

# Digital Quantum Simulation of Few-body Hamiltonians on a Noisy Quantum Computer

Pedro Miguel Miranda Queiroz da Cruz

Mestrado em Física

Departamento de Física e Astronomia

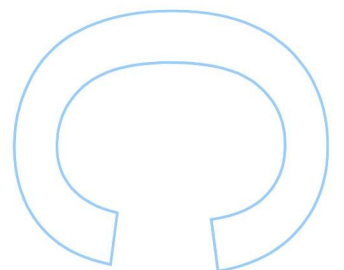
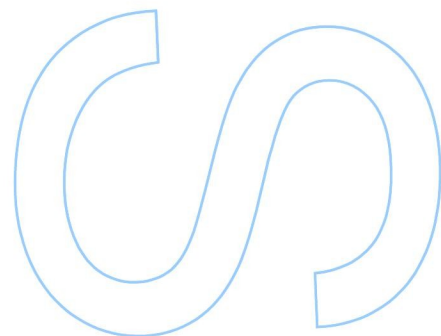
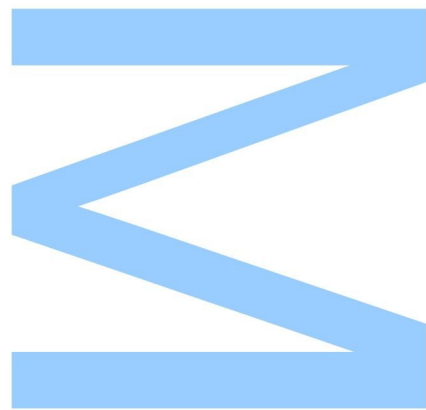
2017/2018

## Orientador

Prof. Dr. Joaquín Fernández-Rossier, Líder do grupo de Teoria de Nanoestruturas Quânticas, QuantaLab - INL (International Iberian Nanotechnology Laboratory)

## Coorientador

Prof. Dr. João Lopes dos Santos, Professor Catedrático, Faculdade de Ciências da Universidade do Porto

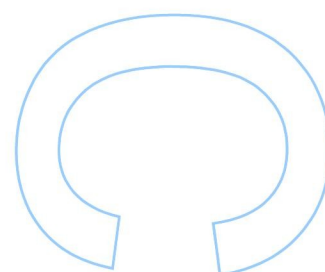
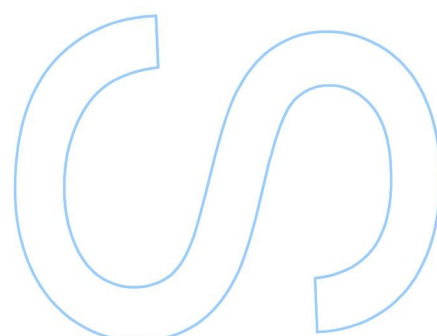
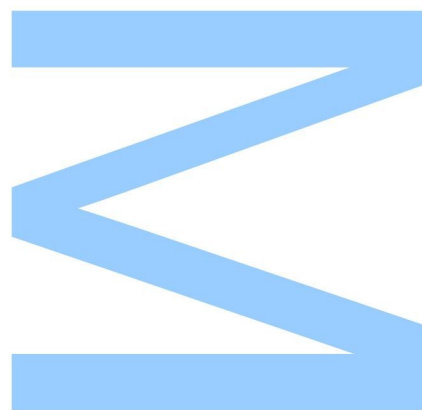




Todas as correções determinadas pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, \_\_\_\_ / \_\_\_\_ / \_\_\_\_



*"Today abstraction is no longer that of the map, the double, the mirror, or the concept. Simulation is no longer that of a territory, a referential being, or a substance. It is the generation by models of a real without origin or reality: a hyperreal. The territory no longer precedes the map, nor does it survive it. It is nevertheless the map that precedes the territory - precession of simulacra - that engenders the territory, and if one must return to the fable, today it is the territory whose shreds slowly rot across the extent of the map. It is the real, and not the map, whose vestiges persist here and there in the deserts that are no longer those of the Empire, but ours. The desert of the real itself."*

— Jean Baudrillard, *Simulacra and Simulation*, 1981



# Agradecimentos

Antes de tudo, devo um profundo agradecimento à minha família, aos meus pais, Jacinta Quinta e Marcelo Cruz, e à minha irmã, Bárbara Cruz. As palavras são incapazes de fazer justiça a todo o apoio que em tudo sempre me deram, em particular neste caminho formativo que nem sempre foi fácil. Sem eles, eu não seria hoje quem sou.

Estou muito agradecido ao Professor Joaquín Fernandez-Rossier, meu orientador, por todas as oportunidades que me ofereceu e por me ter dado a possibilidade de assumir este projeto, arriscando uma área tão nova para si quanto para mim. O seu modo informal e próximo de agir em muito ajudou à abertura e profundidade das nossas discussões. Guardo apreço pelo apoio no direcionamento do meu percurso e agradeço ainda pelo convite para apresentar estes resultados na *IBM-QuantaLab School on Quantum Computing*, no INL. Agradeço igualmente ao Professor João Lopes dos Santos não só pelo apoio ao aceitar fazer a minha orientação do lado da Faculdade, mas também pela preparação que me deu com aulas em várias disciplinas, sempre desafiantes e repletas de *insight* físico. Foi um desejo concretizado conseguir fazer uma dissertação na área que ambicionava aprender desde o início do Mestrado.

Agradeço ao Gonçalo Catarina por acompanhar este projeto de perto desde o início, por ler e me ajudar a rever este documento. Ao Ronan Gautier agradeço igualmente a colaboração no desenvolvimento de objetivos comuns aos deste trabalho no seio do nosso grupo. A vontade de aprendizagem de todos trouxe consigo úteis trocas de ideias que assim contribuíram para tornar melhor esta dissertação.

Deixo também aqui um agradecimento a todos os meus professores de forma geral, com quem direta e indiretamente percorri os caminhos da Física e Matemática, e com quem também entendi que devia aprender mais por mim mesmo. Em particular, deixo uma palavra de agradecimento ao Professor Augusto Rodrigues que, enquanto Diretor do Mestrado, entendeu e permitiu a minha escolha de algumas disciplinas opcionais que me interessava perseguir.

Durante este Mestrado tive o privilégio de participar em outros dois projetos de investigação. Presto um grande agradecimento ao Professor João Pedro Pedroso por ter acreditado em mim e me ter dado a autonomia de trabalhar num projeto aliciante numa altura que para mim foi decisiva. Ficarei para sempre grato por esta iniciação. Ao Rui Rei agradeço igualmente por todas as dicas profissionais, sempre acompanhadas pelos indispensáveis almendrados. Estou também muito grato ao Professor João Nuno Tavares, à Professora Sónia Gouveia e ao Nilson Lima pela aprendizagem, colaboração, amizade e boa disposição que partilhamos durante mais de um ano num projeto igualmente cativante.

Ao longo deste percurso tive o prazer de conhecer e coabitar com vários estudantes estrangeiros, alguns dos quais se tornaram bons amigos. A todos agradeço as inesquecíveis vivências. Fui aprendendo que a parte infeliz deste intercâmbio cultural é que sempre chega ao fim, deixando a amizade dilatada entre a distância. Guardo comigo as pessoas e momentos que mais me marcaram. Em particular, gostava de mencionar o Emílio Trevisani e o Jonathan Toledo.

Termino não pelo fim, mas pelo início da minha sorte. Deixo aqui um sentido agradecimento à Bruna Araújo pela presença permanente e por todo o amor. O seu entusiasmo contagiante pela Física transpareceu sempre para mim da forma certa, realista, e a capacidade inata de me fazer persistir esteve sempre presente quando precisei. Não conheço coração maior que o seu.

Agradeço ainda ao INL pela hospitalidade, ao QuantaLab pela oportunidade de acesso exclusivo ao IBM Q 20 Tokyo, e à IBM Corporation por disponibilizar os computadores quânticos usados neste trabalho à comunidade.



# Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>   | <b>9</b>  |
| <b>2</b> | <b>Computing in the Hilbert space</b>                               | <b>11</b> |
| 2.1      | Historical background . . . . .                                     | 11        |
| 2.2      | Fundamentals of digital quantum computation . . . . .               | 13        |
| 2.3      | Quantum Fourier transform . . . . .                                 | 20        |
| 2.4      | Phase estimation algorithm . . . . .                                | 25        |
| 2.5      | Iterative phase estimation . . . . .                                | 32        |
| 2.6      | Noisy computations . . . . .  | 38        |
| 2.6.1    | Useful information theoretic measures . . . . .                     | 40        |
| 2.7      | Current status and near term prospects . . . . .                    | 41        |
| 2.7.1    | IBM Quantum Experience . . . . .                                    | 42        |
| <b>3</b> | <b>Quantum many-body systems</b>                                    | <b>47</b> |
| 3.1      | The many-body problem . . . . .                                     | 47        |
| 3.2      | Condensed matter <i>theory of everything</i> . . . . .              | 48        |
| 3.3      | Strongly entangled systems . . . . .                                | 49        |
| 3.4      | Simple spin Hamiltonians . . . . .                                  | 50        |
| 3.5      | A first look at the Hubbard model: the 2-site Hamiltonian . . . . . | 51        |
| <b>4</b> | <b>Digital quantum simulation of time-evolution</b>                 | <b>57</b> |
| 4.1      | Jordan-Wigner mapping to the qubit space . . . . .                  | 57        |
| 4.2      | The phase estimation approach . . . . .                             | 60        |
| 4.3      | Trotter-Suzuki decomposition . . . . .                              | 61        |
| 4.4      | State preparation . . . . .   | 63        |
| <b>5</b> | <b>Experiments on a real quantum computer</b>                       | <b>65</b> |
| 5.1      | Fourier transform . . . . .   | 65        |
| 5.2      | Phase estimation . . . . .  | 66        |
| 5.2.1    | Zeeman Hamiltonian . . . . .  | 66        |
| 5.2.2    | Ising Hamiltonian . . . . .   | 70        |
| 5.2.2.1  | Evolving an eigenstate . . . . .                                    | 70        |
| 5.2.2.2  | Evolving a general state . . . . .                                  | 72        |
| 5.2.3    | 2-site Hubbard Hamiltonian . . . . .                                | 74        |
| 5.3      | Iterative phase estimation . . . . .                                | 78        |
| 5.3.1    | Hubbard 2-site 2-electron sector . . . . .                          | 79        |
| 5.3.2    | Benchmarking accuracy with the Zeeman Hamiltonian . . . . .         | 84        |

**6 Conclusions**

**87**

**Bibliography**

**89**



# Chapter 1

## Introduction

*“Science is more than a body of knowledge. It is a way of thinking.”*

— Carl Sagan, last interview with Charlie Rose, 1996

The main goal of this Master’s project is to explore the use of quantum computers to address the solution of many-body quantum problems. I tackle two aspects. First, the understanding of how quantum algorithms can accomplish this task in ideal quantum computers. Second, the implementation of these algorithms in state of the art quantum computers, that have been made available for remote on-line access.

The quantum many-body problem has been formulated many decades ago, and its exact solution remains intractable, except in a few niche cases that can be treated numerically or analytically.

A general brute-force computational approach faces the so-called exponential wall problem: the dimension of the Hilbert space scales exponentially with the number of degrees of freedom. For instance, in a brute force approach, a spin-only model for  $N$  spins with  $S = 1/2$  requires storing in memory  $2^N$  vectors of dimension  $2^N$ . The most powerful supercomputers have managed to do this with  $N = 50$ . Going to  $N = 51$  would require duplicating computational resources.

In the case of the many-body problem, as we discuss below, there is a linear scaling between the number of qubits and the number of degrees of freedom of the system we want to study. Therefore, when it comes to store and study quantum states, quantum computers can mimic complex many-body entangled states and have potential for exponential improvement in information storage performance.

We will stay within the circuit model of quantum computing and mostly deal with the algorithmic side, while also exploring implementations on real quantum computers. The detailed characterization of these real physical systems and platforms used is beyond the scope of this dissertation.

This work is organized as follows:

- In **Chapter 2** we start by reviewing some quantum computing fundamentals and study in depth the three main algorithms to be used in the next sections. Computations that are efficient to carry out on a quantum computer are not necessarily also efficient on a classical computer, and vice-versa. We will first encounter this by learning about the striking efficiency of the quantum Fourier transform which will be an essential building block for the algorithms to follow. This will be our starting point in the study of quantum algorithms. After that, we will move on to introduce two quantum procedures of central importance for the subject of this dissertation - the phase estimation algorithm and its iterative version. We develop those from the ground up and explore some original approaches on how to use them with an eye on the implementation in noisy devices. Finally, we explore the differences involved in computing

on noisy environments and present some benchmarking measures, before going on to an overview of the state-of-the-art quantum computers from IBM that we will be using in this work.

- In **Chapter 3** we present the quantum many-body problem and in particular the fermionic case, which is of great interest in quantum chemistry and condensed matter physics. We explain the special instances in which it is important to approach this with an exact method and put forward the reason why quantum computers are prepared to tackle the solution to this problem. Three simple condensed matter models are then developed to be used as the testbed for the digital quantum simulation methods. The first two are spin models and the third one is a Hubbard model.
- In **Chapter 4** we will explore several technical steps involved in carrying out the digital quantum simulation of the dynamics of many-body initial value problems. We start by deriving the Jordan-Wigner mapping to be used in rewriting second quantized Hamiltonians in the spin language. Then, we will introduce the first and second order Trotter-Suzuki decompositions, with which we will be able to implement a quantum circuit for the time-evolution operator of Hamiltonians having non-commuting terms. Methods are presented for initial state preparation, another key component of the whole simulation procedure.
- Finally, in **Chapter 5** we put it all together and explore several simulations to understand the different methods and Hamiltonians previously presented. We provide implementations on ideal classical emulators of quantum computers and on the real quantum computers from IBM that we could access. Various scenarios are explored and we are able to reach several conclusions.

What is remarkable about quantum information and computation is that it offers a different way of using and looking at quantum theory. By using quantum mechanics through the optics of information processing, the field proposes a novel perspective over the world, leading to new insights not only in interpretational questions but also on the development of new tools to explore quantum mechanics. Furthermore, by showing that the structure of algorithms cannot be separated from the physical process used for computation, a connection is established between the abstractions from the theory of computation and physical reality. In this way, the elusive nature of information is linked with the tangible realm of physics, with deep implications for the understanding of Nature. Personally, I believe the intersection of fundamental physics, information and computation theories is fertile ground to new understandings about the complexity, but also the regularity and repeatability, of the physical world.

Pedro Miguel Miranda Queiroz da Cruz

## Chapter 2

# Computing in the Hilbert space

*“Quantum information processing is so different from what we have thought information processing is like for the last sixty years that it would be a shame not to explore it and find out what we can do with it.”*

— Charles Bennett, IBM, 2016

### 2.1 Historical background

The origins of modern computers date back thousands of years, with the invention of several devices to aid calculation, one of those primordial examples being that of the abacus. An important milestone was achieved in 1837 by Charles Babbage, who made the first design proposal of a general-purpose mechanical computer, the *analytical engine*, which he was never able to build [1].

This early discovery preceded by a century the development of formal mathematical models of computation that emerged from the work of Alan Turing and Alonzo Church in 1936 - the Turing machine [2] and lambda calculus [3]. To this end, many important contributions had also been made by the logicians working on the foundations of Mathematics in the turn of the twentieth century.

The field was originally an abstract construction to study mathematical proofs and the connection to Physics was not understood. It was only with the advent of Quantum Information and Computation that the Theory of Computation was fully made a part of Physics. This came to happen because studying computation entails designing methods to represent and manipulate abstract objects using physical objects. Since the later obey physical laws, the kind of computation that is possible will depend on the physical objects being used.

The understanding of the connections between information processing and Physics started to appear along with the efforts to build classical computers. A seminal insight from Landauer in 1961 shows that a logically irreversible operation<sup>1</sup> always releases a minimum amount of heat and increases entropy in the information-processing system or its environment [4]. In 1973, Bennett is trying to overcome this minimum energy cost and conceives a Turing machine that is both logically and thermodynamically reversible [5, 6]. In the same year, Holevo proves that although a general state of  $n$ -qubits is described by  $2^n$  complex coefficients, the maximum amount of information that can be retrieved from it is only of  $n$  bits (Holevo bound) [7]. The construction of a theory of Quantum Information on the image of Shannon's Information Theory from 1948 [8] (now understood to be “classical”) begins in 1976 with Ingarden [9].

By the late seventies, the first steps on elucidating which implications could quantum mechanics have for the models of computation developed by Turing and Church emerged. In 1979 [10], Benioff studied the use

---

<sup>1</sup>This is any logical function with no one-to-one correspondence between input and output, e.g. AND, OR, XOR, ERASE.

of quantum mechanical phenomena to implement a classical Turing machine. A step ahead was done in 1982 [11] by Feynman when he was asking about the possibility of Physics being simulated by a universal computer; he put forward some thoughts and an incomplete conjecture on what was to be required of his pondered “universal quantum simulator”. Two years before that, in a 1980 book [12] by soviet physicist Yuri Manin, one of the first conceptions of a quantum computer had already been briefly motivated.

However, it was Deutsch who pioneered the generalization of the Turing machine to the quantum realm in 1985 [13], and by showing that we can in principle build universal quantum computers *within* our own Universe he gave one of the earliest signals of the computability of the laws of Nature as we know them today.

Notice that all the papers cited so far in the origins of the field were one-author papers. The year of 1993 saw the important discovery of quantum teleportation [14] by an international team, an effect in which the complete information of an unknown quantum state is sent to a new location to rebuild its replica onto another object of the same type while the original state is destroyed in the process.

With Deutsch the era of quantum algorithms had been born, although the experimental realization of the first universal quantum computer had to wait until 2009 [15]. After some contrived examples of problems that would more efficiently find a solution on quantum computers than on classical ones, in 1994, the field draw great attention when Peter Shor came up with a ground-breaking quantum algorithm to factor integers in polynomial time, something believed impossible with classical computers [16]. In 1996, Lov Grover presented another useful algorithm, this one for searching unsorted databases [17], and Seth Lloyd took on the original Feynman conjecture from 1982 and showed that quantum computers can indeed provide efficient simulation of other quantum systems [18].

In 2002, Lloyd applied the quantum information processing perspective to the whole Universe and understood that it can have performed a maximum of  $10^{122}$  operations on the same number of  $10^{122}$  bits that could be registered on matter, energy, and gravity degrees of freedom [19, 20]. Furthermore, due to the no-cloning and no-deleting theorems we now understand that quantum information is a conserved quantity in the Universe - it can neither be destroyed nor created. Interestingly, an upper bound on the total information content of the observable Universe can also be derived from the very different route of the holographic principle [21, 22] - dividing its spherical surface area by the Planck area one obtains  $10^{123}$  bits.

The field of quantum information and computation has been exploding since the nineties, sub-fields matured and several new ideas saw their development, such as new algorithms for diverse tasks, quantum error correction codes, quantum complexity theory, superdense coding, quantum computing networks, quantum cryptography, quantum machine learning, hamiltonian complexity, tensor networks, etc. Nowadays, with the first universal

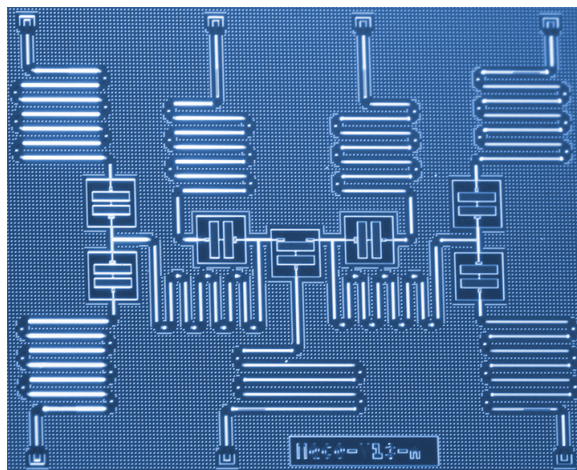


Figure 2.1: Seven-qubit superconducting quantum processor developed by IBM in 2017. (Credit: IBM)

quantum computers with more than a dozen qubits available, efforts have been gathering to find useful applications for the present-day hardware.

Besides the obvious interest in its exciting technological applications, the field has also been proving fruitful in leading to new theoretical tools and developments into fundamental physics, including quantum gravity and black holes, which are expected to soon become possible to study in the lab [23, 24, 25, 26, 27]. Understanding black holes is the bridge to quantum gravity, and these ubiquitous objects are in fact very interesting also from the computational point of view - almost all the information in the Universe exists in black holes, they can be regarded as the fastest and most energy efficient possible computers, and are also the limit for information storage per given size.

## 2.2 Fundamentals of digital quantum computation

Let us start with the big picture.

### ***What is a Quantum Computer?***

Broadly (and abstractly) speaking, a quantum computer is a device which takes advantage of exclusively quantum phenomena, such as *superposition* and *entanglement*, as resources to process information in a fundamentally different way from what classical computers are able to do.

Over the years that the Quantum Computing field has been developed, several different types of quantum computers were proposed in an attempt to design a system that could be experimentally realized. The many different architectures trying to make quantum computers a reality can be divided in some major categories: digital quantum computers [28], quantum annealing devices [29, 30], analog (or adiabatic) quantum computers [31], and measurement based quantum computers [32]. In fact, we have been learning that there seem to be several different ways to process quantum information by exploiting physical systems.

In this dissertation, we will be using the digital circuit model of quantum computing. Here, the computer is built from the assembly of several individual two-level quantum systems, which are the most simple type of quantum mechanical system. Information processing is achieved by quantum operations (referred to as gates) that consist of coherent unitary evolution operators sequentially applied on subsets of this discrete physical system. Readout of the computation results is performed in the end of these sequential evolutions. The component subsystems encode the logical bits used for computation which, being governed by quantum mechanics, are called *qubits*.

On an ideal quantum computer of this type, interactions within this multi-qubit system should be entirely controlled by the outside program. The same is true for the interactions with the environment, which should only consist on the instructions coming from the program to manipulate the device and carry out the computational process itself. Further interactions with the surroundings should be minimized as much as possible to avoid corrupting the computational process, and if they happen to occur at a sufficiently low rate, errors can and should be corrected. Coherence of this quantum system should also be maintained along the full time taken by the computation.

### ***Why is building a quantum computer so hard?***

Building a quantum computer is an incredibly difficult task. The main problem lies in isolating the system from the environment and protecting perturbations to its state. Quantum systems are very unstable due to being constantly in interaction with their surroundings. Remember that our best physical theories tell us that the fundamental building blocks of the universe are quantum fields permeating all universe and moving much more fluidly and interconnected than the localized objects of the classical world. This is at the heart of the

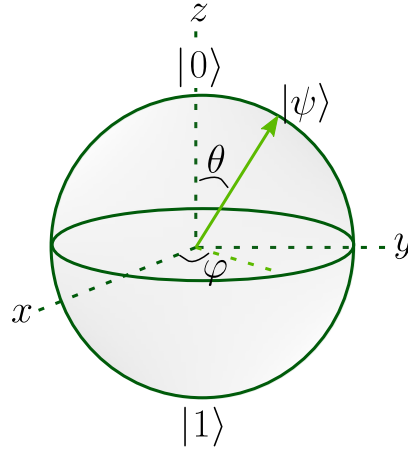


Figure 2.2: Bloch sphere representation of a qubit. When measuring the qubit along the  $\sigma^z$  basis, we collapse the wavefunction and extract information in the form of just one of two possible states:  $|0\rangle$  or  $|1\rangle$ . The probability of each outcome is proportional to the state vector projection on the z-axis, that is on the qubit's "latitude".

challenges in constructing a fully functional quantum computer. To minimize the consequences of interactions, these systems have to operate in very controlled environments and at temperatures lower than outer space.

### What is a Qubit?

The minimal unit of the digital quantum computer is the qubit, the logical element used to store and process information. A qubit is a two-level quantum mechanical system described by a quantum state  $|\psi\rangle$  whose basis states,

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad (2.1)$$

correspond to the classical bits. The distinctive feature of the qubit is that of living in a two-dimensional state space,  $|\psi\rangle \in \mathcal{H} = \mathbb{C}^2$ , in which it can exist in any superposition of the basis states,

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \quad (2.2)$$

where  $\alpha_0$  and  $\alpha_1$  are complex amplitudes obeying the usual normalization constraint  $|\alpha_0|^2 + |\alpha_1|^2 = 1$ . Observables of this quantum system are hermitian linear operators on  $\mathbb{C}^2$ , consisting in  $2 \times 2$  complex matrices that satisfy  $M = M^\dagger$  and are linear combinations of

$$M = c_0 \mathbf{1} + c_1 \sigma^x + c_2 \sigma^y + c_3 \sigma^z \quad (2.3)$$

with  $c_j \in \mathbb{R}$  and the conventional choice of the identity and the Pauli matrices<sup>2</sup> as basis,

$$\mathbf{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (2.4)$$

<sup>2</sup>When useful we will symbolize the Pauli matrices simply as  $X \equiv \sigma^x$ ,  $Y \equiv \sigma^y$ ,  $Z \equiv \sigma^z$ .

The general state of a qubit in Eq.2.2 can also be parameterized by just two real numbers,

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\varphi}\sin\left(\frac{\theta}{2}\right)|1\rangle \quad (2.5)$$

and is represented by a point on the surface of the so called Bloch sphere, depicted in Fig.2.2. Opposite points in this sphere correspond to mutually orthogonal states. In this representation, the  $x$ ,  $y$  and  $z$  axes hold the eigenstates of  $\sigma_x$ ,  $\sigma_y$  and  $\sigma_z$ , respectively.

The generalization to states of  $n$  qubits,  $|\Psi\rangle_n \in \mathbb{C}^{2^n}$ , means that we need to specify  $2^n$  complex numbers to characterize a general state,

$$|\Psi\rangle_n = \sum_{x=0}^{2^n-1} \alpha_x |x\rangle_n, \quad (2.6)$$

with  $\sum_{x=0}^{2^n-1} |\alpha_x|^2 = 1$  and  $x \in \mathbb{N}_0$ . The basis of the  $n$ -qubit state is produced by all the tensor product combinations of the individual basis states of the component qubits. Here, we begin to see a hint of the power of superposition. If you take, say,  $n = 300$ , the number of amplitudes needed to characterize only one state already exceeds the number of atoms in the observable universe. As we will see in the next sections, this means that a state of  $n$ -qubits can work with  $2^n$   $n$ -bit strings simultaneously. By applying operations on this kind of superposition states, several quantum algorithms can produce final states where the results of multiple computations are encoded.

### The computational basis

The computational basis we work with is the  $\sigma_z$  basis and its states will often encode integers or fractional numbers, as we wish. We will use the following notation for the  $n$ -bit representation of an integer  $j$ ,

$$j \equiv j_0 j_1 \cdots j_{n-1} = \sum_{q=0}^{n-1} j_q 2^{(n-1)-q} = j_0 2^{n-1} + j_1 2^{n-2} + \cdots + j_{n-2} 2^1 + j_{n-1} 2^0. \quad (2.7)$$

Similarly, we symbolize a binary fraction representation with  $m - l + 1$  bits by  $0.j_l j_{l+1} \cdots j_m$  and employ the following decode rule

$$0.j_l j_{l+1} \cdots j_m = \sum_{q=l}^m \frac{j_q}{2^{q-l+1}} = \frac{j_l}{2^1} + \frac{j_{l+1}}{2^2} + \frac{j_{l+2}}{2^3} + \cdots + \frac{j_m}{2^{m-l+1}}. \quad (2.8)$$

So for example when we are dealing with the basis states of a computational register with  $n$ -bits, if we write  $|j\rangle_n = |j_0 j_1 \cdots j_{n-1}\rangle$  we mean the integer represented in binary by such basis state, while if we write  $|0.j_0 j_1 \cdots j_{n-1}\rangle$  we are representing the  $n$ -bit binary fraction representation of the same basis state.

**Example 2.1.** For concreteness, let us consider the  $2^3$  basis states of a 3-bit register,

$$|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle.$$

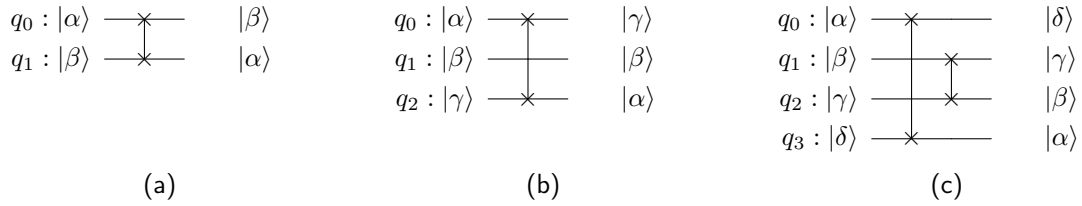
According to the rule in Eq.2.7, these states might encode the integers

$$|0\rangle_3, |1\rangle_3, |2\rangle_3, |3\rangle_3, |4\rangle_3, |5\rangle_3, |6\rangle_3, |7\rangle_3,$$

while if we follow Eq.2.8, they represent

$$|0.0\rangle_3, |0.125\rangle_3, |0.25\rangle_3, |0.375\rangle_3, |0.5\rangle_3, |0.625\rangle_3, |0.75\rangle_3, |0.875\rangle_3.$$

Sometimes we may want to change this binary representations by reading bits in the reverse order. The permutation operations needed to reverse the basis can be accomplished by cascades of *SWAP* gates as

Figure 2.3: Examples of basis reversion using *SWAP* operations.

exemplified in Fig.2.3. Next, we present how operations such as these are described.

### Operations and circuits

In the circuit model, computation is achieved with algorithms that sequentially manipulate the qubit states with discrete operations known as quantum logic gates. These are single-qubit operations and two-qubit entangling operations such as the controlled-NOT,  $cX$ . All the gates acting on a quantum computer must be described by unitary operators, as these preserve the norm of the state vectors they act upon and thus are consistent with the conservation of probability amplitudes of physical states.

**Definition 2.2.** An unitary operator is a linear operator  $U : H \rightarrow H$  on a Hilbert space which satisfies the condition  $UU^\dagger = U^\dagger U = \mathbf{1}$ .

Sequential composition of these unitary gates preserves unitarity. The complete algorithm can then be described as a single unitary transformation which, having an inverse, allows computation to be reversible.

**Theorem 2.3.** An unitary operator over a complex vector space has complex eigenvalues with modulus 1.

*Proof.* \_\_\_\_\_

From the eigenvalue equation,  $U|u\rangle = u|u\rangle$ , we get  $\langle u|U^\dagger U|u\rangle = uu^* \langle u|u\rangle$  and it follows  $|u| = 1$ .

□

**Theorem 2.4.** The action of an unitary operator  $U$  on an orthonormal and complete basis  $\{|a'\rangle\}$  spanning some Hilbert space produces another complete and orthonormal basis  $\{|b'\rangle\}$ .

*Proof.* \_\_\_\_\_

Let us construct the operator and prove it verifies this property. We define

$$U = \sum_j |b^{(j)}\rangle \langle a^{(j)}|. \quad (2.9)$$

Due to the orthonormality of  $\{|a'\rangle\}$ , the basis transformation follows directly,

$$U|a^{(k)}\rangle = |b^{(k)}\rangle, \quad (2.10)$$

and  $U$  is actually unitary, as can be seen by,

$$U^\dagger U = \sum_j \sum_k |a^{(k)}\rangle \langle b^{(k)}|b^{(j)}\rangle \langle a^{(j)}| = \sum_j |a^{(j)}\rangle \langle a^{(j)}| = \mathbf{1}, \quad (2.11)$$

and similarly for  $UU^\dagger = \mathbf{1}$ .

□



The most general form of a single-qubit unitary can be written as

$$U(\theta, \phi, \lambda) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -e^{i\lambda} \sin\left(\frac{\theta}{2}\right) \\ e^{i\phi} \sin\left(\frac{\theta}{2}\right) & e^{i(\lambda+\phi)} \cos\left(\frac{\theta}{2}\right) \end{pmatrix} \quad (2.12)$$

with  $0 \leq \theta \leq \pi$ ,  $0 \leq \phi \leq 2\pi$  and  $0 \leq \lambda < \pi$ . It can easily be checked that this general  $2 \times 2$  unitary matrix can be decomposed as the sequence

$$U(\theta, \phi, \lambda) = R_z(\phi) R_y(\theta) R_z(\lambda), \quad (2.13)$$

known as the *ZYZ decomposition*. A proof is provided by [33].

Tab.2.1 shows some basic quantum gates. Controlled operations will be symbolized starting with the character  $c$  and the qubits in which they act will come as subscripts, first control then target. When working out derivations, we will sometimes use  $R_z(\theta)$  as

$$R_z(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}, \quad (2.14)$$

in which case  $R_z(\theta)(\alpha|0\rangle + \beta|1\rangle) = (\alpha|0\rangle + e^{i\theta}\beta|1\rangle)$ , while on other occasions we will prefer to define

$$R_z(\theta) = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}, \quad (2.15)$$

which yields  $R_z(\theta)(\alpha|0\rangle + \beta|1\rangle) = e^{-i\theta/2}(\alpha|0\rangle + e^{i\theta}\beta|1\rangle)$ . The two definitions are equivalent since the transformed states will only differ by a global phase which is not detectable upon measurement. That is, the probability amplitudes of measuring any basis state are unchanged by a global phase. This is still true when we multiply  $R_z(\theta)$  with other operations on the same qubit or on the tensor product with the other qubits. Global phases propagate and are not observable.

To represent the applied sequence of gates of an algorithm, circuit diagrams are drawn with as many rows as there are qubits, where each qubit subspace is represented by one of these rows (e.g. as in Fig.2.3). Single-qubit gates are placed on the row corresponding to the qubit they act upon, and 2-qubit gates connect the rows of the qubits being targeted. The state of the computer evolves from left to right in the diagram. Note that the order in which the operations are represented in the diagram is the reverse of the order of the mathematical representation. For instance, the mathematical expression for the operations in Fig.2.3 (c) is written as

$$|\delta\gamma\beta\alpha\rangle = SWAP_{12} SWAP_{03} |\alpha\beta\gamma\delta\rangle, \quad (2.16)$$

although in this case getting the order correctly is irrelevant since the operations commute. Note also that in general there is no connection between the order of the tensor product of the Hilbert space of the qubits and the circuit rows, the only thing that matters is that the label of the qubit is properly identified in the figure. If not explicitly stated otherwise, in this dissertation we will adopt the following conventions: the basis of the computer will always be written with the numbering of the qubits increasing from left to right, i.e.  $|q_0, q_1, q_2, \dots, q_{n-1}\rangle$ , and the diagrams will always be drawn with labels increasing from top to bottom, such that we can reconstruct the tensor product by going up-down in a diagram  $|q_0\rangle \otimes |q_1\rangle \otimes \dots \otimes |q_{n-1}\rangle$ .

Provided with some diagram, it is straightforward to find its inverse. We can think about any quantum circuit  $\mathcal{U}_{2^n \times 2^n}$  as a sequence of operators, each one a column in the diagram,  $\mathcal{U} = \mathcal{O}_1 \mathcal{O}_2 \dots \mathcal{O}_m$ ; to transpose it we just do the usual  $\mathcal{U}^\dagger = \mathcal{O}_m^\dagger \dots \mathcal{O}_2^\dagger \mathcal{O}_1^\dagger$ ; now, each of these column operators  $\mathcal{O}_i$  is built from a tensor product  $\mathcal{O}_i = \mathcal{Q}_1 \otimes \mathcal{Q}_2 \otimes \dots \otimes \mathcal{Q}_n$ , and for these we have  $\mathcal{O}_i^\dagger = \mathcal{Q}_1^\dagger \otimes \mathcal{Q}_2^\dagger \otimes \dots \otimes \mathcal{Q}_n^\dagger$ . Therefore, to draw the

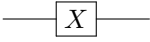
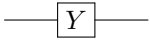



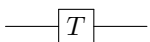
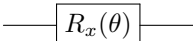
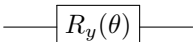
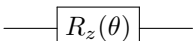
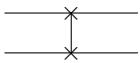
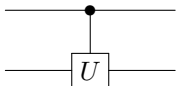
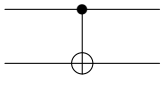
| Symbol        | Matrix   | Circuit   | Description  |
|---------------|--|---|--|
| $X$           | $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$   |    | Rotates the Bloch sphere $\pi$ radians around the $x$ -axis. Maps $ 0\rangle \rightarrow  1\rangle$ and $ 1\rangle \rightarrow  0\rangle$ (bit-flip).    |
| $Y$           | $\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$  |    | Rotates the Bloch sphere $\pi$ radians around the $y$ -axis. Maps $ 0\rangle \rightarrow i 1\rangle$ and $ 1\rangle \rightarrow -i 0\rangle$ .           |
| $Z$           | $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$  |    | Rotates the Bloch sphere $\pi$ radians around the $z$ -axis. Maps $ 0\rangle \rightarrow  0\rangle$ and $ 1\rangle \rightarrow - 1\rangle$ (phase-flip). |
| $H$           | $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$   |    | Rotates the Bloch sphere $\pi$ radians around the $x$ -axis, and $\pi/2$ radians around the $y$ -axis.   |
| $S$           | $\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$   |    | Maps $ 0\rangle \rightarrow  0\rangle$ and $ 1\rangle \rightarrow i 1\rangle$ .  |
| $T$           | $\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$  |  | Maps $ 0\rangle \rightarrow  0\rangle$ and $ 1\rangle \rightarrow e^{i\pi/4} 1\rangle$ .   |
| $R_x(\theta)$ | $\begin{bmatrix} \cos \theta/2 & -i \sin \theta/2 \\ -i \sin \theta/2 & \cos \theta/2 \end{bmatrix}$                 |  | Rotates the Bloch sphere $\theta$ radians around the $x$ -axis.  |
| $R_y(\theta)$ | $\begin{bmatrix} \cos \theta/2 & -\sin \theta/2 \\ \sin \theta/2 & \cos \theta/2 \end{bmatrix}$                      |  | Rotates the Bloch sphere $\theta$ radians around the $y$ -axis.  |
| $R_z(\theta)$ | $\begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}$  |  | Rotates the Bloch sphere $\theta$ radians around the $z$ -axis.  |
| $SWAP$        | $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$                     |  | Exchanges the states of two qubits.  |
| $cU$          | $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{11} & u_{12} \\ 0 & 0 & u_{21} & u_{22} \end{bmatrix}$ |  | Applies the unitarity $U$ to the bottom qubit if the top one is in state $ 1\rangle$ .   |
| $cX$          | $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$                     |  | Applies the $X$ gate to the bottom qubit if the top one is in state $ 1\rangle$ .  |

Table 2.1: Some basic quantum gates.

circuit for  $U^\dagger$  one just mirrors the diagram horizontally and conjugate-transposes each gate.

