Carlos Eduardo Teixeira Tavares

Foundations for Quantum algorithms and Complexity

UMinho | 2020

Universidade do Minho
Escola de Engenharia

Carlos Eduardo Teixeira Tavares

Foundations for Quantum algorithms and Complexity

The MAP-i Doctoral Programme in Computer Science, of the Universities of Minho, Aveiro and Porto

universidade de aveiro
theoria poiesis praxis

Universidade do Minho

U.PORTO

July 2021


FCT
Fundação
para a Ciência
e a Tecnologia
POCH
PROGRAMA OPERACIONAL CAPITAL HUMANO
REPÚBLICA PORTUGUESA
CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR
NORTE2020
CENTRO 2020
ALENTEJO 2020
PORTUGAL 2020
UNIÃO EUROPEIA
Fundo Social Europeu

Universidade do Minho
Escola de Engenharia

Carlos Eduardo Teixeira Tavares

Foundations for Quantum algorithms and Complexity

The MAP-i Doctoral Programme in Computer Science, of the Universities of Minho, Aveiro and Porto

Universidade do Minho

Supervisor:
Professor Doutor Luís Manuel Dias Coelho Soares Barbosa

July 2021

COPYRIGHT AND TERMS OF USE FOR THIRD PARTY WORK

ACKNOWLEDGEMENTS

The first time I heard about quantum computation was in the far far away year of 2002, when I was still in high school, during a conversation with my colleagues in the interval of physics class. It came at a time where I was simultaneously entering my adulthood and taking important decisions about what to do with my professional life. I can say that the idea immediately captured my attention, not only due the potential impact of the technology, which was clear from the very first moment, but mostly because it somehow combined two things I truly enjoyed: physics and computation. For practical reasons, I did not pursue the study quantum computation immediately by then, but many years later, after many twists and turns in my career and life, I decided to enrol on a doctoral programme to study the topic and many years after that moment, I am able to put together this work. The path was long, and somehow tortuous at times, but definitely rewarding. There are, of course, many important people who supported me along the way, helping to make this work a reality, to whom I feel very grateful.

First and foremost, my most sincere thanks to my advisor Prof. Luís Soares Barbosa, for the great opportunity of pursuing this interest of mine, for the uncountable opportunities that truly helped me becoming a better researcher, and also for the patience, generosity, time and dedication to this work, for which I will be forever indebted.

I did not have no one assigned, formally, the role of co-advisor, but definitely there were some people with whom I ended up collaborating more closely. My thanks for Prof. Mikhail Vasilevskiy for the great cooperation, and for always being able, and available, to answer all my questions about physics, from whom I learnt a lot. Also, to professor Alexandru Baltag, for the profound insights on quantum logic and for the warm welcome during my stay in Amsterdam and to Professor Jamie Vicary, my external advisor, for his availability and valuable insights about quantum computation and about doing research and being a researcher.

My time of being student of quantum computation also coincided with the birth of a wide interest on the study of the discipline in the university across many departments, and due to this I had the privilege of collaborating with many people from different backgrounds, who definitely influenced me, as well as the ideas that shape this work. A word of thanks to professors José Espírito Santo, Pedro Patrício, José Nuno Oliveira, Manuel Martins and Rui Soares Barbosa for helping shape my understanding of computation, logic, mathematics and physics. Also, my acknowledgements, to prof. Andrei Postnikov and Tomás Veloz, for providing valuable contributions to my research work. Finally, a word of thanks to prof. Nuno Peres, for letting me assist to his classes and introducing me, seriously, to quantum mechanics, I could not wish for a better introduction.

Teaching was one of the greatest experiences of this past few years, and of course I would thank to all my students, with their young and bright minds, for pushing me forward to learn more about quantum computation, so that I could teach them. Some of them ended up having a closer cooperation with me, and here, my special thanks to José Guimarães, Vitor Gonçalves and Marta Sofia Oliveira.

Not only of science is made a doctoral programme, and I would also to thank the people who shared with me the pains of being a PhD student and have been present in many cheerful events that took place throughout these years: from conferences in Brasil, to a certain summer school in a convent near Braga, our hiking trips to the Gerês region, or simply during our lunches at the canteen of University of Minho. My thanks to my colleagues from the ARCA group: Alexandre, Leandro, Renato, José Proença, Guillermina, who became my friends. Also, to my colleagues of 2.17 laboratory, for the relaxed environment in the laboratory and outside of it, in our numerous field trips to settle the most important research question: which Francesinha is best, Porto or Braga. The quest, despite the extensive field work, the long argumentations about the historical importance of the former and the innovative character of the latter, apparently fostered by an excess of eggs and cream in the region of Braga, remains open, i.e. it remains a question of personal taste. Also, my thanks to the innumerous brilliant people I have met during my stay in HASLAB laboratory, Ana, Ali, Paolo Masci, Vitor Enes and many others, for all the nice conversations about multiple subjects.

A word of thanks also to the people from whom I could always count on to answer, with a smile on their face, about the completely non-trivial, but vital, non-research questions that have presented to me throughout the years: Catarina, Paula, Sofia Santos, Goretti and Conceição, and Rita Pacheco.

Outside research there should be a life too, although severely conditioned at certain stage, due to the demands of academic work, but lately, also due to a unfortunate *pandemic* situation. To my friends and colleagues who have supported me in this endeavour throughout the years, sometimes without knowing it, from toy assignments about quantum computation in the early years of college, to cheerful moments around a table, or simply through kind words in darker days, a big word of thanks.

Finally, to my family, specially to my parents, Ana Maria and Jorge, an infinite word of thanks, for the unconditional love and truthfully *rock-solid* support, but mostly, for *everything*.

I would like also to acknowledge the institutions where I carried this work, University of Minho and the High-Assurance software laboratory (HASLAB/INESCTEC), for providing me with the necessary conditions to do so. Partial finantial support is acknowledged to INESC TEC, through the internal grant of reference 3179/BI_B2C/15 within the project INESC TEC-EEA/50014, funded by national funds and FEDER, in scope of Portugal 2020, as well as to the FCT – Fundação para a Ciência e Tecnologia (FCT) by means of the individual grant SFRH/BD/116367/2016, funded under the POCH programme and MCTES national funds. This dissertation was partially supported by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme and

by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project POCI-01-0145-FEDER-030947.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity.

I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

FOUNDATIONS FOR QUANTUM ALGORITHMS AND COMPLEXITY

Recently, quantum computation has been generating a lot of interesting from both industry and academia, due to the first results on quantum supremacy, i.e. the first time quantum computers were able to perform *efficiently* tasks deemed *unfeasible* to classical computers, made possible by the state-of-art qubit technology. These achievements, despite the unusefulness of the tasks performed (quantum circuit sampling), provide evidence that *real world* quantum computation is not only evolving, but also, that full-scale quantum computers may be a reality in the mid-term future.

The benefits of quantum computation are well-known to be potentially ground-breaking, from making *RSA* cryptography unsecure, to the *efficient* simulation of quantum systems. On theoretical side, the algorithm body of knowledge has evolved, and nowadays, there is already a huge number of algorithms and techniques scattered across a vast realm of applications, from solving certain linear equations to optimization.

Nonetheless, the progress in the development of new algorithms with an *exponential advantage*, rather than *polynomial*, has been quite slow and even the application of the existent quantum computational techniques to new problems is far from being a trivial task. The main motivation for this work was to contribute to this problem, and do so by *following* a foundational approach, i.e. by the understanding of the structures of the computational advantage of quantum algorithms and the conception of formal methods to aid in their application to new problems.

Such an approach has to deal with two somewhat *orthogonal* perspectives of quantum algorithms: complexity and semantics. The contribution of this work can also be divided in these two perspectives: in one hand we try to identify and characterize the structures that carry the so-called quantum advantage and exercise them with new applications, and in the other hand we propose tools to deal with the correction of quantum algorithms, particularly, we discuss a dynamic logic for a class of quantum programs: the ones expressible in the quantum assembly programming language (QASM).

Concerning the algorithmic side of the contribution we propose two new applications for quantum algorithms: the first one from quantum biology, concerning the simulation of the non-radiative effects of electronic transport through a molecular chain in a photosynthesis system; the second one belongs to the field of quantum chemistry, and concerns the calculation of the ground state of the Hydrogen and Lithium-Hydride molecules, under the action of a strong electrical field (the stationary Stark effect). Both applications were implemented and experimented in a real world quantum computer, the IBM Q.

keywords: biology, chemistry, dynamic logic, quantum computation

FUNDAMENTOS DE ALGORITMOS QUÂNTICOS E COMPLEXIDADE

Recentemente, o interesse em computação quântica tem vindo a aumentar exponencialmente, devido ao facto da meta da "supremacia quântica", momento em que os computadores quânticos são capazes de realizar tarefas intratáveis para computadores clássicos, ter sido recentemente atingida por equipas independentes, utilizando arquiteturas de qubits quânticos diferentes. Isto dá evidência da saudável velocidade de evolução da área, e fortalece a ideia de que os computadores quânticos em grande escala, podem vir a ser uma realidade no futuro.

Os benefícios, há muito conhecidos, podem ser realmente transformadores em campos como a criptografia, ao tornar a criptografia RSA obsoleta, assim como tornar possível a simulação de muitos sistemas quânticos, com aplicações em muitas áreas da ciência, como na química, ou na física de partículas. O campo teórico dos algoritmos, baseando-se nos sucessos dos primórdios da computação quântica, como foram os algoritmos de Grover e Shor, tem vindo a crescer, sendo que existe um vasto campo de aplicações, desde a resolução de equações lineares, até variados métodos de resolução de problemas de otimização.

Não obstante, o progresso no desenvolvimento de algoritmos quânticos que consigam tirar completo proveito da vantagem quântica tem sido lento, e mesmo a aplicação das técnicas existentes a novos problemas, revela-se complexa. A motivação deste trabalho é contribuir para a mitigação deste problema, através de uma abordagem "fundacional" dos algoritmos e da sua complexidade, ou seja, através da análise e classificação das estruturas que adicionam a vantagem quântica aos programas quânticos e, se possível, da concepção de técnicas formais que permitam ajudar na engenharia de novos algoritmos.

Esta abordagem oferece o desafio de ter de gerir duas dimensões dos algoritmos quânticos, tradicionalmente estudadas em separado: complexidade e semântica. Assim, a abordagem deste trabalho baseou-se também nessas duas dimensões: por um lado, no estudo das estruturas que permitem a vantagem nos algoritmos quânticos, e por outro na concepção de uma lógica dinâmica para uma classe específica de programas quânticos: aqueles que são exprimíveis na linguagem QASM.

Relativamente à parte mais algorítmica da contribuição, são propostas duas novas aplicações em dois campos ciêntificos diferentes: a simulação do transporte electrónico sem recurso a radiação no campo da biologia quântica; no campo da química quântica, o cálculo do *estado fundamental* da moléculas H_2 e LiH, sob acção de um campo elétrico *forte* (efeito de Stark). Ambas as aplicações foram implementadas e experimentadas num computador quântico real, o IBM Q.

palavras-chave: biologia; computação quântica; química; lógica dinâmica

CONTENTS

1

INTRODUCTION

Quantum mechanics emerged in the beginning of the twentieth century, as a theoretical solution to theoretical inconsistencies faced by classical physics at atomic scales. Later, it was extended to all physical regimes, except to gravity, explaining physical phenomena at small scales with great accuracy. Nowadays, it is considered the most successful theory in the history of physics.

Fifty years after its conception, with its unique properties still being understood, the first quantum-based technologies on the fields of communication and information started to appear, giving rise to the new area of quantum information. In the Eighties, it was introduced the first notion of quantum computing by Feynman [150] and several others, which has quickly extended to actual computer models and new algorithms, such as the Deutsch-Jozsa [133], as well as Grover [178] and Shor algorithms [328], all of them with the potential of significantly boost the performance of many very hard classical computational tasks. These include, for instance, the breaking of Rivest, Shamir, Adelmant (RSA) cryptographic system [75, 328] (one of the cornerstones of industrial cryptograhic systems), exponential increase in the simulation of quantum mechanical systems [248], or the resolution of some types of linear equations [191]. Nowadays, the number of quantum algorithms and techniques has grown significantly, with many potential applications across industry and science [167], with an expected huge impact, for instance in chemistry [89].

Quantum computation faces, however, many very hard technical challenges, which has been preventing the reaches of its full potential. The hardest of these challenges is the so-called *decoherence*, i.e. the hardness of conserving superposition and entangled quantum states, from the errors caused by environmental interaction, which so far, remain uncontrollable, despite the many physical architectures used, error correction strategies and even alternative computer models [161, 147, 83]. In fact, in terms of hardware, quantum computing is still at an early stage of development, and *impactful* implementations of most relevant quantum algorithms are still beyond reach. However, advancements have been made, mostly due to the developments in superconducting qubit technology [227, 235], and of optical systems [384]. It is expected that in the next few years quantum computers start performing tasks deemed hard to classical computers, reaching the so-called *quantum supremacy* [296].

By the time of the development of this work, an important milestone was reached in this regard, by Google claiming a quantum advantage in a circuit sampling task [73, 30], using a 53 *super-conducting*

qubit processor a claim disputed by IBM [284]. More recently a Chinese team, has also claimed *quantum advantage* using a photonic quantum computer, in *boson sampling* [384]. Furthermore, recently another important milestone was attained in the field of simulation of quantum chemistry [299].

Current technology, known as the *Noisy Intermediate-Scale Quantum* (NISQ) [297] era, yields qubits with a controllable amount of error and it is already expected that high-impactant computational tasks will be performed by quantum computers, in the field of optimization [90, 87], or even in very relevant problems such as *nitrogen* or *carbon* fixation, using techniques such as *quantum annealing* or the *variational quantum eigensolver* method.

In conclusion, advancements in the theory of quantum algorithms, have led to large number of quantum algorithms, techniques, and computer models. However, the creation of new algorithms, or even, their adaptation to new contexts, has shown to be non-trivial. The motivation for this work, is exactly to contribute to the engineering of new quantum algorithm, following a foundational approach. The word "foundations" may have several meanings depending on the context, dealing, in general, with trying to find the most fundamental components of the object of study. In the context of mathematics and computer science, the term usually refers to the underlying theory or semantic model, where the object, or objects, of study can be captured in a *sound and complete way*. Following such an approach for the construction of new quantum algorithms is specially challenging, as it has to deal with two, generally orthogonal, dimensions: their performance (complexity wise) and their correctness.

In quantum computation, the *correctness side* is already significantly mature, and many useful tools to reason about quantum processes and programs are available, founded on the consistent work in the fields of semantics of quantum programming languages and logical systems. The *complexity* side is not so mature: while the fundamental components that allow for *quantum advantage* across all quantum computer models, techniques and algorithms are common and identified, *interference* and *entanglement*, and despite the multitude of methods to identify and quantify them, to the best our knowledge there is no *compositional* theory that formalizes them. Nonetheless, the mathematical theory of quantum advantage is a very active research area. Hence, in this thesis we follow a more empirical approach: performance-wise we focused in the characterization of some the efficient quantum algorithms and their primitives, and the conceive new examples; from the point of view of correctness, we provide a logical formalism for dealing with a specific class of quantum programs. More precisely the research work was developed along these lines:

- **Efficient quantum algorithms -** Two classes of problems for which there is a quantum algorithm with exponential advantage are the *soft-simulation* of a wide class of Hamiltonians (local to sparse ones), and the ones that make effective use of the Fourier transform, which range from the Shor to the HHL algorithm. We studied these algorithms and provided a new application to the *soft-simulation* of nonradiative energy in photosynthetic systems;

- **Optimization problems -** Quantum computers may yield some performance advantage in industrial problems, particularly when involving some form of optimization. We review the current quantum computational techniques employed in this kind of problems and, in particular, we explore an example of application of one of these techniques, to the calculation of the ground-state of the H_2 and LiH, under the action of a *strong* electrical field, the so-called *Stark effect*, which, to the best of our knowledge, was not attempted before and whose conception faces non-trivial theoretical and practical challenges;

- **Logic for quantum programs -** We define a dynamic logic, to reason about the QASM programming language, which involves quantum and classical data and measurements.

These lines of research have originated the following publications:

- José Diogo Guimarães, Carlos Tavares, Luís Soares Barbosa, and Mikhail Igorevich Vasilevskiy (2020). **Simulation of nonradiative energy transfer in photosynthetic systems using a quantum computer.** Complexity, 2020 ([179]).

- Carlos Tavares, Sofia Oliveira, Vitor Fernandes, Andrei Posnikov and Mikhail Igorevich Vasilevskiy. **Quantum simulation of the ground-state Stark effect in small molecules: a case study using IBM Q.** Soft Computing 25, 6807–6830 (2021). ([343]).

- Carlos Tavares. **A Dynamic Logic for QASM Programs.** International Workshop on Dynamic Logic. Lecture Notes in Computer Science, vol 12005, 209-217 (2019). Springer, Cham. ([342]).

The structure of the thesis reads as follows: in chapter 2 we revisit the relevant mathematical notions underlying quantum mechanics and provide some background on quantum computation and programming. Chapter 3 aims at classifying the quantum efficient algorithms, as well as, the inherent primitives that allow quantum advantage. We also discuss the example of the simulation of non-radiative energy transfer in photosynthetic systems. Chapter 4, is devoted to analyse complex computational problems, which remain beyond reach to quantum computers, whereas most of them have industrial interest. We also discuss a case study on the calculation of a ground-state Stark effect for small molecules. Finally, chapter 5, introduces a specific quantum dynamic logic to reason about quantum programs.

2

A BRIEF JOURNEY ON QUANTUM COMPUTATION

> I think I can safely say that nobody understands quantum mechanics.
>
> Richard P. Feynman, The Messenger Lectures, 1964, MIT.

This chapter aims at revisiting the fundamental theoretical notions of quantum computation, as a background of the following chapters. We explore the fundamental mathematical concepts of quantum mechanics, the Hilbert space and density operators formalisms, as well as the *distinctive* characteristics of quantum mechanics, i.e. *interference* and *entanglement*, that, arguably, are the main source of the so-called quantum advantage, which play a vital role in many quantum technologies, from communication and cryptography, to computation. We also explore the main quantum computation models and programming languages, including quantum circuits, and we provide a general perspective of the quantum complexity classes.

2.1 Quantum physics

At the beginning of the 20^{th} century, physics was thought to be complete, unifying all known physical forces and regimes, from the small to the very large, under the general relativity theory. However, there were still some issues to solve, such as the divergences verified in the calculation of *black-body radiation*, and the incomplete understanding of the photoelectric effect. The solutions to these problems, led to the introduction of the of the concept of quantum, the smallest amount of a physical quantity, for example, the Planck constant[1], which is the quantum of the orbital angular momentum. It allowed simultaneously the resolution of the divergences of the radiation of black bodies [289] as well as the full understanding of the *photoelectric effect* [140], for which Albert Einstein was awarded the Nobel prize in 1921. This in particular, included the radical idea of discretization of electromagnetic waves, whose *quanta* were later called *photons*.

1 A constant $h = 6.62607004 \times 10^{-34} m^2 kg/s$ introduced by Max Planck, to fit the data of black body radiation experiments

These findings along with several other ideas and experiments, such as the experimental validation of the *wave-particle duality*[2] and the new atomic models proposed by Bohr [71], shaped and gave experimental validation to a radically different theory from classical mechanics: the quantum theory.

This body of ideas required a sound mathematical formulation, and this was initially achieved by two different and *independent* formulations: the Erwing Schrödinger's formulation [318], based on *waves*, and the Heisenberg-Jordan formulation [196], based on *matrices*. Both formulations were, later, shown to be equivalent by Schrödinger [317], and later, mathematically unified under the Hilbert space formalism by Von Neuman [275], as both are canonical examples of a complex Hilbert space : Schrodinger's formulation happens in the space l_2, the space of all square summable complex sequences, and the Heisenberg's in the $L_2(\mathbb{R}^3)$ space, the set of all square integrable complex functions of three real variables [15].

Another important formalism is the one of density matrices, also introduced by Von Neumann in 1927 [360], targeting at the representation of *statistical ensembles* of quantum states, useful to represent uncertainty in quantum states, e.g. fuzziness in preparation of quantum states. The Hilbert and density matrices formalisms, particularly the finite-dimensional ones, are the most commonly used ones in quantum computation. In the following sections we enumerate the main principles of quantum mechanics

2.2 The Hilbert space formulation

Despite of the complexity of its *consequences*, quantum theory is defined by only four postulates, in addition to the algebraic laws governing a Hilbert space, and can be interpreted as state-based systems, where states are unity vectors and transitions correspond to unitary transformations between states, preserving *energy conservation laws*.

Definition 2.2.1. A vector space V over a field $\mathbb{F}$, is a set such that for every $x, y \in V$, the following laws apply:

$$\begin{array}{cc} x+y \in V; & 0_{\mathbb{F}} x = 0_V; \\ x+y = y+x; & 1_{\mathbb{F}} x = x; \\ 0_V + v = x + 0_V = x; & \alpha(x+y) = \alpha x + \alpha y; \\ \alpha x \in V; & (\alpha+\beta)x = \alpha x + \beta x; \end{array}$$

Definition 2.2.2. ([108]) A Hilbert space is a vector space H, over the complex numbers $\mathbb{C}$, with an internal product of type:

$$\langle - | - \rangle : H \times H \to \mathbb{C}\,. \tag{1}$$

In this operation, for all $\psi, \phi, \epsilon \in H$ and $\lambda \in \mathbb{C}$ the following algebraic rules apply:

2 The idea that both waves and particles had similar behaviour, as proposed theoretically by Louie De Broglie [126], and experimentally demonstrated by Compton [112]

$$\langle\psi|\lambda\phi+\epsilon\rangle=\lambda\langle\psi|\phi\rangle+\langle\psi|\epsilon\rangle;\quad \langle\lambda\psi+\phi|\epsilon\rangle=\lambda^*\langle\psi|\epsilon\rangle+\langle\phi|\epsilon\rangle;$$
$$\langle\psi|\phi\rangle=(\langle\phi|\psi\rangle)^*;\qquad \langle\psi|\psi\rangle\geq 0 \text{ for all } \psi\neq 0$$

where $(-)^*$ is the conjugate of complex numbers. Additionally, Hilbert spaces of infinite dimensions must be Cauchy complete, i.e. for any (Cauchy) convergent sequence $(v_i)_i$ of vectors in H, there exists $v\in H$, such that $|v_i-v|\to 0$.

2.2.1 *The state space*

The first *postulate*, defines the set of possible states in a quantum system:

Postulate 1. The state space of an isolated physical system is the set of unitary vectors of an Hilbert space.

We denote an Hilbert space by $\mathcal{H}$. In 1939, Paul Dirac proposed a notation to express vectors in the Hilbert space, the so-called Dirac notation [136]. Following this notation, a vector $v\in\mathcal{H}$, i.e. a state, is expressed as $|\Psi\rangle$. Every vector is generated by a basis, a subset of *linearly independent vectors* $|v_1\rangle,\ldots,|v_n\rangle$. Hence, in its general form, a vector $v\in V$, can be expressed as a linear combination of elements of a basis,

$$|\Psi\rangle=\sum_i \alpha_i|v_i\rangle\ , \tag{2}$$

where $|v_i\rangle$ correspond to vectors of the basis, and α_i are *complex numbers* corresponding to their *amplitudes*. In the matricial form $|v_i\rangle$ corresponds to a *column* vector:

$$|\Psi\rangle=\begin{bmatrix}\alpha_1\\ \alpha_2\\ \vdots\\ \alpha_3\end{bmatrix}. \tag{3}$$

A *basis* may be or not be complete, it is complete if the following condition holds,

$$\sum_i \alpha_i|v_i\rangle=0 \text{ if } \alpha_i=0, \text{ for all i}\ . \tag{4}$$

The dimension of $\mathcal{H}$ is the maximum number of linearly independent vectors in $\mathcal{H}$. A further condition of valid states is *unitarity*, which is provided by the normalization, the norm of the vector is equal to 1, i.e. $||\,|v\rangle\,||=1$, where the *norm* is given as:

$$||\,|\Psi\rangle\,||=\sqrt{\sum_i \alpha_i^\dagger\alpha_i} \tag{5}$$

where α_i and α_i^* are the amplitudes and their conjugates, respectively. Another relevant mathematical object in quantum mechanics is the *bra* $\langle\Psi|$, which corresponds to the *conjugate transpose* of a vector

$$\langle\Psi| = |\Psi^*\rangle^T = [\alpha_1^*, \alpha_2^*, \dots, \alpha_n^*]\,. \tag{6}$$

Conceptually, *kets* correspond to the states of a quantum system, and *bras* to the tests that can be performed over states. Finally, the internal product provides notion of *distance* between vectors. In the Dirac notation, for instance, an internal product is denoted as $\langle\Psi_1|\Psi_2\rangle$ and can be calculated as follows:

$$\langle\Psi_1|\Psi_2\rangle = [\alpha_1^*, \alpha_2^*, \dots, \alpha_n^*].\begin{bmatrix}\alpha_1\\ \alpha_2\\ \vdots\\ \alpha_3\end{bmatrix}. \tag{7}$$

In other words, the internal product provides a measure of the likelihood that a *test* $\langle\Psi_1|$ results *successfully* in a state $|\Psi_2\rangle$. If the internal product $\langle\Psi_1|\Psi_2\rangle = 0$, then $\langle\Psi_1|$ and $|\Psi_2\rangle$ are independent, i.e. *orthogonal*, while if the internal product is one, means they coincide.

It can be easily observed that two kinds of states are possible in an *Hilbert space*: $|\Psi\rangle = |v_i\rangle$ corresponding to a *single element* of the basis, and $|\Psi\rangle = \sum_i \alpha_i |v_i\rangle$ with $\alpha_i \in \mathbb{C}$, corresponding to linear combinations of *basis* elements, both having fundamental differences in terms of physical interpretation. The former ones are called stationary states and correspond to states where the energy of the system is well known, i.e. with no uncertainty associated, corresponding to different *quantum numbers*, and being perfectly distinguishable. The latter ones correspond to the so-called *superposition states*,

$$|\Psi\rangle = \lambda_1|\Psi_1\rangle + \lambda_2|\Psi_2\rangle + \dots + \lambda_m|\Psi_m\rangle\,, \tag{8}$$

i.e. states where the energy is not well-defined, with an *uncertainty* associated. The proper physical understanding of such states is still under debate, due to their *unobservability* in the classical world. However in practice, according to the statistical interpretation of Quantum Mechanics originally proposed by M. Born [314], a *measurement* of such a quantum state can *randomly* yield one of the eigenvalues of its energy, E_n, with the probabilities given by the squared amplitudes of the corresponding basis elements, $|\lambda_n|^2$.

2.2.2 *Interference*

A closer look to the possible state spaces in quantum mechanics, leads to the observation of one of its *distinctive properties*: interference. In general, a quantum state, can be written as

$$Ae^{i\theta_1}|0\rangle + Be^{i\theta_2}|1\rangle \equiv \lambda|0\rangle + \beta|1\rangle \text{ with } |\lambda|^2 + |\beta|^2 = 1\,. \tag{9}$$

Interference concerns the non-independence of certain pairs of events in quantum mechanics, which can be verified, for instance, in the so-called *double slit experiment* [19]. This prevents that the probability of such events is given by the sum of their probability, i.e. $P(A \vee B) = P(A) + P(B)$, as if they were independent. *Interference* requires two or more events to be calculated: considering the state described in equation (9), and that $A \sqcup B$ corresponds to the *quantum union* of the two possible quantum events, i.e. to obtain 0 or 1 upon measurement, the probability of $A \sqcup B$ is calculated as follows:

$$P(A \sqcup B) = (A.e^{i\theta_1} + B.e^{i\theta_2})^{\dagger}(A.e^{i\theta_1} + B.e^{i\theta_2}) = |A|^2 + |B|^2 + A.B.cos(\theta_2 - \theta_1)\,. \qquad (10)$$

It is easily observable that the probability depends on the phase difference between the two *waves* (eigenfunctions) of the stationary states, i.e. by the term $cos(\theta_2 - \theta_1)$. Formally, the actual calculation of interference is given by the difference between the probability of quantum union and the classical union:

$$\begin{aligned} I(A,B) &:= P(A \sqcup B) - P(A) - P(B) \\ &:= \left(A.e^{i\theta_1} + B.e^{i\theta_2}\right)\left(A.e^{i\theta_1} + B.e^{i\theta_2}\right)^{\dagger} - A.e^{i\theta_1}.\left(A.e^{\theta_1}\right)^{\dagger} - B.e^{i\theta_2}.\left(B.e^{\theta_2}\right)^{\dagger} \\ &:= A.B.cos(\theta_2 - \theta_1) \end{aligned}$$

It can be easily concluded, that the events can only be treated as independent only when interference $I(A,B)$ is 0. On the other hand, when the *interference factor* is not 0, the two waves corresponding to the events are said to be *coherent*.

