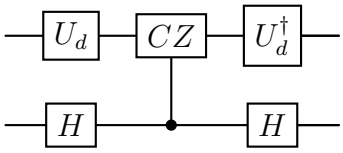


Figure 4.10: Close up of the QPE in the HHL algorithm using the formula from 4.2

We can now simplify the code, saving some calculations, remembering that a controlled-Z gate in between of two Hadamard gates is equivalent to a CNOT. In this particular case we find that

$$A_1 = \begin{pmatrix} \cos\left(\frac{\pi}{8}\right) & \sin\left(\frac{\pi}{8}\right) \\ -\sin\left(\frac{\pi}{8}\right) & \cos\left(\frac{\pi}{8}\right) \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} \cos\left(\frac{\pi}{8}\right) & -\sin\left(\frac{\pi}{8}\right) \\ \sin\left(\frac{\pi}{8}\right) & \cos\left(\frac{\pi}{8}\right) \end{pmatrix},$$

hence $U_d = R^1_{-y}(\frac{\pi}{4})$ and $U_d^\dagger = R^1_y(\frac{\pi}{4})$. For this reason we can couple this gate with the one for the encoding of $|b\rangle$, leaving a $R^1_y(\frac{\pi}{4})$ gate. Additionally the $U_d^\dagger$ gate from QPE simplifies with the U_d from IQPE.

As for the rotation of the ancilla bit, it is possible to express it through

$$RY(\theta) = CZ \cdot R^3_{-x}(\frac{\pi}{2}) \cdot R^3_{-z}(\frac{\pi-\theta}{2}) \cdot CNOT \cdot R^3_z(\frac{\pi-\theta}{2}) \cdot R^3_{-x}(\frac{\pi}{2}), \tag{4.3}$$

where $\theta = 2\arccos(\lambda_1/\lambda_2)$.

Remark 5. Given that A is Hermitian, and hence normal, the condition number can be expressed as

$$\kappa(A) = \left|\frac{\lambda_{max}}{\lambda_{min}}\right|.$$

We know that $\kappa(A)$ is always grater than 1. This implies that we would need work with a decomposition of the form

$$A = U_d \begin{pmatrix} \lambda_{min} & 0 \\ 0 & \lambda_{max} \end{pmatrix} U_d^\dagger,$$

is to say $\lambda_1 = \lambda_{min}$ and $\lambda_2 = \lambda_{max}$. In fact, if this was not the case, then

$$\theta = 2\arccos\left(\frac{\lambda_1}{\lambda_2}\right),$$

would not be well defined. Hence if $\lambda_1/\lambda_2 > 1$ we cannot apply the above method for the implementation of the $RY(\theta)$ gate. △

Finally, substituting in 4.2 and 4.3 the values of the variables we obtain the quantum code which solves the problem:

$$R_y^1(\frac{\pi}{4}) \cdot X^3 \cdot CNOT^{1,2} \cdot CZ^{2,3} \cdot R_{-x}^3(\frac{\pi}{2}) \cdot R_{-z}^3(\frac{\pi}{3}) \dots$$
$$\dots \cdot CNOT^{2,3} \cdot R_z^3(\frac{\pi}{3}) \cdot R_{-x}^3(\frac{\pi}{2}) \cdot CNOT^{1,2} \cdot R_y^1(\frac{\pi}{4}).$$

Moving onto the computer, the code we have described will result in the circuit in Fig. 4.11