### Measurement

Measurement is the only non-invertible and irreversible action that the program ever performs on the quantum computer, and it is the process by which we extract information from the final state obtained after the algorithm has applied all the gates. To measure a qubit, the computer applies some interaction which couples to its wavefunction producing an outcome that can be either 0 or 1. If the final state of the qubits is given by Eq.2.6, after the action of the measurement gate  $M_n$ , we obtain the integer  $x$  with probability  $p = |\alpha_x|^2$ . The wavefunction has collapsed to the state  $|x\rangle_n$ .

Measurement, or the collapse of the wave-function, is arguably the most mysterious piece in the QM puzzle, and one which we don't yet fully understand. Being so, near term quantum computers also offer an opportunity to perform new tests on quantum theory, in particular to try to investigate peculiarities on measurement versus unitary evolution for entangled states with many parties.

**Example 2.5.** The following is a simple example to remind of how to compute measurements on subspaces of quantum systems. Consider the following wavefunction for a system divided into two orthogonal subspaces

$$|\Psi\rangle = \alpha |a_1\rangle |b_1\rangle + \beta |a_2\rangle |b_2\rangle + \gamma |a_1\rangle |b_2\rangle \quad (2.17)$$

where  $|a_i\rangle$  and  $|b_i\rangle$  are some of the eigenkets for the respective subsystems. The probability amplitude for a measurement of  $|\Psi\rangle$  along  $|a_1\rangle$  is given by

$$\begin{aligned} |(\langle a_1| \langle b_1| + \langle a_1| \langle b_2|) |\Psi\rangle|^2 &= |\alpha \langle a_1|a_1\rangle \langle b_1|b_1\rangle + \beta \langle a_1|a_2\rangle \langle b_1|b_2\rangle + \gamma \langle a_1|a_1\rangle \langle b_1|b_2\rangle \\ &\quad + \alpha \langle a_1|a_1\rangle \langle b_2|b_1\rangle + \beta \langle a_1|a_2\rangle \langle b_2|b_2\rangle + \gamma \langle a_1|a_1\rangle \langle b_2|b_2\rangle|^2 \quad (2.18) \\ &= |\alpha + \gamma|^2 \quad (2.19) \end{aligned}$$

### Universal Quantum Computing

Coding a quantum algorithm is an exercise in finding an unitary transformation that can be used to solve the problem under question. Sometimes it is possible to find *exact* decompositions of arbitrary unitary operations using some finite set of elementary gates. We have already encountered one such general decomposition to obtain any 1-qubit unitarity in Eq.2.13. What about circuits for general 2-qubit unitary operations? These can be implemented with a sequence of at most three entangling gates and some single qubit rotations. For instance, a method was provided by [34] based on the method given by [35] to decompose any two-qubit operation.

However, for a general number of qubits, exact decompositions are difficult to find. Nevertheless, we can find *approximate* decompositions, and the accuracy can be made arbitrary at the cost of increasing the number of applied operations. The framework of universal quantum computation is the idea that a quantum circuit can be built out of gates from a finite set to approximate to an arbitrary accuracy *any* unitary operation [13]. An universal quantum computer can perfectly simulate any Turing machine.

Different sets of gates can be universal for quantum computation. A famous example is the Clifford+T gate set consisting of the  $cX$ ,  $H$ ,  $S$  and  $T$  operations. A proof is provided in the classic textbook by Nielsen and Chuang [36]. Furthermore, approximating a given unitarity to accuracy  $\epsilon$  can be done in polynomial time in  $\mathcal{O}(\log^2(1/\epsilon))$ , as shown by the Solovay-Kitaev theorem. A proof is also given in [36]. Later, we will introduce another universal gate set.

## 2.3 Quantum Fourier transform

The Fourier transform was introduced in the beginning of the 19th century by Joseph Fourier while he was seeking for solutions to the heat equation [37]. Since then, it has found a wide range of applications in mathematics and the physical sciences, being employed over continuous or discrete valued functions, scalar or vector valued, periodic or not [38]. For context, we assume some familiarity with this transformation before introducing its quantum version, nevertheless we are going to build into this unitary operation in a self-contained manner. Let us start by reviewing the discrete Fourier transform.

**Definition 2.6.** Let  $|x\rangle$  be a vector on a  $N$ -dimensional complex vector space  $\mathbb{C}^N$ , with components  $\{x_j\}$ ,  $j \in [0, 1, \dots, N-1]$ . The  $N^{\text{th}}$  order *discrete Fourier transform* ( $\mathcal{DFT}$ ) is the linear transformation  $\mathcal{DFT} : \mathbb{C}^N \rightarrow \mathbb{C}^N$  taking  $|x\rangle$  to another vector  $|y\rangle$  with components obtained according to

$$y_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_k \omega_N^{kj}, \quad (2.20)$$

where  $\omega_N = \exp(i2\pi/N)$  is the  $N^{\text{th}}$  root of unity, and  $\omega_N^{kj} = \exp(i2\pi kj/N)$ . We also define the *inverse discrete Fourier transform* ( $\mathcal{DFT}^{-1}$ ) which recovers  $|x\rangle$  given  $|y\rangle$ , i.e.  $\mathcal{DFT}^{-1} \circ \mathcal{DFT} = \mathbf{1}$ ; it reads

$$x_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} y_k \omega_N^{-kj} \quad (2.21)$$

**Example 2.7.** Using Def.2.6, we obtain  $|y\rangle = (2 - 2i, 1 - i, -2, -1 - i)^T$  as the DFT of vector  $|x\rangle = (-2i, 2 - 2i, 0, 2)^T$ , with  $N = 4$ . For example,

$$y_0 = \left( (-2i) e^{i2\pi \cdot 0 \cdot 0/4} + (2 - 2i) e^{i2\pi \cdot 1 \cdot 0/4} + 0 + 2e^{i2\pi \cdot 3 \cdot 0/4} \right) / \sqrt{4} = 2 - 2i.$$

Fourier transforms can be found over the literature in several slightly different definitions. Although this is only a matter of conventions, it can be annoying when we just want to quickly look up some property or compare different results. We use definition in Eq.2.20 to match the usual convention for the quantum version, which is just the same transformation applied over the amplitudes vector of a general  $n$ -qubit quantum state. This is, we start from a general  $n$ -qubit state  $|x\rangle$  described by  $N = 2^n$  amplitude coefficients on the Hilbert space  $\mathcal{H}^{(n)}$  spanned by the computational basis  $\{|j\rangle_n\}_{j=0, \dots, 2^n-1}$ , and by applying the transformation we get to another state  $|y\rangle$  where the new amplitudes are the DFT of the former given as in Eq.2.20, i.e.

$$|x\rangle = \sum_{j=0}^{2^n-1} x_j |j\rangle_n \longrightarrow |y\rangle = \sum_{j=0}^{2^n-1} y_j |j\rangle_n. \quad (2.22)$$

**Definition 2.8.** The  $n$ -qubit *quantum Fourier transform* ( $\mathcal{QFT}$ ) is the unitary transformation  $U_{\mathcal{QFT}}$  defined to act on the computational basis states  $|j\rangle_n$  by

$$U_{\mathcal{QFT}} |j\rangle_n = \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} \omega_n^{kj} |k\rangle_n, \quad (2.23)$$

where we use  $N = 2^n$ ,  $\omega_n = \exp(i2\pi/2^n)$  and  $kj$  is the ordinary multiplication of the integers represented in binary by states  $|j\rangle$  and  $|k\rangle$ . Hence, the  $\mathcal{QFT}$  takes the computational basis to the Fourier (or frequency) basis.

The transformation rule for a general state,  $|x\rangle \rightarrow |y\rangle$ , can be directly obtained by the linearity property

of this operation,

$$U_{\mathcal{QFT}} \left( \sum_{j=0}^{2^n-1} x_j |j\rangle_n \right) = \sum_{j=0}^{2^n-1} \left( \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} x_k \omega_n^{kj} \right) |k\rangle_n, \quad (2.24)$$

and coordinate  $y_j$  can be found by taking the dot product with the  $j$ th basis vector. Besides being linear,  $U_{\mathcal{QFT}}$  is norm-preserving and  $U_{\mathcal{QFT}} |i\rangle$  is orthogonal to  $U_{\mathcal{QFT}} |j\rangle$  for  $i \neq j$ , hence the operator is unitary and a quantum circuit can be designed to implement it. Being unitary, the inverse transformation is  $U_{\mathcal{QFT}}^{-1} = U_{\mathcal{QFT}}^\dagger$ . In this way, it is helpful to think of the quantum Fourier transform as just another unitary operator which we can use to measure any ket along the frequency basis.

**Theorem 2.9.**  $U_{\mathcal{QFT}}$  is an unitary operator.

*Proof.* \_\_\_\_\_

Consider two transformed basis states  $|s\rangle$  and  $|t\rangle$ . Then,

$$\begin{aligned} \langle s|t\rangle &= \frac{1}{2^n} \sum_{u=0}^{2^n-1} \sum_{v=0}^{2^n-1} \omega_n^{tv-su} \langle u|v\rangle_n \stackrel{(a)}{=} \frac{1}{2^n} \sum_{u=0}^{2^n-1} \omega_n^{u(t-s)} \\ &\stackrel{(b)}{=} \frac{1}{2^n} \left( \frac{1-\omega_n^{(t-s)2^n}}{1-\omega_n^{(t-s)}} \right) = \frac{1}{2^n} \left( \frac{1-e^{i2\pi(t-s)}}{1-\omega_n^{(t-s)}} \right) \stackrel{(c)}{=} \frac{1}{2^n} \left( \frac{1-1}{1-\omega_n^{(t-s)}} \right) = 0 \end{aligned} \quad (2.25)$$

where at (a) we use  $\langle u|v\rangle_n = \delta_{uv}$ , at (b) we apply the geometric series sum  $\sum_{n=n_0}^N a^n = \frac{a^{n_0}-a^{N+1}}{1-a}$ , and at (c) we just remember  $t-s \in \mathbb{Z}$ . Hence, we have  $\langle s|t\rangle_n = \delta_{st}$  implying  $U_{\mathcal{QFT}}$  is unitary. □

**Theorem 2.10.** The action of the  $n$ -qubit quantum Fourier transform  $U_{\mathcal{QFT}}$  on the computational basis states  $|j\rangle_n$  as defined in Eq.2.23 can be written in the useful product representation,

$$U_{\mathcal{QFT}} |j_0 \cdots j_{n-1}\rangle = \frac{(|0\rangle + e^{i2\pi 0.j_{n-1}} |1\rangle) (|0\rangle + e^{i2\pi 0.j_{n-2}j_{n-1}} |1\rangle) \cdots (|0\rangle + e^{i2\pi 0.j_0j_1 \cdots j_{n-1}} |1\rangle)}{2^{n/2}}, \quad (2.26)$$

where we use the binary representation  $j \equiv j_0j_1 \cdots j_{n-1} = \sum_{q=0}^{n-1} j_q 2^{(n-1)-q}$  introduced in Eq.2.7 and the binary fraction notation  $0.j_lj_{l+1} \cdots j_m = \sum_{q=l}^m j_q 2^{l-q-1}$  introduced in Eq.2.8.

*Proof.* \_\_\_\_\_

We can show Eq.2.26 by carefully manipulating Eq.2.23,

$$U_{\mathcal{QFT}} |j_0 \cdots j_{n-1}\rangle = \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} \omega_n^{jk} |k\rangle_n \quad (2.27)$$

$$\stackrel{(a)}{=} \frac{1}{2^{n/2}} \sum_{k_0=0}^1 \cdots \sum_{k_{n-1}=0}^1 e^{i2\pi j (\sum_{q=0}^{n-1} k_q 2^{n-1-q}) / 2^n} |k_0 \cdots k_{n-1}\rangle \quad (2.28)$$

$$= \frac{1}{2^{n/2}} \sum_{k_0=0}^1 \cdots \sum_{k_{n-1}=0}^1 \bigotimes_{q=0}^{n-1} e^{i2\pi j k_q 2^{-q-1}} |k_q\rangle \quad (2.29)$$

$$= \frac{1}{2^{n/2}} \bigotimes_{q=0}^{n-1} \left( \sum_{k_q=0}^1 e^{i2\pi j k_q 2^{-q-1}} |k_q\rangle \right) \quad (2.30)$$

$$= \frac{1}{2^{n/2}} \bigotimes_{q=0}^{n-1} (|0\rangle + e^{i2\pi j 2^{-q-1}} |1\rangle) \quad (2.31)$$

$$\stackrel{(b)}{=} \frac{1}{2^{n/2}} \bigotimes_{q=0}^{n-1} (|0\rangle + e^{i2\pi (\sum_{r=0}^{n-1} j_r 2^{n-1-r}) 2^{-q-1}} |1\rangle) \quad (2.32)$$

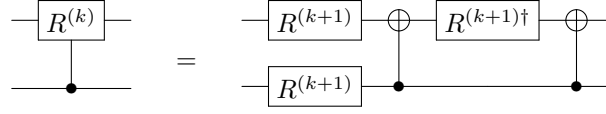


Figure 2.4: Decomposition of the controlled- $R^{(k)}$  gate in terms of single qubit gates and  $cX$ 's. The  $R^{(k)}$  operation should be available in the computer gate toolbox, or else one needs to decompose it in terms of the available gate set.

$$= \frac{1}{2^{n/2}} \bigotimes_{q=0}^{n-1} \left( |0\rangle + e^{i2\pi \sum_{r=0}^{n-1} j_r 2^{n-2-r-q}} |1\rangle \right) \quad (2.33)$$

$$= \frac{1}{2^{n/2}} \bigotimes_{q=0}^{n-1} \left( |0\rangle + e^{i2\pi \left( \sum_{r=0}^{n-2-q} j_r 2^{n-2-q-r} + \sum_{r=n-1-q}^{n-1} j_r 2^{n-2-q-r} \right)} |1\rangle \right) \quad (2.34)$$

$$\stackrel{(c)}{=} \frac{1}{2^{n/2}} \bigotimes_{q=0}^{n-1} \left( |0\rangle + e^{i2\pi \sum_{r=n-1-q}^{n-1} j_r 2^{-r+(n-1-q)-1}} |1\rangle \right) \quad (2.35)$$

$$\stackrel{(d)}{=} \frac{1}{2^{n/2}} \bigotimes_{q=0}^{n-1} \left( |0\rangle + e^{i2\pi 0.j_{n-1-q} \dots j_{n-1}} |1\rangle \right) \quad (2.36)$$

$$= \frac{\left( |0\rangle + e^{i2\pi 0.j_{n-1}} |1\rangle \right) \left( |0\rangle + e^{i2\pi 0.j_{n-2}j_{n-1}} |1\rangle \right) \dots \left( |0\rangle + e^{i2\pi 0.j_0j_1 \dots j_{n-1}} |1\rangle \right)}{2^{n/2}} \quad (2.37)$$

where at (a) we substitute the expression for  $\omega_n$  and the binary expansion for  $k = \sum_{q=0}^{n-1} k_q 2^{(n-1)-q}$ ; at (b) we perform the same expansion for  $j$ ; at (c) we notice that  $\sum_{r=0}^{n-2-q} j_r 2^{n-2-q-r}$  is a non-negative integer since here  $r \leq n-2-q$ , and can be removed from the exponential on account that  $\exp(ij'2\pi) = 1 \forall j' \in \{0, 1, 2, \dots\}$ ; and at (d) we again use the binary fraction notation. □

We have already proven that a quantum circuit can be constructed to implement  $U_{QFT}$ , on account of this transformation being unitary. It turns out a design entirely built out of 1-qubit and 2-qubit gates emerges easily from the product representation Eq.2.26 and is shown in Fig.2.5. To derive it, we introduce the single-qubit gate

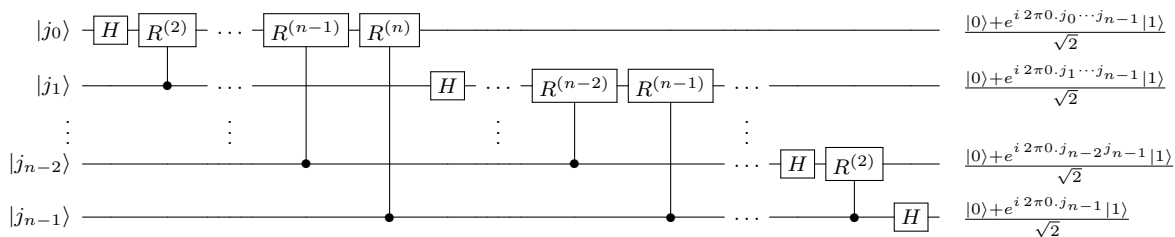
$$R^{(k)} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i2\pi/2^k} \end{pmatrix} \quad (2.38)$$

which adds a phase to state  $|1\rangle$  only, i.e.  $R^{(k)}|0\rangle = |0\rangle$  and  $R^{(k)}|1\rangle = e^{i2\pi/2^k}|1\rangle$ . We will need to apply this operation controlled by the  $|1\rangle$  state of another qubit. In that case, the controlled- $R^{(k)}$  gate is actually symmetric (exchanging target and control qubits results on the same operation) and the phase is added only if both qubits are in state  $|1\rangle$ . The decomposition of  $cR^{(k)}$  is shown in Fig.2.4, where the representation, despite being common in the literature, also pictures the artificial asymmetry just explained.

Let us walk through the circuit and check how it computes the quantum Fourier transform of the input state  $|j_0 \dots j_{n-1}\rangle$ . Since  $e^{i2\pi 0.j_0} = 1$  if  $j_0 = 0$  and  $-1$  if  $j_0 = 1$  we see that the action of the Hadamard gate on the first qubit can be written as

$$H_0 |j_0 \dots j_{n-1}\rangle = \frac{1}{2^{1/2}} \left( |0\rangle + e^{i2\pi 0.j_0} |1\rangle \right) |j_1 \dots j_{n-1}\rangle. \quad (2.39)$$

Now comes a series of controlled operations on this qubit. To understand the state this produces, let us first look at the result of applying the first of these gates,  $cR^{(2)}$ ,

Figure 2.5: Circuit for the quantum Fourier transform without swapping states in the final basis,  $U_{QFT}$ .

$$cR_{10}^{(2)} H_0 |j_0 \dots j_{n-1}\rangle = \frac{1}{2^{1/2}} (|0\rangle + e^{i2\pi 0 \cdot j_0 j_1} |1\rangle) |j_1 \dots j_{n-1}\rangle. \quad (2.40)$$

This state is produced because, as we have seen,  $R^{(k)}$  adds a phase to state  $|1\rangle$  leaving  $|0\rangle$  unaltered, and with this circuit we carefully linked the phase angle that  $R^{(k)}$  adds to the qubit which controls actuation. It is now easy to see that the action of each of the successive controlled- $R^{(k)}$  gates is to add a progressively smaller phase to the  $|1\rangle$  basis state of the first qubit. This nicely adds on the overall phase angle as another bit to its binary fraction expansion in agreement to the state stored by  $|j_k\rangle$ . The final state of this qubit will be

$$cR_{n-1,0}^{(n)} \dots cR_{10}^{(2)} H_{j_1} |j_0 \dots j_{n-1}\rangle = \frac{1}{2^{1/2}} (|0\rangle + e^{i2\pi 0 \cdot j_0 j_1 \dots j_{n-1}} |1\rangle) |j_0 \dots j_{n-1}\rangle. \quad (2.41)$$

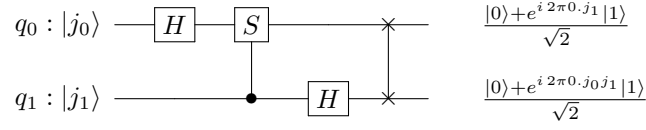
The same kind of operations are applied to the remaining qubits, with the string of controlled- $R^{(k)}$  operations on the  $i$ th qubit always starting with  $cR_{i+1,i}^{(2)}$  and being limited by the number of  $n-1-i$  qubits below. Proceeding this way for each qubit, we can now understand that the final state of all the qubits will be

$$\frac{1}{2^{n/2}} (|0\rangle + e^{i2\pi 0 \cdot j_0 j_1 \dots j_{n-1}} |1\rangle) (|0\rangle + e^{i2\pi 0 \cdot j_1 \dots j_{n-1}} |1\rangle) \dots (|0\rangle + e^{i2\pi 0 \cdot j_{n-1}} |1\rangle) \quad (2.42)$$

This is almost the same product representation we showed in Eq.2.26, however the product is reversed, such that we have the Fourier transformed amplitudes of the input state stored in reverse order on the final state, from qubit  $n-1$  to qubit 0. Depending on what we want to do, we can now reverse this order by applying a series of *SWAP* operations to obtain the final state exactly as derived in Eq.2.26, or else, if the *QFT* is a sub-module of some algorithm we are building, we can skip the reordering and just code it with this order in mind.

Finally, note that coding the circuit for the inverse transformation,  $QFT^\dagger$ , is as simple as following what was described in the previous section: one just mirrors the circuit in Fig.2.5 horizontally and conjugate-transposes each gate.

**Example 2.11.** Let us pause for a moment to build some intuition by looking at the simple example of the 2-qubit quantum Fourier transform,  $U_{QFT}^{(2)}$ . The circuit is



and the unitary matrix for this operation becomes

$$U_{QFT}^{(2)} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}. \quad (2.43)$$

If we feed the circuit with the initial state  $|\psi_{in}\rangle = (\sqrt{1/2}, 0, 1/2, 1/2)^T$  in the basis  $|q_0q_1\rangle \in \{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ , we get  $|\psi_{out}\rangle = \frac{1}{4} (2 + \sqrt{2}, \sqrt{2} - 1 - i, \sqrt{2}, \sqrt{2} - 1 + i)^T$ , which is also the DFT of  $|\psi_{in}\rangle$ , as expected. We could also remove the SWAP gate to perform  $U_{QFT}^{(2)}$  and read the final state in reverse order to obtain the same result. Of course, we can not measure the coefficients of the wave function, but only the modulus square of those. Therefore, if we perform this experiment a large number of times, each time measuring the final state, we obtain the probability vector  $PMF \approx (72.86\%, 7.32\%, 12.50\%, 7.32\%)^T$ , in the same basis. We will come back to this example to experiment on a real device later.

We should also point out there is a further simplification of the circuit in Fig.2.5 which only uses 1-qubit gates. It works by measuring each qubit after the action of each hadamard gate and then replacing the  $cR^{(k)}$  gates by  $R^{(k)}$  gates that act or not depending on the prior measurement (see [39, 40] for further details).

The quantum Fourier transform is an example of a quantum algorithm to compute the  $N^{th}$  order discrete Fourier transform which has a lower time complexity than any classical algorithm performing the same task. We can see that by counting the total number of applied gates in Fig.2.5. On each qubit  $i$  we apply  $n - i$  operations, therefore the algorithm employs  $\sum_{i=0}^{n-1} n - i = n(n + 1)/2$  gates in total. This is really the core of the algorithm, the final SWAP operations are not essential as they only serve the purpose of adjusting the result to our preferred basis. Hence, the derived circuit computes the  $DFT$  on  $N = 2^n$  elements, i.e. the  $QFT$ , with  $\mathcal{O}(n^2)$  time complexity and  $\mathcal{O}(\log n)$  spatial complexity.

If we now look at Eq.2.20 we have  $\sim N = 2^n$  complex numbers to multiply and add for each final component of the computed DFT. Since the final vector will have  $2^n$  coordinates, this amounts to  $N^2 = 4^n$  complex operations, hence the time complexity of the  $DFT$  is exponential,  $\mathcal{O}(4^n)$ . One of the most important computational inventions of the last century is the *fast Fourier transform (FFT)*, a celebrated classical algorithm to speed up the task of computing the  $DFT$  on  $N = 2^n$  elements, achieving a time complexity of  $\mathcal{O}(n2^n)$  [41].

Hence, the  $QFT$  is exponentially faster than the best known classical algorithm to compute the same unitary operation. However, there is a catch. We have no way of learning the Fourier coefficients after running the quantum procedure, as the amplitudes of the wave function are not measurable. We can however, measure the modulus square of those coefficients. This needs to be done probabilistically, which means we need to run the circuit several times to let measurement probabilities converge to proper accuracy. Besides this unfortunate news, preparing the initial state to be Fourier transformed might also be a complex procedure many times. Therefore, the algorithm can not be used to speed up the task of computing the  $DFT$  of classical data. If we were hoping to use the  $QFT$  to process data as it is usually done in many real-world applications with the  $FFT$ , this does not seem the way to go.



Nevertheless, the quantum Fourier transform can be made useful when we only need partial information about the final state, as is usual in so many quantum algorithms<sup>3</sup>. This is the case in several important applications enabled by the phase estimation algorithm (PEA), which uses the quantum Fourier transform as a sub-routine. We introduce this algorithm in the next section.

## 2.4 Phase estimation algorithm

The phase estimation algorithm (PEA) is a general procedure of central importance to quantum information and computation. It relates and provides a solution to important problems such as estimating eigenvalues [42, 43], quantum systems simulation [44, 45, 46, 47, 48], order-finding [36], period-finding and factoring [16, 36], clock synchronization [49], and quantum non-demolition [50], to name a few.

The algorithm allows us to estimate the eigenvalues of a unitary operator  $U$ . One might think that the process is to somehow compute the characteristic equation numerically, in analogy with what is performed in classical computers. This is not the case. The PEA offers a way to get to our much-desired eigenvalues, but there is a sufficiently labored route to go. To get to the answer what we are going to do is to actually simulate the action of  $U$  in the state of our quantum system in such a way that it is then possible to measure its spectrum.

To be able to apply the PEA, we need to already have designed a quantum circuit implementation of the unitarity  $U$ . Although in principle it is always possible to implement  $U$  or approximate it to high accuracy, we might find it a challenging task sometimes. Recall that from theorem 2.3 we can write the eigenvalue of a unitary operator  $U$  associated to some eigenket  $|u\rangle$  as  $e^{i2\pi\varphi_u}$ , where  $\varphi_u \in [0, 1) \in \mathbb{R}/\mathbb{Z}$  is a real number modulo 1. With the PEA we can estimate  $\varphi_u$  under the circumstances we are about to describe.

The construction of the algorithm divides the qubits of the computer into two quantum registers. The *first register* (F) is composed of  $f$  qubits which are initialized in state  $|0\rangle_f$  and will end in the final state from which we read the phase estimation  $\tilde{\varphi}_u$ . The *second register* (S) is made up of as many  $s$  qubits as necessary to implement  $U$  and is initialized in a  $s$ -qubit eigenstate  $|u\rangle$ . Later we will look at what is the behaviour of the PEA when we start the second register from a general state. As in the case of  $U$  one should also have a circuit which is able to prepare the initial state.

When we can initialize the second register at an eigenstate, the procedure represented in Fig.2.6 is divided as follows:

1. Initialize the quantum registers to the state  $|0\rangle_f |u\rangle$ .
2. On the first register, create an equal superposition of all the basis states  $\{|j\rangle_f\}$ ,  $j = 0, \dots, 2^f - 1$ , by applying the hadamard gate to each one

$$H_{f-1} \cdots H_0 |0\rangle_f = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)^{\otimes f} = \frac{1}{2^{f/2}} \sum_{j=0}^{2^f-1} |j\rangle_f. \quad (2.44)$$

3. On the second register, apply a sequence of the unitarity  $U$  raised to increasing powers of 2, with each gate in the sequence being controlled by one of the qubits of the first register as represented on Fig.2.6. This produces the final product state on the first register

$$\frac{1}{2^{f/2}} \left( |0\rangle + e^{i2\pi 2^0 \varphi_u} |1\rangle \right) \left( |0\rangle + e^{i2\pi 2^1 \varphi_u} |1\rangle \right) \cdots \left( |0\rangle + e^{i2\pi 2^{f-1} \varphi_u} |1\rangle \right) \quad (2.45)$$

4. Apply the inverse quantum Fourier transform without the final SWAP steps,  $U_{\mathcal{QFT}}^\dagger$ , on the first register.

---

<sup>3</sup>Note that obtaining partial information efficiently is also where the power of the Deutsch's algorithm comes from.

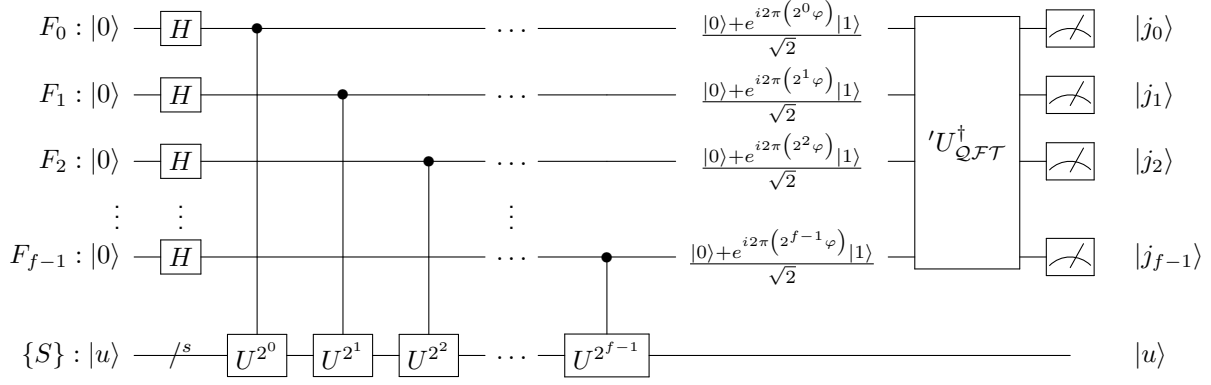


Figure 2.6: Schematic of the phase estimation algorithm.

5. Measure the final state of the first register to learn

$$\tilde{\varphi}_u = 0.j_0j_1 \cdots j_{f-1} = \frac{j_0}{2^1} + \frac{j_1}{2^2} + \cdots + \frac{j_{f-1}}{2^f}, \quad (2.46)$$

an  $f$ -bit binary fraction approximation to the actual  $\varphi_u$  read from the first register.

Let us now describe these steps in greater depth. By now we are already familiarized with the equal superposition of basis states in step 2. Now looking at step 3, it is easy to see why the intermediate state of the first register is the product state in Eq.2.45. Consider for example a first register with only one qubit. The action of the controlled unitarity raised to  $2^j$  yields

$$(cU_{F_0S})^{2^j} \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) |u\rangle = \frac{1}{\sqrt{2}} (|0\rangle |u\rangle + e^{i2\pi\varphi_u 2^j} |1\rangle |u\rangle), \quad (2.47)$$

and this generalizes directly to the mentioned product state when we add more qubits and controlled unitarities as pictured in the circuit. At the end of applying this sequence of  $cU^{2^j}$ 's, the qubits of the second register remain in state  $|u\rangle$ , which is not entangled with the first register. Therefore, let us disregard the second register and move on working expression 2.45 for the first register. We get

$$\frac{1}{2^{f/2}} \bigotimes_{q=0}^{f-1} (|0\rangle + e^{i2\pi\varphi_u 2^q} |1\rangle) = \frac{1}{2^{f/2}} \bigotimes_{q=0}^{f-1} \left( \sum_{k_q=0}^1 e^{i2\pi\varphi_u 2^q k_q} |k_q\rangle \right) \quad (2.48)$$

$$= \frac{1}{2^{f/2}} \sum_{k_0=0}^1 \cdots \sum_{k_{f-1}=0}^1 \bigotimes_{q=0}^{f-1} e^{i2\pi\varphi_u 2^q k_q} |k_q\rangle \quad (2.49)$$

$$= \frac{1}{2^{f/2}} \sum_{k_0=0}^1 \cdots \sum_{k_{f-1}=0}^1 e^{i2\pi\varphi_u \sum_{q=0}^{f-1} 2^q k_q} |k_0 \cdots k_{f-1}\rangle \quad (2.50)$$

$$\stackrel{(a)}{=} \frac{1}{2^{f/2}} \sum_{\underline{k}=0}^{2^f-1} e^{i2\pi\varphi_u \underline{k}} \left| \underline{k} \right\rangle_f \quad (2.51)$$

$$\stackrel{(b)}{=} \frac{1}{2^{f/2}} \sum_{\underline{k}=0}^{2^f-1} \omega_f^{\varphi'_u \underline{k}} \left| \underline{k} \right\rangle_f \quad (2.52)$$

where at (a) we identify  $\underline{k} = \sum_{q=0}^{f-1} 2^q k_q = 2^0 k_0 + 2^1 k_1 + \cdots + 2^{f-1} k_{f-1}$  as the integer represented by a binary expansion in the reverse order from what was defined in Eq.2.7, and at (b) the  $2^f$ -th root of unity  $\omega_f = \exp(i2\pi/2^f)$ , and  $\varphi'_u \equiv 2^f \varphi_u$ .

We could now apply to 2.52 a cascade of SWAP operations (Fig.2.3) to reverse the states of all the qubits.

We would write the result as

$$\frac{1}{2^{f/2}} \sum_{k=0}^{2^f-1} \omega_f^{\varphi'_u k} |k\rangle_f \quad (2.53)$$

and comparing this with Eq.2.23 it is clear that the inverse Fourier transform of this state will give us  $|\varphi'_u\rangle_f$  if the integer  $\varphi'_u$  can be expressed exactly in the computational basis. Since what we perform in the circuit represented in Fig.2.6 is  $U_{QFT}^\dagger$ , we don't need to SWAP the state obtained in 2.52 to get to the same result. Therefore, if  $\varphi_u$  can be expressed exactly in  $f$  bits,  $0.j_0j_1 \dots j_{f-1} = \frac{j_0}{2^1} + \frac{j_1}{2^2} + \dots + \frac{j_{f-1}}{2^f}$ , the output state of the first register is  $\varphi'_u = \varphi_u 2^f = j_0j_1 \dots j_{f-1}$ , a single basis state perfectly encoding the binary fraction expansion of  $\varphi_u$ . A single measurement on F gives us the correct value with probability 1.

To understand what happens when  $\varphi_u$  is any number in  $[0, 1[$ , which more often than not can't be exactly determined in  $f$  bits, let us explicitly compute the inverse Fourier transform of Eq.2.53 to obtain the output state on the first register,

$$|F_{out}\rangle = U_{QFT}^\dagger \left( \frac{1}{2^{f/2}} \sum_{k=0}^{2^f-1} \omega_f^{\varphi'_u k} |k\rangle_f \right) \quad (2.54)$$

$$= \frac{1}{2^{f/2}} \sum_{k=0}^{2^f-1} \omega_f^{\varphi'_u k} U_{QFT}^\dagger |k\rangle_f \quad (2.55)$$

$$= \frac{1}{2^{f/2}} \sum_{k=0}^{2^f-1} \omega_f^{\varphi'_u k} \left( \frac{1}{2^{f/2}} \sum_{j=0}^{2^f-1} \omega_f^{-jk} |j\rangle_f \right) \quad (2.56)$$

$$= \sum_{j=0}^{2^f-1} \left( \frac{1}{2^f} \sum_{k=0}^{2^f-1} \omega_f^{(\varphi'_u - j)k} \right) |j\rangle_f \quad (2.57)$$

$$= \sum_{j_0=0}^1 \dots \sum_{j_{f-1}=0}^1 \left( \frac{1}{2^f} \sum_{k=0}^{2^f-1} \omega_f^{2^f(\varphi_u - 0.j_0 \dots j_{f-1})k} \right) |0.j_0 \dots j_{f-1}\rangle. \quad (2.58)$$

This shows the probability of measuring the  $j$ -th phase estimation  $\tilde{\varphi}_{u,j}$  encoded by one of the  $2^f$  basis states on the first register,  $|\tilde{\varphi}_{u,j}\rangle = |0.j_0 \dots j_{f-1}\rangle$ , is

$$P(\tilde{\varphi}_{u,j}) = \left| \frac{1}{2^f} \sum_{k=0}^{2^f-1} \exp(i2\pi(\varphi_u - \tilde{\varphi}_{u,j})k) \right|^2. \quad (2.59)$$

From this expression it can be reconfirmed that if  $\varphi_u$  can be expressed exactly in  $f$  bits, then  $P = 0$  for all the  $|0.j_0 \dots j_{f-1}\rangle$ 's of the computational basis except the one encoding  $\varphi_u$ , in which case  $P = 1$ . Hence, in the cases where  $\varphi_u$  can not be written exactly in  $f$  bits, i.e. when  $\exp(i2\pi(\varphi_u - \tilde{\varphi}_{u,j})k) \neq 1$  for any  $|\tilde{\varphi}_{u,j}\rangle$ , we can employ the geometric series sum  $\sum_{k=0}^{N-1} a^k = \frac{1-a^N}{1-a}$  to get the general expression

$$P(\tilde{\varphi}_{u,j}) = \begin{cases} 1 & , \varphi_u - \tilde{\varphi}_{u,j} = 0 \\ \frac{1}{2^{2f}} \left| \frac{1 - \exp(i2\pi 2^f(\varphi_u - \tilde{\varphi}_{u,j}))}{1 - \exp(i2\pi(\varphi_u - \tilde{\varphi}_{u,j}))} \right|^2 = \frac{1}{2^{2f}} \frac{\sin^2(2^f \pi(\varphi_u - \tilde{\varphi}_{u,j}))}{\sin^2(\pi(\varphi_u - \tilde{\varphi}_{u,j}))} & , \varphi_u - \tilde{\varphi}_{u,j} \neq 0 \end{cases} \quad (2.60)$$

We have just arrived at an important checkpoint. Eq.2.60 describes exactly what comes out of our experimental measurements, which is everything we see from the quantum computation. From now on we just need to classically post-process this probability mass function (PMF) to extract the relevant information

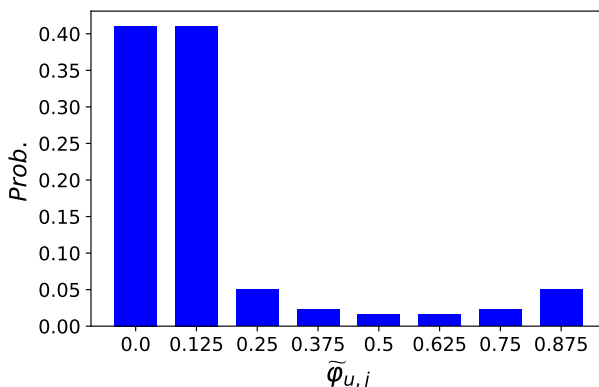
we are looking for. It is thus the time to ask what should we consider as the final computation  $\hat{\varphi}_u$  of the full phase estimation procedure. There are two options:

- **Rule 1:** the phase estimation  $\tilde{\varphi}_{u,j}$  encoded by the basis state corresponding to the maximum probability peak of the PMF.
- **Rule 2:** the mean phase direction of the PMF calculated after decoding the basis states to the decimal basis.

Let us anticipate what we will see to prepare the stage for the coming derivations. Turns out that with a sufficient number of measurements, rule 1 always returns the best possible basis state  $\tilde{\varphi}_{u,b}$  (the one which is closest to the real phase  $\varphi_u$ ) with probability  $P(\tilde{\varphi}_{u,b}) \geq 4/\pi^2 > 0.40$ . Therefore, after running the PEA circuit multiple times to perform a sufficient number of measurements, one could choose the most frequently measured basis state as the determined phase and this choice would have an error bounded by  $|\epsilon| = |\varphi_u - \hat{\varphi}_u| \leq 2^{-(f+1)}$ .