There is always interference in superposition states, i.e. *eigenwaves* of the system are always in coherence, and the probability calculations the only depend on the phase differences (local phases) between them. The common denominator between all phases is the so-called *global phase*, and while essential to the composition of quantum systems, does not have any observational relevance, i.e. it is not relevant to probabilistic calculations. In other words, it can be stated that

$$\lambda\,|\Psi\rangle \cong |\Psi\rangle\,, \text{ with } \lambda \in \mathbb{C} \qquad (11)$$

where λ is a *global* phase factor, i.e. it affects the whole system, and $\cong$ means *observational equivalence* and $|\Psi\rangle$ is an arbitrary state.

Interference is well known to happen with waves in classical physics, but in quantum mechanics it is known to happen with *photons*, but also with particles, e.g. electrons, i.e. e providing evidence that also matter can behave as waves, in agreement with the hypothesis *wave-particle duality*. In fact, interference has also shown to be existent in much larger bodies at mesoscopic scale, such as in large molecules [168], but not at macroscopic level.

This was shown in quantum mechanics resorting to different devices and methods, such as *slit experiments*, observation of electron diffraction or *interferometers* of many kinds. A paradigmatic example of interferometers is the so-called *Mach-Zender* interferometer, depicted in figure 1, initially targeted to photons.

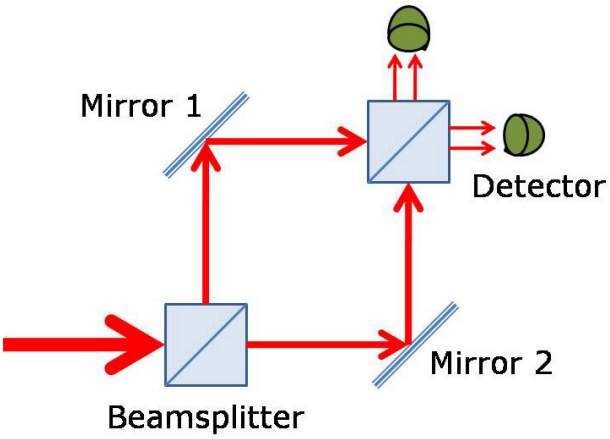


Figure 1: The Mach-Zender interferometer, is a optical device where, upon the application of a beam splitter, photons can travel through two possible paths, until reaching one of two detectors. Such paths can be transversed with equal probability. The detection of the photon in the first detector corresponds to the basis state $|0\rangle$ and in the second detector as $|1\rangle$. Adapted from [132].

In the experiments with the *Mach-Zender* interferometer, the end states read as follows:

$$\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) \tag{12}$$

with $1/2$ probability of obtaining 1 or 0. However, the probability can be controlled by the phase difference, which establishes the difference between a *purely random* distribution, where there is no control over the distribution and it is constant throughout time, and the distributions of quantum mechanics, by-product of coherence between waves, controllable, and oscillatory throughout the action of Hamiltonian. The loss of coherence, which can happen through many processes, causes the decay of a quantum superposition state exhibiting coherence to a classical probabilistic state.

2.2.3 *Unitary evolution*

The evolution of quantum mechanical systems has to be *reversible*, due to the laws of energy conservation, and preserve linearity. Therefore, the evolution in a quantum system is given by a unitary operator U,

$$|\Psi_{t+\delta t}\rangle = U\,|\Psi_t\rangle\ , \tag{13}$$

which preserves linearity:

$$U\left(\sum_i \alpha\,|v_i\rangle\right) = \sum_i \alpha_i U\left(|v_i\rangle\right)\ . \tag{14}$$

Postulate 2. The evolution of a closed system is described by a unitary transformation i.e. an operator U such that $U.U^{-1} = I$.

A corollary of these postulates is that the temporal evolution of a quantum mechanical system is given by the *Schrödinger equation* [317], which is the *cornerstone* of Schrödinger's formulation of quantum mechanics, still very useful in many contexts.

Postulate 2.1. The time evolution of the state of a closed quantum system is described by the Schrödinger equation,

$$i\hbar\frac{\partial}{\partial t}|\Psi(t)\rangle = H|\Psi(t)\rangle\,. \tag{15}$$

The Schrödinger's formulation of quantum mechanics, is centred at the action of the Hamiltonian operator over quantum systems, $H = T + V$, T being the kinetic energy of the constituent particles and V the potential energy of all interactions and fields in the system, both internal and external, whose action yields the total energy of a system if in a stationary state,

$$H|\Psi\rangle = E|\Psi\rangle\ . \tag{16}$$

Notation $|\Psi(t)\rangle$ denotes the system's wavefunction (WF), which represents the timed evolution of an *eigenstate* $|\Psi\rangle$, resulting from the action of an Hamiltonian operator. The set of WF's correspond to all physically meaningful solutions of the Schrödinger equation, which are mutually orthogonal. Usually, there are several possible solutions to the equation, corresponding to different values of the energy (energy levels or eigenvalues, E_n), which are discrete for a confined (or bound) physical system. The wavefunction $|\Psi\rangle$, may also depend on other arguments (such as spatial coordinates and spin components) according to the *representation* used. In section 4.3, on chapter 4, this formalism is further explored, when addressing a problem of simulation of a many-body system for quantum chemistry.

2.2.4 *Observables*

A quantum state, i.e. a *wavefunction*, carries information about all the observations that can be made, by characterizing their probability distribution. In quantum mechanics one may be interested in an infinity of types of observations, e.g. one can be interested in measuring the position of a particle, or its velocity. Observables in quantum mechanics, are operators with a special status, known as *Hermitian* linear operators which *conjugate transpose* (A^*) equals itself,

$$A^* = A\,. \tag{17}$$

It can be easily verified that such operators possess *real* eigenvalues, and indeed correspond to those who are *empirically* meaningful, as in the classical world all physical quantities are *real* values. A difference between *classical* and *quantum* mechanics is that observables, in general, **do not commute**, i.e. the commutator given by

$$[A, B] = AB - BA\,, \tag{18}$$

is not necessarily 0 for all pairs of observables. The most well-known example of this is the *non-commutativity* of *position* and *momentum* operators, which constitute the so-called *Heisenberg uncertainty principle*. The commutator, basically allows the evaluation of the effect of the order of application of the observations, i.e. if the commutator is different from zero, i.e.

$$[A, B] \neq 0\,, \tag{19}$$

the observations interfere with each other and it is impossible to estimate with *arbitrary accuracy* both observables at the same time, meaning that observables are correlated, and even more, that it is impossible to find separable distributions for the statistics of both observables and assign definite values to both of them [72]. On the other hand, if (19) is 0, then observables are independent, and hence, it is possible to estimate both observables with arbitrary accuracy. Einstein defined this impossibility of assigning a definite value to every observable of a quantum system in a given instant as some sort of *anti-realism*, stating that quantum mechanics should be incomplete. This, however, was proven experimentally several times, and, is an important ingredient of several quantum technologies. Niels Bohr stated that this is a very powerful assumption, which invalidates the calculation of joint probabilities for events that do not commute.

2.2.5 *Measurements*

The postulates enumerated so far, describe the possible states connected by transitions in quantum mechanics, which include a special kind of states, the *superposition states*. Such states, cannot be observed in the classical world, i.e. objects at *macroscale* can only be observed in a stationary and not in a superposition of states, although they have been observed at micro and mesoscopic scales. This disparity between classical and quantum worlds is known as the so-called *measurement problem* [374], for which multiple possible theories have been developed, unfortunately, undistinguishable from the experimental point of view with current technology. However, the process of causing the collapse of a superposition state

to a classical state is known as *measurement* and measurements have a stochastic nature in quantum mechanics.

Postulate 3. Measurements cause the collapse of quantum states into classical states. Mathematically, they correspond to projection operators: $|M_m\rangle\langle M_m|$, where m is the desired outcome.

The calculation of the resulting state after a measurement over $|\Psi\rangle$ is given by

$$\frac{M_m|\Psi\rangle}{\sqrt{\langle\Psi|M_m^\dagger M_m|\Psi\rangle}}, \tag{20}$$

and the probability of obtaining a specific measurement by

$$p(m) = \sqrt{\langle\Psi|M_m^\dagger M_m|\Psi\rangle}. \tag{21}$$

The set of measurement operators, for all possible outcomes, respects the condition that $\sum_m M_m^\dagger M_m = I$. Each measurement operator corresponds to only one possible outcome.

Example 2.2.1. Consider state $|\psi\rangle = a|0\rangle + b|1\rangle$ and measurements $M_0 = |0\rangle\langle 0|$, $M_1 = |1\rangle\langle 1|$; then,

$$p(0) = \langle\psi|M_0^\dagger M_0|\psi\rangle = \langle\psi|M_0|\psi\rangle = |a|^2 \tag{22}$$

$$\frac{M_0|\psi\rangle}{|a|} = \frac{a}{|a|}|0\rangle \tag{23}$$

$$\frac{M_1|\psi\rangle}{|b|} = \frac{b}{|b|}|1\rangle \tag{24}$$

2.2.6 *Combining quantum systems*

In quantum mechanics the possibility of combining *smaller* quantum systems can be combined to obtain larger ones, and a profound difference between classical and quantum mechanics, is that such combination is not *trivial*: new possible states arise from the combination of two Hilbert spaces, e.g. $H_1 \otimes H_2$, which did not exist in the original vector spaces, H_1 and H_2, before being combined (more on this on section 2.3.3). The last postulate of quantum mechanics corresponds exactly to this:

Postulate 4. The state space of a composite physical system corresponds to the tensor product of the state spaces of the component physical systems. A tensor is operator

$$\otimes : \mathbb{C}_1^n \times \mathbb{C}_2^n \rightarrow \mathbb{C}^{n_1+n_2} . \tag{25}$$

In fact every Hilbert space of arbitrary dimension can be written as a tensor product

$$H^n = H_1 \otimes H_2 \otimes H_3 ... \otimes H_n , \tag{26}$$

and the calculation of a tensor product of two vector spaces, corresponds to the following linear operation:

$$T = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(v_i w_j \right) (e_i \otimes f_j) \tag{27}$$

$$\begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} \otimes \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} = \begin{bmatrix} a_{1,1} \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} & a_{1,2} \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} \\ a_{2,1} \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} & a_{2,2} \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} \end{bmatrix} \tag{28}$$

$$= \begin{bmatrix} a_{1,1} * b_{1,1} & a_{1,1} * b_{1,2} & a_{1,2} * b_{1,1} & a_{1,2} * b_{1,2} \\ a_{1,1} * b_{2,1} & a_{1,1} * b_{2,2} & a_{1,2} * b_{2,1} & a_{1,2} * b_{2,2} \\ a_{2,1} * b_{1,1} & a_{2,1} * b_{1,2} & a_{2,2} * b_{1,1} & a_{1,2} * b_{1,2} \\ a_{2,1} * b_{2,1} & a_{2,1} * b_{2,2} & a_{2,2} * b_{2,1} & a_{1,2} * b_{2,2} \end{bmatrix} \tag{29}$$

2.3 The density matrix formalism

Density matrices were introduced by Von Neumann in 1927 to represent *mixtures* of physical systems [359], i.e. stochastic combinations of quantum systems. This representation is useful, particularly when there is *uncertainty* over the preparation of quantum states: there is a classical probability of a system being in a quantum state or the other, leading to the so-called **mixed states**. The latter is a common phenomenon in quantum mechanics. Following equation (6), a quantum state can be written as:

$$|\Psi\rangle = \sum_i \alpha_i |\Psi_i\rangle \tag{30}$$

The density operator can be obtained by the projection of a state over itself, which reads as

$$\rho = |\Psi\rangle \langle\Psi| , \tag{31}$$

which results in a matrix of the following type:

$$\rho = \begin{pmatrix} \alpha_1\alpha_1^\dagger & \alpha_2\alpha_1^\dagger & \dots & \alpha_n\alpha_1^\dagger \\ \alpha_1\alpha_2^\dagger & \alpha_2\alpha_2^\dagger & \dots & \alpha_n\alpha_2^\dagger \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1\alpha_n^\dagger & \dots & \dots & \alpha_n\alpha_n^\dagger \end{pmatrix}. \tag{32}$$

In this matrix lines correspond to kets, $|\Psi_i\rangle$, and columns to bras, $\langle\Psi_j|$, and each position of the matrix contains the correspondent eigenvalue of the projection $|\Psi_i\rangle\langle\Psi_j|$, the multiplication of the α_i for $|\Psi_i\rangle$ for its conjugate $\alpha_i^\dagger$. Fue to the spectral decomposition theorems, the matrix operator can be written as

$$\sum_i \left(\alpha_i^\dagger * \alpha_i\right) |\Psi_i\rangle\langle\Psi_i| \ . \tag{33}$$

The elements appearing on the spectral decomposition, $\left(\alpha_i^\dagger * \alpha_i\right)$, are elements of the diagonal of the matrix, and correspond to the probabilities of obtaining state $|\Psi_i\rangle\langle\Psi_i|$ upon measurement, hence the trace[3] of the matrix equals to 1, and elements are always positive. Thus,

1. **(Trace condition)** ρ has trace equal to one;
2. **(Positivity condition)** ρ is a positive operator.

2.3.1 *Pure and mixed states*

States in quantum mechanics are of two types: *pure*, if it is possible to express them as a single *ket*, or *mixed*, when expressed as a *stochastic combination* of quantum states, i.e. the so-called *mixture* of quantum states,

$$\rho = \sum_i p_i |\Psi_i\rangle\langle\Psi_i| \ , \tag{34}$$

where p_i is the probability of obtaining Ψ_i in a measurement. There is a multitude of situations where this kind of representation may be useful, the most obvious case, being the existence of a dependency between the observables of two different quantum systems, which imply that, from the perspective of one of the systems, there is *uncertainty* about the actual state of the system, i.e. none of the two systems alone possesses complete information about the state.

In practice the main difference between *mixed states* and *pure states* is that the latter states present *coherent* effects and the formers do not, being purely *random*. A very simple example is given by the pure state

3 The trace of a matrix can be calculated by summing all the elements in the diagonal $T_r[\rho] = \sum_i a_{ii}$.

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

which reads, in the *density operator* formalism, as follows:

$$\begin{pmatrix} \alpha * \alpha^{\dagger} & \beta * \alpha^{\dagger} \\ \alpha * \beta^{\dagger} & \beta * \beta^{\dagger} \end{pmatrix} \Leftrightarrow \begin{pmatrix} |\alpha|^2 & \beta * \alpha^{\dagger} \\ \alpha * \beta^{\dagger} & |\beta|^2 \end{pmatrix}. \tag{35}$$

The following density operator acting on the same basis, corresponds to a *mixed state*:

$$\begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}, \tag{36}$$

where $A, B \in \mathbb{R}$ and $A = |\alpha|^2$, $B = |\beta|^2$. While both states have $1/2$ of probability of being in states 0 and 1, they have a very different physical meaning. The former corresponds to a *superposition* state and possesses interference; the latter corresponds to a plain classical distribution of two *stationary* states and does not possess any interference (non-diagonal elements are 0). In the latter is impossible to *control* the probabilities of obtaining $0, 1$ by the use of *phase-shifts*, for example. The distinguishability between *pure* and *mixed* states, can be made explicit by the application of the trace operator $T_r(\rho^2)$, where operator ρ^2 is given by

$$\rho^2 = \rho * \rho = \sum_i p_i\,|\psi_i\rangle\langle\psi_i| \sum_j p_j\,|\psi_j\rangle\langle\psi_j| = \sum_i p_i p_j\,|\psi_i\rangle\langle\psi|\psi_j\rangle\langle\psi_j| = \sum_i p_i^2\,|\psi_i\rangle\langle\psi_i| \tag{37}$$

and the *trace* corresponds to the sum of the diagonal elements of ρ^2

$$tr(\rho^2) = \sum_i {p_i}^2 tr(|\Psi_i\rangle\langle\Psi_i|) = \sum_i p_i^2\,. \tag{38}$$

In the case of a *pure state*, the *partial trace* will be equal to 1, while for *mixed states* such trace will be < 1. For instance, if $A = B = 1/2$

$$\begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}^2 = \begin{pmatrix} 1/4 & 0 \\ 0 & 1/4 \end{pmatrix}, \tag{39}$$

the trace of ρ^2 corresponds to $1/2$ and it is lesser than 1, corresponding to a *mixed state*. This technique may also be useful, for instance, in the detection of *non-separable* states (more on this on section 2.3.3).

Moreover, every *mixed state* can be transformed onto a *pure entangled* state of higher dimension, a process, known as *purification*, for which there are an *infinitude* different processes to do it. Simultaneously, it is always possible to produce mixed states, out of *pure states* [354].

2.3.2 *Evolution*

The evolution in the density operator formalism, similarly to what happens in the Hilbert space formulation, is described by both unitary evolution and measurements. However, density operators have to support both *pure* states and *mixed* states. Hence, the evolution in the formalism is not given by unitary and projection operators, but rather by *completely positive maps (CPM)*, which are:

- linear operator on positive matrices, preserving positivity;
- *trace preserving*, or trace non-increasing;
- stable under the addition of qubits, i.e. for F, I CPMs, $F \otimes I$ is still a CPM.

Unitary evolution

The unitary evolution is given by action of the unitary operator U over each of the *kets* of the mixed state:

$$\rho = \sum_i p_i \left|\psi_i\right\rangle \left\langle\psi_i\right| \xrightarrow{U} \sum_i p_i U \left|\psi_i\right\rangle \left\langle\psi_i\right| U^\dagger = U\rho U^\dagger . \tag{40}$$

Measurements

The simplest type of measurements are projection measurements, which correspond exactly to projection operators. Projections have the following characteristics:

- idempotence $\left|i\right\rangle \left\langle i\right| = \left(\left|i\right\rangle \left\langle i\right|\right)^2$;
- orthogonality and $\sum_i \left|i\right\rangle \left\langle i\right| = I$.

On the other hand, the action of a measurement, over a state, reads as,

$$\sum_i \left|\phi_i\right\rangle \left\langle\phi_i|\psi\right\rangle \left\langle\psi|\phi_i\right\rangle \left\langle\phi_i\right| = \sum_i \left|\left\langle\phi_i|\psi\right\rangle\right|^2 \left|\phi_i\right\rangle \left\langle\phi_i\right| , \tag{41}$$

and the probability of the outcome j, using a measurement M over a state $|\Psi\rangle$, is calculated as follows:

$$P_\Psi(M = j) = P_\Psi(M|j) = \left\langle\Psi\right| M^\dagger M \left|\Psi\right\rangle = \left\langle\Psi|j\right\rangle \left\langle j|\Psi\right\rangle = Tr(\rho \left|j\right\rangle \left\langle j\right|) . \tag{42}$$

Measurements do not necessarily need to be expressed as *orthogonal* operators, but rather can be more general, and not correspond to an *orthogonal basis*. Such is the case of the so called *Positive operator valued measurements* (POVM)

$$\sum \left(M_m^\dagger M_m\right) = I \tag{43}$$

In both cases the measurement over a density operator is expressed as

$$M_m \rho M_m^\dagger \,. \tag{44}$$

2.3.3 *Entanglement*

Entanglement is perhaps the most distinctive characteristic of quantum mechanics and, arguably, the most important component of quantum technology. The phenomenon was raised for the first time in a paper by Einstein, Podolsky and Rosen in 1935 [141], as a *by-product* of a thought experiment involving two spatially separated particles with total *spin* 0 (*singlet state*), which interact in a way, such that the outcome of a measurement of the *spin* in one the particles defines the outcome of the same measurement made in the other particle, a direct consequence of the *non-commutativity* of certain observables (discussed in previous section 2.2.4). This phenomenon suggests a sort of *inseparable behaviour* of the two particles, happening *regardless* of the distance between them, in an *apparent* violation of *special relativity*. In a follow up paper by Schrödinger in 1935 [319], the phenomenon was named "entanglement", for the first time, and recognized as the most distinguished aspect of quantum mechanics.

The original argument of Einstein et al. [141], had the objective of criticizing in one hand the purely statistic prediction power of quantum mechanics (indeterminism), and, by finding a case where it is impossible that every observable has, simultaneously, a definite value, its apparent *anti-realism*. The main claim was that the *wave-function* does not possess all the information to avoid these problems, and therefore, quantum mechanics must be incomplete. This originated a new field of study on trying to find alternative theories to quantum theory able to solve these issues, the so-called "Hidden variable theories". A particular important class of candidates of the latter were the "local hidden variable theories", where the problem of entanglement in quantum mechanics was solved resorting to *hidden local information*, available to each side of the entanglement parties. However, these theories were ruled out experimentally, as explained in the following sections.

Bell experiments

The idea of the non-locality, which at first was considered an *insufficiency* of quantum mechanics, ended up developing into one of the fundamental principles of quantum mechanics, particularly as a consequence of the results of Von Neumann [208], and Gleason [174], but mostly of the so-called Bell theorems [51, 52]. The latter allowed the conception of several experiments, which could be performed in laboratory, to validate non-locality as truthful, and a core property of quantum mechanics. The Bell theorems are based in

an experimental setting based on a *singlet* state involving two particles, with global *spin* 0. Due to the commutativity properties of *spin* operators in different axis, it is possible to choose two spin components σ_1, σ_2 in different axis, such that if the action, over the *state vectors* of two separate systems, denoted by $\vec{a}$ and $\vec{b}$, $\sigma_1.\vec{a}$ yields $+1$, the subsequent action $\sigma_2.\vec{b}$ yeilds -1. Hence the expected value of the measuring the system in σ_1, σ_2 is given as

$$\left\langle \sigma\vec{a} \middle| \sigma\vec{b} \right\rangle = -\vec{a}\vec{b}\,. \tag{45}$$

In the other hand, the calculation of the expectation values of $\vec{a}$ and $\vec{b}$, in a *local hidden variables* setting, should read as

$$P(\vec{a},\vec{b}) = \int d\lambda p(\lambda) A(\vec{a},\lambda) B(\vec{b},\lambda)\,, \tag{46}$$

where λ is a tuple of *local variables*. The Bell theorem in [51], shows that there is no choice of variables λ is possible so that the expectation value of equation (46) matches the expectation value of equation (45), as this would be a contradiction, ruling out local *hidden variable* theories. Furthermore, from these works were also derived a set of inequalities, useful for the distinguishability of non-separable (quantum) and separable (classical) correlations, the so-called *Bell inequalities*, which are useful to detect non-locality in actual experiments, namely:

$$|ac| - |ba| - |bc| \leq 1\,, \tag{47}$$

where a, b, c represent the local components on each experiment. Also, other inequalities of this type are given by

$$C_h(a,c) - C_h(b,a) - C_h(b,c) \leq 1 \tag{48}$$

where C_h, represents an arbitrary classical correlation function. It has been shown, multiple times in experiments [31], that these *inequalities* can be violated in many Bell-like experiments. Moreover, many other arguments of this type have arised, showing the existence of non-local effects in quantum mechanics. Hence, *local* hidden variables are excluded as candidates to a complete theory of quantum mechanics. In despite of this there are many non-local *hidden variable theories* candidate to a complete version of quantum mechanics, such as the ones of De Broglie [126], or Bohm [70] theories.

The behaviour of entangled systems

Entanglement is characterized by the non-separability verified in multi-partite states, *witnessed* by appearance by *non-separable* correlations in the observations in two spatially separate quantum systems. A famous example of states in such conditions are the so-called *Bell states*:

$$|\phi\rangle = \frac{1}{\sqrt{2}}\left(|0_1 0_2\rangle + |1_1 1_2\rangle\right) , \tag{49}$$

where the indexes $1, 2$ correspond to two different *two-level* [4] quantum systems. In this case the existence of entanglement is pretty obvious, as one can easily observe that it cannot be written as a *tensor product* of two separable states, and it can be *straightforwardly* confirmed by the evaluation of the partial trace in one of the systems, upon *tracing out* the other one. However, the detection of entanglement in general, is far from trivial and the conception of *Hermitian* operators able to detect entanglement in states, the so-called *entanglement witnesses*, is an area of study on its own.

The most important fact about multi-qubit entangled systems, is that they behave as a *pure state*, i.e. behaving similarly to a *pure state* of a *single* qubit system, exhibiting coherence between the elements of the orthogonal basis. However, the density matrices for any state involving only subset of qubits in the *entanglement*, are *mixed*, i.e. no coherence is exhibited. This is verified, for instance, in the teleportation protocol where two qubits are involved in the entanglement, yielding a pure state, however the density matrix of each of the individual qubit presents itself as mixed state.

Another important fact about entangled and classical states, is that classical states are always convex (separable) and entanglement states are not [355], i.e. a linear combination of a separable state is also a separable state, for instance, given two classical states,

$$\left|\phi_1^{AB}\right\rangle = \sum_i p_i \phi_i^A \otimes \phi_i^B \tag{50}$$

and

$$\left|\sigma_1^{AB}\right\rangle = \sum_i q_i \sigma_i^A \otimes \sigma_i^B , \tag{51}$$

any linear combination of such states yields also a separable state:

$$p\sigma_1^{AB} + (1-p)\sigma_2^{AB} = p\sum_i p_i \phi_i^A \otimes \phi_i^B + (1-p)\sum_i q_i \sigma_i^A \otimes \sigma_i^B . \tag{52}$$

A somehow fundamental method to detect *entanglement*, unfortunately very hard to apply, is exactly by trying to find such a factorization for a given state, a problem clearly intractable from the computational point of view, due to the large number of possible factorizations. The conception of *efficient* entanglement witnesses is a very broad and active area of research [224].

4 Quantum systems possessing only two stationary states, here denoted as $|0\rangle$ and $|1\rangle$, the so-called qubits

Quantifying entanglement

Entanglement may manifest in different degrees. Hence, a very relevant question becomes the definition of *efficient* methods to quantify entanglement in quantum systems, which becomes specially relevant, for instance in quantum computation, as it is considered a fundamental resource in computation. In a way, entanglement is a measure of *non-classicality* of states, i.e. a measure of distance between non-separable states. A basic measure of *entanglement* is mutual information,

$$I_N(\rho_A : \rho_B; \rho_{AB}) = \mathcal{S}(\rho_A) + \mathcal{S}(\rho_B) - \mathcal{S}(\rho_{AB}) \,, \tag{53}$$

where ρ_A and ρ_B are the *density matrices* of two separate systems, and $\mathcal{S}$ is their Von Neumann entropy.

$$\mathcal{S}(\rho_A) := -tr(\rho_A \ln \rho_A) = -tr(\rho_B \ln \rho_B) \,. \tag{54}$$

However, this measure does not behave in *mixed-state* scenarios, due to the existence of processes able to transform them in *pure states* and *vice-versa*, i.e. processes of *purification* [355]. Therefore, a more general definition of measure of *entanglement* is necessary, which is given, for instance, as

$$E(\sigma) := \min_{p \in \mathcal{D}} \mathcal{D}(\sigma || \rho) \,, \tag{55}$$

where D is a measure of distance between two matrices, where σ is an entangled state, subject to *purification* and ρ is a classical state and $\mathcal{D}$ is the set of convex states. Any valid measure of entanglement, shall obey to the following conditions:

- for separable states $E(\sigma) = 0$, i.e. the state is not entangled, and cannot be purified to an entanglement state;
- any measure of entanglement remains unchanged under the action of locally unitary operations[5];
- any measure $E(\sigma)$ cannot be increased by local operations, classical communication and subselection (see figure 2)

$$E(\sigma) \geq \sum_i p_i E(\sigma_i) \,. \tag{56}$$

There are many measures which respects these conditions, for instance: relative entropy, entanglement of formation, or entanglement of distillation, or Schmidt rank [142]. Nowadays the quantification of entanglement is very relevant, specially in many-body systems, which, however, has revealed itself as challenging [353, 316, 23].