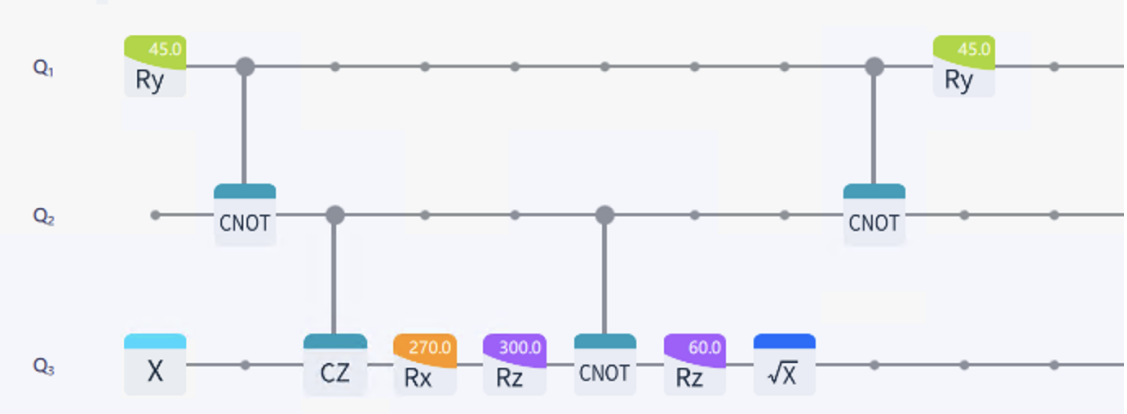


Figure 4.11: HHL circuit on the SpinQuasar software.

In the case of a ill-conditioned matrix, the algorithm presents some criticalities. Consider the following entries to the problem

$$A_2 = \begin{pmatrix} \frac{202-99\sqrt{2}}{40} & \frac{99\sqrt{2}}{40} \\ \frac{99\sqrt{2}}{40} & \frac{202+99\sqrt{2}}{40} \end{pmatrix} \text{ and } |b\rangle = \begin{pmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{pmatrix}$$

For simplicity we have chosen A_2 such that

$$A_2 = U_d \begin{pmatrix} 0.1 & 0 \\ 0 & 10 \end{pmatrix} U_d^\dagger,$$

where U_d is the same as in the last problem. The condition number of A_2 is $\kappa(A_2) = 100$, which is significantly higher than $\kappa(A_1)$. Since U_d is still equivalent to $R_{-y}(\frac{\pi}{4})$, to write the code for this problem we only need to evaluate θ for the expression of the ancilla bit rotation. We find that

$$\theta = 2 \arccos(0.01) \approx 89.4°.$$

The circuit for this problem is presented in Fig. 4.12.

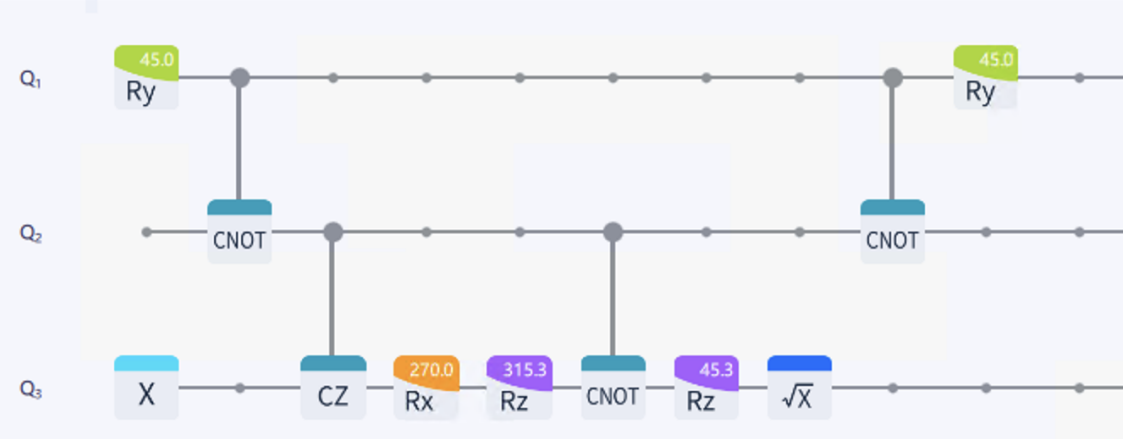


Figure 4.12: HHL circuit for a bad conditioned problem on the SpinQuasar software.

Results

From the theory we can expect to have the final output delivered in the states $|001\rangle$ and $|101\rangle$. We will consider acceptable those results with the highest probability concentrated in those two basis states. Starting with the first circuit, for a well conditioned matrix, in a sample of 5 test we got one totally off result, one almost acceptable result, and three acceptable results, one of which with a fairly good resemblance to the expected data.