On the other hand, not surprisingly the shape of distribution carries information that we can harvest instead of ignore, and using rule 2 we conjecture it is always possible to better localize  $\varphi_u$  than using rule 1.

**Example 2.12.** Consider a phase estimation circuit with  $f = 3$  first register qubits and  $\varphi_u = 0.0625$ . The measurement probabilities on the  $2^3 = 8$  first register basis states at the end of the procedure are given by Eq.2.60 and produce the probability mass function plotted below.



The absolute error of the final estimation  $\hat{\varphi}_u$  using rule 1 is  $|\epsilon_1| = |\varphi_u - \hat{\varphi}_u| = 0.0625 = 1/2^4$ , where we can choose either  $\hat{\varphi}_u = 0$  or  $\hat{\varphi}_u = 0.125$ . If we choose rule 2 we obtain  $|\epsilon_2| = 0$ .

We start by exploring rule 1. In the case that the PEA returns the basis state closest to the real phase, we can write

$$\varphi_u = \tilde{\varphi}_{u,b} + \epsilon \quad (2.61)$$

with  $|\epsilon| \leq 2^{-(f+1)}$  smaller than a half-spacing between basis states. Then, replacing this  $\varphi_u$  in Eq.2.60 and computing the probability of getting the best possible basis estimation  $\tilde{\varphi}_{u,b}$ , it follows that

$$P_{\tilde{\varphi}_{u,b}}(\epsilon) = \frac{1}{2^{2f}} \frac{\sin^2(2^f \pi \epsilon)}{\sin^2(\pi \epsilon)} \quad (2.62)$$

This expression is monotonically decreasing with  $\epsilon$  in  $0 < \epsilon \leq 2^{-(f+1)}$ , and  $\tilde{\varphi}_{u,b}$  is by definition the best possible binary decomposition, therefore the peak of the PMF in Eq.2.60 lies at  $\tilde{\varphi}_{u,b}$ . Furthermore, the

lower bound must sit at  $\epsilon = 2^{-(f+1)}$ , which gives  $P_{\tilde{\varphi}_{u,b}}(2^{-(f+1)}) = 2^{-2f} \csc^2(\pi 2^{-(f+1)}) \xrightarrow{f \rightarrow \infty} 4/\pi^2$ , i.e.  $P(\tilde{\varphi}_{u,b}) \geq 4/\pi^2 > 0.40$ . We conclude this with the understanding that we may have at most two basis states which are closest to  $\varphi_u$ , implying we may sometimes have two peaks close to each other with the same height in the PMF.

Let us now try to follow the path of calculating descriptive statistics on the obtained probability distribution. This is not usually done in the literature. To our knowledge, the following approach is original. Firstly, recognizing the circular nature of the obtained PMF, we review some basic concepts of summary statistics for circular data [51].

**Definition 2.13.** The *mean resultant vector* for the circular probability distribution  $P(\tilde{\varphi}_{u,j})$  is given by

$$\bar{R} = \sum_{j=0}^{2^f-1} P(\tilde{\varphi}_{u,j}) e^{i 2\pi \tilde{\varphi}_{u,j}} = |\bar{R}| e^{i 2\pi \hat{\varphi}_u}, \quad (2.63)$$

where  $0 \leq |\bar{R}| \leq 1$  is the *mean resultant length*, and  $0 \leq \hat{\varphi}_u \leq 1$  is the *mean phase direction*.  $\bar{R}$  is the center of mass of the distribution, and  $|\bar{R}|$  is a measure of its concentration.

**Definition 2.14.** For the purpose of comparison with the statistics for data on the line, let us define the *circular variance*,  $0 \leq V \leq 1$ , as

$$V = 1 - |\bar{R}|, \quad (2.64)$$

and the *circular standard phase deviation*,  $v \in [0, \infty]$ ,

$$v = \frac{\sqrt{-2 \ln |\bar{R}|}}{2\pi}. \quad (2.65)$$

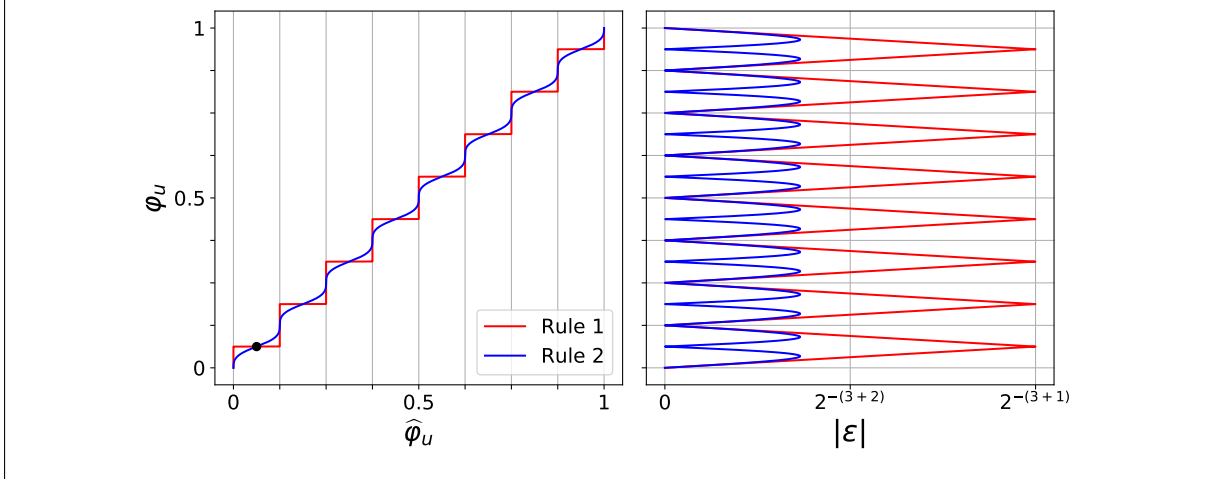
We are now ready to conjecture on the advantage of using rule 2 rather than rule 1 to classically post-process the quantum computing results.

**Conjecture 2.15.** Denote the absolute error of the final determined phase  $\hat{\varphi}_u$  as  $|\epsilon| = |\varphi_u - \hat{\varphi}_u|$ . For a given value of  $\varphi_u$ , let  $|\epsilon_1|$  and  $|\epsilon_2|$  be the errors made using rule 1 and rule 2 to evaluate  $\hat{\varphi}_u$ , respectively. Then,  $|\epsilon_2| \leq |\epsilon_1| \forall \varphi_u \wedge \forall f$ .

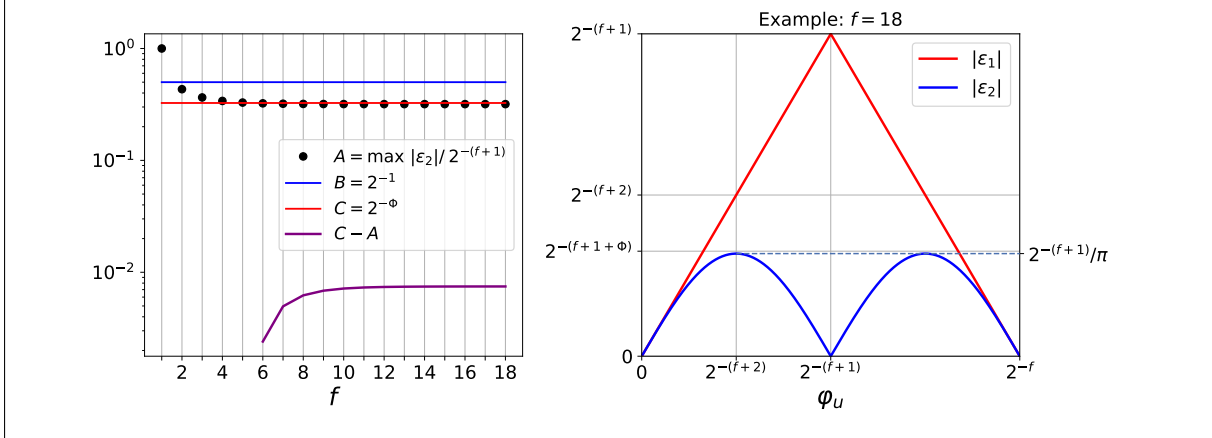
**Conjecture 2.16.** Using rule 2, bounds for the absolute error exist such that

- a)  $|\epsilon_2| \leq 2^{-(f+2)} \forall f \geq 2 \wedge \forall \varphi_u$
- b)  $|\epsilon_2| \leq 2^{-(f+1+\Phi)} \forall f \geq 6 \wedge \forall \varphi_u$ , where  $\Phi = \frac{1+\sqrt{5}}{2}$  is the golden ratio
- c)  $\lim_{f \rightarrow \infty} \max |\epsilon_2| = \frac{1}{\pi} 2^{-(f+1)}$

**Example 2.17.** To justify Conjecture 2.15 we go back to the  $2^3$  first register basis states of Example 2.12. We vary the value of  $\varphi_u$  over the domain  $[0, 1]$  and compute  $\hat{\varphi}_u$  from Eq.2.60 using both rules 1 and 2. On the left plot we show the estimation  $\hat{\varphi}_u$  from both rules, and on the right plot we show  $|\epsilon_2| \leq |\epsilon_1| \forall \varphi_u$ . The black dot represents  $\varphi_u = 0.0625 = \hat{\varphi}_{u,rule_2}$  from Example 2.12.



**Example 2.18.** Here we study the absolute error bounds put forward with Conjecture 2.16 up to  $f = 18$ . For each value of  $f$ , we again vary the value of  $\varphi_u$  over the domain  $[0, 1]$  and compute  $\hat{\varphi}_u$  with rule 2. We are interested in studying the local maxima for the absolute error, which we find numerically. On the left figure, we plot this maximum absolute error as a function of the number of first register qubits  $f$ , together with the lines for the two bounds conjectured, (a) and (b). We also plot the difference between bound (b) and the maxima, which is positive for the values of  $f \geq 6$  studied. On the right plot we choose to visualize one example, for  $f = 18$ . Finally, item (c) in the conjecture was arrived at by looking at  $\max |\epsilon_2| / 2^{-(f+1)}$  and understanding it is progressively approximating  $1/\pi$  up to the first 5 decimal digits recovered when  $f = 18$ . Conjecture 2.16 is confirmed numerically for these cases.



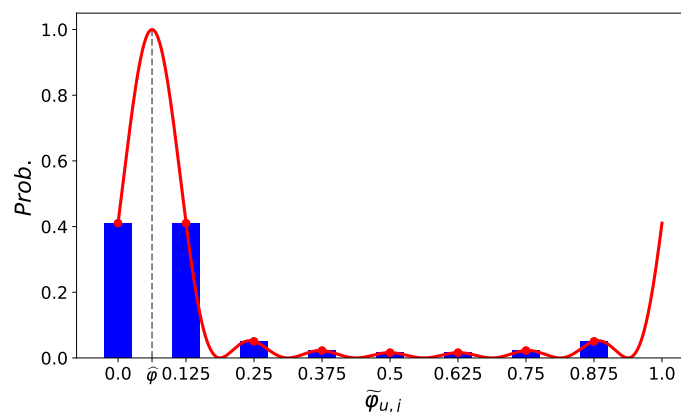
As we have seen, while the error with rule 1 is bounded by  $|\epsilon| \leq 2^{-(f+1)}$ , using rule 2 is always more advantageous, at least by  $|\epsilon| \leq 2^{-(f+2)}$ . One way to look at it is that using rule 2 we can have “at least” one qubit less on the first register and achieve the same accuracy we would obtain with rule 1. This improvement will be more noticeable the smaller  $f$  is.

Furthermore, debating these two rules is quite important from an experimental point of view when we come to consider the stochastic process involved in any measurement procedure as well as the possibility of systematic errors. It becomes even more relevant in the scenario of phase estimation computations with a

noisy quantum computer. In such scenario, on top of the uncertainty coming from the quantum nature of measurements on pure states, we will also have classical uncertainty overlapped on the final PMF obtained in the laboratory. The added randomness of having a classically mixed state will degrade the ideal PMF and further motivate the need to better quantify accuracy and precision. Both  $|\overline{R}|$  and  $v$  may be used to assess dispersion of the data.

Before moving on with a different discussion, we will end by noticing that a third rule can actually be followed. **Rule 3** is very simple in retrospect: we just need to fit the probability points of the PMF to the theoretical curve from which the PMF points are sampled, Eq.2.60. This curve only has two parameters:  $f$  and the real phase  $\varphi_u$ . Since we fix  $f$  from the beginning, we are left with the job of finding the best  $\varphi_u$  to fit the probability values for each  $\tilde{\varphi}_u$ . Thus, we can estimate  $\hat{\varphi}_u$  and its error bar from a simple fitting problem. For this task, we need to look into an appropriate fitting algorithm. Rule 3 is better than the previous two rules as it works perfectly for unitary simulations: it allows estimating  $\varphi_u$  without accuracy error in the whole range  $\varphi_u \in [0, 1]$ . We verified this numerically.

**Example 2.19.** Below, we recover Example 2.12 to uncover where the real phase is hiding. We again see the probability amplitudes; fitting this (ideal) normalized histogram to a function of the type of Eq.2.60 with  $f = 3$  we obtain the phase.



Despite rule 3 looking so promising, when we work with noisy experimental data, the procedure can be challenging. In this scenario, there may be room to develop clever filtering techniques to deal with the noise statistics degrading the curve we are trying to fit. We leave this challenge open.

Let us now understand the operation of the algorithm in a more general case. Often we may need to start the second register from a general state  $|\psi\rangle$ , which we can write as a linear combination of the eigenstates  $|u\rangle$  of  $U$ , with eigenvalue  $e^{i2\pi\varphi_u}$ ,

$$|\psi\rangle = \sum_u \alpha_u |u\rangle. \quad (2.66)$$

We will see that in this situation, the output state on the second register is no longer  $|\psi\rangle$  but one of the eigenstates  $|u\rangle$  with probability given by  $|\alpha_u|^2$ , while on the first register we will have an eigenvalue estimation for this eigenstate with the same probability. To see this, we write the initial state as  $|\psi_{in}\rangle = |0\rangle_f |\psi\rangle$ , and by the simple linear superposition property of quantum mechanics, the output state before measurement may be written

$$|\psi_{out}\rangle = \sum_u \alpha_u \left( \sum_{j=0}^{2^f-1} \left( \frac{1}{2^f} \sum_{k=0}^{2^f-1} \exp(i2\pi(\varphi_u - \tilde{\varphi}_{u,j})k) \right) |\tilde{\varphi}_{u,j}\rangle \right) |u\rangle, \quad (2.67)$$

where  $\tilde{\varphi}_{u,j}$  is still the binary fraction approximation to  $\varphi_u$  corresponding to one of the  $|0.j_0\dots j_{f-1}\rangle$  basis states of  $F$ . Then, a measurement on  $F$  yields the basis state  $|\tilde{\varphi}_{u,j}\rangle$  with probability

$$|\alpha_u|^2 P(\tilde{\varphi}_{u,j}), \quad (2.68)$$

where  $P(\tilde{\varphi}_{u,j})$  is again given by Eq.2.60. The final PMF is generally given in terms of  $F$ 's basis states  $\tilde{\varphi}_j = |0.j_0\dots j_{f-1}\rangle$  by

$$P(\tilde{\varphi}_j) = \sum_u |\alpha_u|^2 P(\tilde{\varphi}_{u,j}). \quad (2.69)$$

This means that if all the  $\varphi_u$ 's of the unitary operator  $U$  can be written exactly in  $f$  bits, the PEA procedure will collapse the wave function on the second register subspace to eigenstate  $|u\rangle$  and read the exact value  $\varphi_u$  on the first register with probability  $|\alpha_u|^2$ . It follows that if we run the same quantum circuit many times we will be able to distinguish in the obtained PMF a number of peaks equal to the number of  $\alpha_u$  coefficients in the originally prepared state, sitting at the phase estimation  $\tilde{\varphi}_u = \varphi_u$  of the corresponding  $|u\rangle$  with an height equal to  $|\alpha_u|^2$ .

Things are not so simple when one or more of the eigenphases  $\varphi_u$  can not be expressed exactly in  $f$  bits. In that case, entropy increases on the distribution  $P(\tilde{\varphi}_{u,j})$  for each of the eigenstates and there are probability values on all the possible basis outcomes. This time, running the circuit multiple times overlaps all the  $P(\tilde{\varphi}_{u,j})$  PMF's into a single one and the structure is in general more difficult to interpret. Nevertheless, if one increases resolution by increasing the number of qubits in the first register, the peaks structure of the probability distribution of Eq.2.69 will become apparent.

It will depend on the application whether or not it is useful to use the PEA starting the second register from a general state. In section 5.2.2.2 we will propose a method to read all the eigenvalues of any unitary operator  $U$  and the modulus square of all the wavefunction coefficients of any initial state  $|\psi\rangle$  at once.

## 2.5 Iterative phase estimation

In this section, we introduce the approach of iterative phase estimation, which allows measurement of the eigenvalues of a unitary operator through the estimation of  $\varphi_u \in [0, 1[$ , just like the PEA, but needs only a single ancillary qubit. Besides the smaller size of the first register, there are also some other differences with respect to the PEA.

Kitaev seems to have been the first one to give an iterative phase estimation procedure [52]. His proposal was based on the Hadamard transform but he didn't provide a complete algorithm for the task. Soon after that, with the introduction by Griffiths and Niu [39] of the semiclassical  $QFT$  mentioned in a previous section, it became possible to implement the  $QFT$ -based PEA with only one ancillary qubit and an iterative procedure. This approach is what we call the iterative phase estimation algorithm (IPEA) in this section. We will start by looking at the IPEA as presented by Dobšiček et al [53] and then we will provide some extra considerations in the end.

To begin with, we decide on the  $m$ -bit resolution for the phase estimation  $\tilde{\varphi}_u$  we aim for and notice that we can write the real phase  $\varphi_u$  in a two-part form containing the binary fraction decomposition  $\tilde{\varphi}_u$  plus some small remainder  $0 \leq \delta < 1$ ,

$$\varphi_u = \frac{j_1}{2^1} + \frac{j_2}{2^2} + \dots + \frac{j_m}{2^m} + \frac{\delta}{2^m} \quad (2.70)$$

$$= \tilde{\varphi}_u + \frac{\delta}{2^m}. \quad (2.71)$$



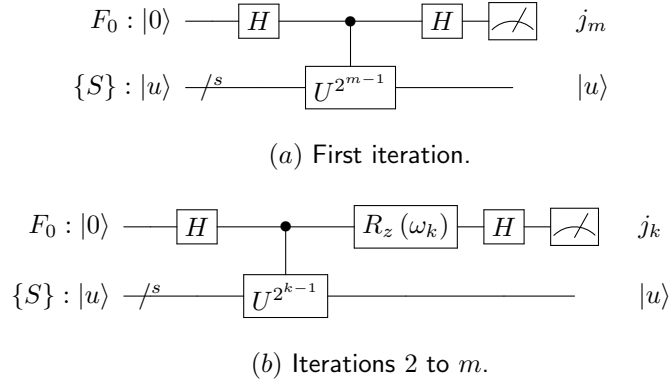


Figure 2.7: Schematic of the iterative phase estimation algorithm (IPEA).

With this representation in mind, we build the simple circuit depicted in Fig.2.7, which we will run for  $m$  iterations, at each one setting the values for parameters  $k$  and  $\omega_k$  using the information obtained in previous steps as will be explained shortly.

To kick-start the algorithm, the first iteration in Fig.2.7 (a) is actually the standard PEA of  $U^{2^{m-1}}$  with only one first register qubit. From what we learned in the previous section, this will give us a one-bit estimation for the phase of the eigenvalue associated with the eigenstate  $|u\rangle$  of  $U^{2^{m-1}}$ . But notice that this eigenvalue is

$$\exp(i2\pi 2^{m-1}\varphi_u) = \exp(i2\pi(j_1 2^{m-2} + j_2 2^{m-3} + \dots + j_{m-1} 2^0)) \exp\left(i2\pi\left(\frac{j_m + \delta}{2}\right)\right) \quad (2.72)$$

$$= \exp(i2\pi(0.j_m + 0.\delta)). \quad (2.73)$$

Therefore, if  $\delta = 0$ , PEA extracts this  $m$ -th bit,  $j_m$ , deterministically. If  $\delta \neq 0$  the case is different, but let us leave that analysis for the end.

Now on the second iteration we set  $k = m - 1$  on Fig.2.7 (b), i.e.  $U^{2^{m-2}}$ , and by the same token as before, the phase of the first register qubit state  $|1\rangle$  after the controlled operation is  $2\pi(0.j_{m-1}j_m + 0.0\delta)$ . This time we introduce an  $R_z(\omega_k)$  gate (Eq.2.14) on the circuit as pictured, and perform the  $z$ -rotation with an angle which is a function of the previously measured  $j_m$ , that is  $\omega_k = -2\pi 0.0j_m$ . This yields the phase  $2\pi(0.j_{m-1}0 + 0.0\delta)$  before the 1-qubit quantum Fourier transform  $H$ , and again considering  $\delta = 0$  we extract the second less significant bit,  $j_{m-1}$ .

A procedural pattern has emerged. Iteration 3, with  $k = m - 2$ , now acts on the phase kicked back by the controlled operation from the second register,  $2\pi(0.j_{m-2}j_{m-1}j_m + 0.00\delta)$ , with the  $z$ -rotation angle  $\omega_k = -2\pi 0.0j_{m-1}j_m$  to produce  $2\pi(0.j_{m-2}00 + 0.00\delta)$ . At each iteration, the  $R_z$  gate adds a relative phase that is just right to remove the phase contribution of all the previous bits. One more time considering  $\delta = 0$ , the hadamard gate extracts  $j_{m-2}$ . Proceeding in this way, all the bits of  $\tilde{\varphi}_u$  are extracted.

To summarize, the IPEA consists in the following steps:

1. Choose an eigenket  $|u\rangle$  from the unitarity to be simulated and define the desired resolution  $m$  for the estimation  $\tilde{\varphi}_u = \frac{j_1}{2^1} + \frac{j_2}{2^2} + \dots + \frac{j_m}{2^m}$ .
2. Starting  $S$  from  $|u\rangle$ , perform a 1-qubit PEA on  $U^{2^{k-1}}$  with  $k = m$  and measure  $F$  in the standard basis to obtain  $j_m$ .
3. Start an iterator variable  $k$  going from  $m - 1$  to 1; for each  $k$ :

(a) run the circuit in Fig.2.7 (b) with

$$\omega_k = -2\pi \sum_{l=2}^{m-k+1} \frac{j_{k+l-1}}{2^l}, \quad (2.74)$$

(b) measure  $F$  to obtain  $j_k$ .

What happens to this analysis if  $\delta \neq 0$ ? Well, going back to iteration 1, if  $\delta \neq 0$  we can use Eq.2.60 with  $\varphi_u = 0.j_m + 0.\delta$  and  $\tilde{\varphi}_u = 0.\tilde{j}_m$  to write

$$P(\tilde{j}_m) = \frac{1}{2^2} \left| \frac{1 - \exp(i 4\pi (0.j_m + 0.\delta - 0.\tilde{j}_m))}{1 - \exp(i 2\pi (0.j_m + 0.\delta - 0.\tilde{j}_m))} \right|^2. \quad (2.75)$$

Let us look at the probability of getting a correct measurement  $\tilde{j}_m = j_m$ . This is

$$P(\tilde{j}_m = j_m) = \frac{1}{4} \left| \frac{1 - \exp(i 4\pi 0.\delta)}{1 - \exp(i 2\pi 0.\delta)} \right|^2 \quad (2.76)$$

$$= \frac{1}{4} |1 + \exp(i 2\pi 0.\delta)|^2 \quad (2.77)$$

$$= \cos^2(\pi\delta/2). \quad (2.78)$$

Since  $0 \leq \delta < 1$ , this shows the probability of getting the correct bit  $P(\tilde{j}_m = j_m)$  is 1 for  $\delta = 0$  but decays to 0 for  $\delta = 1$ . As previously described, for the second iteration and following the  $z$ -rotation we have the state  $2^{-1/2}(|0\rangle + \exp(i 2\pi (0.j_{m-1}0 + 0.0\delta))|1\rangle)$  on the ancillary qubit. Applying the Hadamard gate yields

$$H \left( \frac{|0\rangle + e^{i 2\pi(0.j_{m-1}0+0.0\delta)}|1\rangle}{\sqrt{2}} \right) = \frac{1}{2} \left[ \left(1 + e^{i 2\pi(0.j_{m-1}0+0.0\delta)}\right)|0\rangle + \left(1 - e^{i 2\pi(0.j_{m-1}0+0.0\delta)}\right)|1\rangle \right]. \quad (2.79)$$

Similarly, for any subsequent iteration  $k$ , following the  $z$ -rotation gate, the ancillary qubit state we will have is  $2^{-1/2}(|0\rangle + \exp(i 2\pi (0.j_k0 + \frac{\delta}{2^{m+1-k}}))|1\rangle)$ ; applying it the Hadamard gate yields

$$|F_{k,out}\rangle = \frac{1}{2} \left[ \left(1 + e^{i 2\pi(0.j_k0+\frac{\delta}{2^{m+1-k}})}\right)|0\rangle + \left(1 - e^{i 2\pi(0.j_k0+\frac{\delta}{2^{m+1-k}})}\right)|1\rangle \right], \quad (2.80)$$

and we can look at the probabilities  $P_k(\tilde{j}_k)$ . For  $\tilde{j}_k = 0$ , we take the dot product with  $\langle 0|$  to get  $|\langle 0|F_{k,out}\rangle|^2$  which yields

$$P_k(\tilde{j}_k = 0) = \cos^2 \left( \frac{\pi}{2} (2^{k-m}\delta + j_k) \right), \quad (2.81)$$

while for  $\tilde{j}_k = 1$  we do  $|\langle 1|F_{k,out}\rangle|^2$  and get

$$P_k(\tilde{j}_k = 1) = \sin^2 \left( \frac{\pi}{2} (2^{k-m}\delta + j_k) \right). \quad (2.82)$$

Both equations give the same result for the probability of getting the correct bit  $\tilde{j}_k = j_k$ , which can be seen by replacing  $j_k = 0$  on the former and  $j_k = 1$  on the later. Hence, for any of the  $m$  iterations, the probability of obtaining the correct bit given that all the previous bits have been correctly extracted is

$$P_k(\tilde{j}_k = j_k) = \cos^2(\delta\pi 2^{k-m-1}). \quad (2.83)$$

With that, we can compute the overall probability for the IPEA to measure the correct binary decomposition

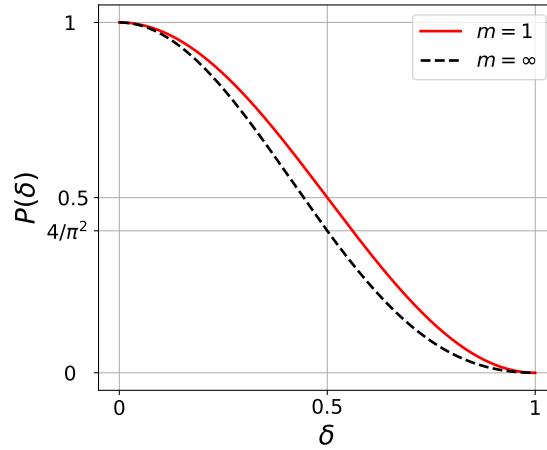


Figure 2.8: Overall probability for IPEA to extract  $\tilde{\varphi}_u$ . By mirroring the functions horizontally we get the overall probability to extract  $\tilde{\varphi}_u + 2^{-m}$ . The lower bound on the probability of getting the best  $m$ -bit decomposition of  $\varphi_u$  is  $4/\pi^2$ .

$\tilde{\varphi}_u$  as a function of the remainder  $\delta$

$$P_{\tilde{\varphi}_u}(\delta) = \prod_{k=1}^m \cos^2(\delta\pi 2^{k-m-1}) \quad (2.84)$$

$$\stackrel{(a)}{=} \left( \prod_{k=1}^m \cos(\delta\pi 2^{-k}) \right)^2 \quad (2.85)$$

$$\stackrel{(b)}{=} \frac{\sin^2(\delta\pi)}{2^{2m} \sin^2(\delta\pi 2^{-m})}. \quad (2.86)$$

where at (a) we note that the product of squares is the square of the products and  $\prod_{k=1}^m P_k = \prod_{k=1}^m P_{m+1-k}$ ; and at (b) we continuously expand the product identity  $\sin(\theta) = 2 \sin(\theta 2^{-1}) \cos(\theta 2^{-1})$  into the string of products  $\sin(\theta) = 2^m \sin(\theta 2^{-m}) \prod_{k=1}^m \cos(\theta 2^{-k})$ .

It is interesting to see this is the same result we obtained for the PEA in Eq.2.62, with  $m = f$  and  $\delta = \varepsilon 2^m$ . It is a monotonically decreasing function of parameter  $m$  for all the values of  $\delta$ , and in the limit  $m \rightarrow \infty$  it approaches

$$P_{\tilde{\varphi}_u, m \rightarrow \infty}(\delta) = \left( \frac{\sin(\pi\delta)}{\delta\pi} \right)^2. \quad (2.87)$$

In Fig.2.8 we plot  $P_{\tilde{\varphi}_u}(\delta)$  for  $m = 1$  as well as the limiting case. All the other functions for different values of  $m$  lie between the two curves.

Now observe that the probability to extract  $\tilde{\varphi}_u$  drops to 0 as  $\delta \rightarrow 1$ , but this is not a problem since only for  $\delta < 0.5$  is  $\tilde{\varphi}_u$  the best  $m$ -bit approximation to  $\varphi_u$ ; for  $\delta > 0.5$  the best  $m$ -bit approximation to  $\varphi_u$  is actually  $\tilde{\varphi}_u + 2^{-m}$  and the probability to get this results is  $P_{\tilde{\varphi}_u + 2^{-m}}(1 - \delta)$ , increasing monotonically up to 1 for  $\delta > 0.5$ . Therefore, we are guaranteed to get the best  $m$ -bit approximation to  $\varphi_u$  with a lower bound of  $P_{\tilde{\varphi}_u, m \rightarrow \infty}(0.5) = 4/\pi^2$ , which is again the same we had for PEA.

Hence, the IPEA outputs a  $m$ -bit binary expansion  $\hat{\varphi}_u$  which is close to  $\varphi_u$  within an error bound of  $|\epsilon| = |\varphi_u - \hat{\varphi}_u| \leq 1/2^{m+1}$  with probability of error  $P_\epsilon < 1 - (4/\pi^2)$ .

There are two main ways of operating the IPEA. These are:

- (a) run an iteration a sufficient number of times to let probabilities converge, choose the most frequently obtained bit and proceed to the next iteration;

- (b) run each iteration only once but repeat the total procedure multiple times.

In (a) we only explore one branch of the full binary tree of the  $m$ -bit string, while on the second format we will stochastically explore multiple branches in this space, just as happened in PEA.

Given that the operational format (a) only outputs a single binary string, the information we obtain is not as complete as what method (b) and the PEA provide, because those instead allow us to explore the complete state space of the  $m$ -bit string. This also means that approach (a) is only useful if we start the algorithm at an eigenket or very close to one; it's not possible to explore general states to obtain all the eigenvalues of  $U$  and the wavefunction coefficients, as we could do with the PEA and still can do using method (b).

To decide on the final computation  $\hat{\varphi}_u$  of the full IPEA procedure, literature usually describes accepting the  $m$ -bit state obtained most frequently, or equivalently, rule 1 from the previous section on the PEA. As already described, with method (a) this is actually the only  $m$ -bit state that we get. However, if we operate the IPEA with method (b) we can also use rule 2 with the same motivation to marginally improve accuracy and obtain a statistical description of the results from noisy platforms. We can't apply rule 2 on method (a) because we don't obtain probabilities for any other state from the full state space of the  $m$ -bit word. However, within method (a), we can still do some useful post-processing by reconstructing an approximation to the full PMF over all the possible  $m$ -bit strings. Let us call it APMF. We can then use the APMF to compute the mean phase direction  $\hat{\varphi}_u$  (rule 2). It turns out we cannot increase accuracy this way because computing the mean phase direction from the APMF gives as result the same phase value encoded by the final  $m$ -bit string already obtained.

Nevertheless, we can still compute the circular standard deviation of the APMF. This allows us to have a measurement of dispersion that takes into account all the 2-state probability functions obtained in the intermediate steps before getting to the final estimation  $\tilde{\varphi}_u$ .

This post-processing step consists in organizing the relative frequencies of measurement obtained for each bit in a binary tree format where we only have 1 full depth branch with chopped nodes at every iteration. This deepest branch encodes the estimated phase  $\tilde{\varphi}_u$ . Then, one uniformly propagates the probabilities obtained for unexplored nodes down to the leafs of the tree to reconstruct an approximation to the PMF on the final level, the APMF. This approximation will not give us sufficient information to discern all the complexity of the PMF across all the different  $m$ -bit states as PEA did, but allows us to compute the circular standard deviation of the APMF, which we can use as a measure of dispersion. An example is provided below.

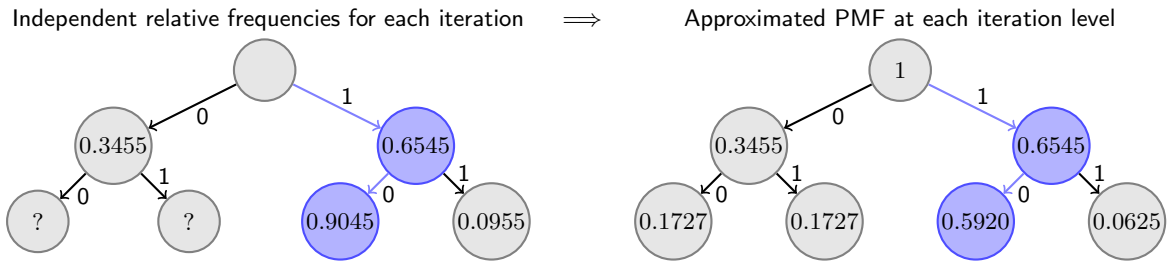
From the physical implementation point of view, there are several advantages of the IPEA with respect to the standard PEA. The first obvious one is that the number of qubits needed on the first register is much smaller. This is a very significant achievement that allows the implementation of the algorithm even on small quantum computers, and it is even more praiseworthy when one considers this is possible without any compromise on the bit resolution which is only dependent on the number of iterations and the ability of the rotation gate to refine the angle ever more precisely. In the case of the PEA, the number of qubits on the first register was a limiting factor on the resolution we could get.

The second obvious advantage is that the circuit depth will also be shallower, which is crucial when the platform being used can not perform well with deep circuits. Going on with the list, the third thing to notice is the fact that overall we use less 2-qubit gates, as the Fourier transform in IPEA never uses those. This is relevant because 2-qubit gates are usually much more difficult to implement and carry higher error rates than 1-qubit ones. Connected with this problem is the issue of imperfect qubit connectivity on the computer, because when using platforms in this form the number of 2-qubit gates increases greatly due to the necessary additional gate decompositions to swap states across the qubit network.

Together, these advantages can indeed make the difference between getting or not results with a phase estimation procedure when using computers with short coherence times, high error rates and low number of

qubits, like the ones currently available.

**Example 2.20.** Let us use a 2-bit precision IPEA and operational method (a) to measure the phase  $\varphi_u = 0.35 = 0/2 + 1/4 + \delta/4$  with  $\delta = 0.4$ . According to Eq.2.83, on the first iteration we get a distribution  $\{0 : \approx 0.3455; 1 : \approx 0.6545\}$ , while on the second iteration the distribution should be  $\{0 : \approx 0.9045; 1 : \approx 0.0955\}$ . Therefore, the phase estimation encoded by the final  $m$ -bit state is  $\tilde{\varphi}_u = 0/2 + 1/4 = 0.25$ . Diagram below shows the raw probabilities on the left. Note that some outcomes on the second level of the tree are not filled because they were never explored. On the right, the reconstructed approximation to the PMF at each iteration level is obtained.



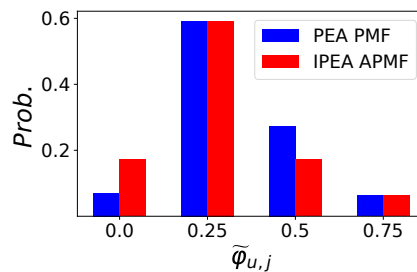
After decoding the binary strings of the leaf level of the tree we are able to reconstruct the final approximation to the PMF,

$$P(\tilde{\varphi}_u) = \{0.0 : \approx 0.1727; 0.25 : \approx 0.5920; 0.50 : \approx 0.1727; 0.75 : \approx 0.0625\} \quad (2.88)$$

Using this distribution we can compute the mean phase direction (Def.2.13) and the circular standard phase deviation (Def.2.14),

$$\hat{\varphi}_u = 0.25, \quad v = 0.17 \quad (2.89)$$

From the result we see that  $\hat{\varphi}_u = \tilde{\varphi}_u$ , hence no improvement in accuracy is obtained. However, we can still get the dispersion measure  $v = 0.17$ . Below, we compare this APMF to the PMF we would obtain with a full exploration of the binary string space that would be possible with the PEA or IPEA with operational method (b). Note that the distributions match at the lowest resolution level - e.g. if we sum the probabilities for the basis states  $|00\rangle$  and  $|10\rangle$  for the PMF we obtain the same value in the APMF at that level.



There have also been other iterative phase estimation proposals. Rudolph and Grover [54] looked at the problem of iterative estimation of the three parameters of a general unknown unitarity  $U \in SU(2)$ , Eq.2.12, and then Zhengfeng et al [55] pointed out to some errors in their method while proposing another approach. More recently, O'Loan [56] identified problems with both the methods in [54] and [55] and provided a new IPEA algorithm with simpler experimental setup that neither requires rotation gates nor entanglement while

it still gets within a logarithmic factor of the Heisenberg limit.

## 2.6 Noisy computations

Up to now we have been considering the case of quantum computing with unitary circuits, and the algorithms we presented work as described for such ideal case. This unitary computation is the goal we hope to achieve experimentally one day, but at the moment the devices we have available are not capable of fault-tolerance, and error correction can not be performed with more than a very small number of qubits. For instance, the connectivity of the IBM Q devices is not yet sufficient to detect and correct a full set of quantum errors, although some techniques have been shown with proof-of-principle experiments [57, 58, 59, 60, 61, 62]. Therefore, we will not be considering error correction.