5 Local unitary operations, are operations which can be written in the form $U_A \otimes U_B$ (note that the CNOT gate is not)

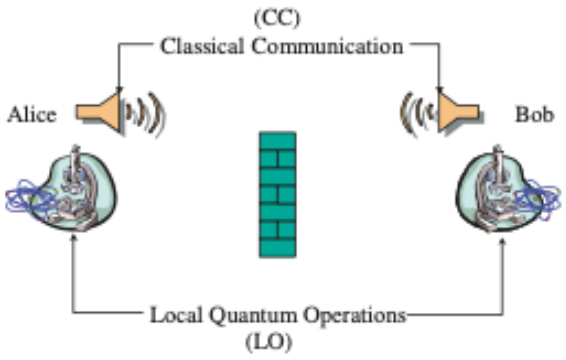


Figure 2: Local operations and classical communication operator definition. Adapted from [290]

2.3.4 *Decoherence*

The nature of the *wave collapse* in quantum mechanics, defined by the so-called *measurement* problem, is still an unsolved problem, and subject to intense debate in many areas of knowledge. *Decoherence* processes, introduced by Zeh in 1970 [383], while not solving the *measurement* problem, provide a very good explanation for the apparent decay of pure (quantum) states to mixed (quantum) states observable at macroscopic level [386].

As discussed in section 2.3.1, every mixed state can be *purified* into an *e*ntangled pure state, i.e. a mixed state is always part of a pure state of larger dimension. The process of *decoherence* is somehow a *purification process*, where due to the introduction of *entanglement* between a *closed system* $\mathcal{H}_s$ and its environment $\mathcal{H}_e$,

$$\mathcal{H}_s \otimes \mathcal{H}_e \,,$$

errors are introduced into the state $\mathcal{H}_s$, which then decays to a (classical) mixed state. Unfortunately, this process happens in an *unpredictable* and *uncontrollable* way, and the incapacity of controlling such *system-environment* interactions, has been the major problem in the conception of quantum computers (more on this on section 2.4). Nonetheless, this type of systems, interacting with their environments (also known as *baths*), constitutes an area of study on their own: *open quantum systems*. While the methods of this area are limited, and only able to deal to small systems, very far from physically meaningful systems for *decoherence* processes, they provide useful insights in many applications, for instance in biology (see more on this on section 3.3).

Decoherent processes

The decoherence observed in the *large* quantum systems, for instance in quantum technology, is an *uncontrollable* and *unpredictable* process, exactly because the number of quantum interactions is completely *intractable* for classical computers, and arguably, will still be too hard even for quantum computers. However, the effects such interactions have in closed systems, due to action of the *bath*, are well-known, which simplifies the conception of strategies to compensate them, and can be divided into three categories [82]:

- amplitude damping;
- dephasing;
- depolarization.

Hereby, we further describe each of these processes.

- **Amplitude damping** interactions encompass the processes that cause loss of the amplitude of one or more system's *eigenstates*. The spontaneous emission of a photon from the system to the environment from a two-level atom is an example of this kind of processes, which can cause the decay of an *excited* state to its *ground state* [276].
- **Phase damping, or dephasing** processes cause the decay of the off-diagonal terms from the system's density matrix, over time, down to zero, removing *superpositions*, i.e. coherences, of the system state. An example of this process can be observed by the interaction of a *two-level* system with its environment, letting the initial state of the latter system be given as $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$. Coupling this system with an environment, and modelling the environmental effect as a *Gaussian distribution* of relative phases θ, with zero mean value and variance 2λ, gives raise to the following *mixed state* [179]:

$$\rho = \begin{pmatrix} |a|^2 & ab^* e^{-\lambda} \\ ba^* e^{-\lambda} & |b|^2 \end{pmatrix}. \tag{57}$$

  It is easily observable, that the off-diagonal elements decay exponentially to 0 as λ increases, and as the variance λ is proportional to the time variable, the coherence in the initial *two-level* system decays over time [276], ultimately, removing any coherence from the system, turning the original coherent probability distribution, into a non-coherent, classical, one. These processes conserve the energy of the system, contrary to what happens with amplitude damping.

- The **Depolarization** changes the system state to a mixed state, with a probability P of another pure state and the probability $(1 - P)$ of the initial state of the system, and it can be thought as a combination of the other two types of decoherence [1].

2.4 Quantum computing

Quantum computation is a scientific discipline born in the eighties, following the idea, proposed Feynman in 1982 [150], to employ quantum mechanical systems in the simulation of quantum mechanical systems themselves. This process, nowadays known as quantum simulation, and as already foreseen by Feynman, yields a huge (exponential) computational advantage in some quantum simulations.

This was the first idea leading to quantum computation, which was followed by the first formal computational models by Benioff [53, 54], based on Hamiltonian evolution and being *continuous*. Later, in 1985, David Deutsch [129] provided a more general quantum universal computing model, the so-called quantum Turing machines, as an extension of their classical counterparts. Borrowing notions from the field of *reversible* computing, he introduced the notion of a quantum circuit. This are, still today, the fundamental notions of the quantum computing models, from which originated a flurry of theoretical work, including many possible implementations, programming languages and formal methods to deal with them.

From the implementation point of view the hardest challenge is *decoherence* [81], i.e. the hardness of conserving the superposition and entangled quantum states, from the errors caused by the environmental interaction. To solve this problem, many physical architectures have been developed, as well as alternative computing models and error correction strategies. However, it cannot be said that this problem has been overcome and quantum computing is a reality.

This section is devoted to the discussion of the main concepts underlying quantum computation, as well as the abstract models for quantum computers, under of a state-based perspective, as proposed by Deutsch [129].

2.4.1 *The state space: finite vs infinite dimension*

The state space for a quantum computation is given by the set of unitary vectors (vectors of norm 1) in a *Hilbert space*, which can be of finite or infinite dimension. Most of computer models, fall in the former category. However, Hilbert spaces with infinite dimensions are relevant, for instance, for quantum computing with *continuous* variables [287, 367] (e.g. the latter work uses the amplitudes of magnetic fields as slices of information), or for dealing with systems whose number of particles in the system changes throughout the computational process.

Nonetheless, the most common form of quantum computation resorts to a notion of *qubit*, 2-dimensional Hilbert spaces, $\mathcal{H}^2$, with $\{|0\rangle, |1\rangle\}$ as basis, i.e. the so-called computational basis, from which spaces of arbitrary dimension can be built. The state space of a *qubit* reads as follows:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \text{ such that } |\alpha|^2 + |\beta|^2 = 1; \lambda\,|\psi\rangle = |\psi\rangle, \lambda \in \mathbb{C} \tag{58}$$

where λ is a *global phase* and does not possess any *observational effects*. Such state space possesses a *geometric* interpretation given by the *Bloch sphere* [276], where a state is described in *polar coordinates* by two angles in a three-dimensional space, as presented in figure 3.

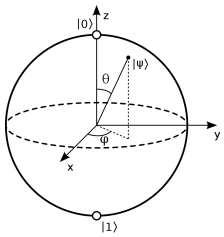


Figure 3: Bloch sphere, adapted from [276].

which is useful to understand *single-qubit* unitary transformations. The state spaces of qubits can be combined with the *tensor* product $\otimes$ to larger quantum systems. For a n-qubit system, the set of possible states is

$$\bigotimes_{i=0}^{n-1} \mathcal{H}_i^2 \,. \tag{59}$$

For instance, the state space of a two-dimensional space, is

$$|\psi\rangle = \alpha\,|00\rangle + \beta\,|01\rangle + \gamma\,|10\rangle + \lambda\,|11\rangle \;, \tag{60}$$

for $\alpha, \beta, \gamma, \lambda$ such that $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\lambda|^2 = 1$. Furthermore, in a purely abstract setting, one can group a set of qubits into a register:

$$\underbrace{\overbrace{q_1 \otimes \cdots \otimes q_l}^{\text{register with l qubits}} \otimes \overbrace{q_{l+1} \otimes \cdots q_{m+l}}^{\text{register with m qubits}}}_{\text{Aggregation with l+m qubits}} \,.$$

A set of registers defines the quantum computer's *memory*, whose semantics is given by the set of all possible *quantum* states. An important property of a quantum memory is that it cannot be *replicated*: as stated in the *no cloning theorem*, proven by Wooters and Zurek in 1982 [376], it is not possible to *copy*, exactly, a quantum state. This is a simple consequence of linearity, but significantly changes the way quantum programs are built.

2.4.2 *Transitions between states: unitarity and time*

In quantum mechanics, transitions preserve *unity* of states and are unitary, hence, programs are given by the class of such linear *unitary* operators ($U.U^\dagger = I$) in a Hilbert space. For a quantum system with n qubits the *signature* of the transition operators reads as follows:

$$U^{\otimes n} : \mathcal{H}^{2^{\otimes n}} \rightarrow \mathcal{H}^{2^{\otimes n}}$$

A particular relevant class of unitary operators used in quantum computation is that of Hamiltonian evolution operators, i.e. operators that characterize the evolution of an Hamiltonian, which possess the form e^{iHt}, where H is an *Hermitian* operator. Many computer models correspond to actual *analog* quantum simulators, i.e. quantum systems, which under appropriate preparation processes, are sufficiently similar to systems to be *simulated*, and hence suitable to be used as *quantum simulators*.

Timed evolution

In quantum mechanics, non-relativistic quantum systems, are captured under the Hamiltonian formalism, where the evolution of a quantum system, along a time t is given by the *exponentiation* of its Hamiltonian operator, given as:

$$|\Psi\rangle = e^{iHt/\hbar} |0\rangle \tag{61}$$

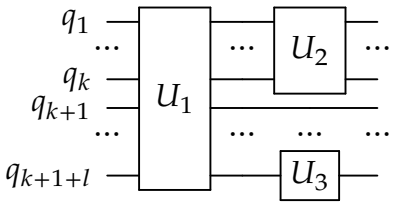
The dimension of the Hamiltonian operator is exponential in terms of the size of the basis of the system it operates, i.e. for a system with N qubits, the dimension of the matrix is exponential

$$2^N \times 2^N .$$

As a result, the dimension of the evolution operator grows very quickly, which makes the calculations intractable even for systems with a small number of particles: procedures such as *matrix multiplication*, or *diagonalization*, vital in the calculation of the evolution, or estimation of *eigenvalues*, while efficient in terms of matrices dimension (complexity $\sim D^3$, where D is system's dimension), become exponential ($D^{2^{N^3}}$) for N particle systems. However, as suggested by Feynman, quantum mechanics yields a huge advantage, i.e. an exponential one, in the simulation of the evolution operators if the evolution operator can be efficiently approximated (more on this on chapter 3).

The circuit model

Following to the discovery of quantum Turing machines, and based on the theory of *Boolean* and reversible circuits, Deutsch proposed the notion of quantum circuits: *acyclic* graphs connecting qubits, used as inputs, and sequences of *unitary operators* acting as transformations, mapping them into outputs. Quantum circuits have a specific notation, where qubits are represented by wires, and transformations by boxes, as in the following example:



,

where U_1, U_2 and U_3 are unitary transformations and q_1 up to q_n are input qubits. Quantum circuits are *analogous* to Boolean circuits: qubits are the quantum counterparts of classical bits, quantum gates are the counterpart of Boolean gates, and quantum circuits are the counterpart of Boolean circuits. Similarly to the classical case, Yao [378] has shown that families of quantum circuits can simulate quantum Turing machines, implying that they provide a complete quantum computational model.

Nowadays, quantum circuits are the useful components of *real world* quantum programs. Hence, it becomes specially important to identify minimal sets of gates from which all possible quantum circuits can be generated, and to find automatic means to approximate generalized unitary operators by them. In this area, one can observe that unitary operators, are fundamentally *reversible operators*, and, hence, some techniques can be *inherited* from the fields of *low power* and adiabatic electronics [128], i.e. subfield of electronics, were electronic circuits have no *dissipation* of energy (and information) to the environment, which can be achieved, for instance, by not using *electronic resistances*. Examples of *Boolean* universal reversible gates are the *Fredkin* and *Toffoli* gates (presented in table 1), which were shown to be (the first) universal gates also in quantum computation [130].

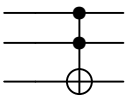
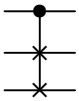
Gate	Circuit form	Matrix form
Toffoli Gate (CCNOT Gate)		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$
Fredkin gate		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Table 1: The original three-qubit, Fredkin and Toffoli, gates.

Later, it was shown that the set of all single-quantum quantum gates (some of them are presented in table 2), along with a *controlled NOT* gate, were universal [47]. Another universal gate set is given in the case of many individual two-qubit quantum gates, as shown by DiVicenzo in [138]. In fact, this type of results has been shown to a wide class of sets of unitary quantum gates [247].

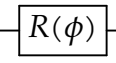
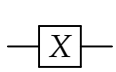
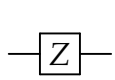
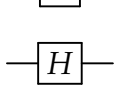
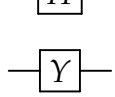
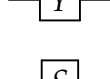
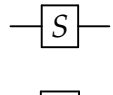
Gate	Circuit form	Matrix form
$R(\phi)$		$\begin{bmatrix} 1 & 0 \\ 0 & e^{-i\phi} \end{bmatrix}$
X		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Z		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
H		$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Y		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
S		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{bmatrix}$
T		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix}$

Table 2: Single qubit gates.

A well-known set of universal quantum gates is formed by the single-qubit gates X, H, and the two-qubit gate $CNOT$ (presented in table 3), and another set, named *standard basis*, is given by the set of gates $\{H, T, CNOT\}$. The latter is particularly important because it is *very close* to the so-called called *Clifford circuits* [263], which replaces T by the *phase gate*, $P = T^2$. The *Clifford circuits*, are efficiently simulatable by classical computer, bearing no quantum advantage, according to Gottesman-Knill theorem [8], hence the presence of the T gate is a *syntactic* indicator of quantum advantage.

However, the base given by the Clifford plus T gate circuits is universal, and there is a wide range of literature about the exact synthesis of unitaries [169], for instance, efficient classical algorithms to approximate unitaries with entries in the $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ ring [228]. Nonetheless, not all operators can be approximated efficiently with these gates and a more accurate description the possible approximations is given in [24]. More general ways of approximating unitaries, are based on the theory of matrix factorization. Relevant works on the subject are given, for instance, in [118], or [352, 244].

2.4.3 *Acceptance states*

An acceptance state in a quantum Turing machine is a quantum state where the correct output can be obtained upon measurement, with a probability that allows the statistical distinguishability of the wrong

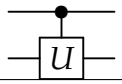
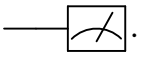
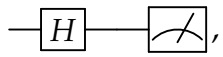
Gate	Circuit form	Matrix form
CNOT		$\begin{bmatrix} I & 0 \\ 0 & X \end{bmatrix} \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
CU (Controlled unitary)		$\begin{bmatrix} I & 0 \\ 0 & U \end{bmatrix}$
SWAP		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Table 3: Two-qubit quantum gates.

one. Measurements are mathematically captured by *projection operators* $Proj_{\varphi}$, *or* $|\varphi\rangle\langle\varphi|$, as discussed in section 2.3.2. In the language of quantum circuits, a measurement is depicted as follows:

 .

A very simple example of a circuit including a measurement is given by

$|0\rangle$  ,

which aims at creating a *quantum coin*, i.e. a qubit in the superposition $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, which upon measurement will randomly yield either $|0\rangle$ or $|1\rangle$, with 0.5 of probability (because $\left(\frac{1}{\sqrt{2}}\right)^2 = 0.5$).

2.4.4 *Semantics of quantum programming languages*

In the previous sections, quantum computation was introduced from the perspective of quantum Turing machines, which somehow provides a (state-based) *operational* semantic model. In this section, we shift attention to programming, focusing on the essentials of quantum programming languages and denotational semantics, in order to obtain a clearer connection between the programming instructions, and the corresponding physical processes.

Broadly speaking, a quantum programming language is composed of two main parts: the control component and the actions. Actions are, in general, measurements or unitary operations. The control statements, which aim coordinating the execution of actions, are the usual of programming languages: *conditions*, *loops* and *recursion*. The control component can be of two types, classical or quantum. The former relies using only on *classical variables*, i.e. variables in quantum stationary states, in control instructions. The latter allows variables in quantum (*superposition*) states, leading to the "strange" idea of *superposition* of fluxes (programs), which under certain conditions becomes *unphysical*, i.e. transitions may become non-completely positive, when measurements are involved. Every quantum programming language fall into one of these schemes.

In this section, we briefly discuss these families of programming languages, their features, associated physical effects and semantics.

Quantum data with classical control

The main paradigm in real-world quantum computation is the so-called *quantum data with classical control*, a term firstly coined by Knill in 1996 [229]. This covers all programming languages involving unitary operations, to be executed in a quantum device, and classical control instructions, to be executed by a *classical agent* that controls the quantum device, as depicted in figure 4.

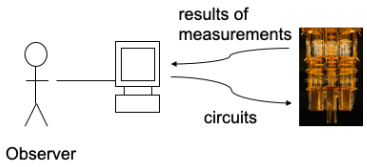


Figure 4: Quantum data with classical control.

In this setting, the execution processes in the quantum device are only a small part of the whole computational process. The latter, in the classical setting, can involve:

- creation and destruction of classical and quantum bits;
- generation of circuits (see section 2.4.2) to be executed in the quantum device;

- controlled execution quantum circuits using classical variables;
- application of measurements in order to obtain results from the quantum computer.

There is extensive research seeking for proper semantics for programming languages in this paradigm, as detailed in table 4.

Quantum Data with Classical control	Amount of bits/qubits	Relevant works
Finite Data types	Fixed	[323, 259]
Infinite Data types	Variable	[103]
Higher-order functions	Variable	[322, 324, 281, 193, 106]

Table 4: Literature on the semantics of programming languages with quantum control

The term *"finite data types"* concerns all data structures that originate from a fixed number of bits/qubits (data + control = data structure), for which a semantics was given by Selinger et al. [323], as *partial superoperators*. On the other hand, *"infinite data types"* concerns all data structures that originate from a variable (potentially infinite) bits/qubits, whose semantics for programs is given by *infinite Hilbert spaces* or the *dual* category of W*-algebras [103]. The semantics of quantum programming languages with *higher-order* functions has been an open problem for a long time, however a definite answer was provided in the works of Pagani [281], or Clairambault [106]. None of those mathematical theories leads to *unphysicalities*, principles of quantum mechanics such as *unitarity* and *complete positivity* are maintained, and they are embeddable in Hilbert spaces.

Quantum data with quantum control (without measurements)

An alternative notion of control is to use quantum variables instead of classical ones to control the flow of instructions, coined as "quantum data with quantum control". A toy language exhibiting this type of control was introduced by Ying in 2014 [381], and is characterized by the existence of an *if statement* which uses a quantum bit as the control variable, where superposition states are possible:

$$\text{if } (q_1) \text{ then action}_1 \text{ else action}_2 \text{ ,}$$

where q_1 is a *quantum bit* and action_1 and action_2 are the actions to be taken if q_1 is equal to 1 and 0, respectively. This type of programs has been studied in a variety of works [313, 22].

Another type of programs that fit this paradigm, are expressed in quantum programming languages that make use of *indefinite* causal structures. Quantum mechanics allows many kinds of *indefinite causal structures* (superposition of causal orders), with the restriction that the resulting transitions still preserve unitarity and complete positivity [28]. This idea led to a new computer model, the *higher-order* quantum computer [189], where an advantage to the circuit model has been identified. Such advantage comes from

the fact that such indefiniteness of causal structures cannot be captured directly by the circuit formalism, only approximated by it. The superposition of causal orders can be achieved by means of a device, known as the *quantum switch*, as discussed by Chiribella et al. [102]. This mechanism has already been experimented without resorting to any exotic physics in, for instance, [298], giving strength to the possibility of using indefinite quantum structures in actual quantum computing, despite the fact that no computer model implementing this kind of structure is known.

Quantum data with quantum control (with measurements)

The semantics of programming languages with quantum data with quantum control, but where the conditional actions involve measurements, apparently, cannot be given even by *superoperators* [38], suggesting that this programming language is not physically acceptable, i.e. complete positivity is violated.

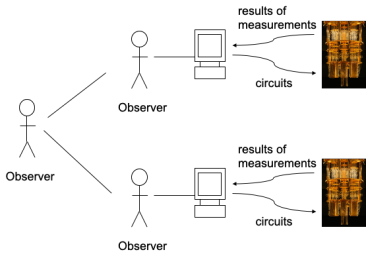


Figure 5: Quantum data with quantum control with measurements

Nonetheless, an interesting question remains on what the appropriate semantics will be for such a language, and if it fits in some variation of quantum mechanics and what would its computational power.

2.4.5 *Computing models*

Arguably, a quantum computer model can be defined by the type of quantum data (qubits, qudits or continuous information), the type of instructions (timed continuous, circuits, classical instructions), and the control strategy (how to prepare, execute and measure quantum procedures) and obviously the type of physical phenomena behind them. These can be diversified as quantum mechanics has shown to be, somehow, *very liberal* in terms of the *physically sound* computational techniques it allows.

Nonetheless, all the valid models can be described using the programming paradigms discussed in previous section, and the *resources* leading to quantum advantage are the same, i.e. *interference* and *entanglement*, regardless the technique used. The main aim of the different computer models is exactly to preserve such resources, against the *uncontrollable entanglement* actions of the environment, the so-called *decoherence*. In this section, we briefly explore the most relevant ones.

The first class of quantum computing models are are the so-called *quantum analogues*. This class of models presupposes the existence of two physical systems, one of which takes the role of *simulator* and the other one of the *simulated*. The idea is that with appropriate preparation, the evolution and measurement of the *simulator* systems, allows for an accurate estimation of the *simulated* one. Hence the simulated system must be similar enough to the simulator, i.e. the mapping,

$$H_{simulated} \rightarrow H_{simulator} \tag{62}$$

can be *efficiently obtained*, and additionally, it must possible to prepare $|0\rangle_{simulated}$ in the simulator, i.e. the state $|0\rangle_{simulator}$, and simulation of $e^{H_{simulated}}$ must be possible using $e^{H_{simulator}}$ and the measurement results are compatible. There are many examples of this kind of simulations, for instance using ultra-cold atoms, to simulate phase transitions [177], or the simulation of many-body physics and certain properties of cosmological bodies using ultra-cold gases [69, 346]. Good surveys of this kind of simulators, are available in [86, 167]. However, in most situations, analogue quantum simulators, are not universal, i.e. not all computations can be executed by them.

Other universal quantum computer models in quantum computation exist, and some of them depicted in table 5. The most obvious one is the one given by the circuit-based quantum computing discussed in section 2.4.2.

Model	Evolution	Dimensions
Circuit-based quantum computing	Discrete	Finite
Discrete quantum walks	Discrete	Finite
Measurement based computing	Discrete	Finite
Topological quantum computing	Discrete	Finite
Hamiltonian quantum computing	Continuous	Finite
Continuous quantum walks	Continuous	Finite
Adiabatic quantum computing	Continuous	Finite

Table 5: Universal quantum computing models

One of them is measurement-based quantum computation [83, 301], which focus on the construction of complex entangled states, and evolution is given by measurements. In this model, the idea is to prepare *entangled* states in such a way that, along with measurements in a suitable basis, a similar effect to the usual quantum gates is obtained. Another model proven to be universal is adiabatic computing [147, 18, 21], which is particularly suited to certain kinds of computational tasks (more on this will be seen on chapter 4). Also, topological quantum computing worths to mention, due to the *resistance* of topological states to *decoherence* [161, 274]. However, it proposes a different model of represent and operate over quantum information. Furthermore, quantum walks are another model that provides with universal quantum computing [252], in some cases more easily implemented, but it is only a particular case of Hamiltonian computing or of the circuit model. Regarding infinite dimensions, LLoyd [250] raised the idea of a universal computer model,

and the physical implementation is based on optics [265]. In last few years, a line of work that has been gaining relevance is the one of *classical-quantum* computational methods, which make heavy use of classical computation, narrowing the usage of quantum resources to the minimum possible. These methods were created mostly to be executed in the current short-term devices and an example of them is the family of quantum variational methods. While such methods are not computationally universal, they can be used in a variety of problems. In chapter 4, we further explore discuss these methods and their application to an actual example.

2.5 Quantum Complexity

Computational complexity is the subfield of theoretical computer science dedicated to the study of the *inherent complexity* of computational problems, by the characterization of the resources necessary to solve them, such as *time* or *memory*, aiming at their classification and the understanding of their *tractability*. Good surveys on the subject are available in [29, 283].

Along with computability, both of these subjects somehow capture, a mathematical notion of *epistemology* [6], and they actually help in the understanding the hardness of problems surrounding our daily lives, from economics and finance to the feasibility of preparation of states for certain physical experiments. And even the problem of determining whether a physical theory can be finitely calculated, or if it can expressed by a finite set of axioms.

This *bidirectional* relationship between physics and these two fields of theoretical computer science, became more evident with the advent of quantum computation. Actually, not only it helps understanding the limits of computation in a certain physical setting, but also the shape of physical theories, given certain computational requirements.

2.5.1 *Quantum complexity classes*

In quantum computation there are several resources generally used to study quantum algorithms, which can be divided in two parts: in one hand the quantum ones, *qubits*, *gates* and quantum oracle calls, and in the other the traditional classical resources used to study the *classical* parts of quantum algorithms. Associated to them there are specific notions of complexity as presented in table 6.

According to these different notions of complexity, algorithms can be classified in several classes. Here we briefly examine, which classes are specific to quantum computation, and how they fit in the traditional complexity hierarchy. The major relevant classes are presented in figure 6.

Resource	Definition
Quantum	
Qubit complexity	Number of qubits necessary to perform a quantum computation
Gate complexity	Asymptote on the number of gates necessary to approximate a quantum operator
Oracle complexity	Number of repetitions of a block of quantum circuits
Query complexity	Number of times the function subject of computation has to be evaluated
Classical	
Time complexity	Asymptote on the number of steps necessary to conclude a quantum computation, including circuit generation, oracle calls, measurement and eventual post-processing
Space complexity	Classical memory used to do the whole computation

Table 6: Most common resources used in quantum computation

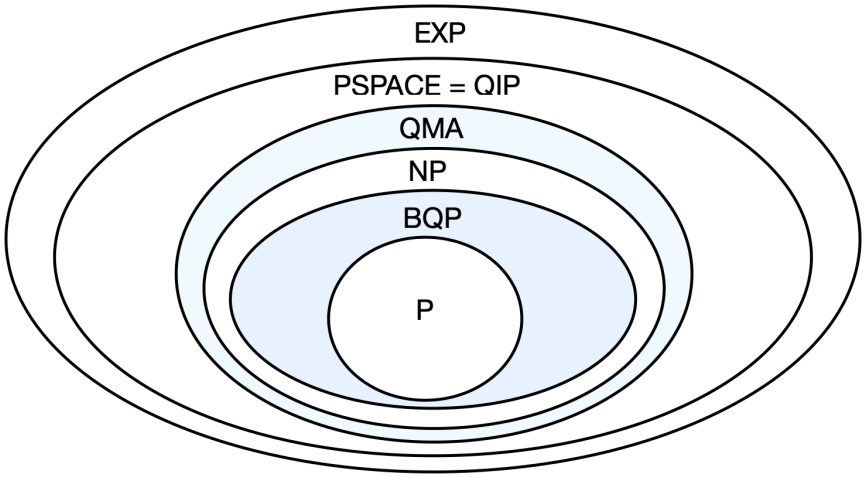


Figure 6: Complexity classes for quantum computation

Polynomial Time

The polynomial time (P) class, encloses all the problems that are *efficiently* solvable by a classical computer. Examples of these are the *graph reachability*, or the shortest path between nodes in a graph [332].