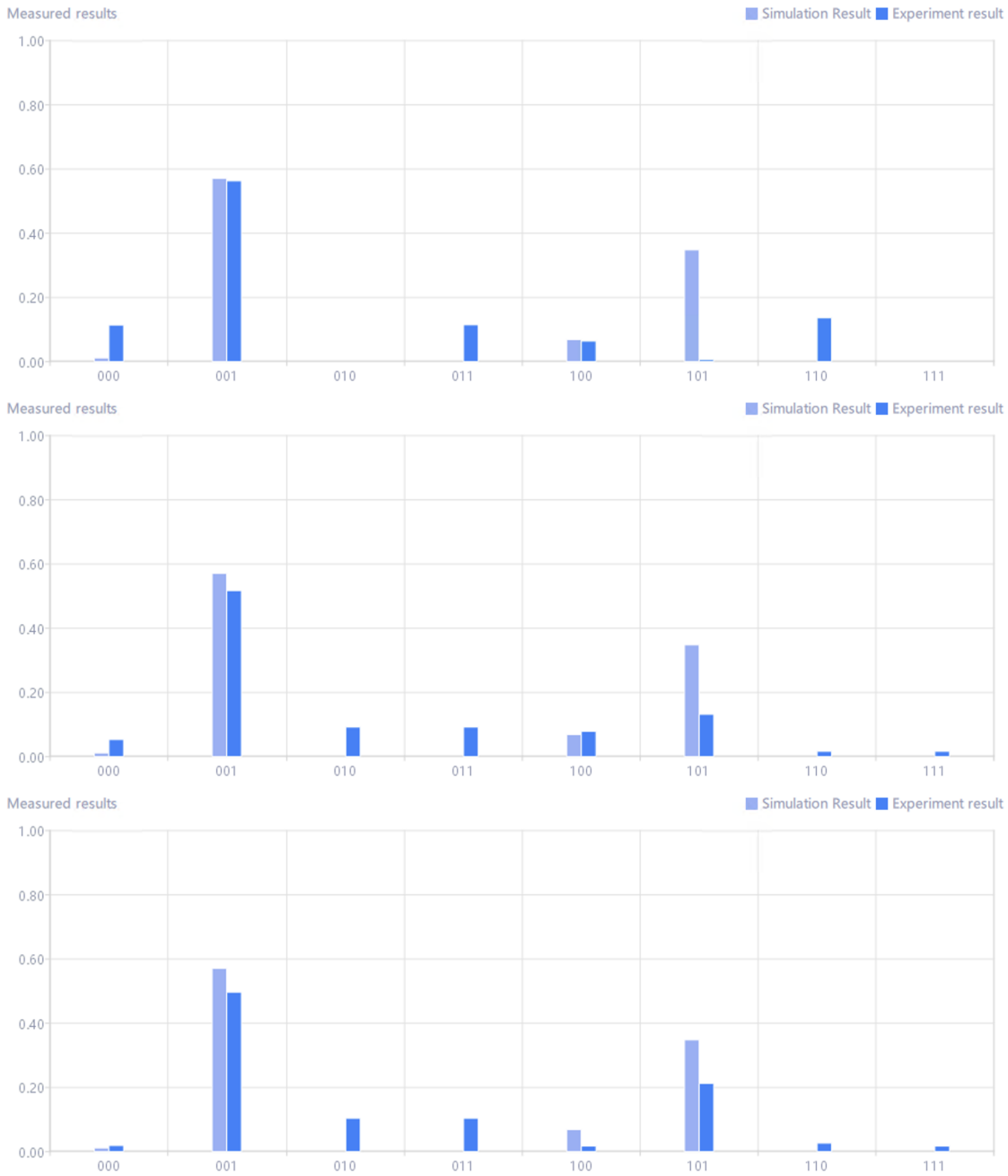


Figure 4.13: Graphics, respectively from experiments n.3 n.1 and n.2, to appreciate the difference between the observed data.