When noise plays a role in the dynamics of the qubits, their evolution is non-unitary, meaning that it doesn't preserve probability within the computer subspace  $\mathcal{H}_c$  of the computer + environment Hilbert space,  $\mathcal{H} = \mathcal{H}_c \otimes \mathcal{H}_e$ . In this case, the quantum processor is not perfectly isolated from its environment and co-evolves with external degrees of freedom due to some non-trivial coupling term  $Q$  on the global Hamiltonian

$$H = H_c \otimes \mathbf{1} + \mathbf{1} \otimes H_e + Q. \quad (2.90)$$

At the end of the algorithm execution, the wavefunction can not be factorized over both subspaces and the output measurements will not correspond to the correct computation. An error has occurred. Therefore, if we run the algorithm multiple times in a platform where noise is present, we should expect that some of the times the final output state will be the correct one corresponding to the unitary evolution of the computer wavefunction, while other times it may be one of multiple possible states corresponding to corrupted computations. On top of the uncertainty coming from the quantum nature of the computation, we have also gained classical uncertainty.

We can model this behavior as an ensemble where we think of the output state  $\rho_c$  as a mixed quantum state in which, in good experimental conditions, we expect to have a higher percentage of the ideal state  $|\psi_0\rangle$ , given in the density matrix formulation by

$$\rho_{c|ideal} = |\psi_0\rangle \langle \psi_0|, \quad (2.91)$$

than of any of the corrupted states  $|\psi_{i>0}\rangle$ . These wavefunctions are classically mixed with weights  $\omega_i \in \mathbb{R}$  that don't carry information about the relative phase of the different pure states (incoherent mixture), and the overall density matrix reads

$$\rho_c \equiv \sum_i \omega_i |\psi_i\rangle \langle \psi_i|, \quad (2.92)$$

with  $\sum_i \omega_i = 1$ . The density operator is hermitian and satisfies the normalization condition  $Tr(\rho_c) = 1$ .

**Definition 2.21.** The *purity* of a quantum state of  $n$ -qubits is a scalar measure of how much the state is mixed, being defined in terms of the density matrix  $\rho_c$  as

$$\gamma = Tr(\rho_c^2) \quad (2.93)$$

and satisfying  $\frac{1}{2^n} \leq \gamma \leq 1$ .

**Definition 2.22.** The *von Neumann entropy* of a quantum state of  $n$ -qubits is given by

$$S = -Tr(\rho_c \ln \rho_c). \quad (2.94)$$

In the basis where  $\rho_c$  is diagonal, it can be shown [63] that the von Neumann entropy has a minimum  $S = 0$  for a pure state, and a maximum  $S = \ln(2^n)$  for a uniformly mixed ensemble. We can also compute  $S$  for some subsystem of qubits (quantum register) disregarding the fact that it may be entangled with the other qubits of the computer. Just like the measure of purity, if we could compute  $S$  it would let us know the reliability of the final results. In fact, purity and the von Neumann entropy are similar measures of the degree of mixing, but purity is easier to calculate as it does not require the density matrix to be diagonalized.

A good output from the computation is one for which  $\omega_0 \gg \omega_i \forall i$ , or equivalently  $\gamma \approx 1$ , or even  $S \approx 0$ . Note however that the two last measures do not guarantee the final state to be  $|\psi_0\rangle$ . In the ideal case of a pure state we have  $\omega_0 = 1$ ,  $\gamma = 1$ , and  $S = 0$ , and the eigenvalues of  $\rho_c$  are 0 or 1.

**Example 2.23.** The mixed state of a qubit can also be visualized in the Bloch sphere. Being hermitian, the density matrix can be written according to Eq.2.3 as

$$\rho = \frac{1}{2} (\mathbf{1} + c_1 \sigma^x + c_2 \sigma^y + c_3 \sigma^z) = \frac{1}{2} (\mathbf{1} + \vec{c} \cdot \vec{\sigma}), \quad (2.95)$$

where each coefficient is associated to a component of Bloch vector such that  $\vec{c}$  is the vector representing the quantum state. Since  $\rho$  has eigenkets  $|r_1\rangle$  and  $|r_2\rangle$ ,  $\vec{c} \cdot \vec{\sigma}$  has the same eigenkets with the eigenvalues  $\pm \sqrt{c_1^2 + c_2^2 + c_3^2}$ . Thus, diagonalizing

$$\rho = \frac{1}{2} \left( 1 + \sqrt{c_1^2 + c_2^2 + c_3^2} \right) |r_1\rangle \langle r_1| + \frac{1}{2} \left( 1 - \sqrt{c_1^2 + c_2^2 + c_3^2} \right) |r_2\rangle \langle r_2|. \quad (2.96)$$

If  $c_1^2 + c_2^2 + c_3^2 = 1$ , we have a pure state and  $\vec{c}$  lies on the surface of the Bloch sphere; else if  $c_1^2 + c_2^2 + c_3^2 < 1$  our state is a mixed and  $\vec{c}$  lives in the bulk of the sphere. In this perspective, purity of a state is represented by how close the state vector is to the surface of the sphere. The completely random state of a qubit for instance corresponds to the central point,  $\rho = \frac{1}{2}$ .

**Definition 2.24.** The *ensemble average* of some observable  $\mathcal{O}$  is the mean value of a measurement of  $\mathcal{O}$  when a high enough number of measurements is performed, being defined as

$$\langle \mathcal{O} \rangle = \text{Tr}(\rho_c \mathcal{O}). \quad (2.97)$$

Note that  $\langle \mathcal{O} \rangle$  can be evaluated using any convenient basis since the trace is independent of the representation used.

During the operation of a quantum gate described algorithmically by its unitary operator  $U$ , errors can occur of two types: *coherent* and *incoherent*. Coherent errors are those that preserve the purity of the state they act upon but apply a slightly different unitary operation

$$\tilde{U} = U + U', \quad (2.98)$$

where  $U'$  is some small perturbation. Differently, incoherent errors do not preserve purity.

To model the dynamic evolution of the density operator of the computer,  $\rho_c$ , during the time of execution of some incoherent quantum gate from the Hamiltonian in Eq.2.90 we would need to consider the density operator,  $\rho$ , over the full Hilbert space  $\mathcal{H}$ , to do

$$i \frac{\partial \rho}{\partial t} = -[\rho, H]. \quad (2.99)$$

This is because the density operator is in general not factorizable, i.e.  $\rho \neq \rho_c \otimes \rho_e$ , and the Hamiltonian

in Eq.2.90 couples both subspaces. This is not practical for several reasons, and the way we represent an evolution, or transformation,  $\Phi$ , between quantum states  $\rho_c$  is through the mapping

$$\Phi(\rho_c) = \sum_{j=1}^n K_j \rho_c K_j^\dagger, \quad (2.100)$$

where the conservation of  $\text{tr}(\rho_c) = 1$  is satisfied by the constraint that the Krauss operators  $\{K_j\}_{j=1, \dots, n}$  obey

$$\sum_{j=1}^m K_j^\dagger K_j = \mathbf{1}. \quad (2.101)$$

This is called a Kraus map, or an operator-sum representation, and it can represent any physical map on a finite dimensional Hilbert space, although in general this representation is not unique.

With the operator-sum representation, calculations are simplified as we only need to know a series of operators  $\{K_j\}_{j=1, \dots, n}$  acting on the computer state space. With this formalism, one doesn't need to explicitly consider the properties of the environment to describe the dynamics of the qubits.

### 2.6.1 Useful information theoretic measures

The mixed state we obtain at the end of a computation performed within a noisy environment will alter the ideal PMF that would be obtained if we had fault-tolerance. The quality of the computation will depend on how resilient is our quantum algorithm and its classical post-processing to an increasing degree of noise.

To assess the quality of computational results, we will use some distance measures for comparing the probability distributions obtained experimentally with the ideal PMF that we expect when a theoretical analysis can give us a prediction.

**Definition 2.25.** Consider a discrete random variable  $X$  over a sample space  $\chi$  and a probability mass function  $p(x) = \text{Pr}\{X = x\}$ ,  $x \in \chi$ . The *entropy* of such variable,  $H(X)$ , is defined [64] in base 2 to be

$$H(X) = - \sum_{x \in \chi} p(x) \log_2 p(x). \quad (2.102)$$

Entropy is a measure of the amount of information required to describe the probability distribution, on the average. We can calculate the entropy for a particular experimental distribution to get a measure for how uncertain it is. Lower entropy is in general a good indicator, as it reveals the PMF is far from uniform.

**Definition 2.26.** The *Kullback-Leibler divergence (KLD)*, also called *relative entropy*, characterizes the degree of inefficiency in assuming a random variable is modeled by a distribution  $q(x)$  when the real distribution is  $p(x)$ . It is defined, in terms of the respective probability mass functions, as

$$D(p||q) = \sum_{x \in \chi} p(x) \log_2 \frac{p(x)}{q(x)}. \quad (2.103)$$

The KLD is always non-negative  $0 \leq D(p||q)$ .

KLD divergence measures a kind of directed "distance" between two probability mass functions. However, it doesn't qualify as a true distance between distributions, i.e.  $D(p||q) \neq D(q||p)$ , since it is not symmetric and doesn't satisfy the triangle inequality. It would be useful to have a symmetric metric to evaluate a true distance between distributions. This is accomplished by the following two definitions.



**Definition 2.27.** The *symmetrized Kullback-Leibler distance (sKL)* between two discrete probability distributions  $p(x)$  and  $q(x)$  is given by

$$sKL(p, q) = D(q||p) + D(p||q). \quad (2.104)$$

**Definition 2.28.** The *Jensen-Shannon divergence (JSD)* is another symmetric and smooth variant of the KLD, defined as

$$JSD(p||q) = \frac{1}{2}D(p||m) + \frac{1}{2}D(q||m), \quad (2.105)$$

with  $m = \frac{1}{2}(p + q)$  written in terms of the probability distributions  $p(x)$  and  $q(x)$ . It is bounded in  $0 \leq JSD(p||q) \leq 1$ .

**Definition 2.29.** The *Fidelity* of the discrete probability distributions  $p(x)$  and  $q(x)$  of two random variables  $X$ , and  $Y$  is a symmetric distance measure given by

$$F(p, q) = \sum_j \sqrt{p_j q_j}. \quad (2.106)$$

It is not a metric but it's bounded in  $0 \leq F(p, q) \leq 1$ , with  $F(p, q) = 1$  iff  $p = q$ .

Both the *sKL*, *JSD* and *Fidelity* can be used to measure the distance between the experimental distribution and the theoretical one. This allows us to estimate the quality of the computation in terms of its overlap with the ideal distribution: lower values of *sKL* and *JSD*, and higher values of *Fidelity* are usually desirable.

## 2.7 Current status and near term prospects

Quantum computers are no longer a future theoretical chimera, but a practical matter which, despite being in its infancy, already hints on spectacular perspectives. The field has been seeing an increased rate of development recently, with significant investments being made worldwide both on the public and private sectors. Several information technology giants, like IBM, Intel, Google, Microsoft and Alibaba are carrying out their own I&D programs to develop quantum computers. Some have shown the first prototypes and made them available for external users via remote on-line connection. In the short run, the leading platforms are based on superconducting qubits, used by IBM and by the spin-off company Rigetti.

We can hope that, in the coming years, frontier science can be done on these quantum computing platforms. Meanwhile, state of the art devices available in the near term have been coined by Preskill [65] in the noisy intermediate-scale quantum (NISQ) computing category. The regime will be characterized by systems with a small number of qubits  $\sim 100$  (thin circuits), where decoherence limits the number of gates that can sequentially be applied (shallow circuits). No full error correction is expected to be possible soon and noise will be part of the operation on these devices. Performance has to be characterized both by the number of qubits and their quality.

The most advanced technologies right now are based on two different physical systems:

1. Superconducting qubits
2. Trapped ions.

The quantum computers used in this work are based on the former. The two level system used to store and manipulate the quantum bit is provided by the collective wave function of superconductor resonators. More specifically, these are based on *LC* circuits, with  $R = 0$  (provided by the superconducting state), that are known to be isomorphic to a harmonic oscillator. The quantum variable in the Hamiltonian is the superconducting

phase. In order to have a non equidistant spectrum, the  $LC$  circuit is in series with a Josephson tunnel junction, that provides a non-quadratic dependence of the Hamiltonian on the phase.

The long-term target of this research field is to fabricate sufficiently coherent qubits that allow for quantum error correction codes. These codes can only be efficient, if the qubits and gate errors perform better than some well defined thresholds.

The short-term goal is to demonstrate the so called *quantum supremacy*. This will happen when an algorithm is implemented in a quantum computer solving a problem that can not be tackled with the best conventional supercomputers in the world. For instance, in the realm of many-body physics, the largest  $S = 1/2$  Heisenberg model diagonalized so far has  $N = 50$  spins. If a quantum computer manages to give the ground state energy of a system larger than this, it will be said that quantum supremacy has been achieved in this context. Of course, as the classical methods can improve with time, quantum supremacy is a moving target. However, in the case of many-body problems, adding just 1 more degree of freedom can entail duplicating the memory of the classical computer, whereas it only requires 1 extra qubit in a quantum computer.

In order to assess the power of a quantum computer, IBM researchers have recently proposed an architecture-neutral metric to characterize the quantum computing power. The so called *quantum volume* [66] takes into account the number and quality of qubits, connectivity, and gate error rates.

An important area of research is to design algorithms that can benefit from the state of the art non-ideal computers. This is the case, for instance, of hybrid algorithms, that combine shallow programs, run on a quantum computer, with parameter optimizations carried out in conventional computers.

### 2.7.1 IBM Quantum Experience

In 2016, IBM launched<sup>4</sup> the Quantum Experience online platform (IBM Q), which permits remote access to several prototype quantum computers. This has allowed thousands of users, worldwide, to run real quantum computations and, despite noise and the limited number of qubits, carry out several novel research projects.

Currently, this platform offers free access to two systems with 5 qubits (*IBM Q 5 Yorktown*, *IBM Q 5 Tenerife*), and two systems with 16 qubits (*IBM Q 16 Rueschlikon*, *IBM Q 16 Melbourne*). There is also available a high performance simulator to emulate a quantum computer with 32 qubits, for public use. To their clients, IBM provides two exclusive 20 qubit computers (*IBM Q 20 Tokyo*, *IBM Q 20 Austin*).

All of these devices are different. They vary in their qubit connectivity graph as well several physical parameters, for instance the  $T_1$  and  $T_2$  time constants, and the error rates. An useful feature that is still not implemented on the hardware is the ability to use measurement feedback or feedforward as well as resetting qubit states. Fig.2.9 and Fig.2.12 show two different architectures.

The physical platform used by IBM's quantum computers consists in superconducting transmon qubits, and despite being so hard and challenging to develop all the technology involved in building a full scale useful universal quantum computer, significant improvements have been continuously made and coherence time of the systems is currently getting to  $\sim 100\mu s$  [67, 68, 69, 70, 71, 72, 73, 74].

However, these quantum computer prototypes still carry high error rates whose improvement is a major challenge. IBM uses simultaneous randomized benchmarking to measure average gate errors per Clifford gate [75]. These readings can then be converted to error rates of different gates according to the set of primitive gates being used.

As an example, Tab.2.2 shows the values of the parameters for one of the computers made publicly available, at some point in time. Notice that the values of these error rates might be quite large for most applications that require a few dozen operations. Applying too many operations destroys coherence of the

<sup>4</sup><https://www-03.ibm.com/press/us/en/pressrelease/49661.wss>

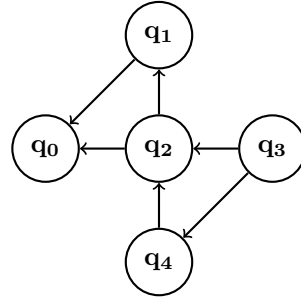


Figure 2.9: IBM Q 5 Tenerife quantum computer (*ibmqx4*). The coupling architecture of this 5 qubit device is represented by the arrows, which indicate the control  $\rightarrow$  target qubits of  $cX$  gates.

|                                      | q0    | q1                     | q2                     | q3                     | q4                     | Average |
|--------------------------------------|-------|------------------------|------------------------|------------------------|------------------------|---------|
| Frequency (GHz)                      | 5.25  | 5.30                   | 5.35                   | 5.43                   | 5.18                   | 5.302   |
| $T_1(\mu s)$                         | 48.90 | 52.90                  | 36.70                  | 43.20                  | 57.90                  | 47.92   |
| $T_2(\mu s)$                         | 35.50 | 14.50                  | 25.40                  | 14.70                  | 14.60                  | 20.94   |
| Gate error ( $10^{-3}$ )             | 0.94  | 6.87                   | 1.46                   | 2.32                   | 1.55                   | 2.628   |
| Readout error ( $10^{-2}$ )          | 7.10  | 6.80                   | 3.00                   | 3.80                   | 7.60                   | 5.66    |
| Multi-qubit gate error ( $10^{-2}$ ) |       | <b>cX<sub>10</sub></b> | <b>cX<sub>20</sub></b> | <b>cX<sub>32</sub></b> | <b>cX<sub>42</sub></b> | 4.99    |
|                                      |       | 3.29                   | 2.79                   | 7.43                   | 6.82                   |         |
|                                      |       |                        | <b>cX<sub>21</sub></b> | <b>cX<sub>34</sub></b> |                        |         |
|                                      |       |                        | 4.65                   | 4.96                   |                        |         |

Table 2.2: IBM Q 5 Tenerife benchmark measurements from calibration on 2018-10-06 8:57:05 am.

quantum state and increases entropy on the final probability distribution obtained, rendering results useless. Therefore, care should be taken in developing circuits that are as shallow as possible.

We performed three quick experiments to probe the reliability of one of the computers made available by IBM and get a feeling for hardware limitations in terms of circuit depth. The first one considers applying two consecutive Hadamard gates on some qubit; the result should be the identity, therefore if we measure the state of the qubit afterwards it should be the same as the initial state. The same is true for any number of repetitions of this group of two Hadamards. We started qubit 8 from *ibmq\_20\_tokyo* in the state  $|0\rangle$  and experimented several times by applying increasingly higher numbers of these couples of hadamards, measuring the probability to get the initial state at the end each time (we call this *series A*). The second experiment (*series B*) is exactly as the first one but starting the qubit from state  $|1\rangle$ . For the third experiment (*series C*) we follow the same approach of repeating a couple of gates but now testing the two-qubit gate  $cX$ . The results are shown in Fig.2.10. Series A and B for instance show an unexpected non-monotonic behavior, while series C is monotonic and shows that after approx. 80  $cX$  gates the probability to measure the initial state has decrease to 0.5. As for the reason why results from series B degrade faster than for series A, we should note that the initial state for series B,  $|1\rangle$ , is an excited state. Therefore, in addition to pure dephasing mechanisms, that kill superpositions, there is an additional mechanism for decoherence if the system is prepared in the excited state: energy dissipation. This is a  $T_1$  process, by which the system jumps from  $|1\rangle$  to  $|0\rangle$ .

Besides the available number of qubits, their error rates and coherence times, the connectivity graph of the computer is also of major importance. For instance, in IBM's 20 qubit Tokyo machine we have 72 coupling pairs. If the computer actually had perfect all-to-all connectivity between qubits we would have 190. Fig.2.11 illustrates how to implement the  $cX_{04}$  gate on *ibmqx4*, showing how incomplete connectivity can disastrously increase the number of operations needed to accomplish some global multi-qubit unitary transformation. Sometimes we may be able to simplify this decompositions a little by considering global simplifications on a circuit containing several gates, but still this reduction is usually only marginal.

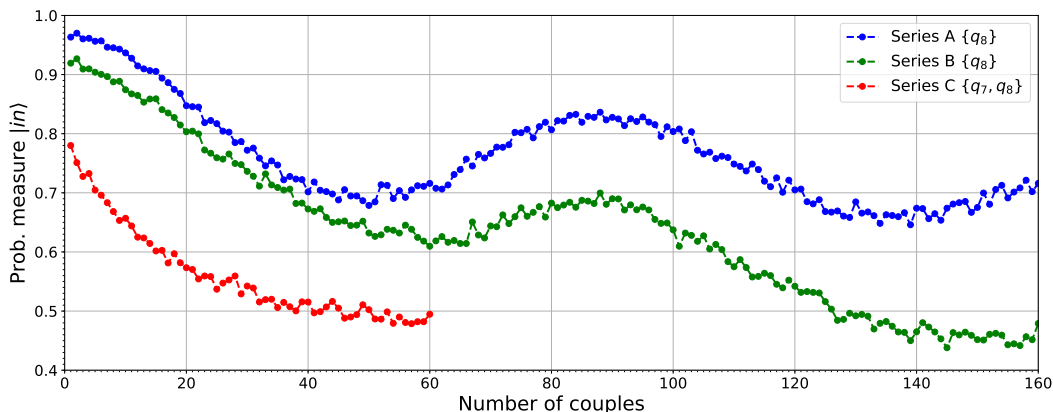


Figure 2.10: Degradation of results due to noise and decoherence on *ibmq\_20\_tokyo*.

Because of this, when we are mapping a quantum algorithm to a physical platform, we should try to find the best possible mapping to reduce the number of necessary operations, as well as choose readout qubits as the ones that carry lower readout errors. For a sufficiently large algorithm (in the number of operations and qubits), this task may easily become an optimization problem as difficult as the ones we would be interested to solve with a quantum computer.

To be fair, some algorithms don't need perfect connectivity, although many others do. Going forward, an expected challenge on the software side of the discipline will be to develop better algorithms for partial coupling architectures. There are already some examples and this effort is also leading into theoretical insight which goes beyond practical considerations [76, 77].

IBM has also been developing all the software stack necessary to control and operate on these machines, from the lowest level of control to the highest level of simplified usability. QISKit<sup>5</sup> is a Python library to code quantum algorithms at high level, compile them and interface with the real devices through the internet. It also allows to perform simulations on classical emulations of quantum computers both unitary and with noise. QISKit is open source and has been benefiting from a growing community of users and contributors worldwide.

In the IBM Q Experience, single-qubit operations are implemented with just three physical gates,

$$\begin{aligned}
 U_1(\lambda) &= U(0, 0, \lambda) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{pmatrix} \\
 U_2(\phi, \lambda) &= U\left(\frac{\pi}{2}, \phi, \lambda\right) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -e^{i\lambda} \\ e^{i\phi} & e^{i(\lambda+\phi)} \end{pmatrix} \\
 U_3(\theta, \phi, \lambda) &= U(\theta, \phi, \lambda)
 \end{aligned} \tag{2.107}$$

These main single-qubit gates are derived from Eq.2.12 and all other gates can be obtained from these. They take in one, two, and three parameters, respectively. Together with the *cX* gate, these form a universal gate set for quantum computation on the real chips of IBM. In fact,  $U_1$  and  $U_2$  are just special cases of  $U_3$ , hence IBM's universal gate set can be reduced to just two gates. However, in practice we have noise, as explained, and universality is not achievable with arbitrary accuracy.

<sup>5</sup><https://qiskit.org/>

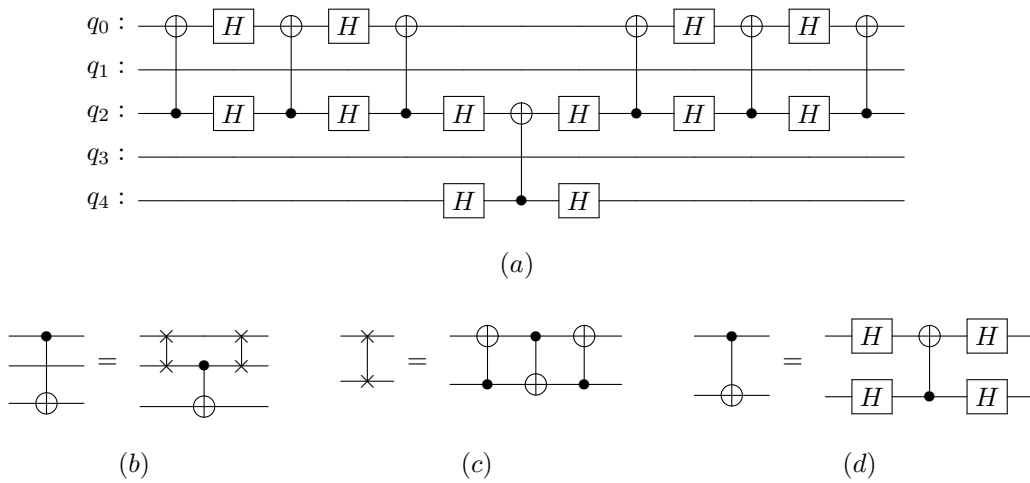


Figure 2.11: The circuit in (a) is the decomposition of the 2-qubit gate  $cX_{04}$  on the incomplete coupling architecture of *ibmqx4*. We need to SWAP states across the  $\{q_0, q_2\}$  qubit pair and then apply  $cX_{24}$ , however, we only have available two 2-qubit gates with these qubits:  $cX_{20}$  and  $cX_{42}$ . The decomposition uses the identities shown in (b), (c) and (d).

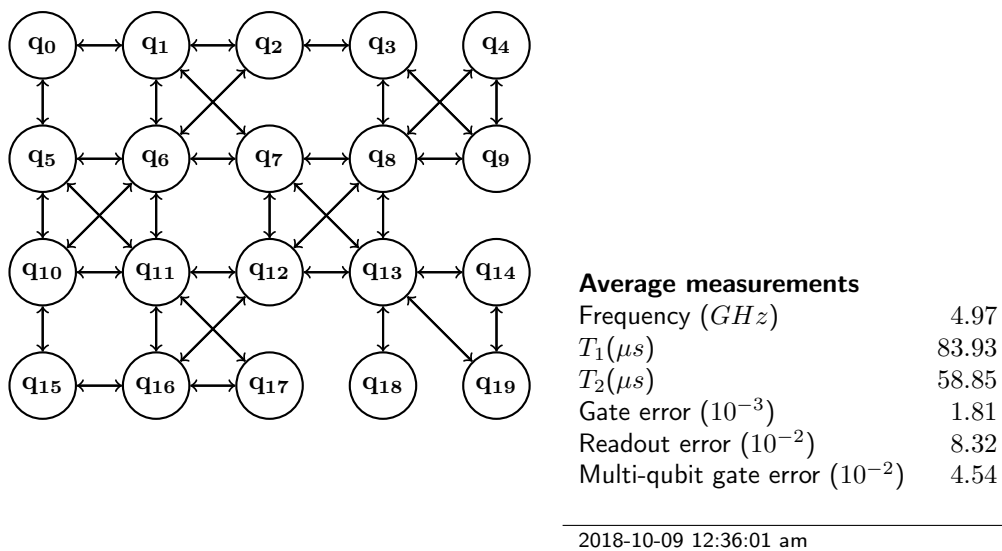


Figure 2.12: IBM Q 20 Tokyo quantum computer (*ibmq\_20\_tokyo*). The coupling architecture of this 20 qubit device is represented by the arrows, which indicate the control  $\rightarrow$  target qubits of  $cX$  gates.



## Chapter 3

# Quantum many-body systems

*"It would indeed be remarkable if Nature fortified herself against further advances in knowledge behind the analytical difficulties of the many-body problem."*

— Max Born, *The Mechanics of the Atom*, 1960

### 3.1 The many-body problem

Most physical systems are made several interacting constituents. These many-body systems exist in a variety of forms across all length scales from the quantum domain, such as the set of molecules in a drop of water, to the astronomical scale, such as the set of galaxies in the Laniakea supercluster [78]. There is usually no analytical solution to describe interacting systems with more than two objects. For instance we can solve analytically the two-body problem of the Earth-Moon system, but not that of the Earth-Moon-Sun system. The situation is similar in Quantum Mechanics, where we can compute analytically the energy levels of the hydrogen atom (a proton and an electron), but not those of the helium atom (two electrons and a nucleus). Therefore, to study complicated systems of this type, we need to resort either to approximation schemes such as perturbation theory, numerical techniques, or both.

Regarding classical many-body systems, we can numerically solve them deterministically from Newton's equations. This is conceptually simple but very dependent on several factors such as the numerical precision and the discrete time-step. The dynamics of the system is highly sensible to small variations in initial conditions, which accumulate drastic effects after a sufficiently long time and lead to a completely different solution from the microscopic point of view (*chaos*). However, there is a difference between few-body problems and many-body problems. In the later we are usually mostly interested in the aggregate behavior, and therefore we can tolerate deviations from the exact solution at small length scales because on the aggregate, statistical mechanical properties are preserved by the simulation when considering non-critical regions of the phase space. In fact, many simulations of materials in classical regimes are not done deterministically but rather using Monte Carlo techniques.

The important thing to note about these classical many-body problems is that even though they may exhibit complexity and chaos, the classical computer can already provide very accurate solutions to many useful instances of these problems. For example, in the field of celestial dynamics one routinely applies numerical techniques to describe the motion of planets and space probes sent around the solar system with an accuracy that can already be made to match observational data.

This is not the case for quantum many-body problems. Quantum many-body systems live in such a large

state space, that one can not hope to solve them accurately from first principles using classical computers for most meaningful sized systems. Consider, for instance,  $N$  classical interacting particles in three dimensions. Such a system has  $3N$  degrees of freedom and its state is described by the Hamiltonian dynamics of a single point moving in a  $6N$ -dimensional phase space. In contrast, a quantum many-body system made of  $N$  parties, each living in a  $d$ -dimensional Hilbert space, is described by a density operator that can give us its probability distribution in any representation and lives in a  $d^N$ -dimensional Hilbert space,

$$\mathcal{H} = \mathbb{C}^{(d^N)} \quad (3.1)$$

Thus, while the state space of a classical system scales only linearly with the number of particles, for quantum systems it scales exponentially. This renders quantum systems exponentially harder to simulate by classical computers and led to Feynman famously proposing the idea of quantum simulation in a 1982 paper [11] cited many times,

*“And I’m not happy with all the analyses that go with just the classical theory, because nature isn’t classical, dammit, and if you want to make a simulation of nature, you’d better make it quantum mechanical, and by golly it’s a wonderful problem, because it doesn’t look so easy.”*

Of course, matter behaves classically in many regimes where there is no need for a quantum description. But when we study systems at very small length scales, or low temperatures or very isolated, quantum behavior has to be taken into account. The way we approach the quantum world really depends on how we look at it. We can think in terms of complex atoms and molecules, quarks and gluons, electrons and protons, even as qubits. Some examples of physical domains where quantum many-body problems exist which don’t yet fully understand include:

**Electronic properties of condensed matter systems.** These invariably require the use of a quantum description of the electronic many-body problem. Whereas the behavior of some systems can be described relying on an independent electron picture, several very interesting problems remain open, such as the mechanism for high  $T_c$  superconductivity.

**Quantum chemistry.** Electrons in a molecule’s orbital take on a very complicated and difficult to understand quantum state. Here, the problem is to describe the electronic ground state of a molecule as a function of the atomic positions. The problem is not tractable using conventional techniques except for tiny molecules, such as  $H_2$ .

**High-energy physics.** We may be interested in studying relativistic regimes where particle number is not even conserved. For instance, in a quark-gluon plasma governed by the strong interaction, which occurs at very high temperatures when protons and neutrons melt producing a new state of matter.

## 3.2 Condensed matter *theory of everything*

A large fraction of the very broad field of condensed matter physics is devoted to the study of the electronic properties of materials. Materials can exhibit a very wide range of electronic personalities. If we consider their electrical conductivity, for instance, it can change by 16 orders of magnitude. This diversity of electronic regimes includes conductors, insulators, semiconductors, metals, superconductors, ferromagnets, antiferromagnets... In principle, the theory for all of these interactions can be encoded in the same Hamiltonian,

$$H = T^{(1)} + V_{ions}^{(1)} + V_{e-e}^{(2)} \quad (3.2)$$



describing the kinetic energy of the electrons, their Coulomb interaction with the ions, and the electron-electron interaction. The kinetic energy of the nuclei could also be considered, but that plays a secondary role, therefore we ignore it, thus performing our first simplification.

This schematic Hamiltonian is sometimes called the “condensed matter theory of everything”. If we deal with very small systems, such as the  $H_2$  molecule or the  $He$  atom, we can solve this Hamiltonian exactly, and this describes these simple systems very well. But we can’t go very far with this “theory of everything” since this quickly takes on a very complicated form where there is no hope to find a numerical solution, let alone an analytical one.

Thus, an array of different approximation techniques are used to deal this, such as the Hartree-Fock method or density functional theory. The problem becomes soluble if we treat the interactions between electrons as interactions of an electron with an effective field

$$H = T^{(1)} + V_{ions}^{(1)} + V_{eff}^{(2)} \quad (3.3)$$

The good news about this approach is that if we ignore electron-electron interactions, or treat them in an approximate manner, we can solve the Hamiltonian, and this describes a very large amount of materials, most often the less interesting, such as insulators, conventional metals, or semiconductors. The bad news is that very important problems can’t be solved this way, such as the origin of high  $T_c$  superconductors, that have challenged theorists for more than 30 years now, and for which the  $e - e$  interaction is crucial and we can not ignore it.

### 3.3 Strongly entangled systems

To begin to understand QMB systems that exhibit complex behavior, we put aside the usual model thinking for a while to recall the most basic property of quantum mechanics. QM is fundamentally non-deterministic in nature and this is what allows the existence of the non-local correlations we can measure in the lab [79]. Non-locality is a basic property of quantum mechanics with no classical counterpart, and it is the distinctive feature of entangled states.

**Definition 3.1.** A state  $|\Psi\rangle \in \mathcal{H}$  of a QMB system  $S$  is said to be entangled when it can not be written as a product

$$|\Psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle \quad (3.4)$$

of states  $|\psi_i\rangle \in \mathcal{H}_i$  from the subsystems  $S_i$  ( $i = 1, \dots, n$ ) making up the composite system with state space  $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_n$ .

We start by noting that almost all states in the Hilbert space are entangled. This is easy to conclude from the fact that we can write superpositions of states. More than that, almost always two particles that start in a separable state go entangled when they are allowed to interact, and almost all Hamiltonians have their eigenstates entangled. For instance, two  $S = 1/2$  spins interacting through

$$H = J\vec{\sigma}_1 \cdot \vec{\sigma}_2 = \begin{pmatrix} J & 0 & 0 & 0 \\ 0 & -J & 2J & 0 \\ 0 & 2J & -J & 0 \\ 0 & 0 & 0 & J \end{pmatrix} \quad (3.5)$$

have an entangled ground state

$$|\Psi_0\rangle = \frac{1}{\sqrt{2}} (|10\rangle - |01\rangle), \quad (3.6)$$

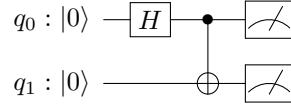
where we know everything about the collective system, but not about the state of each spin individually. In an entangled state, the state's complexity is spread all over the system and each subsystem cannot be described independently of the others.

Furthermore, there can be different degrees of entanglement, and several measures try to quantify it, for instance the von Neumann entropy (Eq.2.94). Highly entangled states occupy a large fraction of the full Hilbert space of the QMB system. Being able to characterize and understand the intricate correlations in the multipartite entanglement structure these states is therefore very important to study materials in these regimes. Since such states require a lot of information to be described, this justifies the need for an exact approach. A quantum computer is one such tool which will enable us to create, manipulate and experiment with novel and exotic entangled states in the laboratory [80, 81, 82].

**Example 3.2.** The minimal example of entanglement is that which can be constructed on a system composed of only two subsystems, each of which with only two energy levels. The *Bell states* are the orthonormal basis set of four maximally entangled states of two qubits,

$$\begin{aligned} |\Phi^+\rangle &= \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle) & |\Psi^+\rangle &= \frac{1}{\sqrt{2}} (|01\rangle + |10\rangle) \\ |\Phi^-\rangle &= \frac{1}{\sqrt{2}} (|00\rangle - |11\rangle) & |\Psi^-\rangle &= \frac{1}{\sqrt{2}} (|01\rangle - |10\rangle) \end{aligned} \quad (3.7)$$

It is trivial to produce these states with a quantum circuit. We illustrate how to do it for  $|\Phi^+\rangle$ ,



$$cX_{01}H_0 |00\rangle = cX_{01} \left( \frac{|0\rangle + |1\rangle}{\sqrt{2}} |0\rangle \right) = \frac{|00\rangle + |11\rangle}{\sqrt{2}} = |\Phi^+\rangle$$

To be able to perform universal quantum computing, at least one entangling gate is necessary (e.g.  $cX$  or  $cZ$ ). It is a remarkable insight how even the most complex entangled states can be made entirely out of a single 2-qubit gate together with the other 1-qubit gates.

When looking for the eigenstates of some Hamiltonian, we are most often interested in the ground state and lower excited states, as they have a predominant weigh in the partition function at low temperature. Furthermore, these are usually the most entangled states since when temperature increases, the states get increasingly more mixed and less entangled. With the energies we can easily compute time evolution, and from the partition function study thermal properties.

### 3.4 Simple spin Hamiltonians

Before dreaming too big, and since the purpose of this project is to actually study Hamiltonians on real quantum computers, let us start by defining two very simple spin Hamiltonians to be used later when we discuss the quantum simulation procedures. The first one is simply a two level system with an energy splitting  $c$ ,

$$H_Z = c \sigma^z. \quad (3.8)$$

As we have seen, this is the model for a qubit, or in the spin language, the Zeeman Hamiltonian of a spin 1/2 particle. Its eigensystem is

$$\begin{cases} \varepsilon_1 = c & , |0\rangle \\ \varepsilon_2 = -c & , |1\rangle \end{cases} \quad (3.9)$$

For the second Hamiltonian we will be increasing the Hilbert space dimension to 4 and consider 2 spins connected by an Ising interaction that preserves the total  $S_z$ ,

$$H_I = c_0\sigma_0^z + c_1\sigma_1^z + J\sigma_0^z\sigma_1^z. \quad (3.10)$$

It can be easily checked that  $H_I$  is characterized by the spectrum

$$\begin{cases} \varepsilon_1 = c_0 + c_1 + J & , |00\rangle \\ \varepsilon_2 = c_1 - c_0 - J & , |01\rangle \\ \varepsilon_3 = c_0 - c_1 - J & , |10\rangle \\ \varepsilon_4 = J - c_0 - c_1 & , |11\rangle \end{cases} \quad (3.11)$$

where the convention we use on labeling the basis states is  $|site_1, site_0\rangle$ .

These are two of the simplest possible spin Hamiltonians but will be our starting point when we come to study quantum simulations on the prototypical quantum computers currently available.

### 3.5 A first look at the Hubbard model: the 2-site Hamiltonian

The simplest yet important class of models to study systems of strongly correlated electrons is the Hubbard model. We will first define the Hamiltonian, briefly mention some of its consequences while skipping many of its interesting features, to jump right into exploring a very simple instance of the problem.