Bounded Quantum Probability

BQP class contains all problems that can be solved in *polynomial* time, by a quantum computer, with a *bounded error probability*. It is an important class in complexity theory, as it is believed to contain all problems *efficiently solvable* by a quantum computer, which is known to include factoring and discrete

logarithm algorithms, and the simulation of many *quantum systems*. Nevertheless, it is proven that BQP is contained in PP [156]. It is a generalization of the classical BPP class [5, 59], and can be also understood as the Polynomial (P) class of quantum computation.

Non-deterministic polynomial class

The non-deterministic polynomial class, encompasses the problems that can be solved *efficiently* by the hypothetical machine *non-deterministic* machine (a machine that can explore all possible solutions at the same time), or as an alternative definition, the class of problems for which the solutions can be verified in polynomial time. It is one of the most studied complexity classes, encompassing a wide class of well-known problems, such as the 3-SAT [113], the knapsack problem, or Hamiltonian path problems [332]. A subset of the problems in NP-class is known to be *complete*, i.e. all problems in the class can be reduced to an them [219]. A well-known example of problems belonging to the NP class that are not complete is the *factoring* and the *graph isomorphism* problems.

The relationship between the NP and P classes is not well-understood (the proof that there are no *polynomial* solutions to all NP problems, the so-called $P \neq NP$ problem [155], is a very famous open problem), so as the relationship between the classes BQP and P. Regarding this, Bennett et al. [56] provided evidence that the NP class is not contained in BQP, i.e. quantum computers, most likely, will not be able to solve NP-complete problems in polynomial time [3]. However, several problems of interest in quantum mechanics, as well as in industry, are NP-complete, such as finding the ground-state of the Ising Hamiltonian, i.e. the state corresponding to the lowest energy of a system governed by Ising Hamiltonians [105]. This connection will be further explored in chapter 4.

Quantum Merlin-Arthur

The Quantum Merlin-Arthur (QMA) complexity class contains the class of problems decidable by *Merlin-Arthur* (prover-verifier) protocols, where the Arthur (verifier) has access to a quantum computer, i.e. it is the quantum version of the MA interactive proof system a computational model based on the interactions between a prover, who possesses *infinite computational power*, but provides the wrong proof of a solution of a problem $1/3$ of the times, and a verifier, which only possesses a computer with PP power. A computation in this model corresponds to an interaction between the prover and the verifier, with a limited number of messages, in which the verifier accepts the proof [170].

The major advantage of this kind of computer models is that they are useful to characterize complexity classes, for instance, the NP class can be seen also as an interactive proof system, where the verifier has access to a computer with polynomial power, and the prover has infinite computational power, which is equivalent to *non-determinism*. The QMA class is a very important one in quantum computation, because it characterizes the class of hard problems to quantum computers [77], similarly to what problems of the

class NP are to classical computers. However, many problems of interest both in quantum mechanics and industry are instances of problems in this class and in section 4, some examples will be reviewed.

Probabilistic Polynomial time, Quantum Interactive Polynomial Time and PSPACE

The probabilistic polynomial time complexity class encompasses all problems that are solvable probabilistically, i.e. that the correct answer can be obtained with more than $1/2$ of probability, and it is known to contain the class QMA.

The Quantum Interactive Polynomial (QIP) is the complexity class that contains all problems that can be solved by a proof-verifier protocol, where the prover has infinite computational power, the verifier has access to a quantum computer, and both can exchange an unbounded number of messages. The PSPACE class is defined by all problems that can be solved with polynomial resources except for time and has shown to be equivalent to QIP [365, 207].

2.5.2 *Quantum advantage*

The algorithmic resources presented in table 6, do not provide any insight on which the structures of quantum theory contribute to the *quantum advantage*, i.e. the structures that make that some problems that fall in the *BQP* class being intractable in classical computing, have *efficient* algorithms in quantum computers. The analysis of such structures brings insight on how to build new quantum algorithms.

One may think that the main advantage of quantum computers resides in the so-called "quantum parallelism", which is partially true. The actual resources available in quantum theory that provide quantum advantage are precisely the features of quantum mechanics that are not available in classical mechanics: entanglement and interference. The former is, as it is well-known, the most distinguishable feature of quantum mechanics and essential to guarantee the existence of states with n-ary qubits, as otherwise only single qubit states would be achievable. It is also the hardest resource to obtain in practice, and the current attempts with short-term devices are basically about obtaining higher degrees of *entanglement*. However, while entanglement provides the notion of parallelism, the differentiating factor for quantum algorithms is *interference*, as argued, for instance, in the works of Fortnow and Lloyd [154, 249]. Further evidence is also provided in the studies made in [80], or [335], where interference of quantum algorithms is measured and compared, and it is concluded that *interference* is more *intense* in Shor algorithm [328], than in the Grover one [178], the former associated with an *exponential advantage* and the latter with a *quadratic advantage*. Hence, *interference* and *entanglement* are the essential resources of the quantum advantage, where they play a complementary role, entanglement being the basic component, and interference the differentiating component.

An interesting discussion in quantum computation, regards the computational power of variations of quantum mechanics, which is not a purely philosophical discussion, as the hypothesis that a theory of quantum gravity fits in some sort of extension of quantum mechanics is real. Two main lines of work shall be considered in this field. One considers variations on the interference patterns of quantum mechanics, which yields a hierarchy of different quantum theories [334, 119, 239]. Another one deals with non-linear quantum theories. The latter were considered to solve the issues of quantum mechanics [374], but that bring problems on their own [66, 369, 368], and have a strong relationship with the existence of *closed timelike curves*, which besides bringing a great computational advantage, pose physical problems under certain conditions [194]. In 1992 David Deutsch proposed a model for quantum mechanics, which, simultaneously, allows the action of curved spacetimes possessing closed time-like curves (D-CTC's), and is free of paradoxes and *superluminal signaling* [131]. This model is based on the additional requirement of self-consistency of the closed time-like curves, namely, only the ones having *fixed points* are considered. In this model it is possible to do perfect cloning of quantum states and distinguish efficiently *any non-orthogonal states* [84]. Exactly because of this, there is a significant quantum advantage, implying *efficient* solutions for all problems in PSPACE [10, 36]. This model is, however, equivalent to a classical theory, and quantum effects are irrelevant for the computation [9], being no more powerful than a classical computer with access to the same timelike curves [25]. Following these lines, based on a model of closed timelike curves using teleportation and post-selection (P-CTC's) [251], Aaronson has shown that quantum computers could solve all problems of the class PP, and hence of the class NP [4], less powerful than D-CTC's.

The calculation of properties such as entanglement and interference are not compositional and may be as complex as conducting the whole simulation of the quantum process, and hence, very complex from the computational perspective. The development of *easier* measurements to detect entanglement and quantum advantage is now a very wide and fruitful line of research, which ranges from complex metrics to trivial properties that can be verified at the syntax level, in the definition of quantum circuits. Some measures that can be applied, are for instance, Schmidt rank [358], factorization into a product state of small subsystem [214], factorization into Clifford gates [176], existence of matchgates [347], small tree width [213, 261] or non-negative Wigner representation [356, 260]. Furthermore, it is also worth looking into the field of *descriptive complexity*, which aims at defining the characteristic languages of complexity classes, , ultimately guaranteeing that a well-formed program in the syntax, is automatically within a certain complexity class. This was, for instance achieved in the work of Dal Lago et al. [120], where a characteristic *lambda-calculus* of the BQP class (see section 2.5.1) was obtained.

2.6 Summary

In this chapter it were reviewed the Hilbert space and density matrices formalism for expressing quantum theory as well as the latter's distinctive properties, such as interference and entanglement, and how they

are crucial to quantum information and computation. It also introduced the fundamentals of quantum computation, such as the circuit model, and other well-known quantum computational models, the semantics of quantum programming languages, and complexity.
Some conclusions can be made in this section:

- Entanglement and interference are the most distinctive features of quantum mechanics.
- They are also the essential components of quantum advantage, where they do play a complementary role. There is also a wide range of methods to quantify them.
- The computational processes allowed in quantum mechanics can be soundly captured, compositionally, in a wide range of quantum programming languages.
- A multitude of methods has been conceived to preserve quantum advantage against environmental effects, from quantum simulators, measurement-based computers, or recently, variational methods.
- Variations of quantum mechanics may yield computational advantage but may also raise other issues from the physics point of view.

3

ON EFFICIENT QUANTUM ALGORITHMS

The number of existent quantum algorithms is now significantly bigger than in the early days of quantum computation, following successes of Grover, Shor, or Simon algorithms. The range of applications has also extended, from machine learning to finance, to simulation of chemistry systems.

However, the progress on the development of efficient quantum algorithms, that bear the so-called *exponential advantage* to classical algorithms has been slower than expected, and the main structures and building blocks of the algorithms in this situation, are short in number and very well studied. Hence, a natural way of trying to find new quantum algorithms is to characterize these structures in order to extend them to new domains.

In this chapter, we intend to so, by characterizing the main quantum efficient algorithms and the structures behind them. We also explore a quantum simulation of the energy transport in a small photosynthetic system, i.e. in the biology domain, as an example of system possessing a structure that can be efficiently handled by quantum computers: it is driven by a local-Hamiltonian. The work was published in co-authorship with Jose Guimarães et al. [179].

3.1 The Bounded Quantum Probability class

The Bounded quantum probability (BQP) class is believed to enclose all the problems that possess an *efficient quantum algorithm*, i.e. algorithms where the number of classical steps, gates and qubits required to do the computation is given by a *polynomial* function. It is believed to contain the classical efficient P (*polynomial*) class, and there is evidence of the existence of problems that pertain to the class BQP, but not to the class P, given by existence of the Shor [328] or Simon algorithms [330], efficient in solving problems known to be in the class NP [1].

Nowadays there is a wide range of efficient quantum algorithms, for which a comprehensive survey seems completely unfeasible, but good starting points are given for instance in [271, 107, 99, 212]. From here, one can identify two types of algorithms possessing quantum advantage, which we denote as the

1 Only evidence because there is no proof that the class P is not the same as the class NP

dynamic ones and the static ones. The former ones encompass the simulation of processes, and the latter the calculation of algebraic properties of certain mathematical objects. From both, it is possible to identify algorithmic building blocks, which, presumably, yield the quantum advantage, by maximizing the effects of interference allowed by sets of entangled qubits.

In one hand, the simulation of quantum processes, i.e. the dynamic algorithms, consists in the *efficient* conception of computational processes that are able to *mimic*, i.e. by being *statistically* indistinguishable, a quantum process to be simulated up to an error ϵ. This type of simulation is also known as *weak simulation*, in opposition to *strong simulation*, where it is expected that both processes produce *exactly* the same results.

On the other hand, the static problems concern the extraction of *generative characteristics* of functions, e.g. the period of a function as it happens in the Shor algorithm [328]. These algorithms seem to resort to the Fourier transform and Phase estimation algorithms, which possess an exponential advantage (more on this on section 3.4). The relationship between both types of algorithms is, however yet unclear.

Finally, there is a class of algorithms, which somehow makes use of both types of strategies, and includes algorithms such as the *eigenvalue estimation* [11] and the HHL algorithm for the resolution of *linear equations* [191].

3.2 Search, sampling and simulation algorithms

The Hamiltonian simulation and quantum walks provide universal models for quantum computation, which imply that not only all quantum algorithms can be put in such forms, but also that they can be reduced to each other. Both of these conceptual models provide a complementary perspective on quantum algorithms, allowing the unified study of, among others, search, sampling and simulation problems and provide insight about the specific structural properties that help in the characterization of their efficiency.

First of all, search and sampling problems are considered to be equivalent [7], i.e. for each problem of sampling there is an equivalent problem of searching and vice-versa, and both can be captured by discrete quantum walks. A search problem can be interpreted as *sampling problem*, where the *correct solution* is expected with high probability, as it happens, for instance, in the Grover algorithm [178], i.e. a generic search algorithm appliable to any search space with known dimension. It yields constant *quadratic advantage*, i.e. it takes $\sqrt{N}$ steps, where N is the size of the search space, and it can be efficiently reduced to a quantum walk, as extensively studied in many works [326, 291].

There are two useful specific measurements of complexity in quantum walks for the study sampling and search problems: the *hitting time*, which concerns the amount of steps a specific marked state is achieved, useful to characterize search problems and the *mixing time*, i.e. the time a quantum walk takes to get to its *stationary distribution*, characteristic of *sampling problems* [258].

Moreover, *discrete* quantum walks can be *always* be reduced to Hamiltonian simulations, i.e. *continuous* quantum walks, where the Hamiltonian corresponds to the diffusion matrix of the discrete quantum walk, as

discussed, for instance, in [95], drawing a common foundation between search, sampling and Hamiltonian simulation algorithms. In the setting of Hamiltonian simulation, one can obtain a clearer idea on the simulation and sampling problems that possess efficient algorithms, due to the extensive work on efficient algorithms in this regard. As discussed in sections 3.2.1 and 3.2.2 there are efficient algorithms for a wide range of Hamiltonians, which encompasses local, sparse and d-sparse ones. Therefore, it can be assumed that the sampling/simulation problems that can be phrased in a local, sparse, or d-sparse Hamiltonian, or graph, most likely possess an efficient quantum algorithm.

For searching problems the criteria are less clear, although it can be stated that simulation problems are, most likely, *simpler* than a search problems, as one can naturally expect that the mixing time is smaller than hitting time, as in the latter one is interested in obtaining a specific element, of set elements, rather than a global distribution. In [96] a translation between a search problem and finding a ground-state of *k-local* Hamiltonian is proposed. Finding the ground state of Hamiltonian is known to be very complex, if it involves components with dimension *greater or equal* to 2, as it will be discussed in chapter 4. Therefore, search problems that cannot be mapped into Hamiltonian simulation ones, which do not involve components with dimension greater or equal to 2, do not possess an efficient algorithm. On the other hand, search problems who can, possess an efficient quantum algorithm, however, it is unclear if this leads to any quantum advantage better than the quadratic one.

3.2.1 *Local Hamiltonians*

The simplest Hamiltonians known to be simulated efficiently with an exponential improvement are local Hamiltonians, as first conjectured by Feynman and then by LLoyd [248]. Local Hamiltonians can be decomposed into their local interactions, i.e. physical interactions happening between all subsets of the particles up to a certain dimension k, which encompass a wide class of physical systems. Example of these is the Fermionic Hamiltonians, which only involve *2*-dimensional *Coulomb interactions* and 1-dimensional *Kinectic components* (in chapter 4 we deal with an Hamiltonian of this type). Mathematically, this class correspond to the Hamiltonians that can be expressed as a sum of their local components:

$$H = \sum_i H_i \tag{63}$$

The simulation of such Hamiltonian requires the existence of an *efficient* classical method to construct a circuit that approximates the evolution operator e^{iHt}, up to an error ϵ. If the Hamiltonians in the sum commute *pairwise* $[H_i, H_j] = 0$ (see section 2.2.4), then the order of application of the Hamiltonians is irrelevant and the evolution of the Hamiltonian operator is simply given by,

$$H = \prod_i e^{H_i} \tag{64}$$

.

The simplest case of these type of Hamiltonians happens when the Hamiltonians are *diagonal*, which can be trivially approximated by quantum circuits. The problematic cases start when the Hamiltonians do not commute, for which a multitude of approximation strategies are available. The simplest one, without diagonalizing the whole operator, is to use the diagonalization matrices in between every Hamiltonian component that does not commute,

$$e^{H_1}De^{H_2}D^{-1} \tag{65}$$

, where D is the diagonalization matrix that makes e^{H_2} diagonal. This approach is quite limited, as the obtention of the diagonalization matrix is computationally complex, however, it is possible to apply it in several cases, for instance, in the simulation of the Schrödinger equation, where the basis transformation matrix (*Fourier transform*) is efficient and expressed as,

$$XFPF^{-1} \tag{66}$$

, where X and P constitute the *position* and *momentum operators*. Furthermore, beyond these conceptually straightforward techniques, there exist a wide range of approximation techniques. The cornerstone of many of such approximation methods is the *Trotter formula*, based in the *Lie product formula* $\lim_{n\to\infty}\left(e^{At/n}e^{Bt/n}\right)^n = e^{i(A+B)t}$, which reads as follows:

$$e^{iHt} \sim (e^{iH_1t/n} \dots e^{iH_nt/n})^n + \sum_{i>j}[H_i, H_j]t^2/2n + \sum_{k=3}^{\infty} E(k) \tag{67}$$

which states that an Hamiltonian operator, where local components do not commute can be approximated by the repetition (through n steps) sequence of local operators $e^{iH_1t/n} \dots e^{iH_nt/n}$ (as if they would commute), where the time is discretized in steps of size t/n,, and with the error bounded by the $E(k)$ formula, always being less than $||n(e^{iHt} - 1 - iHt/n)||$ and the error can be arbitrarily small with the increase of n.

The computational complexity can be estimated by the number of operations needed for the repetition of the operators involved. Each H_j acts on a local Hilbert space, the number of operations needed to simulate $e^{iH_jt/n} \sim {m_j}^2$. Hence the global simulation time obeys the inequality $n(\sum_i^l m_i^2) \leq nlm^2$, where $m = max\{m_i\}$. The error in each of the operations must be minor than ϵ/nlm^2. Now everything depends on l, some number of components of the variable.

3.2.2 *Sparse and d-Sparse Hamiltonians*

Sparse Hamiltonians are a more general setting than local Hamiltonians, encompassing a wider class of physical systems, as well as other algorithmic problems with practical interest, such as quantum walks

with exponential gain [100], or NAND trees approximation [101]. Sparse Hamiltonians are defined as Hamiltonians, with a limited number of non-zero entries per row: $d = poly(logN)$, where N is the dimension of the Hamiltonian. For a $k-$local Hamiltonian with m terms, the Hamiltonian is sparse if $d = 2^k m$.

The first algorithm to deal efficiently with this kind of Hamiltonians was introduced by Aharonov et al. [17], where both the number of gates and of oracle calls are polynomial. Since then, there has been a significant amount of work on the subject, striving to reach optimality of such parameters, where the works of [254] and [62], (almost optimal) and the recent ones of [253, 254] (optimal), shall be highlighted. The methods employed in these approximations are the recent technique of qubitization [255], and the truncated Taylor series method [61].

There is also some work on the simulation of *non-sparse Hamiltonians*, only applied in very limited cases [97], and for which no general efficient quantum algorithm is known [363]. Furthermore, to the best of our knowledge it is not clear how interference is used in these algorithms.

3.3 Case study: Simulation of non-radiative energy transfer in photosynthetic systems using a quantum computer

We explore now the experimental simulation of a local Hamiltonian, which can be also interpreted as discrete quantum walk, that can be built and runned in a quantum computer: the one of transfer energy, by non-radiative means, existent in first stage of photosynthesis. This process has been shown to be influenced both by *quantum coherent* and *decoherent* effects, aspects also explored in this simulation. This exploration also helps understanding, besides the quantum aspects of photosynthesis, how the theoretical aspects of quantum mechanics, particularly the environmental ones introduced in section 2.3.4, work.

Photosynthesis is a vital and pervasive complex physical process in nature, where the radiation of the Sun is captured by certain *living beings*, such as plants and bacteria, and transformed into the necessary *carbohydrates* needed for their survival [266, 237]. From the physics and chemistry perspective, it is a complex process occurring through several stages with several kinds of physical phenomena involved, namely, the light absorption, energy transport, charge separation, photophosphorylation and carbon dioxide fixation [153]. The understanding of such phenomena has greatly progressed in the past 40 years with the physical characterization of the structure of many photosynthetic complexes [127, 320, 93]. The comprehension of such processes would allow for many potential huge-impact industrial *breakthroughs* in the field of energy, from the great efficiency improvement in energy capture of solar panels [243] to the construction of artificial light-harvesting devices and solar fuels [377, 182, 183, 311].

The photosynthesis begins by the absorption of a photon. It occurs via excitation of a pigment molecule, which acts as a *light-harvesting antenna* connected to the rest of the photosynthetic apparatus by protein molecules. Photosynthetic pigment-protein complexes transfer the absorbed sunlight energy, in the form

of molecular electronic excitation, to the reaction center, where charge separation initiates a series of biochemical processes [266]. This work is focused on the first stage of photosynthesis, more precisely, on the transport of the absorbed radiation energy from the antenna to the reaction centre, which proceeds in the form of the so-called Excitonic Energy Transfer (EET), as schematically shown in Fig.7.

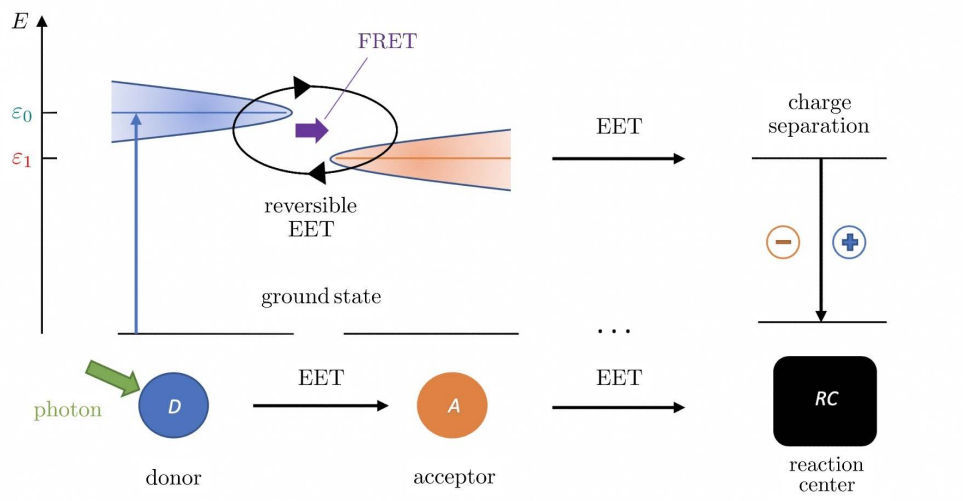


Figure 7: Schematics of the energy transfer process from light-harvesting antenna (the donor) through a chain of acceptor molecules to the reaction center. The excited states of the participating molecules, denoted ϵ_m, are broadened and it allows for resonance energy transfer via irreversible Förster-type resonant process of exciton transfer from donor to acceptor even if $\epsilon_m \neq \epsilon_{m+1}$, which is denoted by the thick arrow labelled FRET. However, if the coupling between the donor and the acceptor molecules is strong enough, the process becomes reversible and the exciton can go to and through many times before it is transferred; this situation is labeled by "reversible EET" and it does not require matching of the energy levels ϵ_m and ϵ_{m+1}. Picture taken from [179].

This transport is known to be very efficient in photosynthesis, as is the whole process, with the overall quantum efficiency of initiation of charge separation per absorbed photon up to 95% [266]. The absorbed photon creates an *exciton* on the antenna molecule, which can eventually transfer it to other molecules. In this context, it is called donor, while the others are called acceptors and the EET process can be described by the following reaction equation:

$$D^\star + A \rightarrow D + A^\star \,. \tag{68}$$

One may be lead to believe that, given the "size" of the physical components involved, the EET is a fully *classical* (i.e. irreversible and unidirectional) process, however experimental results have shown the opposite, with the verification of coherence between molecules over some period of time, evidenced by long-lived oscillatory features in the dynamical response of several photosynthetic systems in many experimental works [143, 240, 199]. However, it is predicted that these processes are still strongly influenced by the environment [302], as the donor-acceptor pairs are not isolated from the rest of the world, and, hence, the

appropriate theoretical setting to deal wih this kind of systems is the one of *quantum open systems*. In this setting, quantum systems are treated as part of a larger system ones, composed by the EET system under study and the environment. The latter is modeled by a thermal bath, which interacts with the EET quantum system, introducing relaxation and dephasing into and, therefore, influencing the efficiency of the energy transport.

The theoretical treatment of such systems is very complex from the computational point view, to which a myriad of methods is available, grouped by the type of regimes they can be applied, characterized by the *coupling strength* between *environment* and *main systems*, as well as the presence of *memory* effects (i.e. whether the system can be considered as Markovian or not) [266, 209, 307, 151, 338, 339, 205]

This case-study proposes a quantum simulation for the EET quantum transport, and the behaviour of the system is evaluated on different environment regimes: from the unexistence of environmental effects (pure) to different system-environment couplings, with the environment being modeled only by the employment of *pure dephasing effects*. The experimentally Hamiltonians defined [20], and already used in other quantum simulations [362], were used and the experimental study was conducted in the commercially available IBM Q of 5 qubits [115], which makes it a different approach from the existent ones [272, 292, 340, 362].

3.3.1 *Modeling the simulation*

Our implementation contains a quantum part, aimed at simulating the unitary part of the system's evolution, and a classical part that simulates the *stochastic* interaction with the environment, the latter only being able to mimic *pure dephasing* environmental effects.

We aim at exploring the energy transport underlying the photosynthesis, throughout time, under two regimes: (i) in an isolated system and (ii) under an action of the environment causing decoherence.

Concerning the particular qubit encoding chosen, a chain of $N = 2^q$ molecules is encoded by a set of q qubits, where $|m\rangle$ corresponds to the excitation (exciton) on the m-th molecule, e.g. for a two-molecule chain, state $|0\rangle$ represents the exciton on the first molecule and $|1\rangle$ on the second one, and a possible successful transport of energy would correspond to the transition of the state $|0\rangle$ to the state $|1\rangle$. We denote this as the *site basis*. The computational Hamiltonians under this encoding for the cases under study are discussed in the following sections. From now on, we shall set $\hbar = 1$. Also, it is convenient to measure the energies/frequencies in cm^{-1}, as it is common in spectroscopy.

3.3.2 *No–decoherence Hamiltonian*

Considering a small chain of N molecules, the system's Hamiltonian in the site basis reads as follows,

$$\hat{H}_S = \sum_{m=0}^{N-1} \epsilon_m |m\rangle \langle m| + \sum_{m \neq n} J_{mn} |m\rangle \langle n| \tag{69}$$

where ϵ_m is the first excited state energy of the molecule m and J_{nm} is the electronic coupling between the molecules n and m. The Hamiltonian (69) for just two molecules (1 qubit), in the 2×2 matrix form, reads:

$$\hat{H}_S = \begin{pmatrix} \epsilon_0 & J \\ J & \epsilon_1 \end{pmatrix} . \tag{70}$$

Its evolution operator is given by

$$|\Psi(t)\rangle = e^{-i\hat{H}_S t} |\Psi(0)\rangle \equiv \hat{U}(t) |\Psi(0)\rangle \ . \tag{71}$$

Although the Hamiltonian (70) possesses non-diagonal elements, finding a good approximation in terms of quantum circuits is relatively straightforward. A possible strategy for this is by finding a diagonalizing transformation, T, of the Hamiltonian, such that,

$$\hat{H}_S = T^\dagger \hat{H}_{S-diag} T \ . \tag{72}$$

where $\hat{H}_{S-diag}$ is the diagonal Hamiltonian. Therefore, the evolution operator can be rewritten as follows:

$$\hat{U}(t) = e^{-i\hat{H}_S t} = T^\dagger e^{-i\hat{H}_{S-diag} t} T \ . \tag{73}$$

The problem now reduces to the approximation of the T operator (and its adjoint) and the Hamiltonian $\hat{H}_{S-diag}$, which can all be efficiently approximated in quantum circuits. The latter operator is diagonal in the site basis, thus the unitary evolution operator can be expressed as

$$\hat{U}(t) = e^{-i\hat{H}_S t} = T^\dagger \left[e^{-i \sum_{m=0}^{1} E_m t} \right] T = T^\dagger \left[\prod_{m=0}^{1} e^{-iE_m t} \right] T \ . \tag{74}$$

The T and $T^\dagger$ matrices can be implemented by simple rotations, $R_y(\theta)$ and $R_y(-\theta)$, for a two-molecule system. However, for a higher number of molecules, a rotational decomposition algorithm together with the Gray code [276], which decomposes a matrix in the multiplication of a single qubit and CNOT gates, has to be used. Using this particular algorithm the *gate complexity* for N molecules is $O(N^2 log^2[N])$ [276]. On the other hand, the diagonalized evolution operator,

$$\hat{U}(t) = \begin{pmatrix} e^{-iE_0 t} & 0 \\ 0 & e^{-iE_1 t} \end{pmatrix} , \tag{75}$$

translates into trivial phase rotations over each of the energy eigenstates $|E_i\rangle$ of the system with the respective energy eigenvalues E_i. This operator can be constructed as a sequence of $CR_Z(\phi_i)$ gates applied to an

ancilla qubit (initialized at $|1\rangle$), where the angle is given by $\phi_i = -2E_i t$, $i = 1, 2$. The X gates are used to "select" the eigenvector to which the controlled rotation is to be applied. The circuit implementation of the operator defined in (75) is illustrated in Figure 8. The gate complexity of this operator, in terms of single qubit and CNOT gates for N molecules, is $O(N \log[N])$.