From Fig. 4.13 we can see how, there appears to be some sort of "scattering" of the probabilities of the second entry of the vector, the one expected in $|101\rangle$. This could be due to the c-register, the second qubit, not reading 0 as it is supposed to after IQPE, or to the approximation of the ancilla bit rotation.

The other collection of tests, performed on the worse conditioned system, pre-

sented more problems. We were in fact unable to achieve an acceptable result in four of the five tests we have performed. We could however observe that the "scattering" problem has intensified, as it can be seen in Fig. 4.14, hinting towards it being a problem due to the approximation of the gate RY, given that in this scenario θ was significantly bigger.

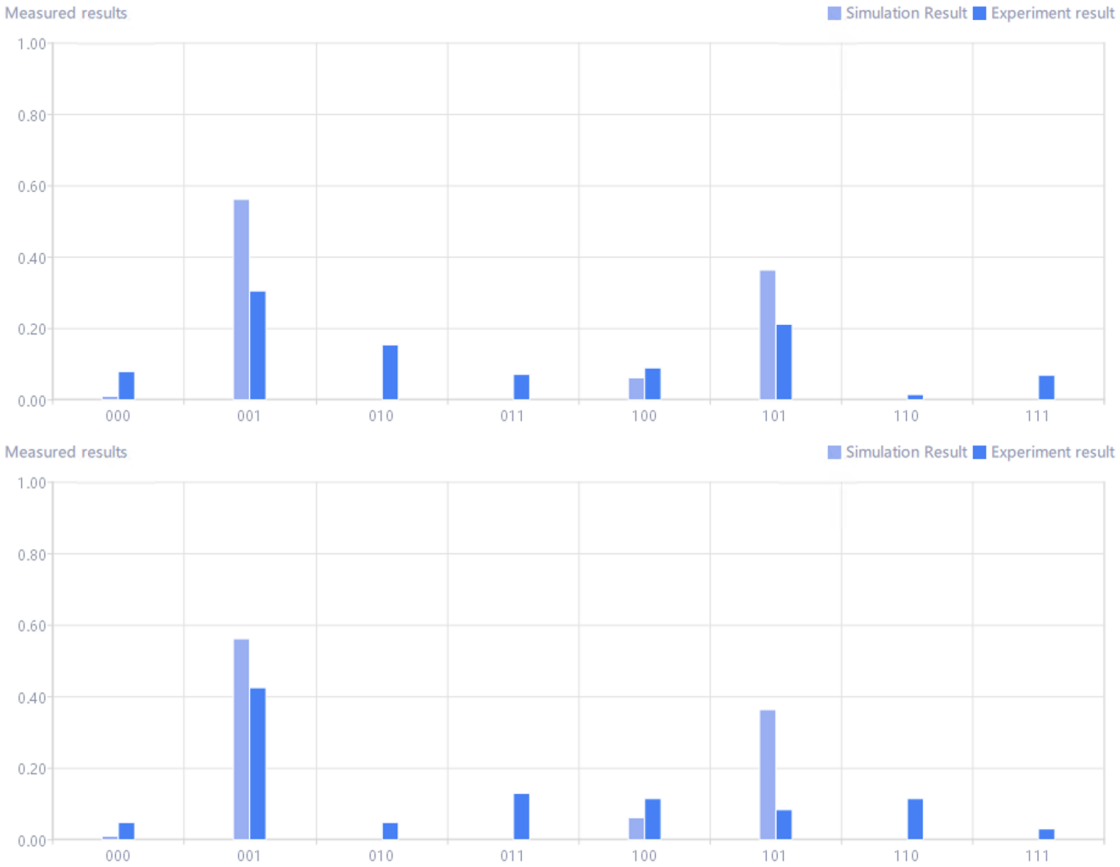


Figure 4.14: In the above graphic, the only one of the bad conditioned tests to be acceptable by our standards. Below we find the graph from experiment n.3 as a comparison.

4.4 Bell's inequality in NMR

As a last test, we will perform a verification of Bell's inequality. This is usually referred to as Bell test, or Bell experiments, and can be done in various ways.

The circuit we will use for the Bell test is depicted in Fig. 4.15.