Take a system of non-interacting electrons on a  $M$ -site lattice where a single level (orbital) sits at each site. The second-quantized Hamiltonian reads

$$\hat{H}_\tau = \sum_{i,j,\sigma} \tau_{ij} c_{i,\sigma}^\dagger c_{j,\sigma}, \quad (3.12)$$

where  $\sigma = \uparrow, \downarrow$  is the  $\hat{S}_z$  spin projection, indices  $i, j$  label the sites and  $c_{i,\sigma}^\dagger$  ( $c_{i,\sigma}$ ) is the creation (annihilation) operator of a fermion in the single particle state  $|\phi_{i,\sigma}\rangle$  at site  $i$ . The matrix of coefficients  $\tau_{ij} = \langle \phi_{i,\sigma} | \left( -\frac{\hbar^2 \nabla^2}{2m} + \hat{V} \right) | \phi_{j,\sigma} \rangle$  accounts for the kinetic and external potential energies that characterize a single-particle hopping from site  $j$  to  $i$ . In the situations we are interested in,  $\tau_{ij}$  decays sufficiently fast that it is acceptable to reduce counting of this  $C_2^M \sim M^2$  pairs to  $\sim M$  by assuming only nearest neighbors have non-zero hopping terms, all equal to  $-\tau$ , with  $\tau > 0$ . With this assumption, the modified free particle Hamiltonian is written in explicit hermitian form as

$$\hat{H}_t = -\tau \sum_{\langle i,j \rangle, \sigma} c_{i\sigma}^\dagger c_{j\sigma} + c_{j\sigma}^\dagger c_{i\sigma}. \quad (3.13)$$

This would be the free electron band theory if only the hopping term were present, but we want to introduce interactions. In the Hubbard Hamiltonian, one considers only an on-site interaction energy,  $U > 0$ , which is a local electrical repulsion between electrons on the same orbital, which we will take as site-independent. Expressed in terms of the number operator of a single particle state,  $\hat{n}_{i\sigma} = c_{i\sigma}^\dagger c_{i\sigma}$ , the interaction term  $\hat{H}_U$  is added to  $\hat{H}_\tau$  to get

$$\hat{H} = -\tau \sum_{\langle i,j \rangle, \sigma} \left( c_{i\sigma}^\dagger c_{j\sigma} + c_{j\sigma}^\dagger c_{i\sigma} \right) + U \sum_i \hat{n}_{i\uparrow} \hat{n}_{i\downarrow}. \quad (3.14)$$

The reason for neglecting long range Coulomb interactions, which might be important for some systems, can be justified for d-shell metals, but could also have been found phenomenologically. The single-band

Hubbard problem definition is complete with this Hamiltonian together with the fermionic anti-commutation relations,

$$\{c_{i\alpha}, c_{j\beta}^\dagger\} = \delta_{ij}\delta_{\alpha\beta}, \quad \{c_{i\alpha}, c_{j\beta}\} = 0, \quad \{c_{i\alpha}^\dagger, c_{j\beta}^\dagger\} = 0. \quad (3.15)$$

At this point it might be of prospective interest to point out the beauty hiding in the simple statement of this model from which can emerge a rich variety of phases for different ratios of  $\tau$  and  $U$ , lattice geometry and electron density. Some of these non-trivial phases and their transitions lend themselves to advanced treatments by strongly-coupled quantum field theories, connecting to gauge theories, conformal invariance and the AdS/CFT correspondence [83].

In its general case, the Schrödinger equation for this model cannot be solved exactly with analytical methods<sup>1</sup>, i.e. its characteristic equation can't be analytically solved, but numerical techniques such as exact diagonalization can in principle provide this solution. This is only possible, however, for instances of the problem with a small Hilbert space, due to the high computational cost of the technique in both time and space complexity. Nevertheless, there are known analytical solutions for a few particularly simple cases of the Hubbard model.

Due to the Pauli exclusion principle, a single lattice site holds four possible states:  $|0\rangle, |\uparrow\rangle, |\downarrow\rangle, |\uparrow\downarrow\rangle$  (explicitly, in the spin occupation direct product basis these are, respectively,  $|0\rangle_\downarrow \otimes |0\rangle_\uparrow, |0\rangle_\downarrow \otimes |1\rangle_\uparrow, |1\rangle_\downarrow \otimes |0\rangle_\uparrow, |1\rangle_\downarrow \otimes |1\rangle_\uparrow$ ). Therefore, there will be  $4^M$  states for the M-site single-band Hubbard model and the Hilbert space dimension scales exponentially in the number of sites, quickly leading to difficulties with the exact treatment of even moderate size clusters numerically.

**Lemma 3.3.** *The number operator,  $\hat{N} = \sum_{i\sigma} \hat{n}_{i\sigma}$ , commutes with the Hamiltonian.*

*Proof.* \_\_\_\_\_

As the number of creation and annihilation operators is the same, term by term, it can be easily seen that  $H$  commutes with total  $N$ . A brute force demonstration of this is given here:

$$\begin{aligned} \hat{H}_\tau \hat{N} &= -\tau \sum_{\langle i,j \rangle \sigma} \left( c_{i\sigma}^\dagger c_{j\sigma} + c_{j\sigma}^\dagger c_{i\sigma} \right) \left( \sum_{i\sigma} \hat{n}_{i\sigma} \right) \\ &= -\tau \sum_{\langle i,j \rangle \sigma k \rho} \left( c_{i\sigma}^\dagger c_{j\sigma} + c_{j\sigma}^\dagger c_{i\sigma} \right) \hat{n}_{k\rho} \\ &= -\tau \sum_{\langle i,j \rangle \sigma k \rho} \left( c_{i\sigma}^\dagger c_{j\sigma} \right) \hat{n}_{k\rho} + \left( c_{j\sigma}^\dagger c_{i\sigma} \right) \hat{n}_{k\rho} \\ &\stackrel{(a)}{=} -\tau \sum_{\langle i,j \rangle \sigma k \rho} \hat{n}_{k\rho} c_{i\sigma}^\dagger c_{j\sigma} + (\delta_{kj}\delta_{\rho\sigma} - \delta_{ki}\delta_{\rho\sigma}) c_{i\sigma}^\dagger c_{j\sigma} + \hat{n}_{k\rho} c_{j\sigma}^\dagger c_{i\sigma} + (\delta_{ki}\delta_{\rho\sigma} - \delta_{kj}\delta_{\rho\sigma}) c_{j\sigma}^\dagger c_{i\sigma} \\ &= -\tau \sum_{\langle i,j \rangle \sigma k \rho} \hat{n}_{k\rho} \left( c_{i\sigma}^\dagger c_{j\sigma} + c_{j\sigma}^\dagger c_{i\sigma} \right) + \delta_{kj}\delta_{\rho\sigma} \left( c_{i\sigma}^\dagger c_{j\sigma} - c_{j\sigma}^\dagger c_{i\sigma} \right) + \delta_{ki}\delta_{\rho\sigma} \left( c_{j\sigma}^\dagger c_{i\sigma} - c_{i\sigma}^\dagger c_{j\sigma} \right) \\ &= \hat{H}_\tau \hat{N} - \tau \sum_{\langle i,j \rangle \sigma k \rho} \delta_{kj}\delta_{\rho\sigma} \left( c_{i\sigma}^\dagger c_{j\sigma} - c_{j\sigma}^\dagger c_{i\sigma} \right) - \tau \sum_{\langle i,j \rangle \sigma k \rho} \delta_{ki}\delta_{\rho\sigma} \left( c_{j\sigma}^\dagger c_{i\sigma} - c_{i\sigma}^\dagger c_{j\sigma} \right) \\ &= \hat{H}_\tau \hat{N} - \tau \sum_{\langle i,j \rangle \sigma} \left( c_{i\sigma}^\dagger c_{j\sigma} - c_{j\sigma}^\dagger c_{i\sigma} \right) - \tau \sum_{\langle i,j \rangle \sigma} \left( c_{j\sigma}^\dagger c_{i\sigma} - c_{i\sigma}^\dagger c_{j\sigma} \right) \\ &= \hat{H}_\tau \hat{N} \end{aligned}$$

---

<sup>1</sup>Except for a 1D chain with the Bethe ansatz.

$$\begin{aligned}
\hat{H}_U \hat{N} &= (U \sum_i \hat{n}_{i\uparrow} \hat{n}_{i\downarrow}) (\sum_i \hat{n}_{i\uparrow} + \hat{n}_{i\downarrow}) \\
&= U \sum_{ij} \hat{n}_{i\uparrow} \hat{n}_{i\downarrow} (\hat{n}_{j\uparrow} + \hat{n}_{j\downarrow}) \\
&= U \sum_{ij} (\hat{n}_{j\uparrow} + \hat{n}_{j\downarrow}) \hat{n}_{i\uparrow} \hat{n}_{i\downarrow} \\
&\stackrel{(b)}{=} (\sum_j \hat{n}_{j\uparrow} + \hat{n}_{j\downarrow}) (U \sum_i \hat{n}_{i\uparrow} \hat{n}_{i\downarrow}) \\
&= \hat{N} \hat{H}_U,
\end{aligned}$$

where at (a) we made use of  $[c_{i\alpha}^\dagger c_{j\beta}, \hat{n}_{k\gamma}] = (\delta_{kj} \delta_{\gamma\beta} - \delta_{ki} \delta_{\gamma\alpha}) c_{i\alpha}^\dagger c_{j\beta}$ , and at (b) of  $[\hat{n}_\alpha, \hat{n}_\beta] = 0$ .

□

Since  $[\hat{H}, \hat{N}] = 0$  both operators can be diagonalized simultaneously and  $\hat{H}$  can be brought to a block diagonal form having the total number of electrons conserved within each sector. Typically one is in fact not interested in the full configuration space of the multi-site problem, but only on those states from an ensemble with fixed number of electrons, which the Hamiltonian operator will keep constant. For  $M$  sites and  $N \leq 2M$  electrons there are  $C_N^{2M}$  available basis states. The configuration in which there are as many electrons as there are lattice sites, i.e. when  $M = N$ , is referred to as half-filling.

Consider the case of a Hubbard model with 2 sites. Table 3.1 shows all possible configuration states. The half-filling case of the 2-site Hubbard Hamiltonian is one of those instances with an exact analytical solution, and it can be used as a toy model for the two electrons interacting in a  $H_2$  molecule when these are restricted to occupy only the  $1s$  shells and one ignores all other small interaction terms<sup>2</sup>. This has been extensively studied already and we will now review it.

|                             | $N = 0$ | $N = 1$ | $N = 2$ | $N = 3$ | $N = 4$ |
|-----------------------------|---------|---------|---------|---------|---------|
| Possible<br>basis<br>states | ○○      | ↑○      | ↑ ↑     | ↑↓ ↑    | ↑↓ ↑↓   |
|                             |         | ↓○      | ↑ ↓     | ↑↓ ↓    |         |
|                             |         | ○↑      | ↓ ↓     | ↑ ↑↓    |         |
|                             |         | ○↓      | ↓ ↑     | ↓ ↑↓    |         |
|                             |         |         | ↑↓○     |         |         |
|                             |         |         | ○↑↓     |         |         |
| # basis states              | 1       | 4       | 6       | 4       | 1       |

Table 3.1: Pictorial representation of the 16 dimensional Hilbert space for the 2-site Hubbard model.

Fermionic commutation relations show it is not enough to say a site is doubly occupied, because such outcome could be achieved with different phase factors by acting on the vacuum in two ways:  $c_{i\uparrow}^\dagger c_{i\downarrow}^\dagger |0\rangle = -c_{i\downarrow}^\dagger c_{i\uparrow}^\dagger |0\rangle$ . The same reasoning applies if we are creating fermions on different sites. Therefore, when defining basis states one should always stick with some convention on the order of creation operators. We will follow the convention on [85]: ( $1^{st}$ ) place spin- $\downarrow$  operators to the left of spin- $\uparrow$  operators, and ( $2^{nd}$ ) place operators on lower index sites to the right (sites being labeled from 1 to  $M$ ). See table 3.2.

Explicitly, the Hamiltonian we are now considering reads

<sup>2</sup>For your amusement, and if you like having a visual intuition of physics models, you might want to check the following to understand how to properly picture the  $H_2$  molecule: [84]

| ② ①                        | Label            | Algebraic   | $ n_{2\downarrow}n_{1\downarrow}n_{2\uparrow}n_{1\uparrow}\rangle$ | $\hat{S}_z$ |
|----------------------------|------------------|---|--|-------------|
| $\uparrow \uparrow$        | $ \phi_1\rangle$ | $c_{2\uparrow}^\dagger c_{1\uparrow}^\dagger  0\rangle$     | $ 0011\rangle$   | 1           |
| $\circ \uparrow\downarrow$ | $ \phi_2\rangle$ | $c_{1\downarrow}^\dagger c_{1\uparrow}^\dagger  0\rangle$   | $ 0101\rangle$   | 0           |
| $\uparrow \downarrow$      | $ \phi_3\rangle$ | $c_{1\downarrow}^\dagger c_{2\uparrow}^\dagger  0\rangle$   | $ 0110\rangle$   | 0           |
| $\downarrow \uparrow$      | $ \phi_4\rangle$ | $c_{2\downarrow}^\dagger c_{1\uparrow}^\dagger  0\rangle$   | $ 1001\rangle$   | 0           |
| $\uparrow\downarrow \circ$ | $ \phi_5\rangle$ | $c_{2\downarrow}^\dagger c_{2\uparrow}^\dagger  0\rangle$   | $ 1010\rangle$   | 0           |
| $\downarrow \downarrow$    | $ \phi_6\rangle$ | $c_{2\downarrow}^\dagger c_{1\downarrow}^\dagger  0\rangle$ | $ 1100\rangle$   | -1          |

Table 3.2: Different representations of basis states for 2 sites and 2 electrons, and their respective  $\hat{S}_z$  value. We see  $|\phi_2\rangle$  and  $|\phi_5\rangle$  are singlet states (all electrons are paired and  $\hat{S}_z = 0$ ) while the other basis states are triplets (two unpaired electrons).

$$\begin{aligned}
\hat{H} &= \hat{H}_\tau + \hat{H}_U \\
&= -\tau \sum_\sigma \left( c_{1\sigma}^\dagger c_{2\sigma} + c_{2\sigma}^\dagger c_{1\sigma} \right) + U \sum_{i=1}^2 \hat{n}_{i\uparrow} \hat{n}_{i\downarrow} \\
&= -\tau \left( c_{1\uparrow}^\dagger c_{2\uparrow} + c_{2\uparrow}^\dagger c_{1\uparrow} + c_{1\downarrow}^\dagger c_{2\downarrow} + c_{2\downarrow}^\dagger c_{1\downarrow} \right) + U (\hat{n}_{1\uparrow} \hat{n}_{1\downarrow} + \hat{n}_{2\uparrow} \hat{n}_{2\downarrow}).
\end{aligned} \tag{3.16}$$

We would now like to find the spectra for this Hamiltonian. The matrix components of  $\hat{H}$  can be found by evaluating its action on the complete basis set in Table 3.2. We first note that the total spin,  $\hat{S}_z = 1/2 \sum_i (\hat{n}_{i\uparrow} - \hat{n}_{i\downarrow})$ , is also conserved by the general Hubbard Hamiltonian,  $[\hat{S}_z, \hat{H}] = 0$ , independently of the number of sites and lattice geometry. This is a consequence of  $[\hat{H}, \hat{n}_{i\sigma}] = 0$ , as already shown. Since  $\hat{H}$  doesn't apply transitions between different total spin states, the 2-electron  $6 \times 6$  block of the 16 dimensional representation of the 2-site Hamiltonian can be further reduced to a direct sum of  $\hat{S}_z = -1, 0, 1$  sectors.

Since there is only one state in the matrix blocks with  $\hat{S}_z = 1$  and  $\hat{S}_z = -1$ , we immediately see  $|\phi_1\rangle$  and  $|\phi_6\rangle$  are eigenstates of  $\hat{H}$ . To find the associated eigenvalues we first verify the kinetic part of the Hamiltonian is zero for these states, because both sites have equal spin and hopping is forbidden by the Pauli exclusion principle, since it would result in an orbital with two equal spin electrons. As for the  $\hat{H}_U$  part, when it is applied to a basis state it counts the number of orbitals with two electrons of such state and returns the state vector multiplied by that number and  $U$ . Therefore,  $\hat{H}_U$  acting on  $|\phi_1\rangle$  and  $|\phi_6\rangle$  also yields zero and one sees these eigenkets have eigenvalues equal to 0. This reasoning can be confirmed by explicitly evaluating  $\hat{H}$  applied to  $|\phi_1\rangle$  and  $|\phi_6\rangle$ .

To determine the matrix elements in the  $\hat{S}_z = 0$  block we must compute  $\hat{H}$  applied to the basis vectors in this sector. For the reason explained previously, we can already expect  $\hat{H}_U$  will only give non zero value when acting on  $|\phi_2\rangle$  and  $|\phi_5\rangle$ . By performing the calculation one obtains the following relations,

$$\begin{aligned}
\hat{H} |\phi_2\rangle &= -\tau (|\phi_3\rangle + |\phi_4\rangle), \\
\hat{H} |\phi_3\rangle &= -\tau (|\phi_2\rangle + |\phi_5\rangle), \\
\hat{H} |\phi_4\rangle &= -\tau (|\phi_2\rangle + |\phi_5\rangle), \\
\hat{H} |\phi_5\rangle &= -\tau (|\phi_3\rangle + |\phi_4\rangle).
\end{aligned} \tag{3.17}$$

Hence, the matrix representation of the Hamiltonian is

$$\hat{H} = \left( \begin{array}{c|cccc|c} 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & U & -\tau & -\tau & 0 & 0 \\ 0 & -\tau & 0 & 0 & -\tau & 0 \\ \hline 0 & -\tau & 0 & 0 & -\tau & 0 \\ 0 & 0 & -\tau & -\tau & U & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right). \quad (3.18)$$

To derive the full set of the eigenvalues of this Hamiltonian one is left with the task of solving the characteristic equation for the middle block,

$$\varepsilon^2 (\varepsilon^4 - 4\varepsilon^2\tau^2 + 4\varepsilon\tau^2U + \varepsilon^2U^2 - 2\varepsilon^3U) = 0, \quad (3.19)$$

which has four exact solutions. Together with the previously found two eigenvalues, the complete spectrum of the two-site two-electron Hubbard Hamiltonian, indexed in increasing order from the ground state to the highest excited state is

$$\begin{aligned} \varepsilon_1 &= U/2 \left( 1 - \sqrt{16(\tau/U)^2 + 1} \right), \\ \varepsilon_{2,3,4} &= 0, \quad (3 - fold) \\ \varepsilon_5 &= U, \\ \varepsilon_6 &= U/2 \left( 1 + \sqrt{16(\tau/U)^2 + 1} \right), \end{aligned} \quad (3.20)$$

and the corresponding eigenkets are

$$\begin{aligned} |\psi_1\rangle &= \left( \frac{(U + \sqrt{16\tau^2 + U^2})^2}{8\tau^2} + 2 \right)^{-1/2} \left( (|\phi_2\rangle + |\phi_5\rangle) + \frac{U + \sqrt{16\tau^2 + U^2}}{4\tau} (|\phi_3\rangle + |\phi_4\rangle) \right), \\ |\psi_2\rangle &= |\phi_1\rangle, \\ |\psi_3\rangle &= |\phi_6\rangle, \\ |\psi_4\rangle &= (|\phi_4\rangle - |\phi_3\rangle) / \sqrt{2}, \\ |\psi_5\rangle &= (|\phi_5\rangle - |\phi_2\rangle) / \sqrt{2}, \\ |\psi_6\rangle &= \left( \frac{(U - \sqrt{16\tau^2 + U^2})^2}{8\tau^2} + 2 \right)^{-1/2} \left( (|\phi_2\rangle + |\phi_5\rangle) + \frac{U - \sqrt{16\tau^2 + U^2}}{4\tau} (|\phi_3\rangle + |\phi_4\rangle) \right). \end{aligned} \quad (3.21)$$

This finishes the diagonalization task, but we can now also show how this problem can be simplified in the limiting region of  $\tau \ll U$ . By Taylor expanding the eigenvalues in the limit  $\tau/U \ll 1$ , we see that the eigenstates fall into two groups: the low energy sector, corresponding to the ground state and the three triplet states (excited), and the high energy sector, constituted by the other excited singlet states.

$$\begin{aligned} \varepsilon_1 &= U/2 \left( 1 - \sqrt{16(\tau/U)^2 + 1} \right) \approx -4U(\tau/U)^2 + O((\tau/U)^3) \approx -4\tau^2/U \approx 0 \\ \varepsilon_{2,3,4} &= 0, \quad \approx 0 \quad \approx 0 \quad \approx 0 \\ \varepsilon_5 &= U, \quad \approx U \quad \approx U \quad \approx U \\ \varepsilon_6 &= U/2 \left( 1 + \sqrt{16(\tau/U)^2 + 1} \right) \approx U + 4U(\tau/U)^2 + O((\tau/U)^3) \approx U + 4\tau^2/U \approx U \end{aligned} \quad (3.22)$$

For  $\tau/U \ll 1$ , charge fluctuations are no longer favorable because of the Coulomb repulsion between two electrons in the same orbital, and one has an one-electron-per-orbital configuration (remember we are in the

half-filling sector of the Hamiltonian). Because of this, we can forget the hopping degree of freedom, being left only with the spin degree of freedom. The low-energy spectrum calculated above can be retrieved by modeling only the spin with the Heisenberg Hamiltonian

$$\hat{H} = \hat{K} + J\hat{\mathbf{S}}_1 \cdot \hat{\mathbf{S}}_2,$$

where  $\hat{K} = (-\tau^2/U) \mathbf{1}_{4 \times 4}$  and  $J = 4\tau^2/U > 0$  is the exchange constant for  $\tau/U \ll 1$ , antiferromagnetic. The spin-1/2 operator of the  $i^{\text{th}}$  orbital is expressed in terms of the Pauli matrices as  $\hat{\mathbf{S}} = (\sigma^x, \sigma^y, \sigma^z)/2$ . Therefore,

$$\begin{aligned} \hat{H} &= \hat{K} + J\hat{\mathbf{S}}_1 \cdot \hat{\mathbf{S}}_2 \\ &= K\mathbf{1}_{4 \times 4} + \frac{J}{4} (\sigma_1^x \otimes \sigma_2^x + \sigma_1^y \otimes \sigma_2^y + \sigma_1^z \otimes \sigma_2^z) \\ &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -\frac{2\tau^2}{U} & \frac{2\tau^2}{U} & 0 \\ 0 & \frac{2\tau^2}{U} & -\frac{2\tau^2}{U} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \end{aligned} \tag{3.23}$$

and one has spectrum  $\{\varepsilon\} = \left\{ -\frac{4\tau^2}{U}, 0, 0, 0 \right\}$ .



## Chapter 4

# Digital quantum simulation of time-evolution

*“If you can’t solve a problem, then there is an easier problem you can solve: find it.”*

— George Pólya, 1957

Already in 1929, Paul Dirac motivated the need to find approximate methods to study quantum systems, on account of the main difficulty left by quantum theory being that the exact application of its laws leads to equations which are too complicated to solve [86]. Motivated by this reasoning, over time we developed many methods and models with which we can find solutions to some problems, sometimes exactly.

When computers became available, the community naturally started wondering how to approach the problem with computation, and the simulation of quantum systems has long become a goal for Physics. In the Soviet Russia of 1975, R. P. Poplavskii argued that classical computers are inherently unable of simulating quantum systems due to the requirements imposed by the superposition principle [87]. Nevertheless, many clever classical computational algorithms have been developed to try and tackle the problem, such as Monte Carlo methods and tensor networks. The later have become quite powerful mathematical tools that are able to find in a subspace of the full Hilbert space of the problem the relevant correlation patterns to efficiently approximate solutions of certain Hamiltonians. However, their efficiency imposes some requirements on the systems that can be tackled, such as that of being only locally entangled [88, 89].

Simulating a generic quantum system with strong correlations appears to be a truly difficult problem for classical computers, as the full Hilbert space description of the problem resists compression. In 1982, Feynman proposed the approach of developing controlled quantum systems with which to simulate other quantum systems [11] and in 1996 Lloyd proposed the first theoretical solution to such problem [18]. The path of developing techniques of simulating quantum systems using other quantum systems has been followed by many since then [44, 90, 45, 46, 47, 48].

### 4.1 Jordan-Wigner mapping to the qubit space

The way in which we want to simulate a fermionic system with a quantum computer is by actually emulating the algebra of fermions with the algebra of qubits, effectively mapping the system we want to simulate into the computational objects we can manipulate. As we have seen, qubits are distinguishable two-level quantum systems which can be described as spin  $1/2$  particles, however, the electronic system we want to simulate is composed of indistinguishable fermions. The quantum states of these two types of systems are governed by

different algebras. While fermions are described by the algebra in Eq.3.15, the  $SU(2)$  algebra of Pauli spin operators describes spin 1/2 particles. As we have seen in section 2.2, these are given by

$$\sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad (4.1)$$

and naturally anti-commute on the same site,

$$\{\sigma^\alpha, \sigma^\beta\} = 2\delta^{\alpha\beta}\mathbf{1}. \quad (4.2)$$

The raising/lowering operators are

$$\sigma^+ = |1\rangle\langle 0| = \frac{1}{2}(\sigma^x - i\sigma^y) = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \quad (4.3)$$

$$\sigma^- = |0\rangle\langle 1| = \frac{1}{2}(\sigma^x + i\sigma^y) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad (4.4)$$

with the usual choice for the representation of basis vectors as

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad (4.5)$$

and

$$\sigma_+ |0\rangle = |1\rangle, \quad \sigma_+ |1\rangle = 0, \quad \sigma_- |1\rangle = |0\rangle, \quad \sigma_- |0\rangle = 0. \quad (4.6)$$

**Definition 4.1.** Labeling the  $n$  components of number basis for the fermionic Fock space in the form

$$|\mathbf{f}\rangle = |f_{n-1}, \dots, f_1, f_0\rangle, \quad (4.7)$$

where  $f_i = \{0, 1\}$ , the Jordan-Wigner transformation of the creation and annihilation operators, originally presented in [91], is given by

$$\begin{aligned} c_j &\Leftrightarrow \mathbf{1}^{\otimes n-j-1} \otimes \sigma^- \otimes \sigma^z \otimes j, \\ c_j^\dagger &\Leftrightarrow \mathbf{1}^{\otimes n-j-1} \otimes \sigma^+ \otimes \sigma^z \otimes j. \end{aligned} \quad (4.8)$$

**Theorem 4.2.** The fermionic algebra is conserved by the operator identity defined in Eq.4.8, i.e.

$$\{c_i, c_j^\dagger\} = \delta_{ij}\mathbf{1}. \quad (4.9)$$

*Proof.* \_\_\_\_\_

We can prove this while also checking that  $\hat{n}_j = c_j^\dagger c_j = \mathbf{1}^{\otimes n-j-1} \otimes \sigma^+ \sigma^- \otimes \mathbf{1}^{\otimes j}$ . First, we compute the number operator  $\hat{n}_j = c_j^\dagger c_j$  as follows

$$\begin{aligned} c_j^\dagger c_j &= (\mathbf{1}^{\otimes n-j-1} \otimes \sigma^+ \otimes \sigma^z \otimes j) (\mathbf{1}^{\otimes n-j-1} \otimes \sigma^- \otimes \sigma^z \otimes j) \\ &= (\mathbf{1}^{\otimes n-j-1})^2 \otimes \sigma^+ \sigma^- \otimes (\sigma^z \otimes j)^2 \\ &= \mathbf{1}^{\otimes n-j-1} \otimes \sigma^+ \sigma^- \otimes (\sigma^z)^2 \otimes j \\ &= \mathbf{1}^{\otimes n-j-1} \otimes \sigma^+ \sigma^- \otimes \mathbf{1}^{\otimes j} \\ &= [\mathbf{1}^{\otimes n-j-1} \otimes (\mathbf{1} - \sigma^z) \otimes \mathbf{1}^{\otimes j}] / 2, \end{aligned} \quad (4.10)$$

where we have used  $(A \otimes B)(C \otimes D) = AC \otimes BD$ ,  $(\sigma^z)^2 = \mathbf{1}$ , and also

$$\begin{aligned}\sigma^+\sigma^- &= (\sigma^x - i\sigma^y)(\sigma^x + i\sigma^y)/4 \\ &= \left( (\sigma^x)^2 + i[\sigma^x, \sigma^y] + (\sigma^y)^2 \right) / 4 \\ &= (\mathbf{1} - \sigma^z) / 2.\end{aligned}\tag{4.11}$$

The  $c_j c_j^\dagger$  term is computed in a similar way, and by combining the two we obtain the same-site anti-commutator

$$\begin{aligned}\{c_j, c_j^\dagger\} &= c_j c_j^\dagger + c_j^\dagger c_j \\ &= (\mathbf{1}^{\otimes n-j-1} \otimes \sigma^- \sigma^+ \otimes \mathbf{1}^{\otimes j}) + (\mathbf{1}^{\otimes n-j-1} \otimes \sigma^+ \sigma^- \otimes \mathbf{1}^{\otimes j}) \\ &= \mathbf{1}^{\otimes n-j-1} \otimes (\sigma^- \sigma^+ + \sigma^+ \sigma^-) \otimes \mathbf{1}^{\otimes j} \\ &= \mathbf{1}^{\otimes n-j-1} \otimes \{\sigma^-, \sigma^+\} \otimes \mathbf{1}^{\otimes j} \\ &= \mathbf{1}^{\otimes n-j-1+1+j} = \mathbf{1}^{\otimes n} = \mathbf{1}_{2n \times 2n},\end{aligned}\tag{4.12}$$

where we made use of  $C \otimes (A + B) = C \otimes A + C \otimes B$  and  $\{\sigma^+, \sigma^-\} = \mathbf{1}$ . Let us now compute the different-site commutator by first considering the case with  $j > i$ . We have

$$\begin{aligned}c_j^\dagger c_i &= (\mathbf{1}^{\otimes n-j-1} \otimes \sigma^+ \otimes \sigma^z \otimes j) (\mathbf{1}^{\otimes n-i-1} \otimes \sigma^- \otimes \sigma^z \otimes i) \\ &= (\mathbf{1} \cdot \mathbf{1})^{\otimes n-j-1} \otimes (\sigma^+ \mathbf{1}) \otimes (\sigma^z \mathbf{1})^{\otimes j-i-1} \otimes (\sigma^z \sigma^-) \otimes (\sigma^z)^2 \otimes i \\ &= \mathbf{1}^{\otimes n-j-1} \otimes \sigma^+ \otimes \sigma^z \otimes j-i-1 \otimes \sigma^z \sigma^- \otimes \mathbf{1}^{\otimes i},\end{aligned}\tag{4.13}$$

and in a similar way one gets to  $c_i c_j^\dagger = \mathbf{1}^{\otimes n-j-1} \otimes \sigma^+ \otimes \sigma^z \otimes j-i-1 \otimes \sigma^- \sigma^z \otimes \mathbf{1}^{\otimes i}$ . Therefore

$$\begin{aligned}\{c_i, c_j^\dagger\}_{j>i} &= c_i c_j^\dagger + c_j^\dagger c_i \\ &= (\mathbf{1}^{\otimes n-j-1} \otimes \sigma^+ \otimes \sigma^z \otimes j-i-1 \otimes \sigma^- \sigma^z \otimes \mathbf{1}^{\otimes i}) + (\mathbf{1}^{\otimes n-j-1} \otimes \sigma^+ \otimes \sigma^z \otimes j-i-1 \otimes \sigma^z \sigma^- \otimes \mathbf{1}^{\otimes i}) \\ &= \mathbf{1}^{\otimes n-j-1} \otimes \sigma^+ \otimes \sigma^z \otimes j-i-1 \otimes \{\sigma^-, \sigma^z\} \otimes \mathbf{1}^{\otimes i} = 0\end{aligned}\tag{4.14}$$

because  $\{\sigma^-, \sigma^z\} = 0$ . Similarly, for  $j < i$

$$c_j^\dagger c_i = \mathbf{1}^{\otimes n-i-1} \otimes \sigma^- \otimes \sigma^z \otimes i-j-1 \otimes \sigma^+ \sigma^z \otimes \mathbf{1}^{\otimes j},\tag{4.15}$$

$$c_i c_j^\dagger = \mathbf{1}^{\otimes n-i-1} \otimes \sigma^- \otimes \sigma^z \otimes i-j-1 \otimes \sigma^z \sigma^+ \otimes \mathbf{1}^{\otimes j},\tag{4.16}$$

$$\{c_i, c_j^\dagger\}_{j<i} = \mathbf{1}^{\otimes n-i-1} \otimes \sigma^- \otimes \sigma^z \otimes i-j-1 \otimes \{\sigma^+, \sigma^z\} \otimes \mathbf{1}^{\otimes j} = 0.\tag{4.17}$$

Thus, in general,  $\{c_i, c_j^\dagger\} = \delta_{ij} \mathbf{1}$ .

□

The Jordan-Wigner transformation is an example of a duality, in that it gives us two equivalent descriptions of the same physical phenomena in terms of the transformed Hamiltonian it allows us to obtain from the original one. In section 5.2.3 we will see one such example with the Hubbard Hamiltonian which when translated with the Jordan-Wigner transformation exhibits non-local interactions.

Besides this transformation, the Bravyi-Kitaev [46] and the Bravyi-Kitaev super fast [92] transformations are the other most common mappings currently under research.

## 4.2 The phase estimation approach

To achieve the goal of estimating Hamiltonian eigenvalues with the PEA and the IPEA, we will need to simulate the evolution operator,  $\mathcal{U}(t)$ , obtained from the Hamiltonian being considered. We will now review elementary quantum dynamics.

In quantum mechanics, time is just a parameter and not an operator. In the Schrödinger representation, a state ket evolves in time as acted upon by the time-evolution operator,

$$|\psi, t_0; t\rangle = \mathcal{U}(t, t_0) |\psi, t_0\rangle \quad (4.18)$$

which necessarily has the properties,

$$\mathcal{U}^\dagger(t, t_0) \mathcal{U}(t, t_0) = \mathbf{1}, \quad (4.19)$$

$$\mathcal{U}(t_2, t_0) = \mathcal{U}(t_2, t_1) \mathcal{U}(t_1, t_0), \quad (t_2 > t_1 > t_0) \quad (4.20)$$

The Hamiltonian operator, hermitian, is the generator of time evolution, and the infinitesimal evolution operator can be written as

$$\mathcal{U}(t_0 + dt, t_0) = 1 - iHdt. \quad (4.21)$$

leading to the Schrödinger equation for the time-evolution operator

$$i \frac{\partial}{\partial t} \mathcal{U}(t, t_0) = H \mathcal{U}(t, t_0). \quad (4.22)$$

The form taken by  $\mathcal{U}(t, t_0)$  will depend on the following cases:

(a) When the Hamiltonian is independent of time we write,

$$\mathcal{U}(t, t_0) = e^{-i(t-t_0)H}; \quad (4.23)$$

(b) If the Hamiltonian is time dependent but commutes at different instants the solution comes from

$$\mathcal{U}(t, t_0) = e^{-i \int_{t_0}^t dt' H(t')}; \quad (4.24)$$

(c) Else, when the Hamiltonian doesn't commute with itself at different times, the Dyson series provides a solution,

$$\mathcal{U}(t, t_0) = 1 + \sum_{n=1}^{\infty} (-i)^n \int_{t_0}^t dt_1 \int_{t_0}^{t_1} dt_2 \cdots \int_{t_0}^{t_{n-1}} dt_n H(t_1) H(t_2) \cdots H(t_n). \quad (4.25)$$

Here the operators are time-ordered, meaning that operators at later times are to the right.

Let us now consider only case (a). Here, when we expand the initial state ket of the system in terms of the eigenkets of the Hamiltonian operator,  $|\varepsilon_n\rangle$ , with eigenvalue  $\varepsilon_n$ ,

$$|\psi\rangle = \sum_n \alpha_n |\varepsilon_n\rangle, \quad (4.26)$$

we can write  $\mathcal{U}(t, 0)$  as

$$\mathcal{U}(t) \equiv \mathcal{U}(t, 0) = \sum_n |\varepsilon_n\rangle e^{-it\varepsilon_n} \langle\varepsilon_n| \quad (4.27)$$

and apply it to the state ket to solve the initial value problem,

$$|\psi\rangle = \sum_n \alpha_n e^{-it\varepsilon_n} |\varepsilon_n\rangle. \quad (4.28)$$

From this result we can see that the phase of the coefficient associated to eigenket  $|\varepsilon_n\rangle$  oscillates with period

$$T_n = \left| \frac{2\pi}{\varepsilon_n} \right|, \quad (4.29)$$

and if the system is initially in an eigenstate of  $H$ , it will remain there forever (stationary state).

To reproduce  $\mathcal{U}(t)$  with a digital circuit, a quantum register with the suitable number of qubits to implement this unitary operator is necessary. From the universality property of quantum computing, it is always possible to construct a circuit to approximate  $\mathcal{U}(t)$  with arbitrary precision using some decomposition on the elementary gates. However, we are not always guaranteed to find efficient decompositions. When we are not lucky enough, this may become a bottleneck for the PEA and the IPEA. In the next section we will approach this problem through the Trotter-Suzuki decomposition.

As we have seen in sections 2.4 and 2.5, the algorithms also need to implement powers of the evolution operator. Since  $H$  commutes with itself, we have

$$\mathcal{U}^p(t) = \mathcal{U}(pt), \quad (4.30)$$

and the exponentiation becomes simple.

As made explicit by Eq.4.27, the eigenvalues of  $\mathcal{U}(t)$  are  $e^{-it\varepsilon_n}$ . We have seen before that the phase estimation procedures output  $\varphi_n = -\frac{t\varepsilon_n}{2\pi}$ . Then, if we define some value for the time parameter and implement phase estimation with the evolution operator  $\mathcal{U}(t)$  associated to the Hamiltonian we are studying, we obtain the eigenvalue from

$$\varepsilon_n = -\frac{\varphi_n 2\pi}{t}. \quad (4.31)$$

We will refer to this method as a *single-shot* estimation, because we only evaluate  $\mathcal{U}(t)$  for a single value of  $t$ . The quality of our estimation of  $\varepsilon_n$  will depend only on the accuracy and precision of the reading for  $\varphi_n$ .

We can improve the quality of the results by doing the computation with a different approach we will call *multiple-shot*. In the multiple shot method we take advantage of the free parameter of time in  $\mathcal{U}(t)$  and perform several estimations for different values of  $t$ . Since  $\varphi_n = -\frac{t\varepsilon_n}{2\pi}$ , we expect to obtain a linear dependence of the phase with time. By classically fitting the results of all the computations with linear regression we can estimate the energy value from the slope  $m$  according to

$$\varepsilon_n = -2\pi m. \quad (4.32)$$

Naturally, the greater the number of time-steps (shots) we can compute inside the time window of a full period, Eq.4.29, the better the linear fitting will be. However, the computation time scales linearly in the number of shots performed, and more importantly, in general we don't know the period, so we might need to approach this with some heuristic.

### 4.3 Trotter-Suzuki decomposition

The name of the game in phase estimation is to implement the operator  $\mathcal{U}(t) = e^{iHt}$ . With the time evolution operator written in terms of Pauli spin matrices we have the conditions to start implementing it on the quantum

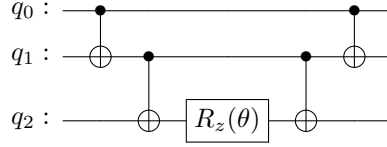


Figure 4.1: An example of how to code the exponential of the tensor product of three Pauli  $\sigma^z$  matrices,  $\exp(-i(\theta/2)\sigma_0^z\sigma_1^z\sigma_2^z)$ .