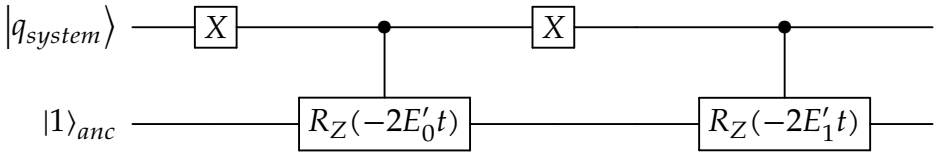


Figure 8: Implementation of the system's evolution operator. $|q_{system}\rangle$ is the state vector of the system's qubit in the energy eigenbasis.

For the whole circuit, resulting from the sequencing of $T^\dagger \hat{H}_{S-diag} T$, the number of qubits required to simulate a molecular chain of N elements is $2 \log_2 N$ and the gate count scales with $O(N^2 \log_2^2 N)$ single qubit and CNOT gates. The transformations T and $T^\dagger$, in the general case, possess a high circuit depth, which makes the system hard to simulate accurately, with low error rate, in the current available quantum computers. More efficient methods are available to do this task, as discussed in section 3.2.1, as the Hamiltonian is clearly local. However, for a matter of simplicity, the option of using the diagonalization method on the operator was used, which due to the size of the system, was not particularly problematic.

3.3.3 *Introducing decoherence into the system*

We shall implement artificial decoherence as pure-dephasing by adding Markovian fluctuations to the Hamiltonian. This approach is considered a good approximation in the *high-temperature* regime for the bath [241, 302, 82]. The actual algorithm to be used is the one of [364], which is used to simulate open quantum systems, with pure dephasing, modeling the action of the decoherence as classical random fluctuations (a telegraph-type classical noise affecting the system). The actual Hamiltonian for this system reads as

$$\hat{H} = \hat{H}_S + \hat{H}_F \tag{76}$$

and it consists of the system Hamiltonian, $\hat{H}_S$, of the previous section and the perturbation of a *bi-stable fluctuator* environment, $\hat{H}_F$. The latter simply shifts the energy by a constant value for each molecule, $\pm g_m/2$, as illustrated in Fig. 9. Explicitly,

$$\hat{H}_F = \sum_{m=0}^{1} \chi_m(t) \hat{A}_m \tag{77}$$

where $\hat{A}_m \left|m\right\rangle \left\langle m\right|$ is the projection operator and considering one fluctuator interacting with each molecule m,

$$\chi_m(t) = g_m \xi_m(t) \ . \tag{78}$$

The function $\xi_m(t)$ switches the fluctuator between the positive and negative values (appearing randomly) at a given fixed rate γ and g_m is the fluctuation strength (or the coupling strength to a molecule m). Physically, the action of the fluctuations is typically stronger for the excited states [241, 14] and g can be larger than the donor-acceptor coupling J.

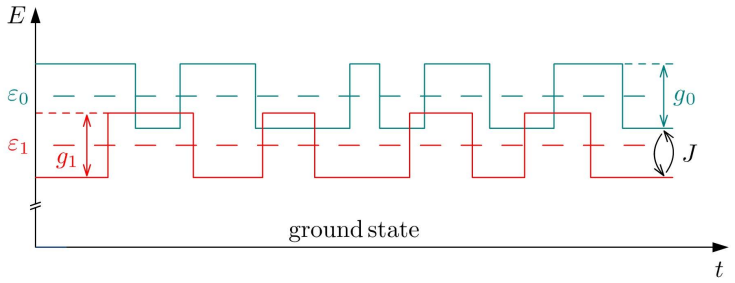


Figure 9: Uncorrelated random fluctuations applied to donor and aceptor's excited state energies, ϵ_0 and ϵ_1. Each molecule is affected by one fluctuator, which generates a telegraph-type classical noise. The fluctuators switch randomly between the positive and negative value at a given fixed rate, so that the periods of time when the molecule energy is constant, $\epsilon_m + g_m/2$ or $\epsilon_m - g_m/2$, are random. J is the coupling strength between the molecules that can be seen as the rate of hoppings between these fluctuating energy levels.

The implementation of such random bi-valued function $\xi_m(t)$, can be done in a straightforward way by a classical pseudo-random numbers generator with a probability of 50% of the values $-1/2$ and $1/2$. For circuit generation purposes, the values resulting from the random sampling have to be provided in advance of the quantum simulation.

The fluctuator interaction Hamiltonian and the system Hamiltonian do not commute, so, in order to generate an appropriate quantum circuit, one needs to apply an approximation technique such as the Trotter product formula [371]. Under this approximation, the unitary evolution operator of the Hamiltonian, for a time $t = N_i \Delta t$, where N_i is the number of iterations and Δt is the iteration time-step, becomes

$$U(N_i\Delta t) = \left(e^{-i\hat{H}\Delta t}\right)^{N_i} = \left(e^{-i\hat{H}_F\Delta t}T^\dagger e^{-i\hat{H}_S\Delta t}T\right)^{N_i} = \left(\left[\prod_{m=0}^{1} e^{\pm i\frac{g_m}{2}\Delta t}\right]T^\dagger\left[\prod_{m=0}^{1} e^{-iE_m\Delta t}\right]T\right)^{N_i}, \tag{79}$$

where E_m denote the eigenvalues of the system Hamiltonian. The whole circuit is presented in Fig. 10 for one iteration.

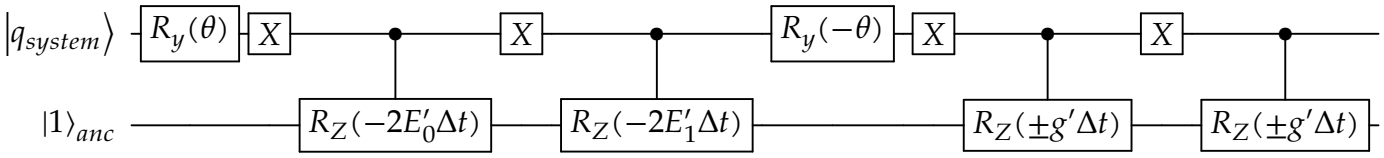


Figure 10: Implementation of one iteration of the system with decoherence algorithm. Here $|q_{system}\rangle$ represents the system's qubit state vector in the site basis.

The fluctuator interaction evolution operator $e^{\pm i\frac{gm}{2}\Delta t}$ is a *selective rotational gate* over a molecule m ($|m\rangle$). The fluctuator waiting time (interval of time between switches), i.e. $\frac{1}{\gamma}$, can only be equal or higher than the iteration time-step, Δt. The switching in the fluctuator-molecule coupling strength is performed at every $\frac{1}{\gamma\Delta t}$ iterations, where $a\Delta t = \frac{1}{\gamma}$, $a \in \mathbb{N}$.

For a time t, and iteration time step Δt, and the gate resource complexity, including single qubit and CNOT gates, , reads, for a single run, as $O(\frac{t}{\Delta t}[N(\log_2 N + F)])$, where N is the number of molecules and F is the number of fluctuators interacting with each one.

In the implementation of the system with decoherence, the algorithm gate resources complexity is $O(\frac{t}{\Delta t}[N^2 \times \log_2^2 N + NF])$ for a single run. This simulation, yet again, possess a very high circuit depth which makes its application unfeasible in quantum computers. The number of necessary qubits is the same as in the no decoherence simulation ($2\log_2 N$).

It also requires $O(NR\sum_{j=0}^{F} t\gamma_j)$ random numbers to be classically generated, where R is the number of runs of the algorithm and γ_j is the switching rate of the fluctuator j interacting with the molecule. The number of required simulation runs to average the results and obtain an error $\epsilon > 0$, is predicted to scale as $O\left([F\frac{t}{\Delta t}]^2/\epsilon^2\right)$. This complexity is calculated based on the possible non-degenerate energy state outcomes of the entire chain in the simulation for a time t.

3.3.4 *Results*

We conducted simulation experiments for the quantum transport in a molecular chain using the algorithm described in the previous section. We executed the simulation for the coherent system on a *real* quantum computer, the IBM Q of 5 qubits, while the *pure dephasing* scenario was simulated on the *QASM* quantum simulator, both in the near-resonant and non-resonant regimes. For the validation purposes, we compared the results for the coherent system with the theoretical predictions obtained by solving the Schrödinger equation (see Supplementary Information A.2).

As for the *decoherent* regime, we used a classical computation of the stochastic Haken-Ströbl model [184, 302]. The simulations and circuits involved, encoded in the Qiskit platform [115], can be tested in the following url: https://github.com/jakumin/Photosynthesis-quantum-simulation.

3.3.5 *Coherent regime*

The scenario for this regime was simulated with a simple chain of two molecules. As discussed in section *Materials and Methods* and using the parameters as proposed in [362], we define the system's Hamiltonian as follows:

(Near-resonant regime)

$$H_S = \begin{pmatrix} 13000 & 126 \\ 126 & 12900 \end{pmatrix} cm^{-1} \; ; \tag{80}$$

(Non-resonant regime)

$$H_S = \begin{pmatrix} 12900 & 132 \\ 132 & 12300 \end{pmatrix} cm^{-1} \; . \tag{81}$$

The results for both regimes were obtained using an actual quantum device (the IBMQ london of 5 qubits) and can be seen in Figs. 11 and 12, respectively. Due to the stochastic nature of quantum computers, the experiments were conducted with 2048 shots for each time value. In the following results, the probability of the donor and acceptor molecules being excited is denoted by $P(0) = \langle 0| \rho_S(t) |0\rangle$ and $P(1) = \langle 1| \rho_S(t) |1\rangle$, respectively.

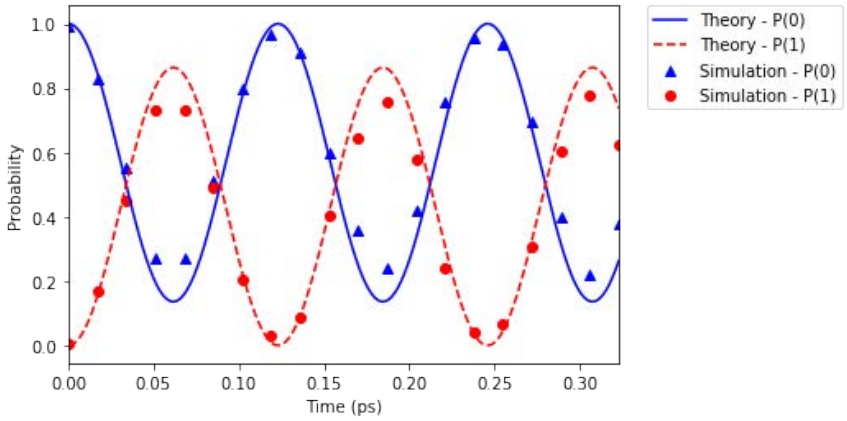


Figure 11: Evolution dynamics of the isolated system obtained by employing the quantum algorithm for the near-resonant system: simulation results (points) and theory (lines).

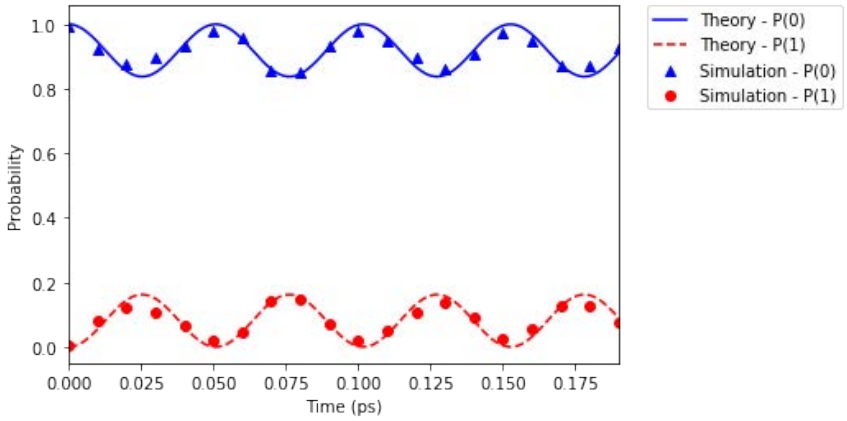


Figure 12: Evolution dynamics of the isolated system obtained by employing the quantum algorithm for the non-resonant system: simulation results (points) and theory (lines).

Taking the fluctuator's switching rate to be $\gamma = 0$ or the fluctuator-molecule coupling strength to be $g = 0$, one has the coherent regime. These simulations show the limiting case of the Redfield regime, i.e. the very weak system-environment coupling, $g \ll J$. The quantum beatings, observed in the simulation results, can be thought of as a *reversible* transfer of energy between the molecules, where the excitation goes back and forth across the molecules [94].

In the performed simulations, the near-resonant and non-resonant regimes have a maximum probability of $\sim 90\%$ and $\sim 20\%$, respectively, of the energy being transferred to the acceptor molecule. Using the quantum Liouville equation [266] (see Supplementary information A.2), the period of the quantum beating is $T_{near-res} \approx 123\, fs$ for the near-resonant regime and $T_{non-res} \approx 51\, fs$ for the non-resonant regime. These periods are in the femtosecond timescale of the experimentally observable quantum beatings [143, 111, 282]. The simulation results show a similar behaviour as those predicted by the Schrödinger and quantum Liouville equations, where the off-curve points are predominantly originated by errors in the quantum hardware.

3.3.6 *Decoherent regime*

The scenario for the regime with decoherence introduced is, in some respect, similar to the one presented for the coherent regime for a chain of two molecules. No further changes are made to the Hamiltonian discussed in the section *Introduction of decoherence in the system*. The quantum simulation results are compared with a theoretical evolution based on the stochastic Haken-Ströbl model, in the form of the Lindbland master equation [184, 302]. The Lindbland equations were solved in a classical computer using *Qutip* [211], a *quantum open systems software framework*. The set of Lindbland equations, correspondent

to the model in this setting, had one free parameter regarding the environment, the *dephasing rate*, γ_{deph}. The Lindbland equation in the Haken-Ströbl model reads:

$$\frac{d\rho}{dt} = \mathcal{L}[\rho] = -i[H_S, \rho] + \gamma_{deph} \sum_m (L_m \rho(t) L_m^\dagger - \frac{1}{2}\rho(t) L_m^\dagger L_m - \frac{1}{2} L_m^\dagger L_m \rho(t)) \tag{82}$$

where $L_m = |m\rangle \langle m|$ are the Lindbland operators, responsible for the system-environment interaction. The system Hamiltonian, H_S, is given by the matrix (80) for the near-resonant system and the matrix (81) for the non-resonant system.

The environment contains only one fluctuator interacting with each molecule with switching rate $\gamma = 125$ THz. As mentioned above, the dephasing rate, γ_{deph}, for the Lindbland equation is adjusted to the behaviour of the system under the action of a fluctuation strength g. For a range of fluctuation strengths of $[100, 1000]$ cm^{-1}, in the quantum algorithm, and the corresponding dephasing rate of the Haken-Ströbl model lies in the $\sim [2.3, 70]$ THz range. Due to the existence of random fluctuations, large number of samples had to be generated. The algorithm was implemented with 250 runs, where 5000 shots were performed for each time t. Figures 13 and 14 present the simulation results for different values of the fluctuation strength, along with the theoretical evolution dynamics, for the near-resonant and non-resonant systems, respectively. It is seen in Figures 13 and 14 that oscillation amplitudes decay over time, as expected, due to the loss of relative phase coherence between the excited states of the two molecules, evidenced by the disappearance of the quantum beatings. This is associated with the irreversible evolution when the system loses its capacity of performing *coherent* transport. Additionally, it is clear that the system is led to a classical distribution of the populations in the site eigenbasis.

In the regime under the study, where the environment is assumed to be at thermal equilibrium, the final probability distribution is calculated in the limit of the classical Boltzmann distribution $\langle m| \rho_S(t \to \infty) |m\rangle = const \times e^{-\frac{\epsilon_m}{k_B T}}$. Here k_B is the Boltzmann constant, T is the temperature of the bath and $const$ is a normalization constant [82]. Taking the limit of very high temperatures, the population terms approach the Boltzmann distribution $\langle 0| \rho_S(t \to \infty) |0\rangle \approx \langle 1| \rho_S(t \to \infty) |1\rangle \approx \frac{1}{2}$, which is compatible with the results obtained. The relaxation can not be fully observed in Figs. 13a, 14a and 14b because a very large number of iterations would be required for this.

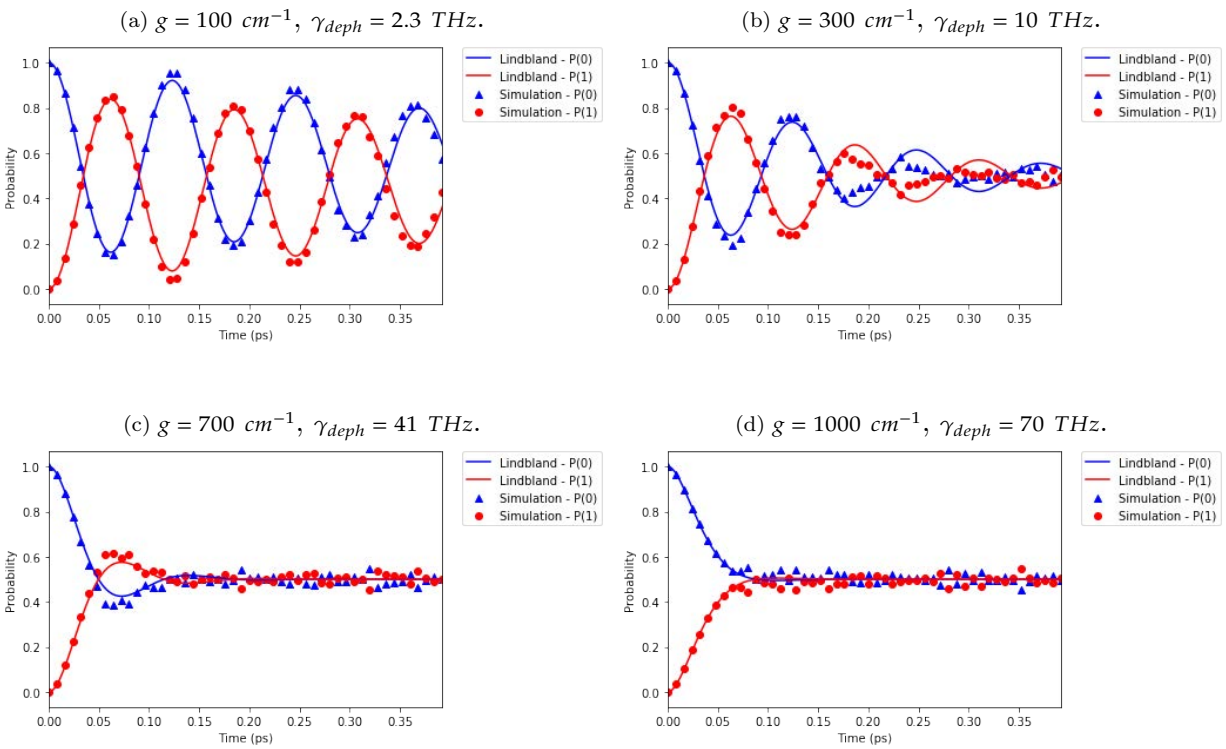


Figure 13: Evolution dynamics of the system with decoherence obtained by employing the quantum algorithm for the near-resonant system: simulation results (points) and theory (lines).

The switching rate must be high enough to observe the dephasing effects. Here we used a value ≈ 33 times larger than the transfer rate, J (that is, the fluctuator waiting time must be shorter than J^{-1}). As observed in the simulations, it is a suitable value for observing the relevant effects of random fluctuations in the system. At very low rates, it leads the system's evolution to a behaviour similar to the previously observed in the no-decoherence regime, Figs. 11 and 12.

The time that coherence lasts in the system is essentially defined by the fluctuation strength, g: in Figs. 13a, 13b, 14a and 14b (lower g) the coherence is maintained for some time, while in Figures 13c, 13d, 14c and 14d (higher g) it is quickly suppressed. In the latter regime, an approximated diffusive motion drives the system's evolution, where quantum beating is practically absent. The time that the quantum beating lasts in these simulations (until it reaches an approximate non-oscillating behaviour), is about $350\,fs$ in Figure 13b (near-resonant system) and $200\,fs$ in Figure 14b (non-resonant system), with a fluctuation strength $g = 300\ cm^{-1}$. At a longer time, it has been experimentally observed to persist ($t > 660\,fs$ [282]), a timescale which could be modeled in the present simulation by changing the environment parameters, i.e. lowering the fluctuation strength g as can be observed in Figures 13a and 14a.

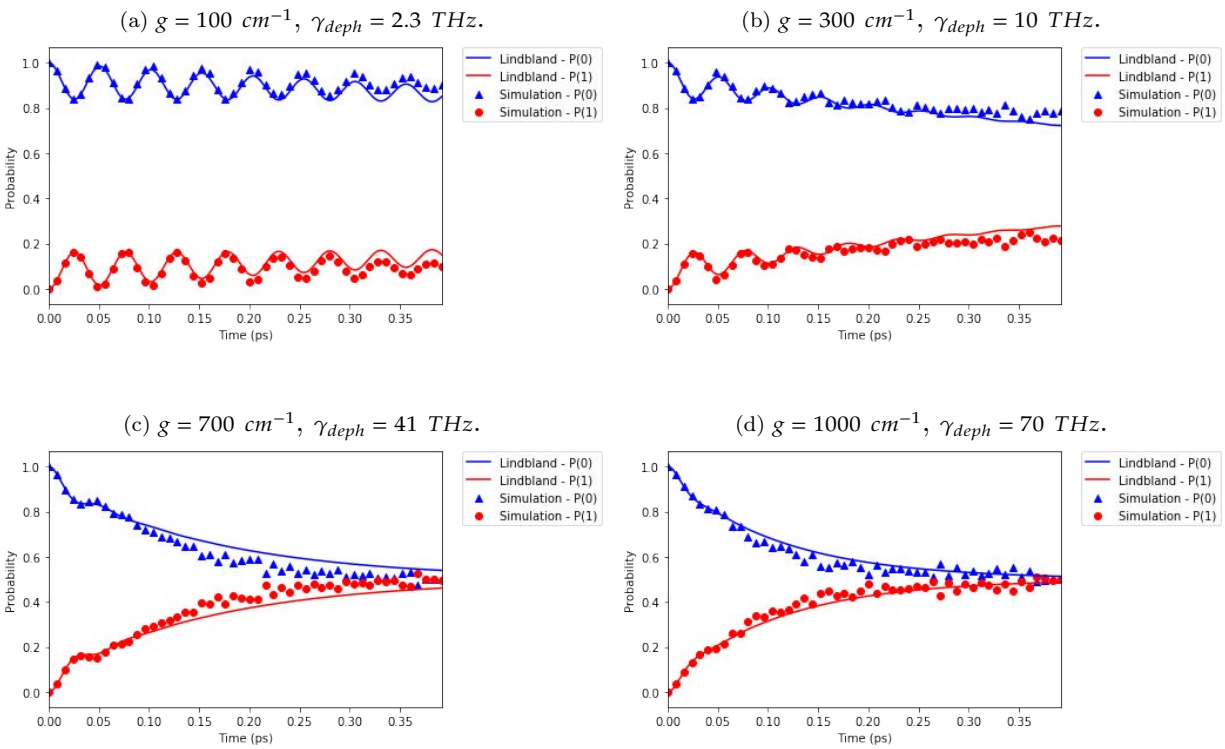


Figure 14: Evolution dynamics of the system with decoherence obtained by employing the quantum algorithm for the non-resonant system: simulation results (points) and theory (lines).

For each quantum simulation performed, a fitting process has been employed by adjusting the dephasing rate of the Haken-Ströbl model, so that the system's evolution in both classical and quantum algorithms have similar behaviours. This enables one to perform a direct comparison between both theories and to find the actual dephasing rate of the modeled environment over the various regimes considered in this work.

The results reflect a good agreement between the data obtained and the theoretical predictions both for the *coherent* case (vs Schrödinger equation) and the *decoherent* case (vs Haken-Ströbl model), for the different regimes definable by the coupling strength [302]. Similar to Ref. [362], this setting revealed itself as an interesting platform for the study the quantum and environmental effects in a small photosynthetic system, and therefore we consider, that the use of quantum simulations may be a feasible alternative in systems with medium-strong coupling and *non-Markovian systems*, in the future, whose main advantage when compared to similar works Ref. [362], is the flexibility on the implementation brought the quantum computer used.

However, the algorithm obtained, possess high requirements in terms of gates and qubits, and, hence, it is feasible to implement to realistic since a realistic quantum simulation of a photosynthetic system would have to involve hundreds of light-harvesting molecules, which is beyond the current quantum technology, and simultaneously the complexity of circuit generation is still $O(N^3)$) and it only involves pure-dephasing

baths. For future work, we aim at extending it to new types of bath, e.g. those allowing for higher exciton recombination rates and non-Markovian effects as well as to new geometries of photosynthetic systems, in particular, to the Fenna-Matthews-Olson complex [266].

In conclusion the coherent case is clearly described by a local Hamiltonian and hence it is efficiently simulatable, although not the best choice of circuit approximation was used in this work. The addition of *environmental noise*, to simulate *decoherence*, quickly raises the complexity of the algorithm, making a realistic simulation of a photosynthetic system unfeasible in current quantum computers. Nonetheless, a very good picture of how the environmental noise interacts with a closed system, destroying coherence, was obtained, complementing the discussion of chapter 2.

3.4 Algorithms based in the quantum Fourier transform

The Fourier transform is exponentially faster in quantum computers than it is on classical ones, and it is a *cornerstone* of several of the most relevant and efficient quantum algorithms, such as the ones of Deutsch-Jozsa [133], Simon [330] and Shor [328]. It seems to be the main cause of quantum advantage in quantum algorithmics, making extensive use of *quantum interference*. Its mathematical foundation can be given in terms of group and representation theory, which allows its generalization to other algorithms, and to obtain further characterization insights about them. So far, only algorithms based on the Fourier transform over functions subject to the action Abelian groups are efficient, i.e. the only ones in which *irreducible* representations are in a one to one relationship with the elements of the group.

From this perspective, efficient algorithms for a wide range of problems have been obtained, however, the attempts to generalize it to non-Abelian groups felt short, and no generalized algorithm for these problems exist, leaving out problems with high industrial impact such as the graph-isomorphism. In this section we explore these issues.