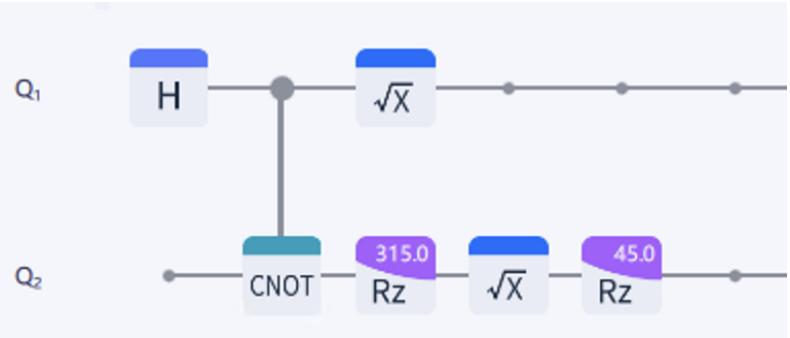


Figure 4.15: Circuit used for the Bell test, realized in SpinQuasar.

Results

Five experiments are performed, with the following outcomes. Tests number 1 and 3 both lean pretty close to the theoretical expectations, as it can be seen in Fig. 4.16.

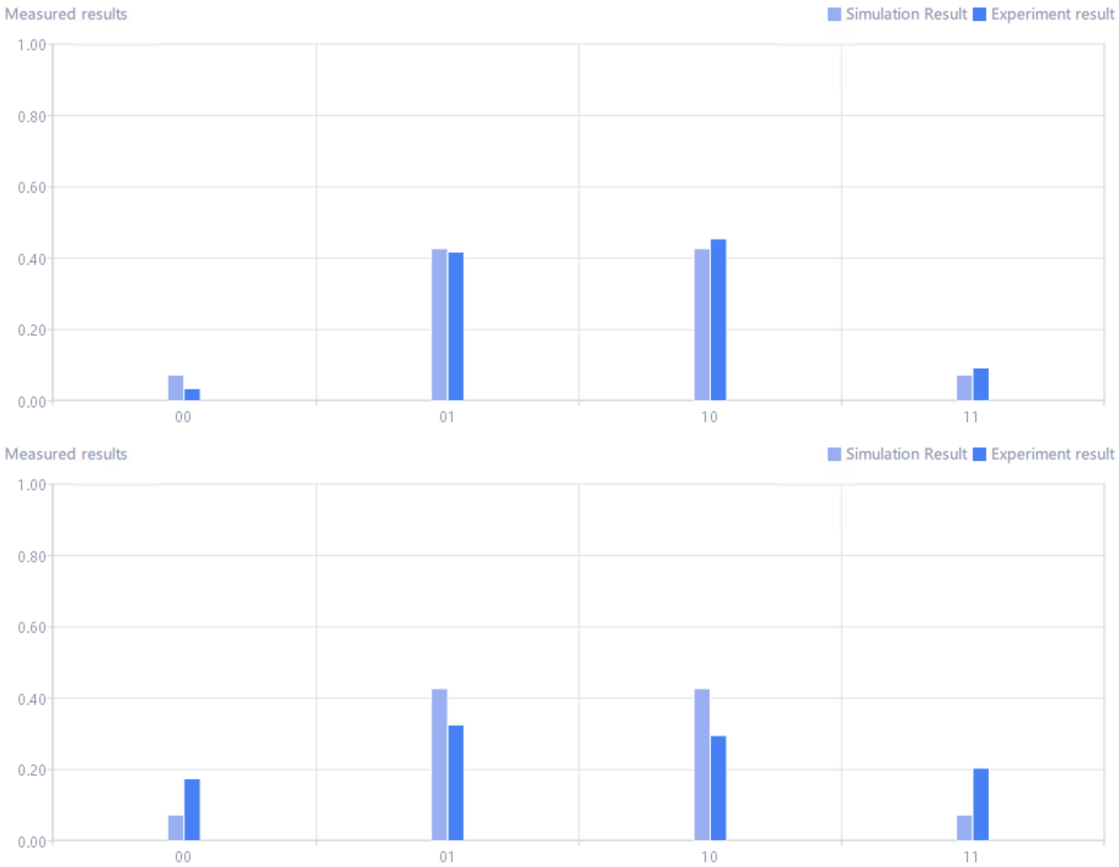


Figure 4.16: Two of the five performed tests with the Bell test circuit.