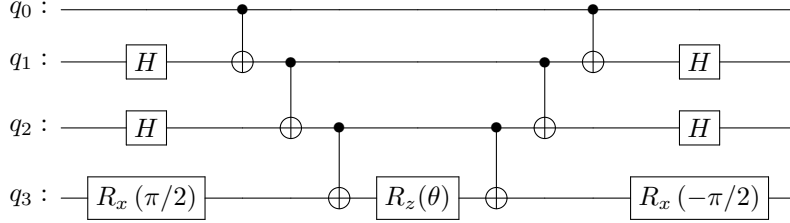


Figure 4.2: An example of how to code the exponential of the tensor product of  $\exp(-i(\theta/2)\sigma_0^z\sigma_1^x\sigma_2^x\sigma_3^y)$ .

computer. Let us first consider the exponential of a tensor product of  $\sigma_z$  gates. Fig.4.1. gives an example. Generally, this will need only one  $R_z$  gate and  $2(n-1)$   $cX$  gates to compute the parity of the qubits. If the exponential contains  $\sigma^x$  or  $\sigma^y$  matrices, we need to change to the  $X$  or  $Y$  basis before computing parity, respectively. This is done by applying an  $H$  or  $R_x(\pi/2)$  gate prior to the action of the  $cX$  gates, and their inverses afterwards, as exemplified in Fig.4.2.

Given that the Jordan-Wigner transformed Hamiltonian is a sum of terms  $H = \sum_m h_m$ , we would like to simulate the time evolution operator by exponentiating each individual term  $h_m$  and multiplying them together, since we already learned a very useful way of translating exponentials of Pauli matrix products into quantum circuits. However, this would only be allowed if all the terms commuted. In general this is not the case, and when not all these terms commute with each other, we can often construct two sets of mutually commuting terms. Call these  $S_1 = \{h_i, \dots, h_j\}$  and  $S_2 = \{h_{j+1}, \dots, h_m\}$ . In each of these sets, all operators commute among themselves but no single operator from one set commutes with all the operators of the other set. Therefore, when we compute the unitary evolution operator for this system,

$$\mathcal{U}(t) = e^{-itH} \quad (4.33)$$

we can break the exponential of the sum into a product of exponentials for the terms in each of these sets, but not when we mix both sets.

Besides the Dyson expansion in Eq.4.25, other expansions exist to calculate approximations to the time evolution propagator. One of these is the Trotter-Suzuki decomposition [93] which reads, to first order,

$$e^{(A+B)t} = \left( e^{At/n} e^{Bt/n} \right)^n + O(t\Delta t), \quad (4.34)$$

and to second order is given by

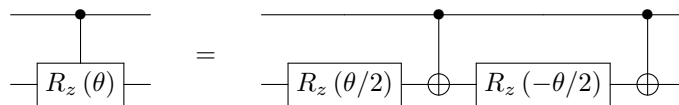


Figure 4.3: Gate decomposition of the controlled- $R_z(\theta)$  operation necessary to apply the controlled unitary evolution operator.

$$e^{(A+B)t} = \left( e^{At/2n} e^{Bt/n} e^{At/2n} \right)^n + O\left(t(\Delta t)^2\right), \quad (4.35)$$

where  $A = \sum \{S_1\}$  and  $B = \sum \{S_2\}$ . Also, note that  $U^p(t) = U(pt)$ , hence

$$\left( e^{(A+B)t} \right)^p = \left( e^{Atp/n} e^{Btp/n} \right)^n + O(t\Delta t) \quad (4.36)$$

$$\left( e^{(A+B)t} \right)^p = \left( e^{Atp/2n} e^{Btp/n} e^{Atp/2n} \right)^n + O\left(t(\Delta t)^2\right). \quad (4.37)$$

In the above expressions we are too thinking abstractly. Remembering the goal of using these expressions to perform efficient computations, we can rewrite the second order Trotter-Suzuki decomposition in a more clever form that will later require a smaller number of gates to be applied,

$$\left( e^{(A+B)t} \right)^p = e^{Atp/2n} \left( e^{Btp/n} e^{Atp/n} \right)^{n-1} e^{Btp/n} e^{Atp/2n} + O\left(t(\Delta t)^2\right). \quad (4.38)$$

Using these decompositions, we can also approximate the time-evolution of a quantum system under a time-varying Hamiltonian by evolving the state discretely at small enough timesteps where the Hamiltonian is kept constant.

## 4.4 State preparation

As we have already seen, if starting the second register from a general state  $|\psi\rangle$ ,

$$|\psi\rangle = \sum_n \alpha_n |\varepsilon_n\rangle, \quad (4.39)$$

the PEA will allow estimation of some eigenvalue with probability proportional to the coefficient associated to such eigenstate in the state decomposition over the basis. Thus, the eigenvalue we get is picked at random from the set of all eigenstates, and by carefully choosing the initially prepared state one can change the probability mass function over this set.

We could of course start the second register from a general state and try to discern all eigenvalues from the final PMF obtained. We already touched on this approach. However, for that we would need enough resolution in the first register to measure a probability distribution with clearly separated peaks at the various estimations for all the eigenkets present in the input state decomposition.

Sometimes we are not interested in doing that but only on finding some eigenvalue. More importantly, it may be very inefficient to do so classically if we are looking for eigenvalues of an unitarity with large dimension  $D$ , since we will have an exponential increase in the number of eigenvalues for each qubit added to the second register. Therefore, if we want to maximize the probability of estimating a specific eigenvalue we should feed the second register with a “good” initial state. For example, if our goal is to estimate the lowest energy level of an Hamiltonian,  $\varepsilon_0$ , we should prepare an initial state with a large overlap with the ground state  $|\varepsilon_0\rangle$ , that is

$$|\psi\rangle = \alpha_0 |\varepsilon_0\rangle + \sum_{j=1}^{D-1} \alpha_j |\varepsilon_j\rangle, \quad (4.40)$$

with  $|\alpha_0|^2 \gg |\alpha_j|^2 \forall j$ . Thus, initial state preparation also plays an important role in the whole simulation process.

There are several methods to prepare an arbitrary state of  $n$  qubits  $|\psi\rangle$ , defined by  $2^n$  amplitudes. One of

those is the method developed by Shende et al. [94]. This consists in constructing a circuit which is the reverse of the series of operations necessary to bring the state  $|\psi\rangle$  to  $|0\rangle_n$ . When we come to explore experiments on real devices we will use this algorithm to prepare arbitrary states. We refer to the original publication for a detailed explanation.

Another approach to initialize states to the PEA or the IPEA would be to use the Variational Quantum Eigensolver [95] or some variation of this method. This may be of interest in some situations if the final circuit to prepare an approximation to the ground state ends up being shallower and using less gates while still getting sufficiently close to the desired state. If the obtained overlap with the ground state is large enough, the phase estimation procedure can extract the ground state eigenvalue precisely.



## Chapter 5

# Experiments on a real quantum computer

*"It is practically certain that the physical laws, in their observable consequences, have a finite limit of precision."*

— Kurt Gödel, [96] p. 206

### 5.1 Fourier transform

Let us begin this chapter by returning to Example 2.11. This will let us illustrate some of the concepts presented previously within this simple example. The circuit to perform the quantum Fourier transform of the input state  $|\psi_{in}\rangle = (\sqrt{1/2}, 0, 1/2, 1/2)^T$  in basis  $|q_0q_1\rangle \in \{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$  is shown in Fig.5.1. The first five gates in the circuit prepare the initial state with the Shende et al. algorithm [94]; the sequence of operations and intermediate states is given by

$$|\psi_0\rangle = |00\rangle \quad (5.1)$$

$$|\psi_1\rangle = R_{y;0} \left( \frac{\pi}{2} \right) |\psi_0\rangle = \frac{1}{\sqrt{2}} (|00\rangle + |10\rangle) \quad (5.2)$$

$$|\psi_2\rangle = cX_{01} |\psi_1\rangle = \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle) \quad (5.3)$$

$$|\psi_3\rangle = R_{y;1} \left( -\frac{\pi}{4} \right) |\psi_2\rangle = \frac{1}{\sqrt{2}} \left( \cos \frac{\pi}{8} |00\rangle - \sin \frac{\pi}{8} |01\rangle + \sin \frac{\pi}{8} |10\rangle + \cos \frac{\pi}{8} |11\rangle \right) \quad (5.4)$$

$$|\psi_4\rangle = cX_{01} |\psi_3\rangle = \frac{1}{\sqrt{2}} \left( \cos \frac{\pi}{8} |00\rangle - \sin \frac{\pi}{8} |01\rangle + \cos \frac{\pi}{8} |10\rangle + \sin \frac{\pi}{8} |11\rangle \right) \quad (5.5)$$

$$|\psi_{in}\rangle = R_{y;1} \left( \frac{\pi}{4} \right) |\psi_4\rangle = \frac{1}{\sqrt{2}} |00\rangle + \frac{1}{2} (|10\rangle + |11\rangle) \quad (5.6)$$

The next three operations correspond to  $U_{QFT}^{(2)}$ , the core of the quantum Fourier transform, and the final three  $cX$  gates would not be necessary because we can send readout to our preferred classical registers. We decide to keep this to illustrate the *SWAP* operation.

Remember that the probability vector expected in an unitary simulator is

$$PMF \approx (72.86\%, 7.32\%, 12.50\%, 7.32\%)^T. \quad (5.7)$$

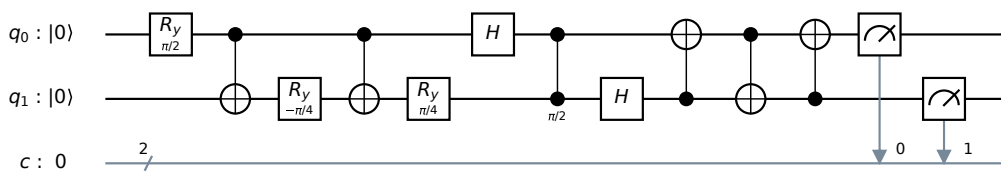
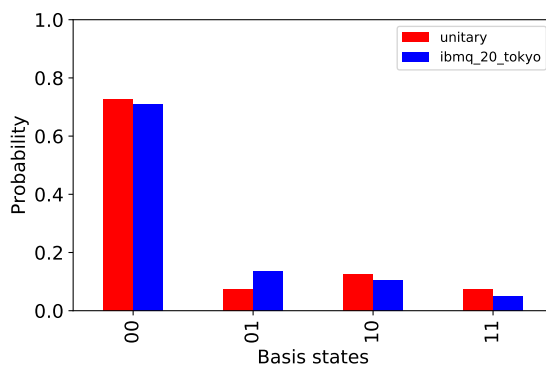
Figure 5.1: 2-qubit  $QFT$  circuit with the initial state from Example 2.11. .

Figure 5.2: Side-by-side comparison of the theoretical and experimental probability distributions for the circuit in Fig.5.1.

We chose 2 qubits,  $\{q_1, q_2\}$ , from the *ibmq\_20\_tokyo* platform (Fig.2.12) and ran the computation 5000 times. The probability vector obtained experimentally is

$$PMF \approx (71.08\%, 13.62\%, 10.40\%, 4.90\%)^T. \quad (5.8)$$

yielding a fidelity of  $F \approx 0.993$  and Jensen-Shannon divergence of  $JSD \approx 0.009$ . Both probability distributions are represented in Fig.5.2.

## 5.2 Phase estimation

Let us skip the *single-shot* experiments and go straight to the more interesting *multiple-shot* ones, i.e. the time-varying method of determining the eigenvalues.

### 5.2.1 Zeeman Hamiltonian

#### Unitary simulation

The unitary evolution operator obtained from the Hamiltonian in Eq.3.8, independent of time, can be seen to be

$$\mathcal{U}_Z(t) = e^{-iH_Z t} \quad (5.9)$$

$$= \begin{pmatrix} e^{-ict} & 0 \\ 0 & e^{ict} \end{pmatrix} \quad (5.10)$$

$$= R_z(2ct), \quad (5.11)$$

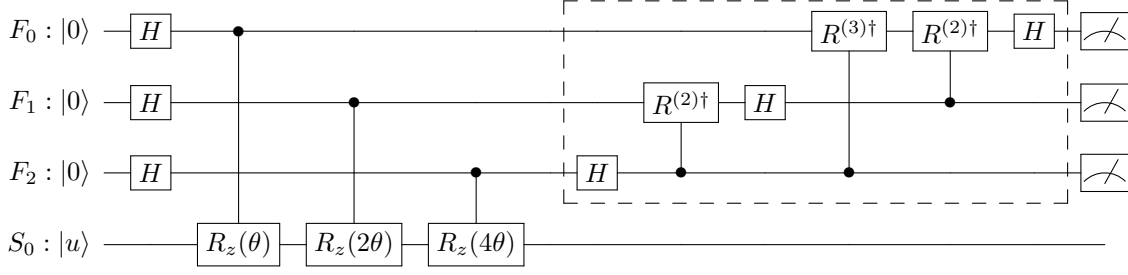


Figure 5.3: Circuit of the PEA implementation with 3 first register qubits for the Zeeman Hamiltonian. The  $z$ -rotation angle is  $\theta = 2ct$  and the boxed operations correspond to  $U_{QFT}^\dagger$ .

where  $R_z(\theta)$  is the  $z$ -rotation operation defined in Eq.2.15. We want to simulate this evolution operator  $\mathcal{U}_Z(t)$  and estimate  $\varphi_u$  for an eigenvalue written in the form  $e^{2\pi i\varphi_u}$ . Therefore, we want to estimate  $\varphi_0 = -ct/2\pi$  if the second register is started at  $|u\rangle = |0\rangle$  or  $\varphi_1 = ct/2\pi$  if  $|u\rangle = |1\rangle$ . With that, we will know the eigenvalues of  $H_Z$ .

To implement  $\mathcal{U}_Z(t)$  we only need 1 qubit in the second register of the PEA. The circuit in Fig.5.3 will simulate  $\mathcal{U}_Z(t)$  exactly, since this operator can be implemented with a single gate from the computer gate set. Note also that  $R_z^m(\theta) = R_z(m\theta)$ . The number of qubits in the first register will determine the resolution of the measurements. We will be using *ibmq\_20\_tokyo* to perform experiments, but even though this computer offers 20 qubits, their limited connectivity greatly increases the number of 2-qubit operations of the PEA circuits we can build in comparison to what would be possible with perfect connectivity. This in turn increases the susceptibility of the final computation to the errors accumulated along execution. Therefore, a compromise on resolution is needed to get results that are useful at all. We will choose a first register with only 3 qubits. With this choice, we can code the PEA for  $H_Z$  with only 4 qubits and take advantage of one of the 2 subsets of 4 fully connected qubits that *ibmq\_20\_tokyo* offers (see Fig.2.12). This also allows us to end up with a circuit with small enough depth.

To study the effectiveness of this quantum algorithm, let us assume, without loss of generality,

$$c = 3.8, \quad (5.12)$$

and start the second register at eigenket  $|u\rangle = |0\rangle$ , which allows us to avoid the need to prepare the initial eigenket  $|u\rangle$ . As we have seen, our single input to the algorithm will be  $\theta = 2ct$  and the output is  $\varphi_0 = -ct/2\pi$ . By running multiple instances of the PEA for different values of  $t$ , we will be able to determine  $c$  from the slope of the estimated  $\hat{\varphi}_0$ .

Since this unitarity is of such a small dimension, we start by performing an unitary simulation with a classical emulation of a quantum computer to understand how the PEA should behave in the absence of noise and decoherence. Fig.5.4 shows the results. The mean phase direction,  $\hat{\varphi}_0$ , of the PMFs obtained for each sampled  $t$  is plotted as a function of time and the circular standard phase deviation from Eq.2.65,  $v_0$ , is represented at each point by the height of the colored region at that point.

Notice there is an oscillation wrapped around the predicted slope. This is expected because the real phase  $\varphi_0$  we are trying to measure travels on the vertical axis but can only be measured exactly at the discrete points of the first register basis states. This changes the  $\hat{\varphi}_0$  and  $v_0$  of the PMF.

Furthermore, we perform linear regression on the  $N = 200$  values of  $\hat{\varphi}_0(t)$ . To do that, we minimize the “chi-squared” function

$$\chi^2 = \sum_{i=1}^N \left( \frac{\hat{\varphi}_i - f(t_i)}{v_i} \right)^2, \quad (5.13)$$

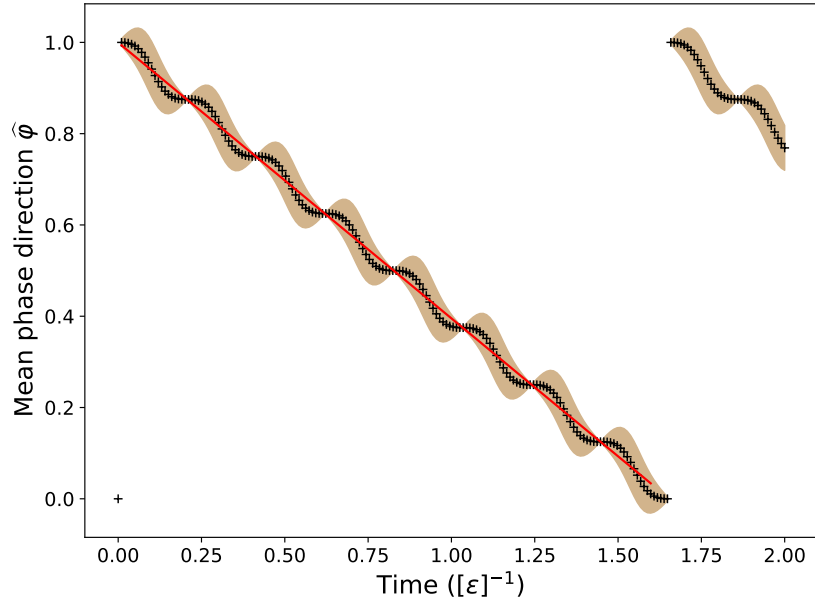


Figure 5.4: Unitary simulation of the evolution operator  $\mathcal{U}_Z(t)$  with  $c = 3.8$ . The mean phase direction is plotted as a function of 200 time values sampled uniformly between  $t = 0$  and  $t = 2$ . At each point, the colored region has height equal to the circular standard phase deviation

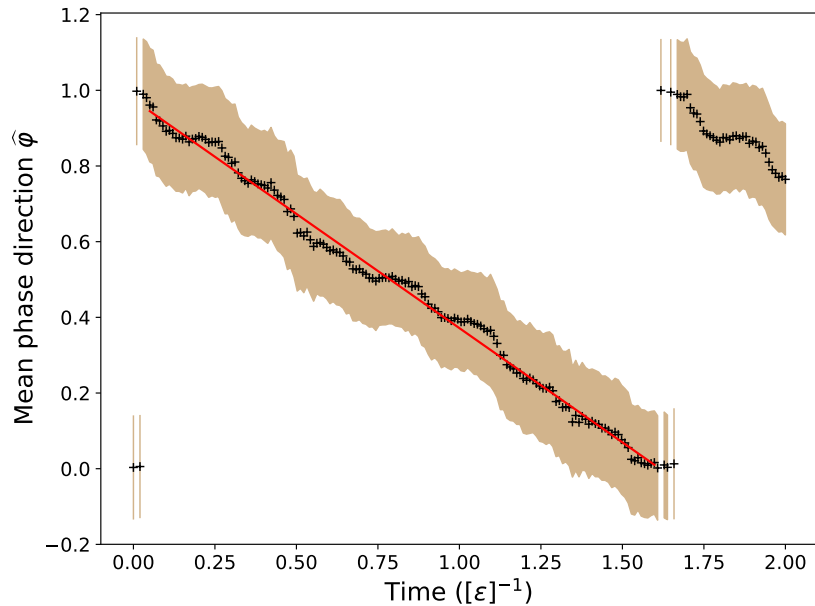


Figure 5.5: Experimental results obtained in the *ibmq\_20\_tokyo* platform for the mean phase direction  $\hat{\varphi}_0 = -\hat{c}t/2\pi$  in  $\mathcal{U}_Z(t)|0\rangle = e^{2\pi i\varphi_0}|0\rangle$ . The circular standard deviation is represented by the colored region, and is centered at  $\hat{\varphi}_0$ .

where  $v_i$  is the circular standard phase deviation error for each data-point  $\hat{\varphi}_i$  and  $f(t_i) = mt + b$  is the fitting function whose parameters  $m$  and  $b$  we would like to find. The fitted slope coefficient we obtain is equal to  $m \approx -0.6044$  with standard deviation  $\sigma_m \approx 0.0017$  and  $\frac{\chi^2}{N-2} = 0.0748$ , which is close to perfect. Hence, we recover the actual eigenvalue associated to eigenstate  $|0\rangle$  of  $H_Z$  with  $\hat{c} = -2\pi m$  and  $\sigma_{\hat{c}} \approx 2\pi\sigma_m$  yielding

$$\hat{c} \approx 3.797 \pm 0.011 \quad (5.14)$$

### Simulation on a real quantum computer

We are now ready to perform exactly the same procedure on the real *ibmq\_20\_tokyo* computer (Fig.2.12). To start with, we need to choose 4 qubits from this device to implement the algorithm. The subset we use is the four fully connected qubits  $\{q_7, q_8, q_{12}, q_{13}\}$  and the average measurements of the physical parameters of these qubits at the time of execution were registered in Tab.5.1. Notice that the biggest source of error comes from readout of the final qubit states.

|                                      |        |
|--------------------------------------|--------|
| Frequency (GHz)                      | 4.76   |
| $T_1$ ( $\mu s$ )                    | 116.25 |
| $T_2$ ( $\mu s$ )                    | 63.36  |
| Gate error ( $10^{-3}$ )             | 1.21   |
| Readout error ( $10^{-2}$ )          | 11.78  |
| Multi-qubit gate error ( $10^{-2}$ ) | 5.97   |

Table 5.1: Average measured parameters of qubits  $\{q_7, q_8, q_{12}, q_{13}\}$  from *ibmq\_20\_tokyo* at the time of the experiment, 2018-10-17 at 1:27:06 am.

After coding the quantum circuit with the high level operations represented in Fig.5.3 using QISKit, we compile it to low level QASM code ending up with a circuit containing 33 gates in total, of which 12 are 2-qubit  $cX$  gates.

For each timestamp we perform 5000 experiment runs, totaling  $10^6$  executions for all simulated instants. The complete computational procedure takes a few minutes. Pause for a moment to consider the technological feat involved here. The measurement results we obtained are shown in Fig.5.5. We compare the 200 experimental PMFs to the theoretical ones by measuring Fidelity and Jensen-Shannon divergence. This is shown in Fig.5.6 as a function of time, and the overall average is marked by the dashed line. We again perform linear regression, selecting the final estimations  $\hat{\varphi}_0(t)$  obtained for  $t \in [0.05, 1.60]$ . The results from linear regression are summarized in Tab5.2.

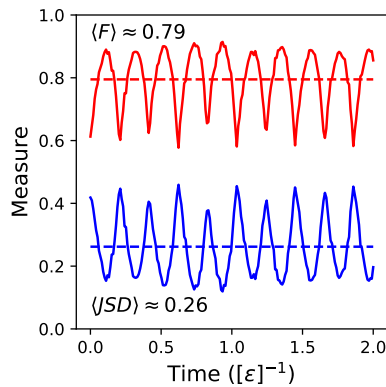


Figure 5.6: Distance measures between the theoretical and experimental probability distributions. Fidelity is represented in red and the Jensen-Shannon divergence in blue. Average values are provided.

| $t$ interval | $m$    | $\sigma_m$ | $b$   | $\sigma_b$ | $\chi^2/(N-2)$ | $\hat{c}$ | $\sigma_{\hat{c}}$ | $\epsilon =  c - \hat{c} $ |
|--------------|--------|------------|-------|------------|----------------|-----------|--------------------|----------------------------|
| [0.05, 1.60] | -0.603 | 0.051      | 0.975 | 0.048      | 0.0068         | 3.79      | 0.32               | 0.006                      |

Table 5.2: Linear regression results obtained with the experimental data of the PEA  $H_Z$  simulation in *ibmq\_20\_tokyo*.

Thus, we estimate  $\hat{c} \approx 3.79 \pm 0.32$  with absolute error  $\epsilon = 0.006$ , which translates to relative errors in precision and accuracy of, respectively, 8.4% and 0.16%.

## 5.2.2 Ising Hamiltonian

### 5.2.2.1 Evolving an eigenstate

We will now simulate the evolution operator  $U_I(t)$  associated to the Ising Hamiltonian  $H_I$  (Eq.3.10). It is trivial to check that the three terms in  $H_I$  commute, hence there is no need for the Trotter-Suzuki decomposition and the evolution operator can be written as

$$U_I(t) = e^{-itH_I} \quad (5.15)$$

$$= e^{-itc_0\sigma_0^z} e^{-itc_1\sigma_1^z} e^{-itJ\sigma_0^z\sigma_1^z} \quad (5.16)$$

$$= \begin{pmatrix} e^{-it\varepsilon_1} & 0 & 0 & 0 \\ 0 & e^{-it\varepsilon_2} & 0 & 0 \\ 0 & 0 & e^{-it\varepsilon_3} & 0 \\ 0 & 0 & 0 & e^{-it\varepsilon_4} \end{pmatrix}. \quad (5.17)$$

We will arbitrarily postulate the following

$$c_0 = 0.33, \quad c_1 = 3.24, \quad J = 1.17, \quad (5.18)$$

which according to Eq.3.11 gives us the eigenstates,

$$\begin{cases} \varepsilon_1 = 4.74 & , |00\rangle \\ \varepsilon_2 = 1.74 & , |01\rangle \\ \varepsilon_3 = -4.08 & , |10\rangle \\ \varepsilon_4 = -2.4 & , |11\rangle \end{cases} \quad (5.19)$$

with basis states again labeled as  $|site_1, site_0\rangle$ . Repeating what was already said before, the goal is to estimate  $\varphi_u = -t\varepsilon_u/2\pi$  in the eigenvalue  $e^{i2\pi\varphi_u}$  associated with eigenket  $|u\rangle$ .

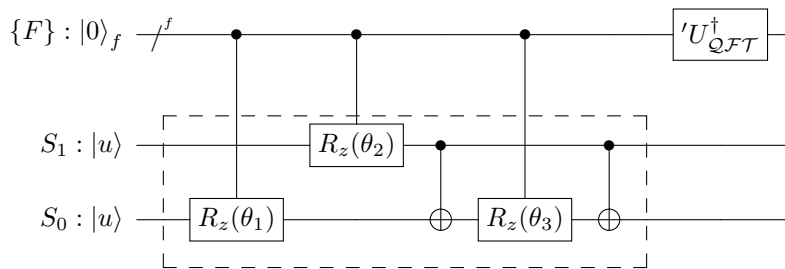


Figure 5.7: Scheme circuit for the PEA implementation of the Ising Hamiltonian. The boxed part corresponds to the unitary evolution operator  $U_I(t)$ , and we have  $\theta_1 = 2c_0t$ ,  $\theta_2 = 2c_1t$ ,  $\theta_3 = 2Jt$ . The first register is represented in condensed form.

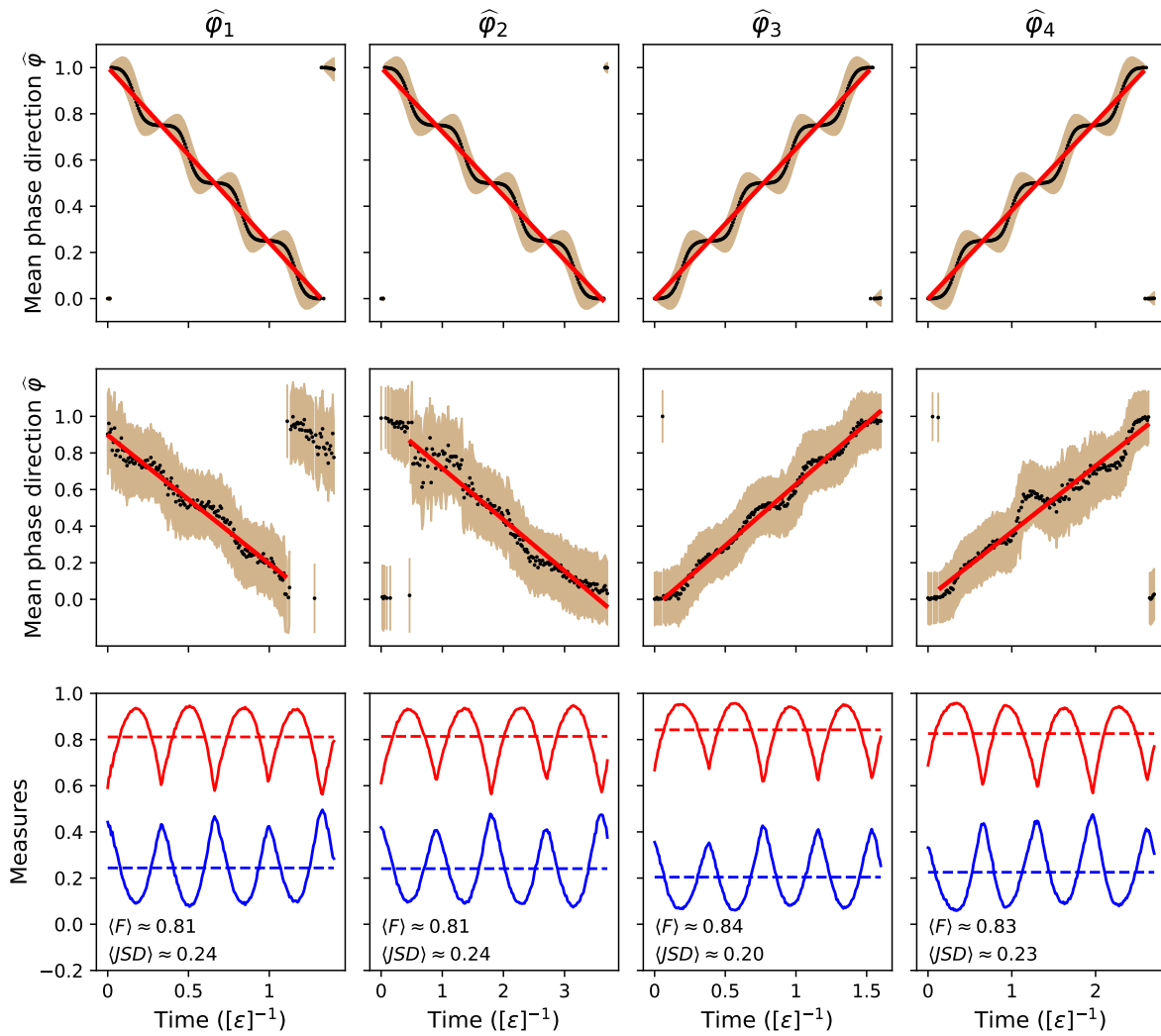


Figure 5.8: PEA simulation results for the  $H_I$  Hamiltonian. The first row of plots shows the results for an unitary simulation, the second row shows what was obtained on *ibmq\_20\_tokyo*, and the third row shows Fidelity in red and Jensen-Shannon divergence in blue (averages are dashed).

This time, circuit construction (Fig.5.7) needs 2 qubits on the second register instead of 1 as before. Preparing the states to feed into the second register is again very simple with these eigenstates by flipping bits using an  $X$  gate.

Since we are again doing a computation in which we already know the actual eigenvalues, we can predict the period  $T_u$  for the  $\varphi_u \bmod 1$  signal of each eigenstate, e.g. the longest one will be  $T_2 = |2\pi/\varepsilon_2| \approx 3.6$ . Because of that, we will vary time between 0 and  $T_u$ , doing experiments for 200 time instants in this interval. Let us now decrease the first register to 2 qubits to keep using the *ibmq\_20\_tokyo* subset of 4 fully connected qubits.

The results from an unitary simulation are shown in the first row of Fig.5.8. Not surprisingly, we recover all of the expected eigenvalues precisely. Notice there are 4 points in time in which the phase is determined with zero standard deviation, corresponding to the case where  $\varphi_u$  can be accurately decomposed in the basis states of the first register.

As for the real experimental computation, the 4 qubits chosen for the implementation were  $\{q_7, q_8, q_{12}, q_{13}\}$ , and at the time of execution, their averaged calibration parameters are shown in Tab.5.3.

|                                      |        |
|--------------------------------------|--------|
| Frequency (GHz)                      | 4.76   |
| $T_1$ ( $\mu s$ )                    | 108.11 |
| $T_2$ ( $\mu s$ )                    | 72.66  |
| Gate error ( $10^{-3}$ )             | 1.18   |
| Readout error ( $10^{-2}$ )          | 11.55  |
| Multi-qubit gate error ( $10^{-2}$ ) | 6.79   |

Table 5.3: Average measured parameters of qubits  $\{q_7, q_8, q_{12}, q_{13}\}$  from *ibmq\_20\_tokyo* in 2018-10-24 at the time of the experiment at 11:43:02 am.

After inspecting the individual error rates,  $\{q_8, q_{13}\}$  were chosen as first register qubits. The prepared circuits ended up having 38 gates in total, of which 18 were  $cX$  gates. The results are plotted in the middle row of Fig.5.8. First, note that the correlation with the ideal scenario is clear, although now we have a higher standard deviation. One can even see the oscillation of the mean phase direction wrapped around the red linear regression line, for example in eigenstates 1 and 3. The eigenvalues can be obtained with linear regression very accurately. The results from are summarized in Tab.5.4.

|              | $t$ interval | $m$    | $\sigma_m$ | $b$    | $\sigma_b$ | $\chi^2 / (N - 2)$ | $\hat{\epsilon}$ | $\sigma_{\hat{\epsilon}}$ | $\epsilon =  \epsilon - \hat{\epsilon} $ |
|--------------|--------------|--------|------------|--------|------------|--------------------|------------------|---------------------------|--|
| $\epsilon_1$ | [0, 1.1]     | -0.702 | 0.094      | 0.897  | 0.061      | 0.011              | 4.41             | 0.59                      | 0.32                                     |
| $\epsilon_2$ | [0.2, 3.7]   | -0.280 | 0.031      | 0.997  | 0.075      | 0.023              | 1.76             | 0.20                      | 0.023                                    |
| $\epsilon_3$ | [0.06, 1.6]  | 0.669  | 0.051      | -0.041 | 0.047      | 0.0094             | -4.20            | 0.32                      | 0.12                                     |
| $\epsilon_4$ | [0.13, 2.63] | 0.360  | 0.033      | 0.0079 | 0.0498     | 0.025              | -2.27            | 0.20                      | 0.13                                     |

Table 5.4: Linear regression results obtained with the experimental data of the PEA  $H_I$  simulation in *ibmq\_20\_tokyo*.

We observe some unexpected experimental behaviors whose possible origin we have not yet understood. For example, the increase dispersion of the data at the region around  $t = 1$  of the plot for  $\hat{\varphi}_2$ , and a small systematic bump of the estimated phase just after  $t = 1$  of the plot for  $\hat{\varphi}_4$ .

### 5.2.2.2 Evolving a general state

In subsection 2.4, we studied the output PMF of the PEA when the second register is started from a general state. Let us now perform an experiment to probe this behavior. We will use the same Hamiltonian parameters and start from the general state

$$|\psi\rangle = \alpha_1 |00\rangle + \alpha_2 |01\rangle + \alpha_3 |10\rangle + \alpha_4 |11\rangle, \quad (5.20)$$

choosing  $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\} = \{1/\sqrt{2}, 1/\sqrt{4}, 1/\sqrt{8}, 1/\sqrt{8}\}$  as an example. Doing this on a real device would require a circuit module for state preparation. For now, we are only concerned with understanding how the PEA works with a practical example and how we could exploit the output PMF. We will skip the complications associated with initial state preparation and perform an emulation of the quantum circuit on an unitary simulator starting the second register from  $|\psi\rangle$  already.

We begin by experimenting with a circuit selecting 4 qubits for the first register. As can be seen in Fig.5.9, a structure becomes apparent when we stitch the computed PMFs for all the time-steps. We can make this even clearer if we increase resolution. In Fig.5.9 we also show the probability map obtained from repeating the procedure with 5 bits of resolution on the first register. Notice how precisely defined the probability peaks start to become. That is an enlightening plot.

In principle, by classically analyzing this data from all the PMFs we can determine the four eigenvalues



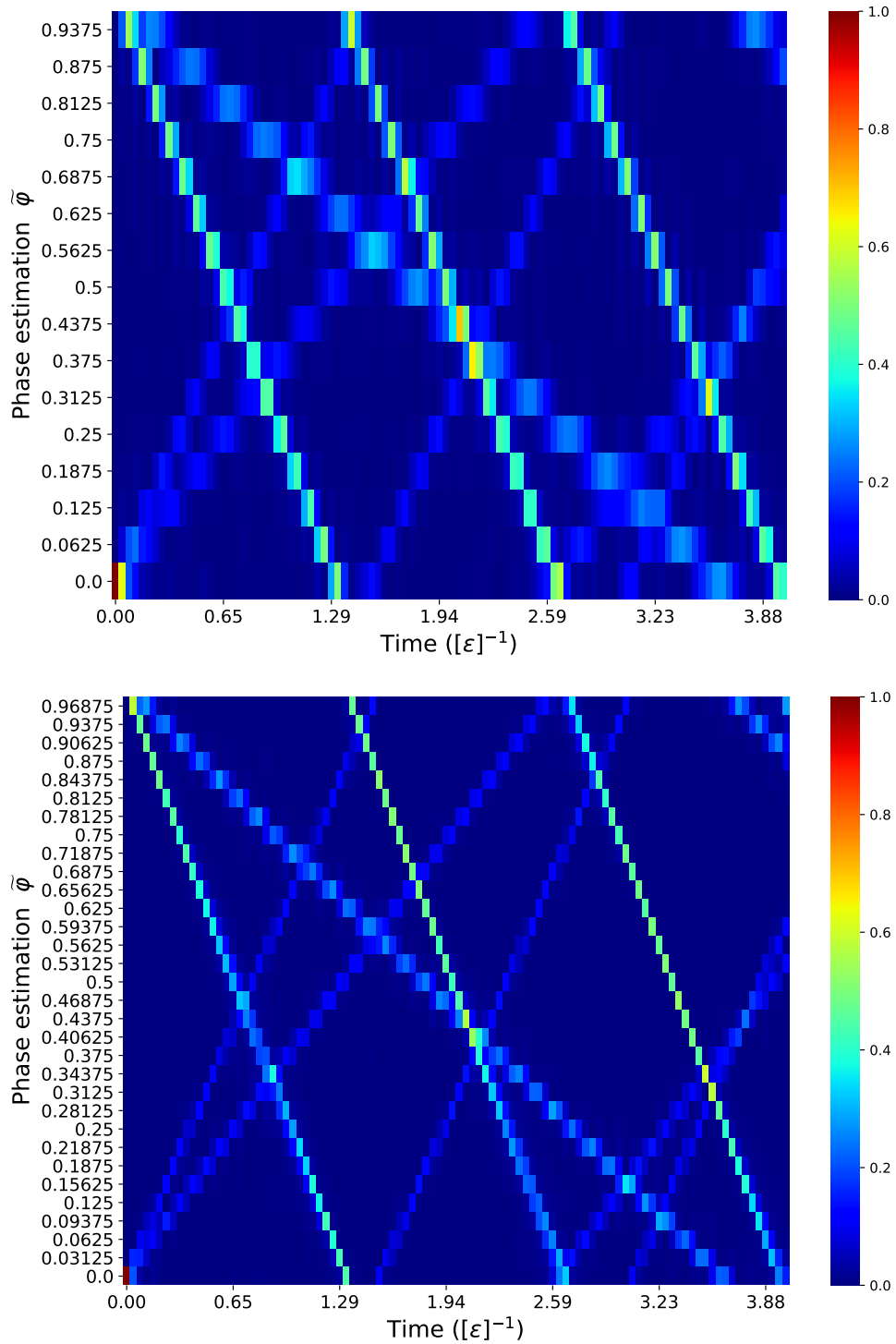


Figure 5.9: Unitary simulation evolving the general state in Eq.5.20 with  $\mathcal{U}_I(t)$  and measuring the phase with 4-bit (top) and 5-bit (bottom) resolution PEA.

and the modulus square of the four  $\alpha_u$  coefficients, which would also allow us to closely delimitate the Hilbert space of the original state. We could do that because we know our unitarity has dimension 4, therefore we could devise a method like the following:

1. Rank the four highest peaks in the sampled PMF at each time-step  $t$ , and match them together for the various time-steps;
2. Perform linear regression on each of the four slopes to determine the four eigenvalues;
3. Average the probability of each peak along time to estimate each coefficient  $\alpha_u$  from the general state.