3.4.1 *The Quantum Fourier transform algorithm*

The Fourier transform is a vital tool in modern mathematics, physics and engineering with a wide range of applications from quantum mechanics to signal processing. In quantum computation, it also plays a central and vital, role as it is the most important building block of the most well-known efficient quantum algorithms, and definitely, a source of *quantum advantage*: the fastest classical Fourier transform has a time complexity of order $Nlog(N)$, while the quantum one, has only $(logN)log(logN)$, i.e. an exponential advantage. This makes the application of the quantum Fourier transform feasible, to problems where the classical Fourier transform is not, as, for example, to find the period of a function as in Shor algorithm. The most common form of the Fourier transform, used in fields such as signal processing, reads as follows:

$$\hat{f}(x) = \int e^{-2\pi ikx} f(x)\, dx, k \in \mathbb{R} \tag{83}$$

In this field of application, the Fourier transform, can be interpreted as a function between the domain of times and the domain of frequencies, where $e^{-2\pi ikx}$, corresponds to a sinusoidal function characteristic of a frequency k, and $\hat{f}(x)$ corresponds to the *frequency response* of the timed function at a given point, decomposing the timed signal in its frequencies. The same intuitions are valid, for instance in quantum mechanics: the Fourier transform is a function from the position domain to the *momentum* domain. A more abstract treatment of the Fourier transform, phrases it in terms of group theory, where it is interpreted as a function between two groups, the group of the original function $f(x)$, G, and the group of unitary linear operators $GL(V)$, which is given by the following expression:

$$\hat{f}(p) = \sum_{x \in G}^{n-1} f(x)p(x), p \in GL(V)\,, \tag{84}$$

where $f(x)$ is function over x, where $x \in G$, and $p(x)$ the *irreducible representation* at point x. The first *efficient* algorithm for the Fourier transform, the fast Fourier algorithm (FFT) was discovered by Cooley and Tuckey [114], and its translation and application to the quantum realm was firstly made by Simon [330] and Shor [327]. The QFT algorithm starts by creating a superposition state, in which the amplitudes of the states are the actual values of $f(x)$, as follows:

$$|\Psi\rangle = \sum_{x=0}^{2^n-1} f(x)\,|x\rangle\ . \tag{85}$$

Upon the application of the quantum Fourier transform, the resultant state is given by:

$$|\Psi\rangle = \sum_{y} g(y)\,|y\rangle \tag{86}$$

where function g(y), corresponds to the amplitude of each of the elements of the y basis, and corresponds to the actual Fourier transform formula

$$g(y) = a(\Phi \to y) = \langle y|\Phi\rangle = \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} e^{2i\pi xy/2} f(x)\,. \tag{87}$$

The *statistics* of the quantum Fourier transform, is *invariant* to *constant* shifts affecting the function $f(x)$, i.e. for functions whose resultant Fourier transform form

$$\frac{1}{2^{n/2}} \frac{1}{\sqrt{K}} \sum_{k=0}^{K-1} e^{2i\pi y(c+kr)/2^n}\,|y\rangle\ , \tag{88}$$

the statistics of of the Fourier transform, reads as

$$p(y) = \frac{1}{2^n}\frac{1}{\sqrt{K}}\left|\sum_{k=0}^{K-1} e^{2i\pi kyr/2^n}\right|^2, \tag{89}$$

i.e. it is not affected by c. The reason for this is that c only affects the global phase, and, hence, does not affect the statistics of the measurements. By reasoning about these statistics it is possible to retrieve important properties of the original function $f(x)$, such as its period as done in the Shor algorithm of the following section.

3.4.2 *The Shor algorithm and the hidden subgroup problem (HSP)*

The Shor algorithm [329] is the most important in quantum computation, due to its industrial impact, as it can *break* RSA cryptography [75]. In practice, what the algorithm does is finding the (large) period of a function, from which an attack on the cryptographic scheme can be built. The algorithm of Shor is based on Simon algorithm for the calculation of discrete logarithms [330], and in fact, these algorithms can be phrased in a more generic way as instances of the *Hidden subgroup problem* [74, 225], a problem from the domain of group theory. Later, other instances of this problem were found, as depicted in table 7. The formal definition of the *Hidden subgroup problem* goes as follows:

Definition 3.4.1. Given a function $f : G \to R$, where G is a finite group and R an arbitrary finite range, and the assumption that there exists a subgroup $H \leq G$, where f is constant and distinct on the left cosets of H, find the generating set of H.

The definition may sound somewhat *puzzling* for non-mathematicians, so the easiest way is to look into an example, which goes as follows:

Example 3.4.1. Consider a periodic function f on a group G presented in the following table:

x	0	1	2	3	4	5	6	7	8	9
f (x)	0	1	2	0	1	2	0	1	2	0

assumed to be of type $f : G \to \mathbb{R}$, where $+$ is the operation of G. The structure of the group, i.e. including its actual number of elements is not known, but according to the definition 3.4.1, $f(x)$ is constant in the cosets of the group. Hence, from the simple observation of the function, one concludes the cosets must correspond to the following sets:

$$\{0,3,6,9,...\},\{1,4,7,...\},\{2,5,8,...\}\,. \tag{90}$$

Each coset corresponds to an equivalency class, resulting from the action of an element of group, over the hidden subgroup. From group theory, it is well known that any element of

a group generates a subgroup. Hence, from the information available about the function (elements and operation), the possible subgroups read as follows:

$\langle 0\rangle = \{0\}$	$\langle 1\rangle = \{0,1,2,3,4,5,6,7,8,9,...\}$
$\langle 2\rangle = \{0,2,4,6,8,...\}$	$\langle 3\rangle = \{0,3,6,9,...\}$

So, having calculated all the subgroups, is trivial is to see what is the one that fits the coset structure presented in equation (90), is the subgroup generated by element 3, $\langle 3\rangle$, which can be confirmed by operating a arbitrary element of each of the cosets with the subgroup determined, allowing to retrieve the cosets back again:

$0 \cdot \{0,3,6,9,...\} = \{0,3,6,9,...\}$	$1 = \{1,4,7,...\}$
$2 \cdot \{0,3,6,9,...\} = \{2,5,8,...\}$	....

Hence the so-called hidden subgroup is $\{0,3,6,9,...\}$ and its generator is 3, which also corresponds to the period of the function, which can be trivially verified. It is also the number needed to perform the attack on RSA encryption scheme.

In the RSA cryptographic scheme, the modular *exponentiation* is used, instead of the modular *summing* operation, as explored in example 3.4.1, and the computational solution explored in such example is clearly *computationally inefficient*, as it requires the calculation of all possible subgroups, being only a toy example. The Shor algorithm is able to determine the right coset structure and the generator of the correct subgroup with high probability in a very efficient way, with an exponential advantage to the best classical algorithms. The same quantum algorithm used in Shor is common and efficient for all Abelian groups with many applications [186, 185, 270]. It has also been shown that the algorithm works *efficiently* for all groups, which even not Abelian, if they possess *normal subgroups* [187], only.

Problem	Query Complexity	Main Technique
Abelian Stabilizer Problem	Polynomial	Fourier Sampling/Transform [225]
Shor algorithm for factoring and discrete logarithm	Polynomial	Fourier Sampling/Transform [328]
Simon's XOR-mask finding a	Polynomial	Fourier Sampling/Transform [330]
Pell's equation and Principal Ideal	Polynomial	Fourier Sampling/Transform [186]
Unit Group and Class group	Polynomial	Fourier Sampling/Transform [185]

Table 7: Some problems with efficient quantum algorithms using the Fourier transform

The algorithm starts with the state preparation, where two logical registers are used, to encode the domain and codomain of a *homomorphic* function between Abelian groups. The first step corresponds to the preparation of the domain of the function in the first register, building a *homogeneous* superposition of values of the domain:

$$|\Phi\rangle = \frac{1}{2^{n/2}} \left(\sum_{x=0}^{2^n-1} |x\rangle \right) \otimes |0 \dots 0\rangle \tag{91}$$

The second step corresponds to the application of an oracle f, which based on the domain elements of the first register, constructs the codomain elements in the second register:

$$\left|\Psi_f\right\rangle = U_f|\Phi\rangle = \frac{1}{2^{n/2}}\left(\sum_{x=0}^{2^n-1}|x \otimes f(x)\rangle\right). \tag{92}$$

At this stage, due to the requirement of being constant on the cosets, for each $f(x)$ is associated a *coset* of the function, i.e. a potential measurement of the second register will yield the corresponding coset in the first register:

$$|\Psi_0\rangle = \frac{1}{\sqrt{K}}\sum_{k=0}^{K-1}|x_0 + kr\rangle\ . \tag{93}$$

In equation (93), kr, corresponds to the exponentiation of the generator of the *hidden subgroup*, r^k, and x_0 is the *coset* representative, which is unknown, and thus, prevents direct measurement of the generator. However, its effect can be discarded, if one goes instead onto the *phase* domain, which can be done systematically by the use of the Fourier transform. The application of the *Fourier transform* to the *coset* state yields the following state:

$$\frac{1}{2^{n/2}}\frac{1}{\sqrt{K}}\sum_{k=0}^{K-1}e^{2i\pi y(x_0+kr)/2^n}|y\rangle \tag{94}$$

While it can be observed that the *coset* representative x_0 is still part of the amplitude of the basis elements $|y\rangle$, i.e. by looking to the amplitude of an element of the basis y given by

$$a(\Phi \mapsto y) = \frac{1}{2^{n/2}}\frac{1}{\sqrt{K}}\sum_{k=0}^{K-1}e^{2i\pi y(x_0+kr)/2^n}\ , \tag{95}$$

it is also observable that the element x_0 does not have any *statistical* effect on the measurement of elements, as it only affects the *global phase* of the state, *disappearing* when the amplitude is *squared*:

$$p(y) = \frac{1}{2^n}\frac{1}{\sqrt{K}}\left|\sum_{k=0}^{K-1}e^{2i\pi kyr/2^n}\right|^2. \tag{96}$$

Hence, the generator r can be retrieved, by the use of a statistical inference technique named *Fourier sampling*. The equation above is equal to the geometric series,

$$p(y) = \frac{sin^2\left(\frac{2\pi.c_s.r.M}{2^n}\right)}{sin^2(\frac{2\pi.c_s.r}{2^n})} \tag{97}$$

, if $y = jK$, and 0 in the other case

$$p(y) = \frac{1}{2^n K} \frac{sin^2(\pi y)}{sin^2(\pi y/K)} = \frac{1}{r}. \tag{98}$$

It can be concluded that $j/r = y/2^n$, and one can obtain j and r from the irreducible form of $y/2^n$, which can be obtained by measuring y and expanding the result towards an irreducible fraction using the *Newton* algorithm. The algorithm is probabilistic and the period obtained must be validated by testing it: $f(x) = f(x + r)$, and the execution of the algorithm repeated, if necessary. The probability of obtaining a stated reducible to j/r is high, when compared to the demanded precision, which means the algorithm in sound. The algorithm corresponds to finding the Abelian group generator.

3.4.3 *Non-Abelian Hidden subgroup problem*

After the success of the hidden subgroup problem algorithm for Abelian groups, there has been an extensive research effort in extending the algorithm to non-Abelian groups. Examples of relevant groups in these conditions are the *dihedral* and *symmetry group*, for which efficient algorithms for their hidden subgroup problem will have an important impact, yielding efficient solutions to lattice-based cryptography [306] and graph isomorphism [50, 144]. However, despite of some successes on this side, [203, 206], the hidden subgroup problem for the most relevant groups is still beyond reach, even, as demonstrated, with the query complexity always being $log(|G|)$ for these algorithms [145]. However, for some groups the information within cosets is not enough to capture the hidden subgroup and further processing may be required, which is generally timewise inefficient. For some groups it may be sufficient to enhance the HSP with the application of a proper measurement strategies [37], however it has also been shown that the application of the technique is bounded by physical limitations [188]. Such information shall then be filtered classically in order to retrieve the hidden subgroup, which may include solving linear equations [310] or other alternative techniques [236]. Moreover, an effective *pretty good measurement* able to distinguish such states, will also need to involve $NlogN$ states, which makes it almost unfeasible from the physical point of view.

Due to impossibility of reaching an efficient algorithm for relevant groups with potential impact in industry, it was also attempted to explore different algorithms, such as hidden-shifts [98, 351], hidden-polynomials, hidden translations [162], hidden cosets [162] and the generalized study of symmetry groups ,which may have many different applications.

3.5 Hybrid algorithms

There is another class of quantum algorithm, which makes use of quantum simulation and of the exponential advantage of the Fourier transform, as it is the example of the estimation of eigenvalues of Abrams and LLoyd [11] and the HHL algorithm for the resolution of linear equations, proposed by Harrow, Hassidim

and LLoyd [191]. The former consists of two steps, namely, the simulation of the Hamiltonian and the application of phase estimation [124] (an algorithm relying on the Fourier transform) to obtain the eigenvalues corresponding to a given *eigenvector*, i.e. the composition

$$\text{Phase estimation} \circ \text{Hamiltonian simulation} \tag{99}$$

Therefore, the efficiency of the algorithm is directly dependent on the efficiency of the Hamiltonian simulation, given that the phase estimation step is known to be efficient. The latter algorithm, the HHL one, can be seen as a more generalized form of finding a particular eingenvalue, and also depends on the efficient simulation of the exponential of an Hermitian operator. Briefly, the algorithm is able to solve systems of equations of the sort:

$$A\vec{x} = \vec{b} \tag{100}$$

where A and b define a set of linear equations. The algorithm requires the simulation of e^{iAt} for different values of t, which possess a potential exponential advantage, as explored in section 3.2.1. The matrix A, may not always be a Hermitian operator, but it can be translated into one, by the following construction,

$$\begin{pmatrix} 0 & A \\ A^t & 0 \end{pmatrix} \tag{101}$$

and whenever A^t is an *Hermitian operator*, e^{iAt} is an unitary operator. Technically, the algorithm prepares $\sum_{i=1}^{N} b_i |i\rangle$, and then using *quantum simulation*, calculates e^{iAt} for a superposition of different times t, and further, with the help of the phase estimation algorithm, is able to decompose $|b\rangle$ in the eigenbasis of A and find the corresponding *eigenvalues* λ_j. The result of these operations reads as:

$$\sum_{j=1}^{N} \beta_j \left|u_j\right\rangle \left|\lambda_j\right\rangle \tag{102}$$

where $\{\left|u_j\right\rangle | j \in 1 \ldots N$ N is the dimension of system$\}$ is the eigenvector basis of A, and $|b\rangle = \sum_{j_1} \beta_j \left|u_j\right\rangle$. The only thing required right now is to find the inverse of each of the eigenvectors, producing the state $C\lambda_j^{-1} \left|\lambda_j\right\rangle$. Finally, one obtains the state

$$\sum_{j=1}^{N} \beta_j \lambda_j^{-1} \left|u_j\right\rangle = A^{-1} |b\rangle = |x\rangle \ . \tag{103}$$

Besides the possibility of the efficient simulation the Hamiltonian, the performance of the algorithm also depends of the differences of eigenvalues, captured by a parameter k. The actual elements $|x\rangle$, can be obtained individually in a single execution, which requires N repetitions of the algorithm, if one is interested

in every value. Nonetheless, one can also obtain the expected eigenvalues $\langle \vec{x} | A\vec{x} \rangle$ efficiently, which useful in many applications.

The applications of the algorithm are limited, due exactly to the caveats of the dependency of the Hamiltonian simulation, to which, as previously discussed, are only efficient for sparse or local Hamiltonians. Unfortunately, dense matrices are very relevant to specific problems in machine learning, from kernel methods to convolutional neural networks. However, there exist alternative approaches for the algorithm, which replace the condition on the sparsity on the matrix, to being approximable by *low-rank matrices* [321]. This algorithm exponentiates a matrix no sparse, positive definite low-rank [304].

There are many solvable systems of equations, that can be handled by this family of quantum algorithms: [60, 63]. The case of Hamiltonian simulation is simply a special case of ODE, where A is *anti-Hermitian* ($A \neq A^{\dagger}$) and f is zero, in the following equation.

$$\frac{dx(t)}{dt} = A(t)x(t) + f(t) \tag{104}$$

Besides solving systems of equations, the algorithm has many applications particularly in the field of machine learning, from data fitting, such as in [372, 321], or support vector machines algorithm [303].

3.6 Summary

In this chapter two main types of efficient quantum algorithms were studied: the *dynamic ones*, which rely on the ability of quantum computers to approximate *efficiently* a wide class of Hamiltonians, i.e. from local Hamiltonians to d-sparse ones, as well a wide class of search and sampling problems which may be reduced to them. Furthermore, another important branch of efficient algorithms is given by the Fourier transform and its derivates, which encompass, for instance, the Shor algorithm and HHL, which were briefly discussed in this chapter. From the analysis we concluded that:

- All sampling problems that can be reduced to the simulation of local, sparse or d-sparse Hamiltonian possess efficient quantum algorithms;
- All search problems that *cannot* be reduced to finding a ground-state, involving only local components with less than dimension 2, does not have an efficient quantum algorithm.

We applied this finding in a conception of a case study for the simulation of a local Hamiltonian: the simulation of non-radiative electronic transfer existing in the preliminary state of photosynthesis. It was also studied the impact of the environment in the quantum process of photosynthesis,, where it was concluded that it may, indeed, play an important role in the process, by enforcing the non-reversibility of the process, by killing interference. The results also bring insight about the quantum mechanical concepts introduced in chapter 2.

4

QUANTUM OPTIMIZATION AND QUANTUM CHEMISTRY

> "What we know is a drop, what we don't know is an ocean."
>
> Sir Isaac Newton.

The human understanding of reality is limited by intractable problems from all perspectives: from down below, by the *daunting* complexity of many-body problems in physics, the intractability of molecular problems, arising in medicine and biology and the unpredictability of geological events, such as *earthquakes*; from the upper side by the inability of predict the emergent behaviour of the individual actions taken by a large number of individuals, either in cooperation, or in competition, vital in areas such as economics, finance, or sociology.

Many of such problems are computable, but completely inaccessible to classical computers, which despite the advancements of science, have fallen (very) short to solve many of them. With the upcoming era of quantum computation, one might ask if quantum computers can make a definite contribution to any of those issues, help shedding light to some of the hardest questions in human understanding. The answer to this question, seems to be that it is not that so, i.e. as quantum computers cannot solve efficiently most of the problems already inaccessible to classical computers. However, some advantage, which can be quite significant is some of them, may be expected, as shown by recent results in complexity theory. A particular example of this is quantum chemistry, which rephrases chemistry in purely quantum mechanical terms, allowing a better understanding of the role of quantum effects at chemical level [242]. Here, a huge advantage of quantum computation methods, in comparison with classical *ab-initio* ones, is expected. The application of nowadays short-term devices to these kinds of problems, particularly in problems with high-societal impact, as in industry, finance, and quantum science, have raised a lot of interest from the research and industrial communities, and in this chapter, we discuss some of the techniques employed.

The calculation of ground-state of quantum physical systems, i.e. more generally the calculation of their *points* of equilibrium, can be understood as *very complex* optimization problems, for which many *optimization* problems can be reduced to. The main contribution of this chapter lies along these lines, in

the field of quantum chemistry, and consists in the conception of the calculation of the ground-state of small molecules, *hydrogen* H_2 and *lithium-hydride* LiH, under the action of *stationary* electrical field using *variational methods*, which, to the best of our knowledge, was not directly addressed in the literature before. The work done in the sequel involved the application of the VQE method in the IBM Q experiment and was published in [343]. A relevant part of the contribution were the calculations made for the matrix elements of the Fermionic operators, available in appendix A.

4.1 Search, constraint satisfaction and optimization

Search problems, i.e. problems concerned with finding a solution that matches a certain *criteria* from a wide set of possible solutions, are pervasive in computer science, and *span* all the complexity hierarchy, i.e. some of them are computationally easy to solve, while some others are really hard, for which no *efficient solution* is likely to appear in the near future, even with quantum computers. Arguably, every computational problem can be *phrased* as some sort of search problem, i.e. *reducible* to the *extensive* evaluation of all the possible cases that compose it.

Particular examples of search problems, that are found very frequently in practice and are, in general, hard, are the ones of *constraint satisfaction* and *optimization*. The latter reads as in definition 4.1.1.

Definition 4.1.1. (Optimization Problem)

$$\begin{aligned} &\text{minimize: } f_o(x) \\ &\text{subject to: } f_i(x) \leq b_i, i = 1, \dots, m. \end{aligned}$$

where x is a vector of variables, f_o is a valuation function and $f_i(x) \leq b_i$ are restrictions over variables.

In fact, *optimization* problems can be seen as generalization of *constrain satisfaction* ones, i.e. in the latter, all solutions that respect the constraints are equally valid, while in the former, the idea is to find the solutions that minimize/maximize a cost function, among those which respect the constraints. Examples of *constraint satisfaction* problems are the SAT problems, regarding the satisfactions of logical propositions by Boolean variables.

There is a very rich relationship between constraint satisfaction problems and physical problems, laid, for instance, by Barahona et al. in 1982 [46], where it was discovered that finding the minimum level of energy (ground-state) of an Ising system, was NP-complete. This way it was possible to assign a physical meaning to NP-completeness, i.e. the costs of the physical processes that could be used for that calculation. With the advent of quantum computation, it supplied the foundations for the reduction the classical NP-complete problems to the ground state of the Ising Hamiltonians [257], in which a computational advantage is expected. However, this advantage is not expected to make the resolution of such problems efficient [56].

This connection between theoretical physics and computation, which now falls under the field of *Hamiltonian complexity*, has progressed, and Kitaev introduced the *k-local Hamiltonian* problem, as the quantum generalization of classical constraint satisfaction problems [226], to which many optimization problems can be reduced. The problem consists of finding the *ground-state* under the action of a local-Hamiltonian which is in generally known to be very hard [117], i.e. while the *soft simulation* of systems governed by local Hamiltonians, which includes the sampling, calculation of expected eigenvalues and etc., is *efficient* (see section 3.2.1), calculating the ground-state of such systems is not.

In fact, this problem is known to be characteristic of the Quantum Merlin-Arthur class [76, 223], which is the analogue of the classical NP-class, containing problems believed to be hard even for quantum computers, i.e. just as the NP class contains problem hard for classical computers. However, the k-local Hamiltonian is also an optimization problem, i.e. finding the ground-state of a quantum system is to find the wave function that minimizes the energy of a system and a possible strategy of resolution of classical optimization problems is exactly to reduce them to a k-local Hamiltonian problem.

The most natural form under which *hard* computational problems appear in industry and science is as optimization problems. Hence in order to have a better understanding of how quantum computational techniques apply to these problems, in this section we briefly review the background of the field (further information can be obtained in the following references: [180, 64]).

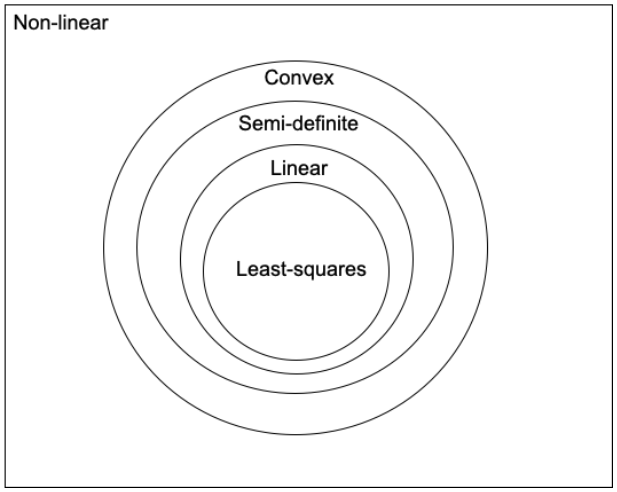


Figure 15: Hierarchy of continuous optimization problems.

A first division between optimization problems regards the type of variables, which define the type of program, e.g. if x is a vector of variables, and n an integer denoting the number of variables, the following classes popup:

- *Continuous:* $x \in \mathbb{R}^n$
- *Mixed integer:* $x \in \mathbb{R}^n \times \mathbb{Z}^n$;

- *Binary optimization:* $x \in \{0,1\}^n$;
- *Integer programming:* $x \in \mathbb{Z}_0^+$.

A further division of optimization problems concerns the type of functions and restrictions to optimize, which can be hierarchically organized, in terms of *expressiveness*, as follows (also presented in in figure 15): *least-squares*, *linear*, *semi-definite*, *convex* and *non-linear*.

Linear programs are the most common form of optimization problems found in practice, and for *continuous variables*, there are efficient algorithms that can find the *optimal solution*. On the other hand, for the discrete cases, i.e. binary, or integer variables, the worst-case is NP-HARD. In quantum computing a relevant class of optimization problems is the one of *semi-definite* programs (SDP's), which correspond to a generalization of linear programs, where restrictions can be expressed as matrices.

Definition 4.1.2. (Semi-definite program)

$$\begin{aligned} &\text{minimize } Tr(CX) \\ &\text{subject to } Tr(A_i X) \leq b_i, i = 1, \dots m. \end{aligned}$$

where X is a $n \times n$ Hermitian matrix defining the variables, so as C, which defines the objective function. The A_i's are also $n \times n$ Hermitian matrices and define program restrictions.

Linear programs correspond to a particular case of an SDP, where all matrices involved are diagonal. A convex program is even a more general than linear program, or an SDP program, where both the cost function and the restrictions are *convex*.

Definition 4.1.3. (Convex Program)
Is an optimization program where the function f_0 specifies the objective function, and f_i and b_i specify the restrictions, all of them being convex functions, i.e. satisfying

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y) \tag{105}$$

Convex problems have the important property that every *local minimum*, corresponds to the *global minima*, and for this reason, there are *polynomial* solutions for these problems, at least for the *continuous form*, based, for instance, in gradient descent methods [78].

Optimization programs involving discrete variables are, in general, harder than continuous ones, and actually, the latter ones usually correspond to *relaxations* of the former. An important sub-type of discrete optimization problems is combinatory optimization, where the solution space is given by the *powerset* of the set of possible components of a solution, e.g. all possible sets of edges in a graph, induced by the set of all possible assignments of binary variables $x \in \{0,1\}^n$, e.g. taking the example of edges in a graph,

0 determines an edge is not present, 1 determines its presence and a string is a possible solution of the problems.

Definition 4.1.4. (Combinatory Optimization)

$$\text{minimize } {}_{S\subseteq N}\{\sum_{j\in S} c_j : S \in \mathcal{P}\}$$

where $\mathcal{P}$ is the set of possible solutions, S is a solution, j a component of the solution and c_j its cost.

Moreover, a particular important class of binary optimization programs, is the quadratic unconstrained binary optimization (QUBO) one, particularly important, due to its proximity with finding the ground-state of the Ising Hamiltonian:

Definition 4.1.5. (Quadratic unconstrained binary optimization (QUBO))

$$\text{minimize } c^\top x + x^\top Q x$$
$$\text{subject to } x \in \{0,1\}^n, \text{ with } c \in \mathbb{R}^n, Q \in \mathbb{R}^{n\times n}$$

A nice survey about QUBO problems can be found in the work of [232]. Furthermore, a relevant reference about integer programming is given in [197].

4.2 Optimization using quantum computers

The seek for optimization algorithms has been very active in the last few years, and there exist many different approaches to tackle optimization problems using quantum computers. As presented in figure 16, such techniques can be organized in three main *families* of techniques: *universal algorithms*, i.e. circuit-based algorithms designed for universal computers, which exhibit polynomial quantum advantages to their classical counterparts; *quantum annealers*, i.e. techniques based in the annealing and adiabatic computing models; Variational methods, which are targeted to short-time devices, and mostly are classical optimization algorithms taking advantage of the exponential gain in Hamiltonian sampling. Throughout the following sections these notions are discussed with more detail.

4.2.1 *Universal quantum algorithms*

There are efficient quantum algorithms to solve least-squares, linear and *semi-definite* programs, up to *convex programs*, with at least polynomial advantage. The algorithm for the least-squares problem is based

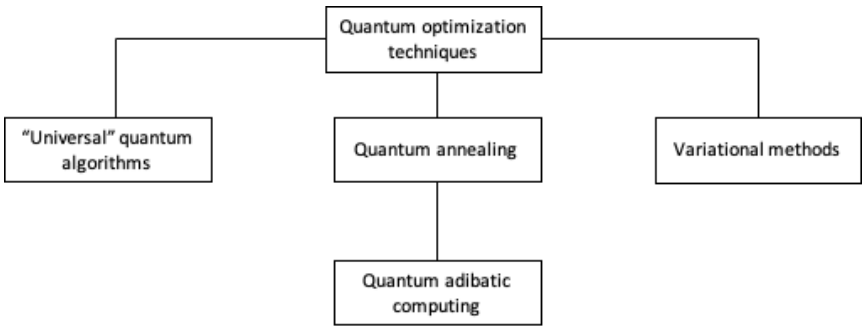


Figure 16: Types of approaches to optimization problems

on the HHL algorithm [372], where an exponential advantage to classical approaches can be obtained, despite the existence of *polynomial* classical algorithms to this kind of problems.

Another very relevant line of work is the one started by Brandao and Svore [79] providing an algorithm for *Semi-definite programs*, yielding a *polynomial advantage* to *linear programs*, based on several geometrical parameters. This line of work was also pursued in [349, 216, 348].