The remaining tests, while still presenting somewhat good results, all have 0 as an entry on the first basis state.

Table 4.1: Results of the experiments with the Bell test. Each column contains the probabilities for every basis state to be measured, obtain from the five respective tests. The only exception is the first column which has the values from the theory.

	T	$\mathbf{E}_1$	$\mathbf{E}_2$	$\mathbf{E}_3$	$\mathbf{E}_4$	$\mathbf{E}_5$
$p(\|00\rangle)$	0.073	0.035	0	0	0	0
$p(\|01\rangle)$	0.427	0.418	0.466	0.485	0.455	0.471
$p(\|10\rangle)$	0.427	0.454	0.487	0.497	0.493	0.491
$p(\|11\rangle)$	0.073	0.093	0.048	0.018	0.052	0.038

4.5 Discussion

While the results for the 2-qubits tests have been almost precise, this is not the case for the 3-qubits algorithms. We have seen how in Grover and HHL the readouts produced were not acceptable a third of the times. This unreliability may be due to a variety of different reasons. Additionally, it is probably the case that the results depend on various small problems rather than on a big mistake. This makes it difficult to identify the causes of the errors.

Basing on the collected evidence, we propose here three different scenarios:

- Considering the calculations that we made to be correct, one could argue that most of the problem relies on the quantum computer, which could either malfunction or be miscalibrated. If this was the case, than one should focus on identifying and fixing the issue, in a work more suited for physicists or engineers.

- If we instead consider the machine to be working correctly, then the error must lie in the way of operating it. It could be the case that we worked under some wrong assumptions, and hence a more rigorous mathematical approach is required.

- Finally, there is also the possibility that our work is correct and the computer is working as it should, but our understanding of the machine its in some way off. This calls for a more comprehensive study on the quantum advantage of this algorithms or on the probability theory underlying the measurement process. By doing so, we would then be able to better judge the performance of the computer and hence this results.

Conclusions and outlook

In this master's thesis we have studied some of the most well known quantum algorithms, focusing on Grover and HHL, to understand the key ideas behind their practical operation. In the first one, we have seen how using the superposition state of qubits can allow for faster oracle queering. The key idea in the HHL was instead to exploit the natural way of implementing a Fourier transform on qubits to encode the eigenvalues of a given vector in the register. From there we only need to rotate the other registers in the right position to obtain the solution vector.

We then discussed the physical principles of quantum computing. In particular we talked about what are the current challenges of their realization and the various proposals that have caught researcher's interest. We presented the fundamental ideas behind superconducting, photonic and ion traps quantum computers, explaining why this aim to meet DiVincenzo's criteria. In this same chapter, we also introduced NMR computing. Studying this last proposal more in depth, we have seen how this model is different from the others, basically presenting an ensemble of computers.

In last chapter, we tested the expected results we had from the theory in an actual real quantum computer. The access to the a 3-qubit NMR computer proved fundamental to obtain a more complete view of the subject. The experimental tests, raised in fact questions on both the reliability of small scale NMR quantum computers and the framework of quantum information theory. While this may as well be a fitting conclusion for this work, it certainly leaves room for interesting outlooks. We in fact intend to perform other tests on the computer, as well as develop a more comprehensive mathematical theory.

In particular, what we intend to do, is to obtain a different formalization for quantum probability, which would in turn allow us to perform better measurements. The classical approach of measure theory and Kolmogorov's axioms could not be the most fitting in QM. Using as the mathematical framework for QM the work of Von Neumann and others on operators algebras may be a better idea.