However, this project is very rudimentary. The height of the peaks can be very similar sometimes, and with less measurement precision the more mixing we will have between different peaks. Furthermore, this fails when we have two or more equal  $\alpha_u$  coefficients in the general state, because when ranking the peaks we would mix the points belonging to different slopes. Adding more eigenstates would make matters even worse. It seems worth exploring more sophisticated signal processing techniques to examine the obtained PMFs.

It should be said, however, that exploring general states in this way may only be useful for small systems, since the post-processing step is classical and the probability map gets exponentially denser with the information of the eigenvalues with every qubit added to the second register.

### 5.2.3 2-site Hubbard Hamiltonian

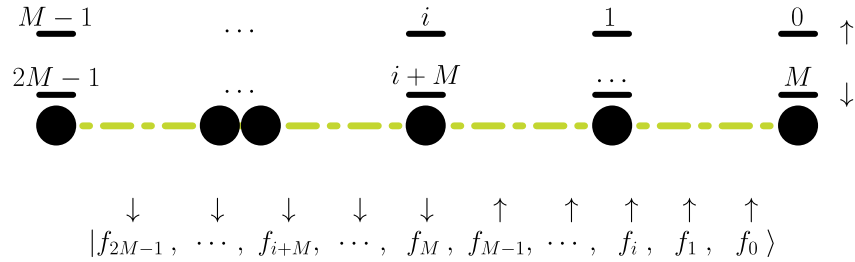


Figure 5.10: Index labeling for the 1D  $M$ -site Hubbard model, with  $2M$  fermionic modes.

With the identification in Fig.5.10, and for a one-dimensional chain, the  $M$ -site Hubbard Hamiltonian in Eq.3.14 becomes

$$H = -\tau \sum_{i=0}^{M-2} \left( c_i^\dagger c_{i+1} + c_{i+1}^\dagger c_i + c_{i+M}^\dagger c_{i+M+1} + c_{i+M+1}^\dagger c_{i+M} \right) + U \sum_{i=0}^{M-1} \hat{n}_i \hat{n}_{(i+M)}. \quad (5.21)$$

In particular, the 2-site Hamiltonian is

$$H = -\tau \left( c_0^\dagger c_1 + c_1^\dagger c_0 + c_2^\dagger c_3 + c_3^\dagger c_2 \right) + U (\hat{n}_0 \hat{n}_2 + \hat{n}_1 \hat{n}_3). \quad (5.22)$$

By employing the Jordan-Wigner transformation to express Eq.5.22 in terms of Pauli operators we get to

$$H_H = -\frac{\tau}{2} (\sigma_1^x \sigma_0^x + \sigma_1^y \sigma_0^y + \sigma_3^x \sigma_2^x + \sigma_3^y \sigma_2^y) + \frac{U}{4} (2\mathbf{1} - \sigma_3^z - \sigma_2^z - \sigma_1^z - \sigma_0^z + \sigma_2^z \sigma_0^z + \sigma_3^z \sigma_1^z). \quad (5.23)$$

*Proof.* \_\_\_\_\_

To compute the first term of the Hamiltonian we need Eq.4.13 and Eq.4.15 to obtain

$$\begin{aligned}
H_\tau &= -\tau [(\mathbf{1}^{\otimes 2} \otimes \sigma^- \otimes \sigma^+ \sigma^z) + (\mathbf{1}^{\otimes 2} \otimes \sigma^+ \otimes \sigma^z \sigma^-) + (\sigma^- \otimes \sigma^+ \sigma^z \otimes \mathbf{1}^{\otimes 2}) + (\sigma^+ \otimes \sigma^z \sigma^- \otimes \mathbf{1}^{\otimes 2})] \\
&= -\tau [(\mathbf{1}^{\otimes 2} \otimes \sigma^- \otimes \sigma^+) + (\mathbf{1}^{\otimes 2} \otimes \sigma^+ \otimes \sigma^-) + (\sigma^- \otimes \sigma^+ \otimes \mathbf{1}^{\otimes 2}) + (\sigma^+ \otimes \sigma^- \otimes \mathbf{1}^{\otimes 2})] \\
&= -\frac{\tau}{4} [\mathbf{1}^{\otimes 2} \otimes (\sigma^x + i\sigma^y) \otimes (\sigma^x - i\sigma^y) + \mathbf{1}^{\otimes 2} \otimes (\sigma^x - i\sigma^y) \otimes (\sigma^x + i\sigma^y) + \\
&\quad + (\sigma^x + i\sigma^y) \otimes (\sigma^x - i\sigma^y) \otimes \mathbf{1}^{\otimes 2} + (\sigma^x - i\sigma^y) \otimes (\sigma^x + i\sigma^y) \otimes \mathbf{1}^{\otimes 2}] \\
&= -\frac{\tau}{2} (\mathbf{1}^{\otimes 2} \otimes \sigma^x \otimes \sigma^x + \mathbf{1}^{\otimes 2} \otimes \sigma^y \otimes \sigma^y + \sigma^x \otimes \sigma^x \otimes \mathbf{1}^{\otimes 2} + \sigma^y \otimes \sigma^y \otimes \mathbf{1}^{\otimes 2}),
\end{aligned} \tag{5.24}$$

where we have used  $\sigma^+ \sigma^z = \sigma^+$  and  $\sigma^z \sigma^- = \sigma^-$ . As for the second term, using Eq.4.10 it is also straightforward to see that

$$\begin{aligned}
H_U &= \frac{U}{4} [(\mathbf{1}^{\otimes 3} \otimes (\mathbf{1} - \sigma^z)) (\mathbf{1} \otimes (\mathbf{1} - \sigma^z) \otimes \mathbf{1}^{\otimes 2}) + (\mathbf{1}^{\otimes 2} \otimes (\mathbf{1} - \sigma^z) \otimes \mathbf{1}) ((\mathbf{1} - \sigma^z) \otimes \mathbf{1}^{\otimes 3})] \\
&= \frac{U}{4} [\mathbf{1} \otimes (\mathbf{1} - \sigma^z) \otimes \mathbf{1} \otimes (\mathbf{1} - \sigma^z) + (\mathbf{1} - \sigma^z) \otimes \mathbf{1} \otimes (\mathbf{1} - \sigma^z) \otimes \mathbf{1}] \\
&= \frac{U}{4} (2\mathbf{1}^{\otimes 4} - \sigma^z \otimes \mathbf{1}^{\otimes 3} - \mathbf{1} \otimes \sigma^z \otimes \mathbf{1}^{\otimes 2} - \mathbf{1}^{\otimes 2} \otimes \sigma^z \otimes \mathbf{1} - \mathbf{1}^{\otimes 3} \otimes \sigma^z \\
&\quad + \mathbf{1} \otimes \sigma^z \otimes \mathbf{1} \otimes \sigma^z + \sigma^z \otimes \mathbf{1} \otimes \sigma^z \otimes \mathbf{1}).
\end{aligned} \tag{5.25}$$

□

The Hamiltonian we obtained in Eq.5.23 has 11 terms. Let us label the  $i$ -th term,  $h_i$ , in the the same order it appears in this equation. It is straightforward to compute the commutation relations between all the pairs of terms, for instance,

$$[h_1, h_2] = -\frac{\tau^2}{4} (\sigma_1^x \sigma_0^x \sigma_1^y \sigma_0^y - \sigma_1^y \sigma_0^y \sigma_1^x \sigma_0^x) \stackrel{(a)}{=} -\frac{\tau^2}{4} (\sigma_1^x \sigma_0^x \sigma_1^y \sigma_0^y - \sigma_1^y \sigma_1^x \sigma_0^y \sigma_0^x) \tag{5.26}$$

$$\stackrel{(b)}{=} -\frac{\tau^2}{4} (\sigma_1^x \sigma_0^x \sigma_1^y \sigma_0^y - \sigma_1^x \sigma_1^y \sigma_0^x \sigma_0^y) \stackrel{(c)}{=} -\frac{\tau^2}{4} (\sigma_1^x \sigma_0^x \sigma_1^y \sigma_0^y - \sigma_1^x \sigma_0^x \sigma_1^y \sigma_0^y) = 0 \tag{5.27}$$

where at (a) and (c) we commute Pauli operators on different sites, and at (b) we use the usual Eq.4.2. Below, we represent the commutation relations for all of the pairs of terms obtained with the transformation. Commuting terms are marked with a dot.

|          | $h_1$ | $h_2$ | $h_3$ | $h_4$ | $h_5$ | $h_6$ | $h_7$ | $h_8$ | $h_9$ | $h_{10}$ | $h_{11}$ |
|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|----------|
| $h_1$    |       | •     | •     | •     | •     | •     | •     |       |       |          |          |
| $h_2$    |       |       | •     | •     | •     | •     | •     |       |       |          |          |
| $h_3$    |       |       |       | •     | •     |       |       | •     | •     |          |          |
| $h_4$    |       |       |       |       | •     |       |       | •     | •     |          |          |
| $h_5$    |       |       |       |       |       | •     | •     | •     | •     | •        | •        |
| $h_6$    |       |       |       |       |       |       | •     | •     | •     | •        | •        |
| $h_7$    |       |       |       |       |       |       |       | •     | •     | •        | •        |
| $h_8$    |       |       |       |       |       |       |       |       | •     | •        | •        |
| $h_9$    |       |       |       |       |       |       |       |       |       | •        | •        |
| $h_{10}$ |       |       |       |       |       |       |       |       |       |          | •        |
| $h_{11}$ |       |       |       |       |       |       |       |       |       |          |          |

As shown, we can construct two sets of mutually commuting terms of this Hamiltonian:  $S_1 = \{h_1, h_2, h_3, h_4\}$ ,  $S_2 = \{h_5, h_6, h_7, h_8, h_9, h_{10}, h_{11}\}$ . In each of these sets, all operators commute among themselves, but no

single operator from one set commutes with all the operators from the other set. We will use these sets to build different Trotter decompositions of the time-evolution operator.

Furthermore, we only need to simulate 10 of the 11 terms in this Hamiltonian, since one of them is a constant, which amounts to a phase we can easily replace back in the end. Therefore, the Hamiltonian we want to study is

$$H'_H = H_H - \frac{U}{2}\mathbf{1} \quad (5.29)$$

and the eigenvalues we are going to get are given in terms of the original eigenvalues in Eq.3.20 by

$$\varepsilon'_i = \varepsilon_i - \frac{U}{2} \quad (5.30)$$

We will simulate the evolution operator

$$\mathcal{U}'_H = e^{-itH'_H} = e^{-it(\sum S_1 + \sum S_2)} \quad (5.31)$$

and estimate  $\varphi'$  for an eigenvalue written as  $e^{2\pi i\varphi'}$ . That will let us know  $\varepsilon_i$ .

The simulation of this Hamiltonian requires 4 qubits in the second register. For the first register, as usual, we can choose as many qubits as we want according to the resolution we want to achieve. We need to build the circuit for the time-evolution operator  $\mathcal{U}'_H(t)$ , and since not all terms in  $H'_H$  commute, this time the Trotter decomposition has to be performed and it plays an important role in the process, as we will see. The two pieces of the time-evolution operator which we combine to build the Trotter order we need, are represented by the circuits in Fig.5.11. Here, the angles should be adjusted properly, accounting for time  $t$ , the Trotter number  $n$ , the exponent  $p$  of the unitarity applied according to the control qubit on the first register, and the Hamiltonian parameters  $\tau, U$ . These are given by

$$\begin{aligned} \theta_1 = \theta_2 = \theta_3 = \theta_4 &= -\tau tp/n \\ \theta_6 = \theta_7 = \theta_8 = \theta_9 &= -\theta_{10} = -\theta_{11} = -Utp/(2n) \end{aligned} \quad (5.32)$$

We will be performing estimations with the multiple-shot method by assuming the following Hamiltonian parameters

$$\tau = 0.34423, \quad U = 1.28473 \quad (5.33)$$

which gives us the eigenstates (refer back to Eq.3.21),

$$\left\{ \begin{array}{ll} \varepsilon_1 = -0.299235 & , |\psi_1\rangle \\ \varepsilon_2 = 0 & , |\psi_2\rangle \\ \varepsilon_3 = 0 & , |\psi_3\rangle \\ \varepsilon_4 = 0 & , |\psi_4\rangle \\ \varepsilon_5 = 1.28473 & , |\psi_5\rangle \\ \varepsilon_6 = 1.58396 & , |\psi_6\rangle \end{array} \right. \quad (5.34)$$

We will vary time inside the period of the phase signal obtained with each eigenstate (Eq.4.29), doing experiments for 100 time instants in this interval. This is the first example in which state preparation is not trivial with some of the eigenstates, namely  $|\psi_1\rangle, |\psi_4\rangle, |\psi_5\rangle$  and  $|\psi_6\rangle$ . We will prepare the eigenstates with the method by Shende et al. [94] and estimate directly the eigenvalue associated to each of them. As has already happened in the previous two examples, this is again not the approach one would follow for a really unknown

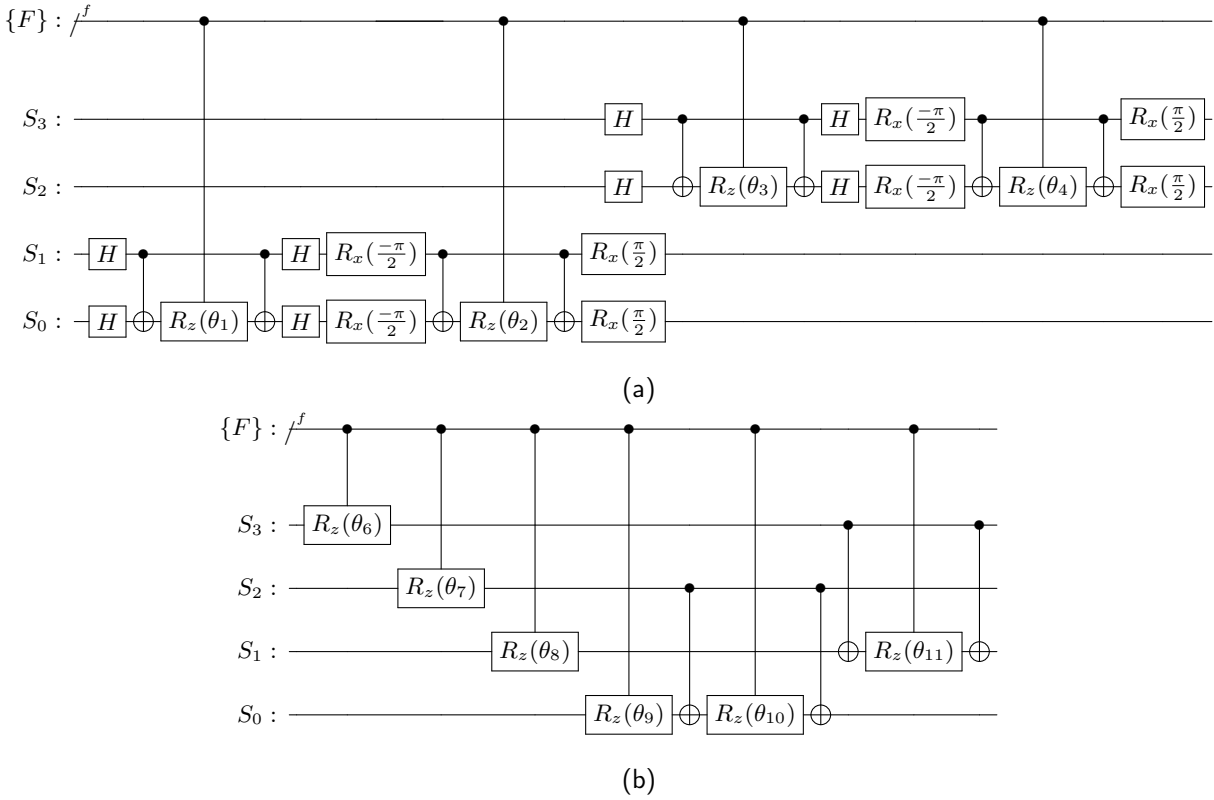


Figure 5.11: PEA Hubbard 2-site Hamiltonian.

eigensystem of some Hamiltonian. In the future, when we have sizable quantum computers available, we will need an extra step of guessing a good initial state for the second register close to the eigenstate we want to determine.

Right now we are only concerned with finding experimental proof for phase estimation algorithms. Before that, however, we want to understand how the circuit for the PEA with this model will perform on an ideal scenario. We start with unitary simulations on a classical emulator assuming full qubit connectivity. Let us choose 3 first register qubits for 3-bit resolution, which yields an algorithm on 7 qubits in total.

## Results

The method correctly approximates all eigenvalues from the slope coefficient  $m$  with,

$$\widehat{\varepsilon}_i = -2\pi m_i + \frac{U}{2} \quad (5.35)$$

for both Trotter orders. We ran 12 experiments: computing the 6 eigenvalues with the 2 Trotter expansions, always choosing  $n = 5$  and sampling 50 timestamps, at each one performing 1000 experiments. We calculate the absolute error each time,  $\varepsilon_i = \|\varepsilon_i - \widehat{\varepsilon}_i\|$ , where  $\varepsilon_i$  is the theoretical eigenvalue, and summarize the obtained results in Tab.5.5.

This serves the purpose of preliminary inspection of results, but now we want to better understand the effect of the first 2 Trotter orders on the final estimations. In Fig.5.12 we show the obtained results when varying the Trotter number from 1 to 9 to study its influence in the estimation of the ground state eigenvalue  $\varepsilon_1$ , and on the total number of gates obtained in the final circuits. We see the error of the estimation decreasing but not a clear distinction between 1<sup>st</sup> and 2<sup>nd</sup> Trotter orders. Regarding the total number of gates of the circuits we ran for 5 Trotter steps, having 3 first register qubits and all-to-all qubit connectivity, we got to

|                      | $\epsilon_i$ ( $1^{st}$ order) | $\epsilon_i$ ( $2^{nd}$ order) | $\epsilon_i$ |
|----------------------|--------------------------------|--------------------------------|--------------|
| $ \epsilon_1\rangle$ | 0.14713                        | 0.014224                       | -0.299235    |
| $ \epsilon_2\rangle$ | 0.01351                        | 0.01351                        | 0            |
| $ \epsilon_3\rangle$ | 0.01351                        | 0.01351                        | 0            |
| $ \epsilon_4\rangle$ | 0.01351                        | 0.01351                        | 0            |
| $ \epsilon_5\rangle$ | 0.01864                        | 0.01864                        | 1.28473      |
| $ \epsilon_6\rangle$ | 0.06364                        | 0.06173                        | 1.58396      |

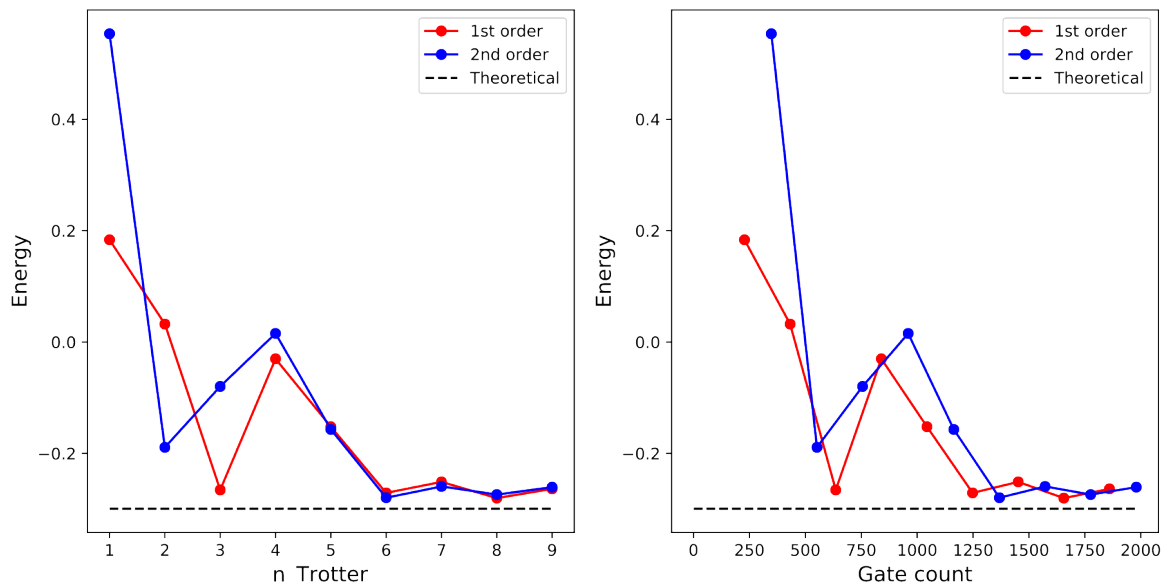
Table 5.5: All eigenvalues obtained in independent experiments with  $n = 5$  and both Trotter-Suzuki orders.

Figure 5.12: The 36 simulations represented here are unitary and assume full qubit connectivity.

- $1^{st}$  Trotter order (Eq.4.36): 1044 gates in total, of which 486 were two-qubit gates.
- $2^{nd}$  Trotter order (Eq.4.38): 1164 gates in total, of which 534 were two-qubit gates.
- $2^{nd}$  Trotter order “naive” (Eq.4.37): 1644 gates in total, of which 726 were two-qubit gates.

Bearing in mind that our goal is to actually implement this algorithm on the real device of IBM, these plots don't show good news. As we can see, the total number of gates quickly becomes very high for present-day devices, even considering total qubit connectivity. For just 1 Trotter step, using  $1^{st}$  order Trotter decomposition, the compiled PEA program with 3 first register qubits already has 228 gates in total, of which 102 are 2-qubit gates. These numbers are only considering the PEA itself and don't even count the circuit that needs to be built to prepare the initial state. When we include those numbers and take into consideration that in real devices the gate count increases due to imperfect connectivity, it becomes clear that with current error rates an coherence times this circuit will not produce good results experimentally.

### 5.3 Iterative phase estimation

In this section we will understand some of the advantages of using the IPEA instead of the PEA when we initialize the second register from an eigenstate. We will be performing experiments on real devices making use of the operational method (a) described in section 2.5. We chose this method for logistical reasons; since

the interaction with the devices is done at a distance and suffers from a communication bottleneck, this is the most suitable method for performing a sufficient number of computations.

### 5.3.1 Hubbard 2-site 2-electron sector

We are not giving up on simulating the Hubbard dimer on a real quantum computer with an exact quantum algorithm. We will now improve our strategy by switching from the PEA to the IPEA and further reducing the number of qubits needed to simulate the time-evolution operator. We will use 2 qubits instead of 4 with the compromise that we consider only the 2 electron sector of the 2 site Hubbard Hamiltonian.

As we proved before, by employing the Jordan-Wigner transformation we can express the 2-site Hubbard Hamiltonian in terms of Pauli operators, which allows us to mimic the algebra of fermions with the algebra of qubits, Eq.5.23. This Hamiltonian lives in a 16 dimensional Hilbert space but we are only interested in the half-filling sector, thus we would like to concentrate on the 6 dimensional half-filling block in Eq.3.18. As shown in [97], it is possible to write a simpler spin Hamiltonian which is iso-spectral to the 6 dimensional half-filling block of the Jordan-Wigner transformed Hamiltonian. This is

$$H_{HHF} = -\tau (\sigma_0^x + \sigma_1^x) + \frac{U}{2} (\mathbf{1} + \sigma_0^z \sigma_1^z) \quad (5.36)$$

$$= \begin{pmatrix} U & -\tau & -\tau & 0 \\ -\tau & 0 & 0 & -\tau \\ -\tau & 0 & 0 & -\tau \\ 0 & -\tau & -\tau & U \end{pmatrix} \quad (5.37)$$

and has the eigensystem summarized in Table 5.6.

| Label                   | Energy  | Eigenket  | Normalization constant  |
|-------------------------|---|---|---|
| $ \varepsilon_1\rangle$ | $\frac{U}{2} \left( 1 - \sqrt{16 \left(\frac{t}{U}\right)^2 + 1} \right)$ | $ 00\rangle +  11\rangle + \frac{U + \sqrt{16t^2 + U^2}}{4t} ( 10\rangle +  01\rangle)$ | $\left( \frac{(U + \sqrt{16t^2 + U^2})^2}{8t^2} + 2 \right)^{-1/2}$ |
| $ \varepsilon_2\rangle$ | 0   | $ 10\rangle -  01\rangle$   | $1/\sqrt{2}$  |
| $ \varepsilon_3\rangle$ | $U$   | $ 11\rangle -  00\rangle$   | $1/\sqrt{2}$  |
| $ \varepsilon_4\rangle$ | $\frac{U}{2} \left( 1 + \sqrt{16 \left(\frac{t}{U}\right)^2 + 1} \right)$ | $ 00\rangle +  11\rangle + \frac{U - \sqrt{16t^2 + U^2}}{4t} ( 10\rangle +  01\rangle)$ | $\left( \frac{(U - \sqrt{16t^2 + U^2})^2}{8t^2} + 2 \right)^{-1/2}$ |

Table 5.6: Eigensystem of the 2-site 2-electron Hubbard Hamiltonian in the compact 2-qubit representation.

This Hamiltonian will only need 2 qubits to be implemented. Clearly, we have one more improvement here:  $H_{HHF}$  only has 4 terms, and can even be reduced to only 3 terms similarly to what we did in the previous section, since one of them is a constant. The Hamiltonian we are interested in inspecting is then

$$H'_{HHF} = H_{HHF} - \frac{U}{2} \mathbf{1} \quad (5.38)$$

and again, by simulating  $\mathcal{U}'_{HHF} = e^{-itH'_{HHF}}$ , the eigenvalues we are going to get are given as

$$\varepsilon'_i = \varepsilon_i - \frac{U}{2}. \quad (5.39)$$

To apply the Trotter-Suzuki decomposition we can trivially group these 3 terms, labeled in the order they

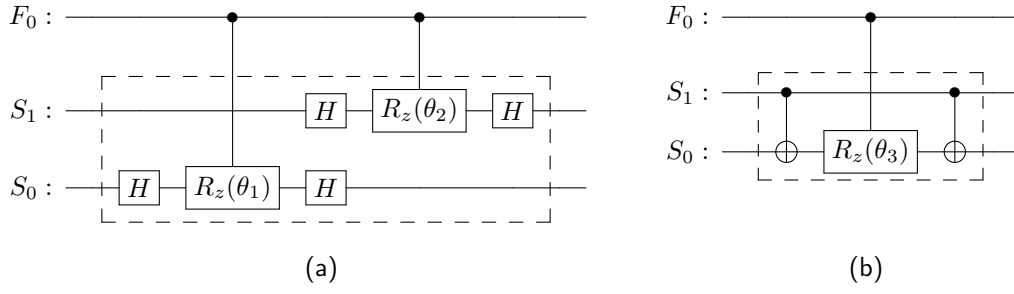


Figure 5.13: IPEA implementation scheme of the 2-site Hubbard Hamiltonian at half-filling. The simulation requires only 3 qubits: 1 in the first register, and 2 in the second register. The first register qubit line is represented on top. The scheme circuits in (a) and (b) represent the two pieces (boxed) of the time-evolution operator which we combine to build the needed trotter order. Rotation angles  $\theta_1$ ,  $\theta_2$  and  $\theta_3$  should be adjusted properly, accounting for time, Trotter number  $n$ , exponent  $k$ , and the Hamiltonian parameters.

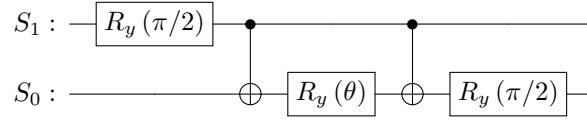


Figure 5.14: With  $\theta = 0.141897054604164$ , this is the initialization circuit for the ground state obtained with the parameters in Eq.5.42. It was built following the algorithm in [94].

appear in Eq.5.36, into two non-commuting sets

$$S_1 = \{h_1, h_2\}, \quad (5.40)$$

$$S_2 = \{h_4\},$$

such that implementing the time-evolution propagator will use the decomposition according to

$$\mathcal{U}'_{HHF} = e^{-itH'_{HHF}} = e^{-it(\sum S_1 + \sum S_2)}. \quad (5.41)$$

Let us experiment with different Hamiltonian parameters this time and choose

$$\tau = 0.35, \quad U = 0.2 \quad (5.42)$$

which gives us the eigenstates,

$$\begin{cases} \varepsilon_1 = -0.607107 & , |\psi_1\rangle \\ \varepsilon_2 = 0 & , |\psi_2\rangle \\ \varepsilon_3 = 0.2 & , |\psi_3\rangle \\ \varepsilon_4 = 0.807107 & , |\psi_4\rangle \end{cases} \quad (5.43)$$

We will only focus on the ground state. To prepare this initial eigenket, we will again use the same method we have been using in the previous sections (see Fig.5.14). With the defined Hamiltonian parameters, this state is

$$|\varepsilon_1\rangle \approx (0.463298, 0.534187, 0.534187, 0.463298)^T \quad (5.44)$$

We will be performing estimations with the multiple-shot method. For the ground state, the phase signal will have period a little over  $T = 10$ . Let us increase resolution to  $m = 6$  and simulate 100 time points, while also testing both the first and the second Trotter-Suzuki orders.



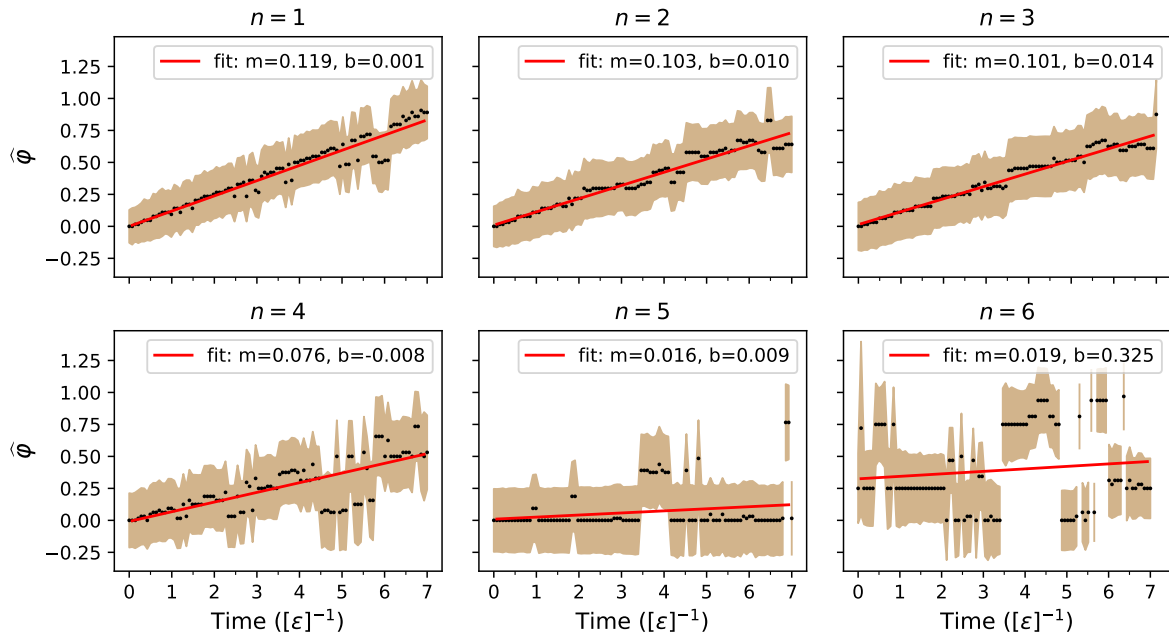


Figure 5.15: 1<sup>st</sup> Trotter order results for the APMFs obtained on *ibmq\_20\_tokyo*.

## Results

The results for the ground state eigenvalue estimation are presented in Fig.5.15 for first Trotter-Suzuki order and in Fig.5.16 for the second order. From left to right, top to bottom, the figures show Trotter exponents ranging from  $n = 1$  to  $n = 6$ . The phase estimation is plotted for each time-step, corresponding to the binary string constructed by taking the basis state obtained most frequently at each iteration. For each time-step, the standard deviation of the reconstructed APMF is also represented by the brown region centered at each data-point.

Let us start with a general inspection of these plots to try and understand the experiment. First, notice that for both Trotter-Suzuki orders, the results for  $n = 1$ ,  $n = 2$  and  $n = 3$  are the only usable ones. For the first order decomposition,  $n = 4$  results are already quite dispersed, while for second order they are useless. This behavior is understood when we take into account the number of gates per iteration yielded by the program for each of the 12 experiments. Increasing the Trotter-Suzuki number linearly increases the total number gates needed per iteration. This, in turn, gives the system enough time to decohere and increases the vulnerability of the final computation to the errors accumulated due to noise.

Thus, on the plots, it is clear that a threshold number of gates has been exceeded rendering results for higher Trotter-Suzuki exponents very bad in both orders of the decomposition, starting to deteriorate earlier for the second order on account of this one needing more gates.

We perform linear regression on the data-points and the fit line is also plotted. The optimal parameters of the fit, as well as their statistical quantifiers, are shown in Tab.5.7 for the first order approximation and in Tab.5.8 for the second order approximation. Notice that we always decided to evolve the system up to a time  $t = 7$  smaller than the expected period  $T$  of the  $\varphi \bmod 1$  signal. This choice was made to avoid the later times for which the Trotter-Suzuki decompositions get increasingly far from optimal, hence limiting regression to the region where we expect the decomposition to best match the actual evolution operator  $\mathcal{U}'_{HFF}$ .

These linear regression results are represented and summarized in Fig.5.17 together with some other analysis. The conclusions become apparent by looking at this.

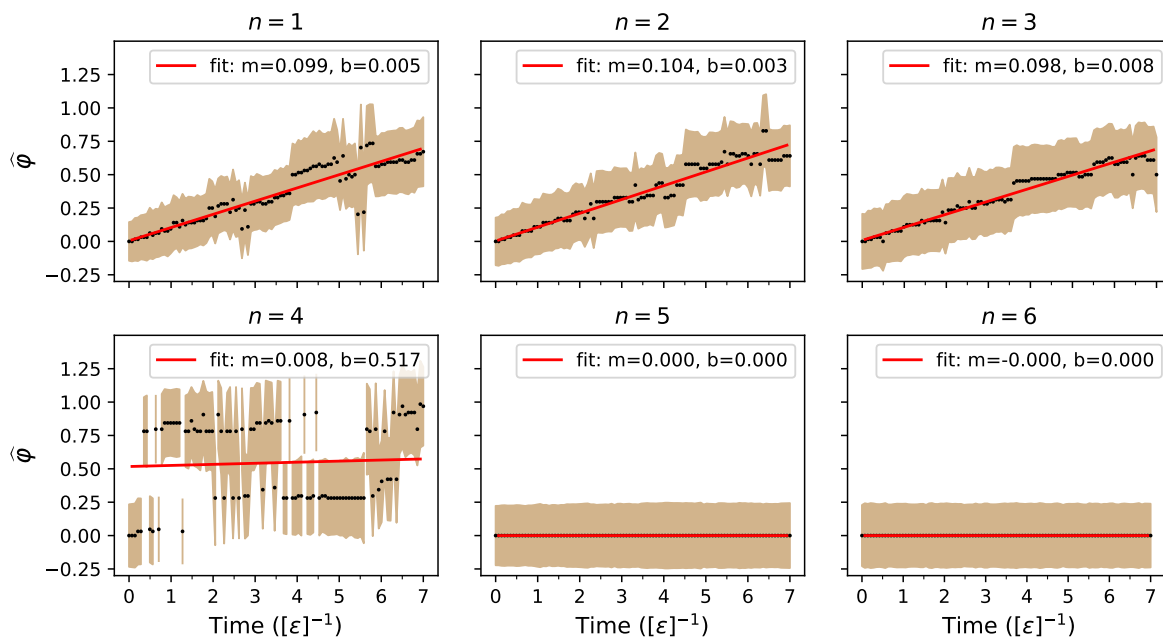


Figure 5.16:  $2^{nd}$  Trotter order results for the APMFs obtained on *ibmq\_20\_tokyo*.