There are also some results within this field, concerning more general convex optimization problems, which contrary to *least-squares* and *SDP* problems, have no natural quantum structure associated. However, an algorithm with a quadratic improvement to the classical *state of the art* algorithm is available [350, 92].

In an even more general setting, shall be considered the work on *quantum gradient descent algorithms* [172, 305], which can be applied to any optimization problem, including *non-convex* and *non-linear* optimization [92], in some cases with polynomial quantum advantage.

In a recent line of work, it has also started to be explored the application of the Grover Adaptive algorithm in optimization problems [171, 48].

4.2.2 *Quantum adiabatic computing and quantum annealing*

The quantum computational technique of *quantum annealing* was firstly introduced in 1989 by Apolloni et al. [27], being one of the first quantum computational techniques conceived. In this work it was established a correspondence between the Schrödinger operator, $\frac{d^2}{dx^2} + V(x)$ on a semi-classical regime, and a function to be *minimized*, corresponding to a combinatorial optimization problem, by modeling the latter on the p-otential $V(x)$ of the former. In this setting, the local minima correspond to the points of lower potential, and, physically, the *wave function* will yield greater probabilities at such points. However due to the, exclusively quantum, *tunneling effects*, it also becomes possible that particles break *potential barriers* and *jump* to the region of other local minima, eventually finding *global minima*. In such work, this is clearly shown by the analysis of the semi-classical dynamics of the ground-state of the system: in one hand it is verified that the equilibrium points (points to which the dynamics tends to) of the dynamics correspond to local minima (zeros of the derivate function), and that *quantum tunneling* allows the jump between

equilibrium points. It can also be shown that if enough time passes the dynamics tends to go to the *global minima*. Hence, quantum annealing can be understood as the quantum counterpart of the classical method of *simulating annealing*, where thermodynamical oscillations guarantee the escape from local minima, and if enough time passes the system will converge to the global minimum.

However, it is believed that this method produces a quantum advantage, by converging more quickly than simulate annealing to global minimum, but most importantly, by doing so in semi-classical regimes, i.e. being more tolerant to environmental noise. The idea has already yielded several short-term devices as documented in [152, 215, 125].

Moreover, adiabatic quantum computing, was firstly proposed by [147], and can be understood as a particular case of *quantum annealing*, applicable when the *environmental conditions* (i.e. temperature and dissipative conditions), are compatible with the *adiabatic theorem*, i.e. allow for an *adiabatic transition* where no exchange of information with the environment happens [222].

The main idea of the technique is to prepare the system in the *ground state* of an easy to prepare Hamiltonian (H_I), and then, *slowly* introduce the action of a second Hamiltonian (problem Hamiltonian), in a perturbative way, in the system, so that in the end the system is encountered in the ground state of the sum of Hamiltonians. The total Hamiltonian of the system shall read as:

$$H = H_I(t_T) + H_P(1 - t/T) \tag{106}$$

If the transition occurs *slowly enough* (large T), the global system will stay in the *ground-state*, which will be equal, in the end of the transition, to the ground state of Hamiltonian H_P. Hence, all that is necessary is that the solution of the problem P, coincides with the global *ground state* of Hamiltonian H_P. There are many methods of doing this mapping, and adiabatic computing, despite being a universal computational method [18], is particularly suited for solving optimization problems [148].

These techniques find many fields of application, such as in traditional computer science, e.g. search engines [165], or even board games [344], or in a large spectrum of industrial applications. The latter includes pharmaceutics [285]), or a multitude of problems in finance, e.g. *portfolio optimization* [312], risk analysis [375], or the prediction of financial crisis [278]. A good review on the applications of quantum computers in finance is available in reference [279]. Finally, the spectra of applications encompass traditional areas of optimization, such as *planning and scheduling problems* [309, 357], instantiated in fields such as health [202] or logistics [135]. The quantum advantage of quantum annealers is yet under discussion, as while it may not be possible to obtain an *exponential* advantage in *exact* solutions of optimization problems, it may be possible to obtain advantages in *approximate* versions [315].

4.2.3 *Variational methods*

Quantum variational methods are hybrid quantum-classical approximation algorithms, targeted at optimization problems [268], which can be applied in traditional optimization fields, such as combinatorial optimization, as well as in quantum physical problems, such as finding the ground-state of quantum physical systems [149, 146]. The former ones are, however, particular cases of the latter, i.e. the variational methods can be applied to classical optimization problems, by finding a QUBO formulation for them, which can be trivially mapped onto an Ising Hamiltonian (there are many software tools to aid in this translation available, for instance, in the qiskit platform [115]), over which the variational method is used, to obtain its ground-state. Recently, these methods have started to be used, also, in *mixed-integer* optimization problems, by the use of decomposition techniques [164].

Moreover, variational methods have also been used to obtain *short-term* versions of the most popular quantum algorithms, such as the Grover [269], or Factoring [26], although the quantum advantage is less clear in these settings. This class of methods became a cornerstone of nowadays quantum computation, due to relationship between the amount of quantum resources required and the quantum advantage obtained (more on this in section 4.2.5), i.e. low resources are required to obtain quantum advantage, turning devices based on this technique, good candidates to *quantum supremacy* and proof-of-concept for industrial and quantum physics optimization problems.

4.2.4 *The Variational Quantum Eigensolver*

The *Variational Quantum Eigensolver* (VQE) method, is a *variational method*, used to estimate the lowest eigenvalue (the ground state energy) of a Hamiltonian, introduced by [286], which has been gaining relevance in recent literature on quantum computation, through its application to Hamiltonian *ground-states* search, and general optimization tasks – see, e.g., [267].

The method is a quantum version of the variational method for the calculation of the ground state energy, extensively used in Physics (also known as the Rayleigh-Ritz method) and has also been widely used for a long time in Quantum Chemistry – see, e.g., [242].

$$E[\Psi(\vec{\theta})] = \frac{\langle \Psi(\vec{\theta}) | H | \Psi(\vec{\theta}) \rangle}{\langle \Psi(\vec{\theta}) | \Psi(\vec{\theta}) \rangle} . \tag{107}$$

The optimization consists of the determination of the set of parameters $\vec{\theta}$ that minimize the function $E[\Psi(\vec{\theta})]$ function, defined by equation (107), consisting of the expectation value of the action of the *Hamiltonian operator*. The VQE method is an iterative method, wherein each iteration a quantum state,

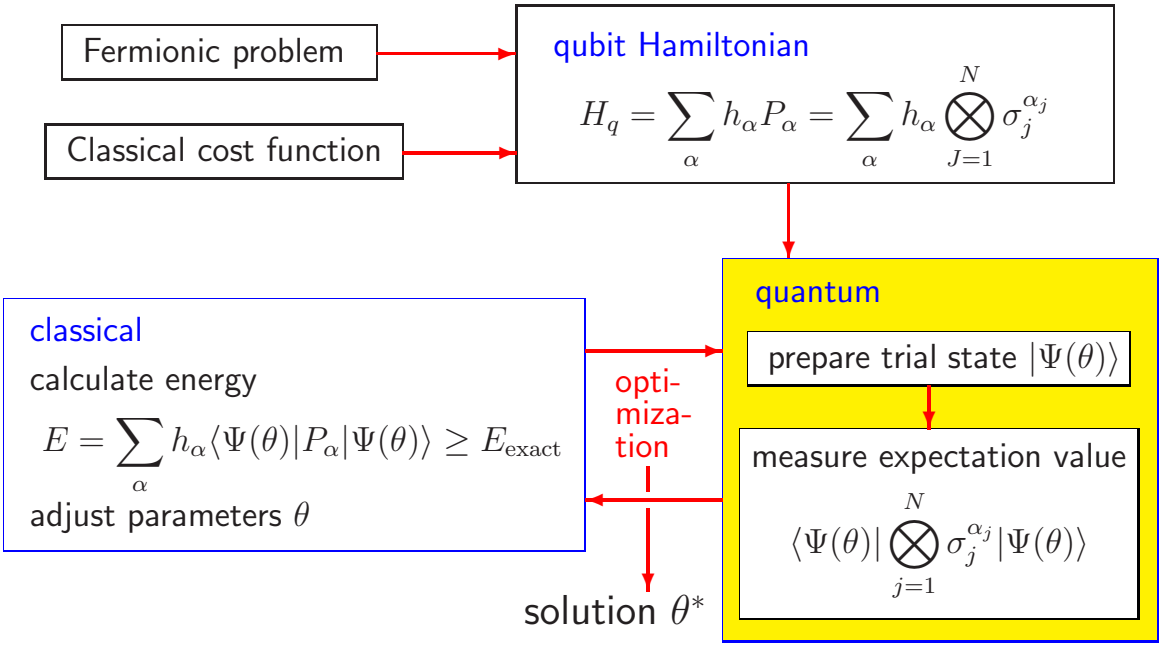


Figure 17: Application of the variational method to fermionic problems, adapted from [267].

corresponding to a parameterized trial function $\Psi(\vec{\theta})$ (see section 4.3.5 for an example), is prepared and the expected eigenvalue with respect to the system's Hamiltonian is calculated.

Then a classically implemented algorithm updates the parameters $\vec{\theta} \in \mathbb{R}^n$ of the quantum state using a classical optimization routine. The previous step is repeated until some convergence criteria (e.g., in energy and/or iteration number) are satisfied. Any optimization method able to perform this task can, in principle, be used. On IBM Q [115], a few methods are available for this purpose. For instance, the *Simultaneous Perturbation Stochastic Approximation Algorithm* [SPSA, see 65], which is characterized by a very good performance under noise, or the Cobyla method [294].

The scheme of the method is depicted in Fig. 17, adapted from the latter work. Even if the optimization part is mostly classical a quantum advantage is, still, obtained, as further discussed in section, 4.2.5. A good additional discussion of this method can be found in [264].

4.2.5 *The quantum advantage of the VQE method*

Each iteration in the VQE method, requires the evaluation of the action of the *Hamiltonian* over the *ansatz*, i.e. the estimation of the eigenvalue for the current *ansatz*, which corresponds to the *cost* function. As it is well-known, this can be done, with an exponential advantage, by the use of the *quantum phase estimation* (QPE) algorithm [11], which has also several other applications, such as in the resolution of linear equations [191].

The QPE method requires an approximation of the evolution operator, $\hat{U} = \exp\{(-iHt\})$ (t is time), and its application to the initial state an appropriate number of times. For an eigenstate, the application of $\hat{U}$ results in adding a phase $(-Et)$, so that the energy eigenvalue E can be estimated. Unfortunately, despite its theoretical attractivity and a broad scope of possible applications, it poses serious technical difficulties, which makes its practical realization unlikely at the present level of maturity of quantum computers: it requires a very large number of entangled qubits and quantum gates to be effective.

Alternatively, one can adopt a strategy of applying the Hamiltonian over a state several times, measuring the result (i.e., performing the *quantum sampling*), to obtain an estimation of the expected *eigenvalue*, for which effective algorithms are available, particularly the *Quantum Expected Eigenvalue Estimation* (QEE) method. The method requires that the Hamiltonian operator can be decomposed into *polynomial* (M) independent n-qubit operators and consists of the "measurement" of the expectation values of such operators for a trial state $|\Psi\rangle$ (also known as the *ansatz*):

$$\begin{aligned} \langle H|H\rangle &= \langle\Psi|H|\Psi\rangle \\ &= \sum_{i;q} h_q^i \left\langle \sigma_i^{(q)} \middle| \sigma_i^{(q)} \right\rangle + \sum_{\substack{i_1,i_2;\\ q_1,q_2}} h_{q_1,q_2}^{i_1,i_2} \left\langle \sigma_{i_1}^{(q_1)} \otimes \sigma_{i_2}^{(q_2)} \middle| \sigma_{i_1}^{(q_1)} \otimes \sigma_{i_2}^{(q_2)} \right\rangle + \cdots \end{aligned} \quad (108)$$

The estimation of the expectation values, $\langle\cdots\rangle$, requires repeated measurements on the polynomial number of independent terms. An objective comparison of the QPE and QEE methods is presented by [264] and summarized in Table 8.

Table 8: Comparison of resources needed for two methods, QPE and QEE. M: the number of independent terms of the Hamiltonian approximation, p: the precision chosen, $O(...)$: asymptotic lower bound of the associated resource function. See text for details.

Method	Number of state preparations	Coherence time	Number of steps
QEE	$O(M)$	$O(1)$	$O(\|h_{max}\|^2 M p^{-2})$
QPE	$O(1)$	$O(p^{-1})$	$O(p^{-1})$

A main advantage of the QEE, when compared with QPE, is that it largely reduces the need for gates, but, more important, the amount of time the *entanglement* over sets of qubits has to be maintained, i.e. the *coherence time*, is $O(1)$ (*independent of precision, p*), which is within the grasp of existing quantum computers, while it grows *linearly* with p, $O(p^{-1})$, for QPE. However, QEE introduces the need to prepare more copies of the *ansatz* to maintain the independence of the terms in Eq. (108) – $O(M)$ against $O(1)$ for QPE – requiring polynomially more memory, i.e. qubits. Furthermore, for the desired precision p, the number of necessary *sampling* steps is $O(|h_{max}|^2 M p^{-2})$, where h_{max} is the term with the maximum norm in the decomposition of the Hamiltonian. In summary, the QEE method reduces the required minimum

coherence in the QPE, while preserving the significant quantum advantage in comparison to classical methods, introducing however a polynomial penalty, both in time and memory, i.e. number of qubits.

4.3 Case study: Calculation of the ground–state Stark effect in small molecules

The work of LLoyd et al. [248] on the simulation of *local* Hamiltonians, initiated the *modern* era of quantum simulation research, by providing a realizable quantum process to the idea introduced by Feynman in 1982 [150]. The path from that point to the one in which it will be possible to simulate quantum chemistry systems, includes the independent works of Wiesner et al. [373], and Zalka et al. [382], and the work of Ortiz et al. [277], on the simulation of Fermionic Hamiltonians. Jointly, they provide the foundation for the simulation of quantum chemistry.

Since then the number of quantum algorithms in quantum chemistry has grown exponentially, making it one of the disciplines where quantum computation can have more impact. From the technical point of view, a very relevant issue to the simulation of quantum chemistry systems, as seen in section 3.2, is to be able to efficiently approximate Hamiltonians for quantum chemistry using quantum gates and circuits. The processes for doing this have greatly improved throughout the years both in terms of the resources required [366, 192, 293, 33] and in terms of the Hamiltonian representation chosen [35, 34, 256], with an exponential advantage in many cases.

The process of quantum simulation aims at the calculation of relevant properties, as for instance what was achieved in the seminal works of Lidar [245], on the calculation of thermal rates in chemical reactions, and Aspuru-Guzik et al. [32] on the calculation of ground states of simple molecules. These two works are also characteristic examples of the two types of properties of interest in quantum chemistry, the *dynamic* ones, which concern aspects of the evolution of the system, e.g. chemical reactions, and the static ones, which concern properties about the eigenvalues. Examples of the former can be found, for instance, in works on the process of nitrogen fixation [221, 308], and of the latter in the characterization of molecular *energy spectra* [370], useful, for instance, to the optimization of molecular geometries [220]. A particular important *static property* is the calculation of the minimum eigenvalues, which can be calculated in many ways, including modified versions of phase estimation, the so-called *iterative phase estimation* [280, 238].

More recently, the so-called *short-term* devices started to appear, and quickly gained relevance in the landscape of quantum simulation of chemistry research, namely the development of specific methods targeted to this kind of devices, e.g. based on the VQE method. On the field, these methods are already extensively studied, both from the theoretical and experimental points of view (see [89] for an excellent review on the subject). The state of the art on this particular subject encompasses the calculation of the ground state of small molecules, namely, H_2, LiH, BeH_2 [218], or the *charged* HeH^+ [325, 286], in physical systems possessing 2 to 6 physical qubits. Still on this field with small quantum devices, one shall

also highlight the very recent breakthrough of [299], on the simulation of the *isomerization of diazene*, using chains of H_2 molecules, on physical devices with 12 physical qubits.

The hydrogen molecule is the simplest one existing in nature, and the LiH molecule is just a bit more complex than H_2, lacking its *mirror symmetry*, which makes the calculation slightly more complicated. Both have been the natural test case for experimental and theoretical research, particularly concerned by the calculation of their ground state properties and the dissociation curves, which have recently been recalculated using advanced classical [361] and quantum [110] algorithms (the latter with extension to excited states).

The following sections are focused on the exploration of the simulations of such two molecules *hydrogen* (H_2) and *lithium hydride (LiH)*, under the action of strong stationary electric fields (Stark effect) [181], using the commercially available quantum computer, the IBM Q, accessed through the QuantaLab UMinho Academic Q Hub, and programmed using the QISKit platform [115]. To the best of our knowledge, it has not been studied directly in a NISQ machine and constitutes the main contribution of this work.

We aim also at revisiting the necessary steps, as well as the relevant theory, for the construction of a quantum simulation for a small molecule using the VQE method, explored in section 4.2.4, as it poses many conceptual and practical challenges, far from trivial [370]. The VQE method, is in some sort, an algorithmic framework, requiring the following input:

- A qubit Hamiltonian, which requires the calculation of the matrix elements of the fermionic Hamiltonian, and an appropriate mapping of the resultant matrix onto qubit Hamiltonians;
- The choice of an appropriate *ansatz* and its implementation on quantum circuits;
- The choice of a classical *optimization* method and the application of the VQE routine.

Hence, over the next sections, we will discuss the relevant theory for these quantum simulations, namely, the Quantum Hamiltonian formalism for many-body systems, the Hartree-Fock approximation and the second quantization representation. A good introduction to the subject is offered, for instance, by [242] and [337]. The necessary items to the application of the VQE method are also reviewed, bearing in mind that the VQE method itself was already explored with detail in 4.2.4. The actual calculation of the specific matrix elements is extensive, but it is available in the associated publication [343] and in appendix A. We will also explain the mapping onto a system of qubits and the design of the quantum circuit corresponding to the initial Hamiltonian, and the execution of the VQE method to this case study, so as the correspondent results obtained.

4.3.1 *Many-particle systems*

The Schrödinger's equation for a system of non-interacting particles can be decomposed into a set of uncoupled equations for each particle and the system's WF suitably factorized. A combination of two non-interacting and non-entangled systems can be described by applying the tensor product to the two vector spaces,[1] with resultant basis given as follows:

$$\begin{aligned} |\Psi^{(1)}\rangle \otimes |\Psi^{(2)}\rangle &= \sum_{\alpha}^{M_1} \sum_{\beta}^{M_2} \lambda_\alpha \mu_\beta |\Psi_\alpha^{(1)}\rangle \otimes |\Psi_\beta^{(2)}\rangle \\ &= \sum_{\alpha}^{M_1} \sum_{\beta}^{M_2} \lambda_\alpha \mu_\beta |\Psi_\alpha^{(1)} \Psi_\beta^{(2)}\rangle \,. \end{aligned} \tag{109}$$

In Eq. (109), $\Psi_\alpha^{(s)}$ denotes an eigenfunction of a state $\alpha = 1, \ldots, M_s$ of the system $\Psi^{(s)}$ ($s = 1, 2$). The dimension of the product vector is $\dim(\Psi^{(1)}) * \dim(\Psi^{(2)}) = M_1 \cdot M_2$.

When the particles constituting the system are *identical*, their *spin* becomes highly relevant. The spin, which is an intrinsic angular momentum of the particle, distinguishes between two different types of particles, bosons (e.g. photons) and fermions (e.g. electrons and protons). For fermions, the Pauli exclusion principle states that the system's WF must be antisymmetric with respect to the permutation of any two particles. This entails an important restriction upon the WF, namely that the product vector (109), if applied to a pair of non-interacting electrons, is not compatible with the Pauli principle.

In Quantum Chemistry, a single-electron WF is called orbital [337]. One can distinguish *spatial orbitals* $\phi(\mathbf{r})$, where r corresponds to spatial coordinates, and *spin orbitals* $\chi(\mathbf{x})$, where $\mathbf{x} = (\mathbf{r}; s)$ and $s \in \{\uparrow, \downarrow\}$ stands for two possible orientations of electron's spin. For two electrons, the Pauli principle means that

$$\chi(\mathbf{x}_1, \mathbf{x}_2) = -\chi(\mathbf{x}_2, \mathbf{x}_1) \tag{110}$$

or, equivalently,

$$\phi(\mathbf{r}_1, \mathbf{r}_2) = \mp \phi(\mathbf{r}_2, \mathbf{r}_1)\,, \tag{111}$$

where the upper (lower) sign corresponds to parallel (antiparallel) spins of the two electrons. If the electron-electron interaction is neglected, the correct (i.e. compatible with the Pauli principle) two-electron WF is written in the form of the so-called *Slater determinant*,

$$|\chi_\alpha^{(1)}\, \chi_\beta^{(2)}\rangle = \frac{1}{\sqrt{2}} \begin{vmatrix} \chi_\alpha(\mathbf{x}_1) & \chi_\beta(\mathbf{x}_1) \\ \chi_\alpha(\mathbf{x}_2) & \chi_\beta(\mathbf{x}_2) \end{vmatrix}\,, \tag{112}$$

1 For interacting or entangled systems, the total WF cannot be written as a product ofits parts. Entangled parts of a system, even if they do not interact physically, may not be described by a wave function, only represented by a density matrix.

where $\chi_\alpha(\mathbf{x})$ and $\chi_\beta(\mathbf{x})$ designate different spin orbitals. A Slater determinant can be straightforwardly generalized towards the case of N identical non-interacting particles. It vanishes when any two electrons "occupy" the same spin orbital, as required by the Pauli exclusion principle.

The Slater determinant is a simple way of constructing a many-electron WF from spin orbitals representing non-interacting electrons. Complete neglection of the Coulomb interaction between the electrons would be too crude an approximation, while solving directly the many-electron Schrödinger equation is an intractable problem. A compromise is achieved by a self-consistent field method also called Hartree-Fock (HF) approximation. An effective one-electron operator is introduced, $v^{HF}(\mathbf{x})$, called the Fock operator, which includes, as a part of the single electron potential energy, the electron's interaction with all other electrons whose positions are averaged under an assumption that the WF representing the system of N electrons is a single Slater determinant. An explicit expression for $v^{HF}(\mathbf{x})$ will be presented below.

4.3.2 *Molecular Hamiltonian and Hartree-Fock approximation*

The general form of a molecular Hamiltonian is (in atomic units):

$$H_{\text{mol}} = -\sum_{i=1}^{N}\frac{1}{2}\nabla_i^2 - \sum_{A=1}^{M}\frac{1}{2M_A}\nabla_A^2 - \sum_{i=1}^{N}\sum_{A=1}^{M}\frac{Z_A}{r_{iA}} + \sum_{i=1}^{N}\sum_{j>i}^{N}\frac{1}{r_{ij}} + \sum_{A=1}^{M}\sum_{B>A}^{M}\frac{Z_A Z_B}{r_{BA}}. \tag{113}$$

The first and second terms of (113) correspond to the kinetic energy of the electrons (numbered by i and $j = 1,\dots,N$) and nuclei (numbered by $A = 1,\dots,M$), respectively. The third one represents the Coulomb attraction of each electron to each nucleus with r_{iA} being the electron-nucleus distance and Z_A the nucleus charge. Finally, the fourth and fifth terms correspond to the repulsion among the electrons and the nuclei, respectively. It is common and well justified to use the Born-Oppenheimer approximation, which neglects the motion of the nuclei because they are much heavier than electrons, whereby the potential energy of the nucleus-nucleus interactions becomes a constant (for fixed placement of the nuclei) hence a parameter for the electron problem. With this, the electron Hamiltonian (113) reduces to:

$$H_{el} = -\sum_{i=1}^{N}\frac{1}{2}\nabla_i^2 - \sum_{i=1}^{N}\frac{Z_A}{r_{iA}} + \sum_{i=1}^{N}\sum_{j>i}^{N}\frac{1}{r_{ij}}. \tag{114}$$

For the H_2 molecule, the Hamiltonian (114) depends on a single parameter, the distance between the protons d. If the lowest eigenvalue of (114), $E_0(d) < 0$, is larger in absolute value than the proton-proton repulsion energy, $E_{rep}(d) = d^{-1}$, the molecule is bound, as illustrated in Fig. 18.

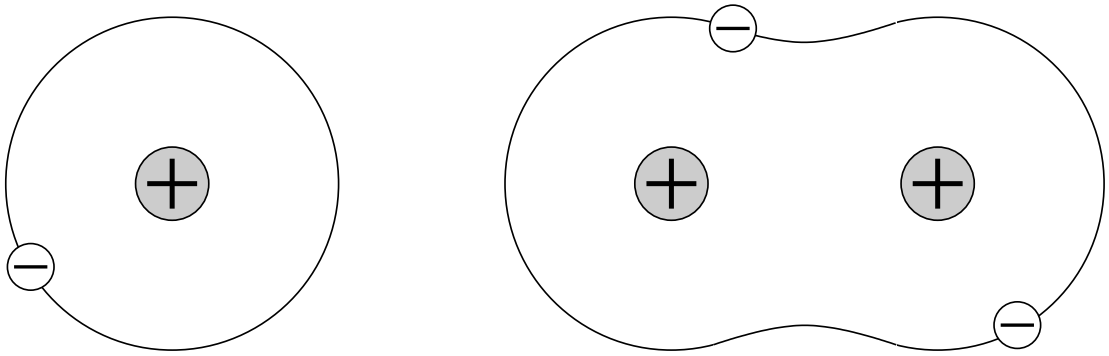


Figure 18: Left: the hydrogen atom consists of a single electron and a proton and has the energy -0.5 a.u. in the ground state. Right: in the hydrogen molecule H_2, made of two nuclei and two electrons, the total energy can be lower than -1 a.u., which makes the molecule stable.

The Hamiltonian (114) has to be reduced to a single-electron one in order to proceed with the determination of its eigenvalues, which is achieved by means of the HF approximation, where one takes an average over the positions and spins of all but one (to be labelled by $i = 1$) electron. This is done by multiplying equation (114) by $|\chi_\alpha^{(1)}\chi_\beta^{(2)}\ldots\chi_\gamma^{(N)}\rangle$ and the corresponding "bra", both in the form of Slater determinants of dimension N (the number of electrons in the system), and integrating over $\mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_N$, which leads to

$$\left(-\frac{1}{2}\nabla_1^2 - \sum_{A=1}^{M}\frac{Z_A}{r_{1A}} + v_1^{HF}\right)\chi_\alpha(\mathbf{x}_1) = \epsilon_\alpha\chi_\alpha(\mathbf{x}_1)\,, \tag{115}$$

where v_1^{HF} is the average potential experienced by the "chosen" electron, and ϵ_α is the single-electron energy. The HF potential can be written in the form:

$$\begin{aligned} v_1^{HF} \;=\; & \sum_\beta \int |\chi_\beta(\mathbf{x}_2)|^2 \frac{1}{|r_{12}|}\, d\mathbf{x}_2 \\ & - \frac{\displaystyle\sum_\beta \int \chi_\alpha^*(\mathbf{x}_1)\chi_\beta^*(\mathbf{x}_2)\frac{1}{|r_{12}|}\chi_\beta(\mathbf{x}_1)\chi_\alpha(\mathbf{x}_2)\, d\mathbf{x}_2}{|\chi_\alpha(\mathbf{x}_1)|^2}\,. \end{aligned} \tag{116}$$

The two terms in Eq. (116) are called Coulomb and exchange energies, respectively. The latter poses the main difficulty in solving Eq. (115); however, its neglection (known as the Hartree approximation) results in an unsustainable error. Due to the nonlinearity of the HF approximation, the equations are solved in practice by self-consistent (iterative) methods, using a finite set of spatial basis functions, $\phi_\mu(\mathbf{r})$ ($\mu = 1, 2, \ldots, K$) – see, e.g., [337]. The solution yields a set HF spin orbitals $\{\chi_\alpha\}$ with corresponding energies $\{\epsilon_\alpha\}$, $\alpha = 1, 2, \ldots, 2K$. The number of electrons in the system must be $2K \geq N$. The possibilities to place N electrons over $2K$ spin orbitals gives rise to $(2K)!/(N!(2K-N)!)$ Slater determinants, one of which

represents the ground state of the system and the others correspond to excited states. The HF approximation takes into account the Quantum Mechanical correlation caused by the Pauli principle, however, only of electrons with parallel spins. The difference between the approximate HF energy and the exact energy of the system is known as the *correlation* correction (or energy).