This can be done by first defining an observable as a self-adjoint matrix in the algebra $\mathbb{C}^{N\times N}$ and a quantum state as a particular linear functional on the same algebra. With this approach, one could then define quantum analogues of events and random variables. In this context a quantum computer can be thought of as a pair (A, ω_0), where A is an observable and ω_0 a state. The computer would run a quantum algorithm by applying the code to the state ω_0. For the readout, the

operator A would me measured on the resulting final state ω_l.

We conclude by restating that, even if the results from the tests and the theory did not coincide, we can still make use of this outcome as a starting point for further work.

Bibliography

[1] Adriano Barenco. "A universal two-bit gate for quantum computation". In: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 449.1937 (1995), pp. 679–683.

[2] Nishanth Baskaran et al. "Adapting the harrow-hassidim-lloyd algorithm to quantum many-body theory". In: *Physical Review Research* 5.4 (2023), p. 043113.

[3] Samuel L Braunstein et al. "Separability of very noisy mixed states and implications for NMR quantum computing". In: *Physical Review Letters* 83.5 (1999), p. 1054.

[4] Isaac L Chuang et al. "Bulk quantum computation with nuclear magnetic resonance: theory and experiment". In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 454.1969 (1998), pp. 447–467.

[5] Juan I Cirac and Peter Zoller. "Quantum computations with cold trapped ions". In: *Physical review letters* 74.20 (1995), p. 4091.

[6] David Deutsch. "Quantum theory, the Church–Turing principle and the universal quantum computer". In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 400.1818 (1985), pp. 97–117.

[7] David Deutsch and Richard Jozsa. "Rapid solution of problems by quantum computation". In: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 439.1907 (1992), pp. 553–558.

[8] David P DiVincenzo. "The physical implementation of quantum computation". In: *Fortschritte der Physik: Progress of Physics* 48.9-11 (2000), pp. 771–783.

[9] Richard Feynman. "There's plenty of room at the bottom". In: *Feynman and computation*. CRC Press, 2018, pp. 63–76.

[10] Lov K Grover. "A fast quantum mechanical algorithm for database search". In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 1996, pp. 212–219.

[11] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. "Quantum algorithm for linear systems of equations". In: *Physical review letters* 103.15 (2009), p. 150502.

[12] Jack D Hidary and Jack D Hidary. "A brief history of quantum computing". In: *Quantum Computing: An Applied Approach* (2021), pp. 15–21.

[13] Navin Khaneja et al. "Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms". In: *Journal of magnetic resonance* 172.2 (2005), pp. 296–305.

[14] Emanuel Knill, Raymond Laflamme, and Gerald J Milburn. "A scheme for efficient quantum computation with linear optics". In: *nature* 409.6816 (2001), pp. 46–52.

[15] Thaddeus D Ladd et al. "Quantum computers". In: *nature* 464.7285 (2010), pp. 45–53.

[16] Joseph B Lambert, Eugene P Mazzola, and Clark D Ridge. *Nuclear magnetic resonance spectroscopy: an introduction to principles, applications, and experimental methods*. John Wiley & Sons, 2019.

[17] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. "Quantum algorithms for supervised and unsupervised machine learning". In: *arXiv:1307.0411* (2013).

[18] Michael A Nielsen, Emanuel Knill, and Raymond Laflamme. "Complete quantum teleportation using nuclear magnetic resonance". In: *Nature* 396.6706 (1998), pp. 52–55.

[19] Mary A Rowe et al. "Experimental violation of a Bell's inequality with efficient detection". In: *Nature* 409.6822 (2001), pp. 791–794.

[20] John Shalf. "The future of computing beyond Moore's Law". In: *Philosophical Transactions of the Royal Society A* 378.2166 (2020), p. 20190061.

[21] Lieven MK Vandersypen et al. "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance". In: *Nature* 414.6866 (2001), pp. 883–887.

[22] Yaakov S Weinstein et al. "Implementation of the quantum Fourier transform". In: *Physical review letters* 86.9 (2001), p. 1889.

[23] Anika Zaman, Hector Jose Morrell, and Hiu Yung Wong. "A Step-by-Step HHL Algorithm Walkthrough to Enhance Understanding of Critical Quantum Computing Concepts". In: *IEEE Access* (2023).