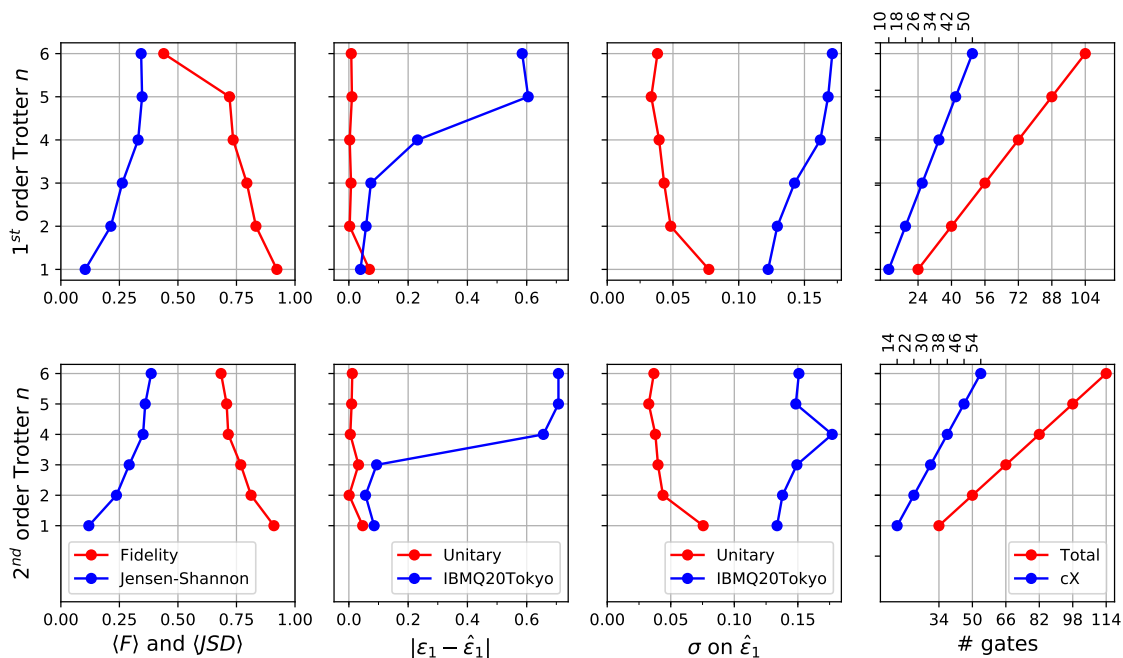


Figure 5.17: Here we show the average Fidelity and Jensen-Shannon divergence of the *APMF*'s from all the time-steps on each Trotter number; the absolute error on the estimated ground state energy obtained from the linear regression; the standard deviation error on the determined energy; and the gate counts on the circuits built for each Trotter order and number.

|         | $t$ interval | $m$   | $\sigma_m$ | $b$    | $\sigma_b$ | $\chi^2/(N-2)$ | $\hat{\varepsilon}_1$ | $\sigma_{\hat{\varepsilon}_1}$ | $\epsilon =  \varepsilon_1 - \hat{\varepsilon}_1 $ |
|---------|--------------|-------|------------|--------|------------|----------------|-----------------------|--------------------------------|--|
| $n = 1$ | $[0, 7]$     | 0.119 | 0.019      | 0.001  | 0.072      | 0.023          | -0.65                 | 0.122                          | 0.039  |
| $n = 2$ | $[0, 7]$     | 0.103 | 0.021      | 0.010  | 0.079      | 0.012          | -0.559                | 0.129                          | 0.058  |
| $n = 3$ | $[0, 7]$     | 0.101 | 0.023      | 0.014  | 0.088      | 0.006          | -0.533                | 0.142                          | 0.074  |
| $n = 4$ | $[0, 7]$     | 0.076 | 0.026      | -0.008 | 0.096      | 0.053          | -0.376                | 0.162                          | 0.232  |
| $n = 5$ | $[0, 7]$     | 0.016 | 0.027      | 0.009  | 0.104      | 0.081          | -0.002                | 0.168                          | 0.605  |
| $n = 6$ | $[0, 7]$     | 0.019 | 0.027      | 0.325  | 0.116      | 0.384          | -0.022                | 0.171                          | 0.584  |

Table 5.7: Linear regression results obtained with the experimental data of the IPEA  $H_{HHF}$  simulation for 1<sup>st</sup> order Trotter-Suzuki decomposition on *ibmq\_20\_tokyo*.

|         | $t$ interval | $m$   | $\sigma_m$ | $b$   | $\sigma_b$ | $\chi^2/(N-2)$ | $\hat{\varepsilon}_1$ | $\sigma_{\hat{\varepsilon}_1}$ | $\epsilon =  \varepsilon_1 - \hat{\varepsilon}_1 $ |
|---------|--------------|-------|------------|-------|------------|----------------|-----------------------|--------------------------------|--|
| $n = 1$ | $[0, 7]$     | 0.099 | 0.021      | 0.005 | 0.077      | 0.027          | -0.522                | 0.134                          | 0.085  |
| $n = 2$ | $[0, 7]$     | 0.104 | 0.022      | 0.003 | 0.084      | 0.012          | -0.551                | 0.138                          | 0.056  |
| $n = 3$ | $[0, 7]$     | 0.098 | 0.024      | 0.008 | 0.092      | 0.005          | -0.513                | 0.149                          | 0.094  |
| $n = 4$ | $[0, 7]$     | 0.008 | 0.028      | 0.517 | 0.109      | 0.308          | 0.049                 | 0.177                          | 0.558  |
| $n = 5$ | $[0, 7]$     | 0.000 | 0.023      | 0.000 | 0.094      | 0.000          | 0.100                 | 0.148                          | 0.507  |
| $n = 6$ | $[0, 7]$     | 0.000 | 0.024      | 0.000 | 0.097      | 0.000          | 0.100                 | 0.151                          | 0.507  |

Table 5.8: Linear regression results obtained with the experimental data of the IPEA  $H_{HHF}$  simulation for 2<sup>nd</sup> order Trotter-Suzuki decomposition on *ibmq\_20\_tokyo*.

We simulated classically the unitary evolution of the circuits in all of the 12 experiments, comparing the obtained APMFs with the experimental ones by calculating Fidelity and Jensen-Shannon divergence. The plots show what is expected: both measures indicate that increasing  $n$  deteriorates the computational results, in both Trotter-Suzuki orders.

Looking at the absolute error  $|\varepsilon_1 - \hat{\varepsilon}_1|$  we confirm that for  $n \geq 4$  it becomes increasingly very bad. While for  $n \leq 3$  we always have  $|\varepsilon_1 - \hat{\varepsilon}_1| < \sigma_1$ , for  $n \geq 4$  we always verify  $|\varepsilon_1 - \hat{\varepsilon}_1| > \sigma_1$ . This is true for both orders of the decomposition.

One could expect that, for both orders of the Trotter-Suzuki decompositions, we would first see that, for lower  $n$ , the energy estimation would deviate from exact due only to the inefficiency of the Trotter expansion to approximate the time-evolution operator accurately; and then, as  $n$  was increased, the estimation would start improving until a threshold number of applied gates was achieved in which noise would progressively degrade results and not allow for higher Trotter exponents to be implemented. Degradation of the computation after the  $n = 4$  threshold is clear from the results. However, initial improvement is only mildly visible for second order, and even though it is predicted by the unitary simulation for the first order, experimentally there was a fluctuation that actually favored a good result for  $n = 1$ . In fact the standard deviation errors on the estimations of  $\varepsilon_1$  cover these fluctuations and we cannot clearly conclude this initial improvement.

The most important reason why it is difficult to conclude that results get initially better as we increase the Trotter-Suzuki number can be seen from the unitary curve in the plots for  $|\varepsilon_1 - \hat{\varepsilon}_1|$ . From these absolute errors we can actually see that convergence is very fast with the Hamiltonian parameters (Eq.5.42) we chose. However, note that we are not exploring all the time interval of one period of the phase signal. If we were, maybe we would see worst results for lower Trotter numbers, which would get better for higher ones.

We conclude that, for the Hamiltonian parameters we chose (Eq.5.42), first and second order Trotter-Suzuki decompositions and low Trotter exponents ( $\leq 3$ ) yield very similar results in the analyzed time interval, going from 0 to  $t_{max} < T$ .

### 5.3.2 Benchmarking accuracy with the Zeeman Hamiltonian

In this section we ask the question of how accurate we can get using the IPEA on *ibmq\_20\_tokyo*. We will use the Zeeman Hamiltonian (Eq.3.8) and probe its  $\varepsilon_1$  eigenvalue by setting the Hamiltonian parameter

$$c = 3.82736029384758, \quad (5.45)$$

which of course gives

$$\varepsilon_1 = 3.82736029384758. \quad (5.46)$$

For different number of IPEA iterations  $M$ , that is, different resolution, we will estimate this eigenvalue and compute errors in accuracy and precision. With that, we want understand how errors scale with  $M$  and possibly infer something about the physical processes in the device. Note that with the Zeeman hamiltonian, an IPEA iteration only uses 4 one-qubit gates and 2  $cX$  gates, thus we expect that noise won't play a very large role, and the scaling will maybe depend on something else, like accuracy and precision of the angle of the rotation gates implemented on the hardware.

#### Results obtained by doing linear regression considering the error-bars

Once again we performed linear regression by minimizing the chi-squared function 5.13 to obtain the slope coefficient  $m$  and its standard error. The experimental points we use for fitting are just like the ones in the previous section, the  $\hat{\varphi}$  with the error-bars given by the circular standard phase deviation obtained from the APMF. From those we compute the eigenvalue and its relative errors in precision and accuracy. We perform this procedure for each independent experiment, each one done by choosing a different resolution  $M$  for the estimation. Fig.5.18 shows the results obtained for the first 13 levels of resolution.

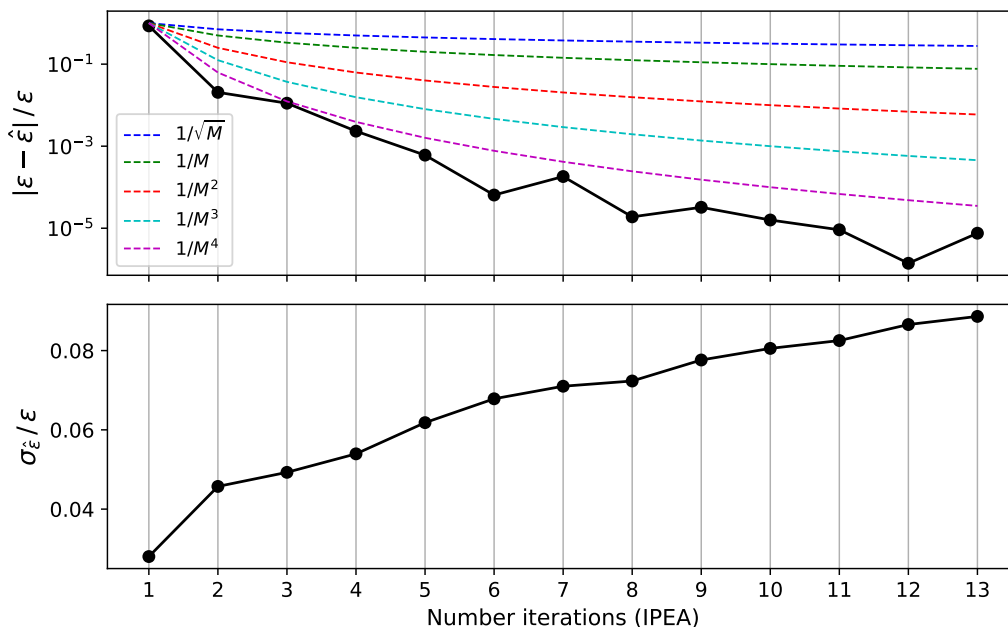


Figure 5.18: All the results obtained by doing linear regression to minimize the chi-squared function taking into account the circular standard phase deviation as error-bars for each data-point.

The results we obtain show a very good scaling of the accuracy error with  $M$ , with a monotonic decreasing trend better than  $1/M^4$ . Surprisingly, we verify the standard error in precision is increasing as we increase the

resolution of the IPEA. However, the fact that the error in accuracy is correctly and steadily decreasing hints that the error in precision is somehow overestimated. Because of that we re-do the linear regression ignoring the error-bars in the data-points next.

**Results obtained by doing linear regression without considering error-bars on the data-points**

This time we took all the experimental results and performed linear regression by minimizing simply the squares of the difference between the observed  $\hat{\varphi}_i$  values and the fitting function evaluated at  $t_i$ , i.e.

$$\sum_{i=1}^N (\hat{\varphi}_i - f(t_i))^2. \tag{5.47}$$

This method still allows us to obtain the best fitting parameters  $m$  and  $b$  in  $f(t_i) = mt + b$ , with  $\sigma_m$  and  $\sigma_b$ . Therefore we are able to obtain accuracy and precision errors for all the eigenvalue estimations, which we plot in Fig.5.19.

Looking at the results, we now see that the sub  $1/M^4$  scaling of the relative error in accuracy is now accompanied by a sub  $2^{-M}$  relative error in precision. These are very good results that also show that for most of the tested values of  $M$  the accuracy error is safely contained in the range of the precision error.

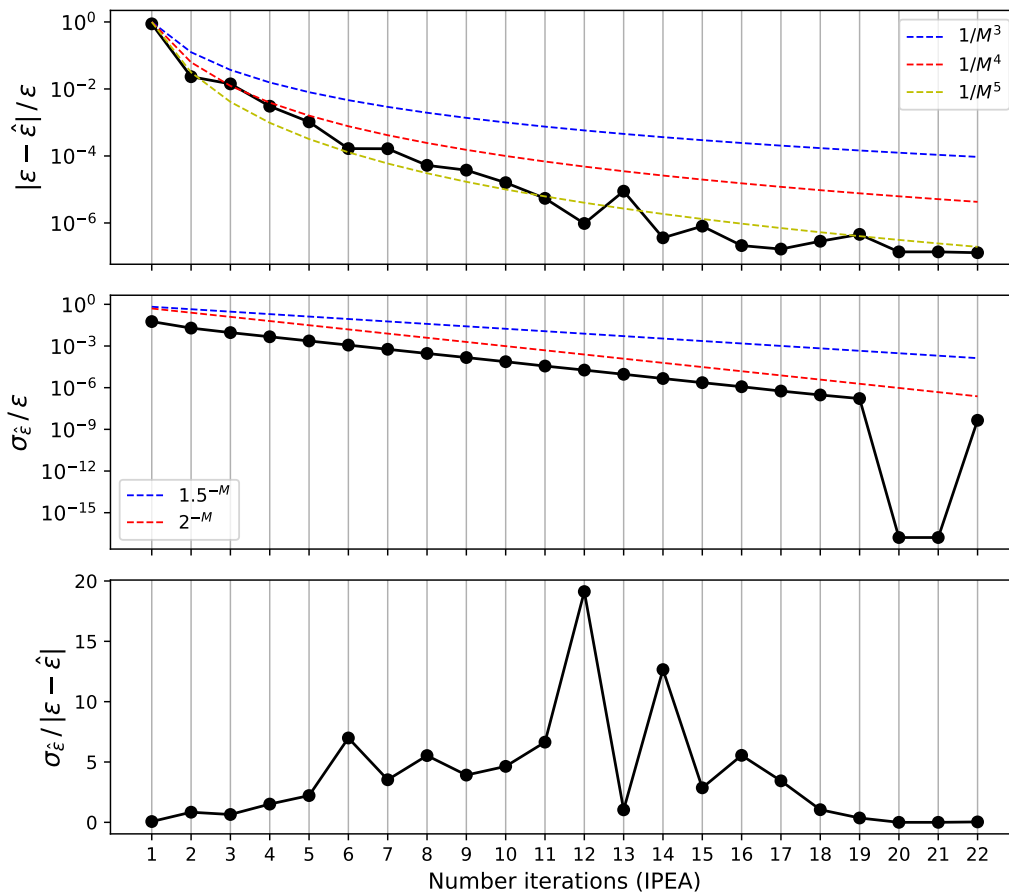


Figure 5.19: Here we show the results obtained by doing linear regression without considering the error-bars on data-points for each  $M$ . The scaling of the relative errors in accuracy and precision are shown as well as the ratio between both.



## Chapter 6

# Conclusions

*“Heavier-than-air flying machines are impossible.”*

— Lord Kelvin, 1895

In this work, we have encountered the need to use quantum computers to efficiently simulate strongly correlated systems. We understood that the exponential wall problem in classical exact diagonalization will not allow us to understand exactly some important quantum mechanical systems living in Hilbert spaces whose dimension is intractable to current and predicted classical computers. To that end, we studied the approach of digital quantum simulation as an alternative. In particular, we explored how to simulate Hamiltonian dynamics on quantum computers through the use of the phase estimation (PEA) and iterative phase estimation (IPEA) algorithms, which we explored thoroughly.

The simulated systems were of two types: spins and fermions. For the fermionic case, we needed to additionally map fermionic states into qubit states, which we did using the Jordan-Wigner transformation. We also presented the Trotter decomposition, necessary to implement the unitary evolution operator when not all of the Hamiltonian terms commuted, and provided the first two orders of this expansion. In the case of the second order decomposition, we showed two ways of coding it. Furthermore, we understood how essential the initial state preparation step is for the success of the PEA and the IPEA: to better determine some of the eigenvalues, these algorithms rely on being fed with a state with a large overlap with the respective eigenket.

On the theoretical side, we have come up with some original contributions for the PEA and the IPEA. Firstly, we proposed a modification to the post-processing part of the PEA which allowed us to increase the accuracy of the estimated phase. One way to look at rule 2 is that we can have one qubit less on the first register and achieve the same accuracy bound we would obtain with rule 1 (even better, in fact). This improvement may not be too relevant when we attain the luxury of coding the PEA with many first register qubits, but it makes a noticeable difference for the experimental implementations constrained to have this number small, as the ones we have available at the moment.

Secondly, we exposed the two operational methods of using the IPEA ( $a$  and  $b$ ) and, for the non-optimal one, IPEA- $a$ , we proposed a way to reconstruct an approximation of the PMF over all the possible sequences of the intermediate basis states, which was useful to quantify the circular standard phase deviation, thus characterizing dispersion of the distribution. Experiments were performed on unitary simulators confirming all our methods.

Besides the methodological contributions, we ventured with newly available quantum computers to test our implementations in real experimental conditions, making use of one of IBM's quantum computers with 20 qubits, to which we had exclusive access.

Initializing the second register qubit states from the eigenvectors of the tested Hamiltonians, we were able to determine all of the eigenvalues. In the case of the Hubbard model, for which a Trotterization procedure is required, we studied the optimal number of Trotter steps, with the first and second orders of the decomposition, before noise and decoherence destroyed the information that could be extracted from the simulation. Moreover, we also looked into the limits for precision and accuracy achievable in the hardware when simulating the Zeeman Hamiltonian with the IPEA. Results gave experimental proof for our methods and confirmed quantum computers can begin to solve some real problems.

However, we also found that digital quantum simulations based on phase estimation can only handle trivial case Hamiltonians with state-of-the-art hardware. In this context, the IPEA proved to be better suited for present-day devices, since it can be implemented with only one auxiliary qubit and achieve much greater accuracy.

From the theory, we understood the superiority of the IPEA-b relatively to the PEA in the sense that it can use only one auxiliary qubit and recover the full PMF over all binary string encodings of the phase, just like the PEA, on top of allowing for arbitrary resolution not limited by the number of qubits. Hence, we understood that simulations based on phase estimation are essentially only limited by the connectivity and depth needed to implement the unitarity to be studied, and this is the crucial part of the algorithm that needs to be improved.

Despite understanding the potential of the IPEA-b, due to logistical reasons and to the architecture of the current devices by IBM, which doesn't allow for measurement feedback and state re-setting, we could only implement our simulations with IPEA-a, and were therefore not able to make full exploration of the IPEA potentialities. Nevertheless, new platforms are currently under development which may allow overcoming these limitations. Besides that, for the researchers with physical access to these computers, it will be much easier to perform these computations even without improving these issues, since the long waiting times of going through the queue thousands of times for obtaining solutions with IPEA-b are eliminated.

This work has also been productive in uncovering new problems that would deserve further investigation. One example are the two open conjectures in chapter 2 regarding the error associated with rule 2. Another one is that we also made evident the possibility of using the PEA (or IPEA-b) to determine multiple eigenvalues of an Hamiltonian from a single circuit with a single initial prepared general state, as well as the coefficients of the decomposition of this state on the basis of the respective eigenkets. We are not aware of a general method like what we have touched on having already been proposed in the literature. In the future, we expect to take a closer look into this problem and improve the method with appropriate signal processing techniques. Possibilities include making use of established methods such as the Hough transform, or exploring other approaches tailored to this specific task, such as fitting the data to the probabilistic model for the PMF at all the time-steps after filtering noise, with appropriate constraints on its parameters, or even other approaches.

We also plan to further improve on the methods we presented. For instance, the approach we took to determine the Hamiltonian eigenvalues relied on linear fitting the measured phase as we varied the time parameter of the unitary evolution operator. To do that, we looked at the data to identify a region where the fitting could be done, since what we measured was  $\hat{\varphi}_u \bmod 1$ . This approach could be improved to avoid the need to perform preliminary inspection of the data to determine the domain where linearity is observed, automating data analysis; one possibility would be to apply some periodic function to the measured phases, such as a cosine, and fit the data to such function.

The PEA and the IPEA proved to be very useful and important tools to extract eigenvalues exactly for the full energy spectrum of several Hamiltonians. However, their usefulness on current hardware is limited by the depth of the circuits needed to implement the unitary matrix to be studied, specially when Trotterization is needed.



Looking forward, variational quantum eigensolver (VQE) algorithms seem a more feasible approach to obtain the ground state energies of more complex hamiltonians with state-of-the-art hardware. This hybrid classical-quantum algorithm can be used to implement different trial ansatz and invariably achieves lower circuit depths for larger systems, being explicitly constructed for low depth architectures. This approach formulates the problem of finding ground state eigenvalues as an optimization problem on the Hilbert space. Future work will also include exploration of this method with the Hamiltonians presented in this work or more complex instances of those.

For a final outlook on the future of the field, we point out that while the computers used in this dissertation are based on superconducting circuits, the ion trap technology is rapidly entering the race - a computer with more than 10 fully connected qubits and unprecedented performance is expected to be deployed soon. Personally, I find it easy to see published results in the near future of an IPEA-b simulation using an Hamiltonian with a record-breaking dimension.



# Bibliography

- [1] A. G. Bromley. Charles babbage's analytical engine, 1838. *Annals of the History of Computing*, 4(3):196–217, July 1982.
- [2] A. M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265.
- [3] Alonzo Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58(2):345–363, 1936.
- [4] R. Landauer. Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 5(3):183–191, July 1961.
- [5] C. H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17(6):525–532, Nov 1973.
- [6] Charles H. Bennett. The thermodynamics of computation—a review. *International Journal of Theoretical Physics*, 21(12):905–940, Dec 1982.
- [7] Alexander Semenovitch Holevo. Bounds for the quantity of information transmitted by a quantum communication channel. *Problemy Peredachi Informatsii*, 9(3):3–11, 1973.
- [8] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, July 1948.
- [9] Roman S. Ingarden. Quantum information theory. *Reports on Mathematical Physics*, 10(1):43 – 72, 1976.
- [10] Paul Benioff. The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines. *Journal of Statistical Physics*, 22(5):563–591, May 1980.
- [11] R. P. Feynman. Simulating Physics with Computers. *International Journal of Theoretical Physics*, 21:467–488, June 1982.
- [12] Manin Yuri. *Computable and Uncomputable*. Sovetskoye Radio, Moscow, 1980.
- [13] D. Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 400(1818):97–117, 1985.
- [14] Charles H. Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K. Wootters. Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels. *Phys. Rev. Lett.*, 70:1895–1899, Mar 1993.

- [15] David Hanneke, Jonathan P Home, John D Jost, Jason M Amini, Dietrich Leibfried, and David J Wineland. Realization of a programmable two-qubit quantum processor. *Nature Physics*, 6(1):13, 2010.
- [16] P. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science(FOCS)*, volume 00, pages 124–134, 11 1994.
- [17] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219. ACM, 1996.
- [18] Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, 1996.
- [19] Seth Lloyd. Computational capacity of the universe. *Physical Review Letters*, 88(23):237901, 2002.
- [20] Seth Lloyd. *Programming the Universe: A Quantum Computer Scientist Takes on the Cosmos*. Vintage, 2007.
- [21] Gerard't Hooft. Dimensional reduction in quantum gravity. *arXiv preprint gr-qc/9310026*, 1993.
- [22] Leonard Susskind. The world as a hologram. *Journal of Mathematical Physics*, 36(11):6377–6396, 1995.
- [23] Seth Lloyd. A theory of quantum gravity based on quantum computation. *arXiv preprint quant-ph/0501135*, 2005.
- [24] Brian Swingle. Entanglement renormalization and holography. *Physical Review D*, 86(6):065007, 2012.
- [25] Gia Dvali and Mischa Panchenko. Black hole based quantum computing in labs and in the sky. *Fortschritte der Physik*, 64(8-9):569–580, 2016.
- [26] Leonard Susskind. Dear qubitizers,  $gr = qm$ . *arXiv preprint arXiv:1708.03040*, 2017.
- [27] Leonard Susskind. Pitp lectures on complexity and black holes. *arXiv preprint arXiv:1808.09941*, 2018.
- [28] D. Deutsch. Quantum computational networks. 425:73–90, 09 1989.
- [29] A.B. Finnila, M.A. Gomez, C. Sebenik, C. Stenson, and J.D. Doll. Quantum annealing: A new method for minimizing multidimensional functions. *Chemical Physics Letters*, 219(5):343 – 348, 1994.
- [30] Tadashi Kadowaki and Hidetoshi Nishimori. Quantum annealing in the transverse ising model. *Phys. Rev. E*, 58:5355–5363, Nov 1998.
- [31] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106*, 2000.
- [32] Richard Jozsa. An introduction to measurement based quantum computation. *NATO Science Series, III: Computer and Systems Sciences. Quantum Information Processing-From Theory to Experiment*, 199:137–158, 2006.
- [33] Adriano Barenco, Charles H Bennett, Richard Cleve, David P DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Physical review A*, 52(5):3457, 1995.
- [34] Farrokh Vatan and Colin Williams. Optimal quantum circuits for general two-qubit gates. *Physical Review A*, 69(3):032315, 2004.
- [35] B Kraus and JI Cirac. Optimal creation of entanglement using a two-qubit gate. *Physical Review A*, 63(6):062309, 2001.

- [36] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2011.
- [37] Joseph Fourier. *Théorie analytique de la chaleur*. Firmin Didot Père et Fils, 1822.
- [38] Elias M. Stein and Guido Weiss. *Introduction to Fourier Analysis on Euclidean Spaces. (PMS-32)*. Princeton University Press, 1971.
- [39] Robert B. Griffiths and Chi-Sheng Niu. Semiclassical Fourier transform for quantum computation. *Phys. Rev. Lett.*, 76:3228–3231, 1996.
- [40] N. D. Mermin. *Quantum Computer Science*. August 2007.
- [41] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.
- [42] D. S. Abrams and S. Lloyd. Quantum Algorithm Providing Exponential Speed Increase for Finding Eigenvalues and Eigenvectors. *Physical Review Letters*, 83:5162–5165, December 1999.
- [43] H. Wang, L.-A. Wu, Y.-X. Liu, and F. Nori. Measurement-based quantum phase estimation algorithm for finding eigenvalues of non-unitary matrices. 82(6):062303, December 2010.
- [44] Alán Aspuru-Guzik, Anthony D. Dutoi, Peter J. Love, and Martin Head-Gordon. Simulated quantum computation of molecular energies. *Science*, 309(5741):1704–1707, 2005.
- [45] James D. Whitfield, Jacob Biamonte, and Alán Aspuru-Guzik. Simulation of electronic structure hamiltonians using quantum computers. *Molecular Physics*, 109(5):735–750, 2011.
- [46] Jacob T. Seeley, Martin J. Richard, and Peter J. Love. The bravyyi-kitaev transformation for quantum computation of electronic structure. *The Journal of Chemical Physics*, 137(22):224109, 2012.
- [47] Peter O'Malley, Ryan Babbush, Ian Kivlichan, Jonathan Romero, Jarrod McClean, Rami Barends, Julian Kelly, Pedram Roushan, Andrew Tranter, Nan Ding, Brooks Campbell, Yu Chen, Zijun Chen, Ben Chiaro, Andrew Dunsworth, Austin Fowler, Evan Jeffrey, Anthony Megrant, Josh Mutus, Charles Neil, Chris Quintana, Daniel Sank, Ted White, Jim Wenner, Amit Vainsencher, Peter Coveney, Peter Love, Hartmut Neven, Alán Aspuru-Guzik, and John Martinis. Scalable quantum simulation of molecular energies. *Physical Review X*, 6:031007, 2016.
- [48] Dave Wecker, Matthew Hastings, Nathan Wiebe, Bryan Clark, Chetan Nayak, and Matthias Troyer. Solving strongly correlated electron models on a quantum computer. June 2015.
- [49] Mark de Burgh and Stephen D. Bartlett. Quantum methods for clock synchronization: Beating the standard quantum limit without entanglement. *Phys. Rev. A*, 72:042301, Oct 2005.
- [50] T. C. Ralph B. C. Travaglione, G. J. Milburn. Phase estimation as a quantum nondemolition measurement. arXiv:quant-ph/0203130.
- [51] Kanti V. Mardia and Peter E. Jupp. *Directional Statistics*. Wiley, 1999.
- [52] A. Y. Kitaev. Quantum measurements and the Abelian Stabilizer Problem. *eprint arXiv:quant-ph/9511026*, November 1995.
- [53] Miroslav Dobsicek, Goran Johansson, Vitaly Shumeiko, and G Wendin. Arbitrary accuracy iterative quantum phase estimation algorithm using a single ancillary qubit: A two-qubit benchmark. 76, 09 2007.

- [54] Terry Rudolph and Lov Grover. Quantum communication complexity of establishing a shared reference frame. *Phys. Rev. Lett.*, 91:217905, Nov 2003.
- [55] Z. Ji, G. Wang, R. Duan, Y. Feng, and M. Ying. Parameter estimation of quantum channels. *IEEE Transactions on Information Theory*, 54(11):5172–5185, Nov 2008.
- [56] Caleb J O’Loan. Iterative phase estimation. *Journal of Physics A: Mathematical and Theoretical*, 43(1):015301, 2009.
- [57] Robin Harper and Steven Flammia. Fault tolerance in the ibm q experience. *arXiv preprint arXiv:1806.02359*, 2018.
- [58] Joschka Roffe, David Headley, Nicholas Chancellor, Dominic Horsman, and Viv Kendon. Protecting quantum memories using coherent parity check codes. *Quantum Science and Technology*, 2018.
- [59] Christophe Vuillot. Is error detection helpful on ibm 5q chips? *arXiv preprint arXiv:1705.08957*, 2017.
- [60] JR Wootton and D Loss. A repetition code of 15 qubits. arxiv preprint. *arXiv*, 1709, 2017.
- [61] Maika Takita, Andrew W Cross, AD Córcoles, Jerry M Chow, and Jay M Gambetta. Experimental demonstration of fault-tolerant state preparation with superconducting qubits. *Physical review letters*, 119(18):180501, 2017.
- [62] AD Córcoles, Easwar Magesan, Srikanth J Srinivasan, Andrew W Cross, M Steffen, Jay M Gambetta, and Jerry M Chow. Detecting arbitrary quantum errors via stabilizer measurements on a sublattice of the surface code. *arXiv preprint arXiv:1410.6419*, 2014.
- [63] J. J. Sakurai and Jim J. Napolitano. *Modern Quantum Mechanics (2nd Edition)*. Pearson, 2010.
- [64] Thomas Cover and Joy Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition edition, 2006.
- [65] John Preskill. Quantum computing in the nisq era and beyond. *arXiv preprint arXiv:1801.00862*, 2018.
- [66] Lev Bishop, Sergey Bravyi, Andrew Cross, Jay M. Gambetta, and John A. Smolin. Quantum volume. Technical report, 2017.
- [67] John M. Martinis, K. B. Cooper, R. McDermott, Matthias Steffen, Markus Ansmann, K. D. Osborn, K. Cicak, Seongshik Oh, D. P. Pappas, R. W. Simmonds, and Clare C. Yu. Decoherence in josephson qubits from dielectric loss. *Phys. Rev. Lett.*, 95:210503, Nov 2005.
- [68] K. Geerlings, S. Shankar, E. Edwards, L. Frunzio, R. J. Schoelkopf, and M. H. Devoret. Improving the quality factor of microwave compact resonators by optimizing their geometrical parameters. *Applied Physics Letters*, 100(19):192601, 2012.
- [69] Hanhee Paik, D. I. Schuster, Lev S. Bishop, G. Kirchmair, G. Catelani, A. P. Sears, B. R. Johnson, M. J. Reagor, L. Frunzio, L. I. Glazman, S. M. Girvin, M. H. Devoret, and R. J. Schoelkopf. Observation of high coherence in josephson junction qubits measured in a three-dimensional circuit qed architecture. *Phys. Rev. Lett.*, 107:240501, Dec 2011.
- [70] R. Barends, J. Wenner, M. Lenander, Y. Chen, R. C. Bialczak, J. Kelly, P. Lucero, E. andand O’Malley, M. Mariani, D. Sank, H. Wang, T. C. White, Y. Yin, J. Zhao, A. N. Cleland, John M. Martinis, and J. J. A. Baselmans. Minimizing quasiparticle generation from stray infrared light in superconducting quantum circuits. *Applied Physics Letters*, 99(11):113507, 2011.

- [71] Antonio D Córcoles, Jerry M Chow, Jay M Gambetta, Chad Rigetti, James R Rozen, George A Keefe, Mary Beth Rothwell, Mark B Ketchen, and Matthias Steffen. Protecting superconducting qubits from radiation. *Applied Physics Letters*, 99(18):181906, 2011.
- [72] Chad Rigetti, Jay M. Gambetta, Stefano Poletto, B. L. T. Plourde, Jerry M. Chow, A. D. Córcoles, John A. Smolin, Seth T. Merkel, J. R. Rozen, George A. Keefe, Mary B. Rothwell, Mark B. Ketchen, and M. Steffen. Superconducting qubit in a waveguide cavity with a coherence time approaching 0.1 ms. *Phys. Rev. B*, 86:100506, Sep 2012.
- [73] Matthew Reagor, Hanhee Paik, Gianluigi Catelani, Luyan Sun, Christopher Axline, Eric Holland, Ioan M. Pop, Nicholas A. Masluk, Teresa Brecht, Luigi Frunzio, Michel H. Devoret, Leonid Glazman, and Robert J. Schoelkopf. Reaching 10 ms single photon lifetimes for superconducting aluminum cavities. *Applied Physics Letters*, 102(19):192604, 2013.
- [74] Josephine B. Chang, Michael R. Vissers, Antonio D. Córcoles, Martin Sandberg, Jiansong Gao, David W. Abraham, Jerry M. Chow, Jay M. Gambetta, Mary Beth Rothwell, George A. Keefe, Matthias Steffen, and David P. Pappas. Improved superconducting qubit coherence using titanium nitride. *Applied Physics Letters*, 103(1):012602, 2013.
- [75] Jay M Gambetta, AD Córcoles, Seth T Merkel, Blake R Johnson, John A Smolin, Jerry M Chow, Colm A Ryan, Chad Rigetti, S Poletto, Thomas A Ohki, et al. Characterization of addressability by simultaneous randomized benchmarking. *Physical review letters*, 109(24):240504, 2012.
- [76] Ian D Kivlichan, Jarrod McClean, Nathan Wiebe, Craig Gidney, Alán Aspuru-Guzik, Garnet Kin-Lic Chan, and Ryan Babbush. Quantum simulation of electronic structure with linear depth and connectivity. *Physical review letters*, 120(11):110501, 2018.
- [77] Zhang Jiang, Kevin J Sung, Kostyantyn Kechedzhi, Vadim N Smelyanskiy, and Sergio Boixo. Quantum algorithms to simulate many-body physics of correlated fermions. *Physical Review Applied*, 9(4):044036, 2018.
- [78] R Brent Tully, Helene Courtois, Yehuda Hoffman, and Daniel Pomarède. The laniakea supercluster of galaxies. *Nature*, 513(7516):71, 2014.
- [79] Ji-Gang Ren, Ping Xu, Hai-Lin Yong, Liang Zhang, Sheng-Kai Liao, Juan Yin, Wei-Yue Liu, Wen-Qi Cai, Meng Yang, Li Li, et al. Ground-to-satellite quantum teleportation. *Nature*, 549(7670):70, 2017.
- [80] Nicolai Friis, Oliver Marty, Christine Maier, Cornelius Hempel, Milan Holzäpfel, Petar Jurcevic, Martin B Plenio, Marcus Huber, Christian Roos, Rainer Blatt, et al. Observation of entangled states of a fully controlled 20-qubit system. *Physical Review X*, 8(2):021012, 2018.
- [81] Xi-Lin Wang, Yi-Han Luo, He-Liang Huang, Ming-Cheng Chen, Zu-En Su, Chang Liu, Chao Chen, Wei Li, Yu-Qiang Fang, Xiao Jiang, et al. 18-qubit entanglement with six photons three degrees of freedom. *Physical review letters*, 120(26):260502, 2018.
- [82] Chao Song, Kai Xu, Wuxin Liu, Chui-ping Yang, Shi-Biao Zheng, Hui Deng, Qiwei Xie, Keqiang Huang, Qiujiang Guo, Libo Zhang, et al. 10-qubit entanglement and parallel logic operations with a superconducting circuit. *Physical review letters*, 119(18):180511, 2017.
- [83] Subir Sachdev. The landscape of the Hubbard model. In *Proceedings, Theoretical Advanced Study Institute in Elementary Particle Physics (TASI 2010). String Theory and Its Applications: From meV to the Planck Scale: Boulder, Colorado, USA, June 1-25, 2010*, pages 559–620, 2010. [559(2010)].

- [84] M Waitz, R Y. Bello, D Metz, J Lower, Florian Trinter, C Schober, M Keiling, U Lenz, Martin Pitzer, K Mertens, Michael Martins, J Viehhaus, Stephan Klumpp, T Weber, Lothar Schmidt, J B. Williams, M S. Schöffler, Vladislav Serov, A Kheifets, and Reinhard Dörner. Imaging the square of the correlated two-electron wave function of a hydrogen molecule. *Nature Communications*, 8:2266, 12 2017.
- [85] S. A. Jafari. Introduction to Hubbard Model and Exact Diagonalization. *ArXiv e-prints*, July 2008.
- [86] Paul Adrien Maurice Dirac. Quantum mechanics of many-electron systems. *Proc. R. Soc. Lond. A*, 123(792):714–733, 1929.
- [87] R. P. Poplavskii. Thermodynamic models of information processes. *Phys. Usp.*, 18(3):222–241, 1975.
- [88] Norbert Schuch. Condensed matter applications of entanglement theory. *arXiv preprint arXiv:1306.5551*, 2013.
- [89] Román Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 349:117–158, 2014.
- [90] Frank Verstraete, J Ignacio Cirac, and José I Latorre. Quantum circuits for strongly correlated quantum systems. *Physical Review A*, 79(3):032316, 2009.
- [91] P. Jordan and E. Wigner. Über das paulische äquivalenzverbot. *Zeitschrift für Physik*, 47(9):631–651, Sep 1928.
- [92] K. Setia and J. D. Whitfield. Bravyi-Kitaev Superfast simulation of fermions on a quantum computer. *ArXiv e-prints*, December 2017.
- [93] Masuo Suzuki. Generalized trotter’s formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems. *Communications in Mathematical Physics*, 51(2):183–190, Jun 1976.
- [94] Vivek V Shende, Stephen S Bullock, and Igor L Markov. Synthesis of quantum-logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(6):1000–1010, 2006.
- [95] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, 2016.
- [96] Hao Wang. *A Logical Journey: From Gödel to Philosophy (MIT Press)*. A Bradford Book, 2016.
- [97] Nikolaj Moll, Andreas Fuhrer, Peter Staar, and Ivano Tavernelli. Optimizing qubit resources for quantum chemistry simulations in second quantization on a quantum computer. 49, 10 2015.