It is common to use, as initial approximation basis sets to represent molecular orbitals (MO) in the HF equations, the linear combinations of atomic orbitals (LCAO). Since the exact atomic orbitals for a given many-electron atom are difficult to construct, the so-called *Slater-type orbitals* (STO) are sometimes used, which are inspired by the (exactly known) radial asymptotics of spatial orbitals of the hydrogen atom,[2]

$$\phi(\mathbf{r}) \sim r^{n-1} e^{-\zeta r} Y_{l,m}(\theta, \varphi)$$

(here $Y_{l,m}$ is a spherical harmonic). For instance, one can use

$$\phi_{1s}^{\mathrm{STO}}(\zeta, \mathbf{r} - \mathbf{R}_{\mathrm{A}}) = \left(\frac{\zeta^3}{\pi}\right)^{\frac{1}{2}} e^{-\zeta |\mathbf{r} - \mathbf{R}_{\mathrm{A}}|}$$

for s-states, where ζ is the Slater orbital exponent. As the STO functions are difficult to handle in many-center integrals, one practical resort consists of approximating these functions with linear combinations of *Gaussian functions*, known as STO-LG functions. The calculation of the necessary matrix elements is then greatly facilitated, because the multi-center integrals with Gaussian functions can be evaluated analytically as discussed in appendix A. In this work, a set of such functions with $n = 3$ Gaussians mimicking each STO function, named STO-3G basis, is used. For the $1s$ state, such a function is

$$\begin{aligned} \phi_{1s}^{\mathrm{STO-3G}}(\zeta, \mathbf{r}) = c_1 \left(\frac{2\alpha_1}{\pi}\right)^{\frac{3}{4}} e^{-\alpha_1 r^2} \\ + c_2 \left(\frac{2\alpha_2}{\pi}\right)^{\frac{3}{4}} e^{-\alpha_2 r^2} + c_3 \left(\frac{2\alpha_3}{\pi}\right)^{\frac{3}{4}} e^{-\alpha_3 r^2}, \end{aligned} \tag{117}$$

where α_i are the Gaussian orbital exponents that have been optimized for the best possible approximation of $\phi_{1s}^{\mathrm{STO}}(\zeta, \mathbf{r})$ for a given ζ [195]. The corresponding spin orbitals, $\chi_\alpha(x)$, are obtained from $\phi_\mu^{\mathrm{STO-3G}}$ by multiplying them with a spinor $\psi(s)$, $s \in \{\uparrow, \downarrow\}$.

4.3.3 *Second quantization*

In the quantum mechanics of systems consisting of a number of identical particles (electrons, in our case), it is common to use the formalism called *second quantization*, originally introduced by P. Dirac – see, e.g.,

2 The STO includes a simple power function of the radius instead of a polynomial, and hence do not possess radial nodes.

[137]. This formalism deals with the whole system of particles, instead of with each particle individually, by introducing a new way of describing states, by the latter's *occupation numbers*. Let $\{\chi_\alpha(\mathbf{x})\}$ be a complete set of one-electron (atomic or molecular) spin orbitals that constitutes the Hilbert space of a single particle. If the particles were non-interacting bosons, a state of the whole system could be entirely specified by indicating the numbers of particles, n_α, occupying each of these orbitals. Such an occupation number state can be designated by a state vector $|n_1, n_2, ...\rangle$. If the particles interact with an external field or with each other (but still assuming that they are bosons and no restrictions are imposed by the particle's spin), the state vector in the occupation number representation will evolve with time, obeying the time-dependent Schrödinger equation (15) with the Hamiltonian written in terms of the occupation numbers

$$H = H_1 + H_2 = \sum_{\alpha,\beta} \tau_{\alpha\beta} a_\alpha^\dagger a_\beta + \frac{1}{2} \sum_{\substack{\alpha,\beta,\\ \gamma,\delta}} \mu_{\alpha\beta\gamma\delta} a_\alpha^\dagger a_\gamma^\dagger a_\delta a_\beta \,. \tag{118}$$

The sum is made over states in the single-particle Hilbert space, e.g., $1s$-, $2p$-like, etc., $\tau_{\alpha\beta}$ being a matrix element of the single-electron energy,

$$\tau_{\alpha\beta} = \int d\mathbf{x}_1 \chi_\alpha^*(\mathbf{x}_1) \left(\frac{-\nabla^2}{2} + \sum_A \frac{Z_A}{|r_{A1}|} \right) \chi_\beta(\mathbf{x}_1) \,. \tag{119}$$

The second term in (118) represents the Coulomb interactions between the particles, with the matrix element given, according to the convention used in Quantum Chemistry [337], by:

$$\mu_{\alpha\beta\gamma\delta} = \int d\mathbf{x}_1 d\mathbf{x}_2 \chi_\alpha^*(\mathbf{x}_1) \chi_\beta(\mathbf{x}_1) \left(\frac{1}{|r_{12}|} \right) \chi_\gamma^*(\mathbf{x}_2) \chi_\delta(\mathbf{x}_2) \,. \tag{120}$$

The integration in Eqs. (119) and (120) is over coordinates (and sum over spins) of one or two electrons labelled 1, 2.

The Hamiltonian (118) is written in terms of so-called *creation*, $a^\dagger$, and *annihilation*, a, operators, which add one particle to (or, remove from) an orbital α, respectively:

$$\begin{aligned} a_\alpha^\dagger |n_1, n_2, ...\rangle &= \sqrt{n_\alpha + 1}\, |n_1, n_2, ...\rangle \,; \\ a_\alpha |n_1, n_2, ...\rangle &= \sqrt{n_\alpha}\, |n_1, n_2, ...\rangle \,. \end{aligned} \tag{121}$$

The product $a_\alpha^\dagger a_\alpha$ is the occupation number operator for the orbital α. In the case of bosons, the creation and annihilation operators for different α and β commute, because different orbitals are filled independently. This is not the case of fermions, because of the Pauli exclusion principle. By virtue of this, the following (anti-commutation) relations hold for the electron operators:

$$a_\alpha\, a_\beta^\dagger + a_\alpha^\dagger\, a_\beta = \delta_{\alpha\beta} \,. \tag{122}$$

It can be shown that (122) guarantees that the occupation numbers can take only values 0 and 1 in accordance with the Pauli principle [137]. Therefore, the Hamiltonian (118) has the same form for bosons and fermions, the only difference being in the (anti-)commutation relations of the creation and annihilation operators. For fermions, each state $|n_1, n_2, ...\rangle$ of this Hamiltonian corresponds to a Slater determinant in the *Fock space* (of dimension $2K$), with the number of columns and rows equal to the number of electrons in the system, $N = \sum_{\alpha=1}^{2K} n_\alpha$.

The choice of single-electron basis functions $\chi_\alpha^*(\mathbf{x})$ is, in principle, arbitrary, but if we "guess" their form close to the "true" WFs of the system (which actually are not well-defined in the single-electron form!), the non-diagonal elements of the matrices $\tau_{\alpha\beta}$ and $\mu_{\alpha\beta\gamma\delta}$ will be much smaller than the diagonal ones. For practical calculations of these integrals, the basis functions are expressed in terms of the STO-3G sets explained in the previous section. The choice of molecular orbitals is based on the MO-LCAO approximation. One can improve this initial approximation by solving first the HF equation (115) and using its solutions to calculate the matrix elements. Then the diagonalization of Eq. (118) amounts to the evaluation of the correlation energy.

In the sequel we also consider the stationary Stark effect described by the following (single-electron) Hamiltonian:

$$H_S = -\mathbf{E} \cdot \mathbf{r}\,, \tag{123}$$

where $\mathbf{E}$ is the electric field intensity. Its second-quantization representation is identical to H_1 in (118), and the corresponding matrix element is written as

$$\tau_{\alpha\beta}^S = \int d\mathbf{x}_1 \chi_\alpha^*(\mathbf{x}_1)\,(-e\,\mathbb{E}\,r\cos\theta)\,\chi_\beta(\mathbf{x}_1)\,, \tag{124}$$

where $\mathbb{E} = |\mathbf{E}|$, and the z-axis is assumed to be directed along $\mathbf{E}$. The use of the second quantization formalism is facilitated, for instance, by the PyQuante [273] and the PySCF [336] tools, two Python libraries targeted to quantum chemistry calculations. We present the matrix elements (119), (120) and (124) calculated for $1s$, $2s$ and $2p_z$ atomic orbitals in appendix A.

4.3.4 *Mapping the fermion Hamiltonian onto a qubit representation*

In order to perform quantum computations, one needs to map the second-quantization Hamiltonian onto a qubit (spin) representation and then design the corresponding quantum circuit that implements it. The basic idea is to replace the fermionic operators a and $a^\dagger$ by tensor products of the Pauli matrices,

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \qquad \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

which can be done in a number of ways, such as the Jordan-Wigner or Bravyi-Kitaev transformations [88]. The former, addressed in this section, is a specific method based on the isomorphism between the creation and annihilation operators and the algebra of the Pauli matrices [370].

In the case of a single (one-electron) state, the Jordan – Wigner (JW) mapping is simple. Following the convention of Fig. 19, common in Physics,

$$a^\dagger \Leftrightarrow \sigma^+ \equiv \frac{1}{2}\left(\sigma_x + i\sigma_y\right) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}; \tag{125}$$

$$a \Leftrightarrow \sigma^- \equiv \frac{1}{2}\left(\sigma_x - i\sigma_y\right) = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}; \tag{126}$$

$$a^\dagger a - \tfrac{1}{2} \Leftrightarrow \frac{1}{2}\sigma_z = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & -\frac{1}{2} \end{bmatrix}. \tag{127}$$

The matrices $\sigma^\pm$ represent the spin-raising and spin-lowering operators, respectively, while σ_z is related to the occupation number operator.

However, usually another convention is used in quantum information, as the computational basis is defined as follows:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

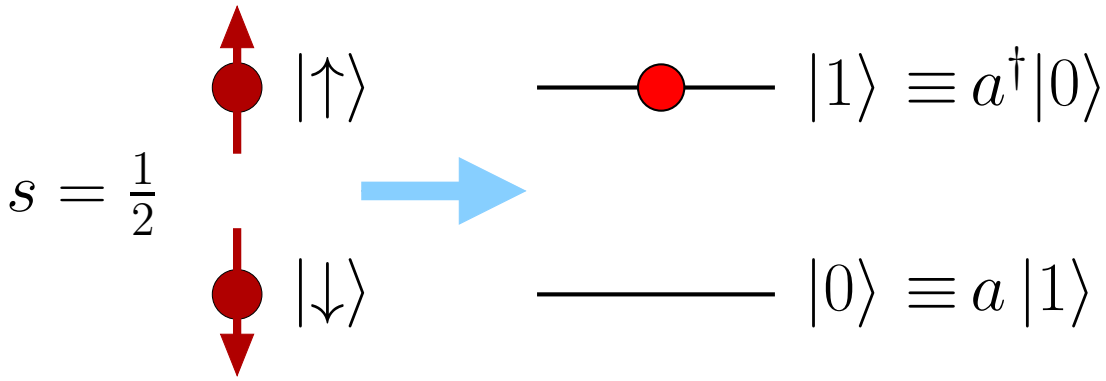


Figure 19: A scheme illustrating the mapping of a fermion onto a qubit. The arrows indicate two spin projections. The two states of the Hamiltonian $\hat{H} = C\hat{\sigma}_z$ (C is a positive real constant) are usually denoted by $|0\rangle$ (lower energy state) and $|1\rangle$.They are connected by the operators a and $a^\dagger$. Considered as spin states, they may be denoted as $\begin{pmatrix} 0 \\ 1 \end{pmatrix} = |0\rangle$ and $\begin{pmatrix} 1 \\ 0 \end{pmatrix} = |1\rangle$, which leads to the correspondence between the a and $a^\dagger$ operators and the Pauli matrices given by Eqs. (125) - (127).

Accordingly,

$$a^{\dagger} \Leftrightarrow \hat{\sigma}^{-} \qquad \text{and} \qquad a \Leftrightarrow \hat{\sigma}^{+} . \tag{128}$$

In case of $N > 1$ fermions, the mapping becomes slightly more complex. In order to satisfy the anti-commutation relations (122) between *any* pair of fermionic operators, one indexes the states by a single index (α) and adds the *string*, i.e. [spin]=[fermion]×[string], taking into account the occupation numbers, n_α, of states with $\beta < \alpha$, for a given α:

$$\sigma_\alpha^{+} \Leftrightarrow a_\alpha e^{i\pi \sum_{\beta<\alpha} n_\beta} , \qquad \sigma_\alpha^{-} \Leftrightarrow a_\alpha^{\dagger} e^{-i\pi \sum_{\beta<\alpha} n_\beta} . \tag{129}$$

The relation (129) holds for multiple fermions and the phase factors (compare to (128)) can be represented by the Pauli matrices, $(\sigma_z)_\beta$, acting on the corresponding fermionic states. Therefore, the fermionic operators are mapped onto direct products of Pauli matrices as follows:

$$\begin{aligned} a_\alpha \quad &\Leftrightarrow \quad \mathbf{1}^{\otimes(\alpha-1)} \otimes (\sigma^{+})_\alpha \otimes (\sigma_z)^{\otimes(N-\alpha)} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{\otimes(\alpha-1)} \otimes \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}_\alpha \otimes \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}^{\otimes(N-\alpha)} ; \end{aligned} \tag{130}$$

$$\begin{aligned} a_\alpha^{\dagger} \quad &\Leftrightarrow \quad \mathbf{1}^{\otimes(\alpha-1)} \otimes (\sigma^{-})_\alpha \otimes (\sigma_z)^{\otimes(N-\alpha)} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{\otimes(\alpha-1)} \otimes \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}_\alpha \otimes \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}^{\otimes(N-\alpha)} . \end{aligned} \tag{131}$$

Thus, any Hamiltonian operator written in the second quantization representation can be rewritten in terms of the raising and lowering spin operators and the Pauli matrix σ_z. A catalogue of such translations can be found in Table A2 of the work [370]. For a Hilbert space of $2K$ spin orbitals, a system of $2K$ fermions (i.e. qubits) is required for the JW mapping. The resulting qubit Hamiltonian has the following generic form:

$$H = \sum_{i;q} h_q^i \sigma_i^{(q)} + \sum_{i_1,i_2;q_1,q_2} h_{q_1,q_2}^{i_1,i_2} \sigma_{i_1}^{(q_1)} \otimes \sigma_{i_2}^{(q_2)} + \cdots \tag{132}$$

where the indices i refer to the type of the Pauli matrix (x, y or z), the indices q run over qubits and each h in coefficients. The resultant is amenable to the treatment with the VQE method, explored in section 4.2.4.

4.3.5 *Trial wave functions (ansätze)*

The ground state energy estimation requires an appropriate *ansatz*. If the number of electrons in the system, N, is fixed, one may use the Slater determinant solution of the HF problem for the molecule at hands, corresponding to its ground state. We shall denote it by $|\Psi_0\rangle$ and write as

$$|\Psi_0\rangle = \prod_{\alpha}^{N} a_\alpha^\dagger |\text{vac}\rangle \,,$$

where α runs over the occupied orbitals and $|\text{vac}\rangle$ denotes the vacuum (absence of particles). Alternatively, one may start by defining a new "vacuum" state in the N-particle sector of the Fock space, which can be chosen as $|\Psi_0\rangle$ and used to prepare the parametrized trial quantum state [49]. This can be done by a quantum circuit implementing a unitary operator, $\hat{U}$, representing a set of perturbations to state $|\Psi_0\rangle$:

$$|\Psi(\vec{\theta})\rangle = \hat{U}(\vec{\theta})|\Psi_0\rangle \,. \tag{133}$$

The parametrized ansatz will be used to estimate the energy with respect to the Hamiltonian. Note that $\vec{\theta}$ stands for the whole set of parameters (also called "gate angles" in this context) that can be adjusted and used in the optimization procedure (see Sec. 4.2.4 below).

There are several possible choices for constructing this operator, leading e.g. to the so-called Unitary Coupled Cluster (UCC) and heuristic approaches that are discussed in by [88] and [49]. There are several options for choosing different *ansätze* implemented in the QISKit package. Let us briefly consider the UCC approach, which has mainly been used in this work.

A flexible way to generate *multideterminantal* (hence overcoming the HF approximation) reference states within the Coupled-cluster (CC) method, suggested by [210], has been translated by [49] (specifically under an angle of quantum algorithms for electronic structure calculations) into the *unitary version* of the CC approach (UCC). The operator acting on the "vacuum state" according to Eq. (133) is chosen as follows:

$$\left|\Psi(\vec{\theta})\right\rangle = e^{\hat{T}(\vec{\theta})-\hat{T}^\dagger(\vec{\theta})}\,|\Psi_0\rangle \,. \tag{134}$$

Here $\hat{T}$ is an operator representing excitations from occupied to unoccupied states (labelled below by Greek and Latin indices, respectively), composed of hierarchical (given by the dimension of the interaction operator, which corresponds to the i in $\hat{T}_i$) terms,

$$\hat{T} = \hat{T}_1 + \hat{T}_2 + \dots \,,$$

corresponding to n-particle excitations, namely,

$$\hat{T}_1(\vec{\theta}) = \sum_{\alpha,a} \theta_\alpha^a a_a^\dagger a_\alpha \,, \tag{135}$$

$$\hat{T}_2(\vec{\theta}) = \frac{1}{2} \sum_{\alpha,\beta;\, a,b} \theta_{\alpha\,\beta}^{a\,b} a_a^\dagger a_b^\dagger a_\alpha a_\beta \,, \tag{136}$$

$$\ldots$$

The UCC ansatz usually retains only the two first terms in the expansion of $\hat{T}$, i.e. if neglects 3-particle and higher-order excitations. The expansion coefficients in (135), (136) can be interpreted as matrix elements of a certain excitation operator between occupied and unoccupied orbitals. They can be assumed to be *real*, i.e., $\{\theta_\alpha^a,\ \theta_{\alpha\,\beta}^{a\,b}, \ldots\} \in \mathbb{R}$.

The anti-Hermitian combination $\hat{T} - \hat{T}^\dagger$ in (134) makes the exponential operator unitary. Unitary operations are natural on quantum computers, yet the implementation into quantum circuits is not straightforward because of the non-commutation of different parts of the Hamiltonian. So, the order in which the different terms are written in the exponent is indeed important. This difficulty is bypassed by using the Trotter identity:

$$e^{(\hat{A}+\hat{B})} = \lim_{n\to\infty} \left[e^{\hat{A}/n} \otimes e^{\hat{B}/n} \right]^n \,, \tag{137}$$

where $\hat{A}$ and $\hat{B}$ are two non-commuting operators, e.g. $\hat{A} = \hat{T}_1 - \hat{T}_1^\dagger$ and $\hat{B} = \hat{T}_2 - \hat{T}_2^\dagger$. This is exact in the limit $n\to\infty$, it is an approximation for finite n. Different Trotter approximations of the operator (134) can be implemented on a quantum computer by transforming it into the qubit representation and using standard circuit compilation techniques for the "exponentiation" of the Pauli matrices [88]. Some examples of such circuits and comparison of results obtained for different orders (n) of the Trotter approximation can be found in the work by [49].

4.3.6 *Results and Discussion*

We used the procedure outlined in previous sections to calculate the ground state energy (which can be straightforwardly converted into the dissociation energy) of two molecules, hydrogen (H_2) and lithium hydride (LiH). This has also been addressed, in which is presumably a novel result, under the action of stationary electric fields of four different magnitudes ($\mathbb{E} = 0.0001,\ 0.001,\ 0.01,\ 0.1$ atomic units; 1 a.u. $\approx 5 \cdot 10^{11}$ V/m). These calculations were performed for the interatomic distances, d, from 0.2 to 4 Å with the step 0.1 Å.

The actual computational environment, where these experiments were conducted, was the IBM Q. Such computational environment is available remotely through the internet and can be accessed and programmed using the QISKit framework, written in the Python language. The actual code developed in this thesis is

Table 9: The set of technical parameters used for the quantum calculations.

parameter	value
shots[a]	4096
Max. number of iterations of Cobyla	15000
Max. number of iterations of PySCF	5000
optimization level	3
mapping method	Jordan-Wigner
QISkit version	0.13.0

[a]number of times the execution of circuits is to be performed due to the stochastic nature of quantum computers

available from the following github repository: https://github.com/arcalab/experiments_quantum_chemistry/tree/master/Qiskit_Programmatic_version_src. It makes use notably of the QISkit and the PySCF python framework.

The PySCF tool was used to specify the molecules and calculate the respective one-body and two-body integrals, encompassing already the action of electric fields, using the theory developed throughout appendix A. Both molecules were assumed to have zero global charge and spin zero; the STO-3G basis (117) was used to calculate the integrals.

The evaluation of the corresponding integrals can then be reformulated into an assembling of quantum circuits, to be executed using the set of *software* packages available e.g. in the QISkit framework: Terra, Aer, Aqua and Ignis. The calculation of the dissociation curves requires the calculation of the ground state energies (discussed in section 4.2.4) over a range of distances, to be able to identify the minimum (bound molecule) and the asymptotics (separated atoms) ones. For this purpose, we used two methods: the *Exact Eigensolver* (classical *matrix-multiplication* method, as a benchmark) and the VQE.

We used the UCC (discussed in section 4.2.5) as the *variational method*, i.e. as the technique to build the *ansätze* for the molecules under study, and the HF approximation to obtain the initial solution for the VQE method. In this relation, several parameters had to be considered: the maximum number of iterations with the Cobyla method,[3] the optimization level (an IBM Q -specific parameter determining the degree of optimization of the circuits generated), the mapping method to use, such as the Jordan-Wigner (131), Bravyi-Kitaev, or parity methods (see [88] for more information on these methods), each offering different (precision) / (circuit size) relationships. The technical parameters of calculation, selected after a course of trial and error, are summarized in Table 9.

3 In this quantum computation setting, an iteration in the Cobyla method is an expensive operation in terms of computation time, and therefore one may be interested in limiting the number of iterations. However, the method stops if convergence is achieved and in our particular case, the method always converged before 15000 iterations.

The use of quantum or *hybrid* (such as VQE) procedures in the IBM Q require that a *backend* is specified, i.e. an actual processing node able to execute the quantum circuits, which can be either a classical computer able to perform a simulation, with or without *simulated quantum noise*, or a real quantum device, with a number of qubits from 2 to 53. The results of this work were obtained using a *simulator*, the *qasm_simulator*.

4.3.7 *Results: H_2 molecule*

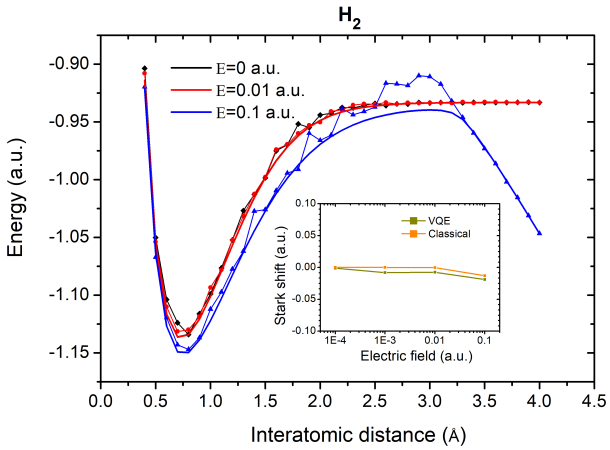


Figure 20: Dissociation curve of the H_2 molecule, as calculated with a classical solver (full lines) and with the VQE (symbols connected by lines), for several values of the external electric field $\mathbb{E}$ marked by color. The Stark effect (i.e. the shift of the minimum energy with electric field) is shown in the inset.

The total energy as a function of the interatomic distance, hence the molecule's dissociation curve for different values of the electric field, is depicted in Figure 20. The effect of the electric field on the shape of the dissociation curve remains negligible at small values of the field inspected yet results in a drastic change of the $d \to \infty$ asymptotic (slope) and in a noticeable shift of the equilibrium position for $\mathbb{E} = 0.1$ a.u. The abrupt change in the $E(d)$ dependence slope at large distances, for very large electric field $\mathbb{E} = 0.1$ a.u., can be related to the onset of the molecule's dissociation, which becomes possible via tunneling through the energy barrier (blue curves in Fig. 20).

The inspection of the VQE results, represented by symbols connected by lines in Fig. 20, reveals a numerical noise that apparently increases with the electric field magnitude. Possibly, the HF approximation used as input for the quantum calculation becomes unstable under the action of a strong electric field.

The inset of Figure 20 shows the Stark effect for the molecule under study, that is, the shift between the ground-state energy calculated under the action of the electric field and at $\mathbb{E} = 0$. The distance at which the respective energies have been extracted was the energy minimum position yielded by the classical solver

at $\mathbb{E} = 0$, $d_{eq} = 0.7$ Å. We took this option because of the fluctuations of $E(d)$ obtained with the quantum solver.

For a non-polar molecule without intrinsic dipole moment, as is the case of the H_2, the stationary electronic Stark effect should be quadratic in the electric field. However, with the limited minimal basis used, it looks even weaker and becomes noticeable only for very strong fields.

4.3.8 *Results: LiH molecule*

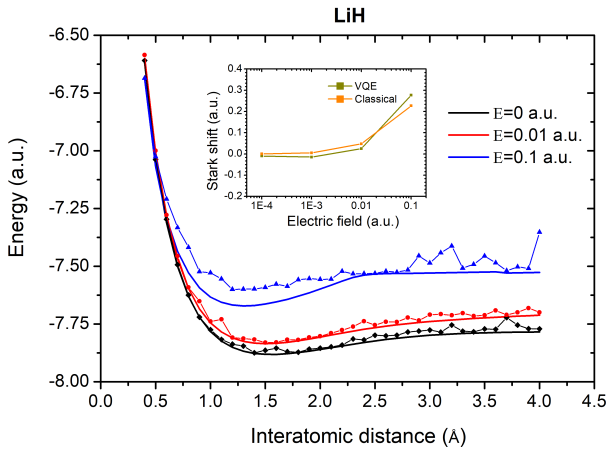


Figure 21: Same as in Fig. 20 for the LiH molecule.

The results for the lithium hydride molecule are shown in Fig. 21, where the effect of the applied electric field is quite noticeable. The displacement of the $E(d)$ curve increases with the electric field: already for 0.01 a.u. the shift of the dissociation curve becomes appreciable. The Stark effect (inset in Fig. 21) increases with the field magnitude much faster then for the case of the H_2 molecule. This is due to the intrinsic dipole moment the LiH molecule already possesses in the ground state. The Stark effect is linear in $\mathbb{E}$ for small fields but than starts growing much faster because of the additional polarization of the ground state induced by the external field.

Similar to what happens in the case of the H_2 molecule, the numerical noise is visible in the results and becomes more pronounced in stronger electrical fields. Also, the ground state energy obtained with the different solvers results in different values for the equilibrium distance, d_{eq}, obtained for the quantum and classical solver, – 1.5 Å and 1.6 Å, respectively, at $\mathbb{E} = 0$. Again, the latter was taken as the reference value for the Stark effect evaluation.

4.4 Summary

In this chapter, the landscape of the application of quantum computers to very hard computational problems, was explored. The focus were on the optimization ones, which are pervasive in industry and can be divided in many types, from linear to convex and non-convex. It were also reviewed the quantum computational techniques and algorithms employed for these types of problems.

The main contribution of this chapter was the calculation of the ground-state of two small molecules H_2 and LiH under the action of a *strong* electric field, which configures the so-called *Stark effect*. The study of the ground-state of such molecules was extensively studied in literature, including by using quantum computers, however, the inclusion of the *electric field* in the calculation of the ground-state on a quantum computer, poses many non-trivial challenges, in theory and in practice, which were not addressed directly in existent literature, but are addressed in the sequel. Hence, it was attempted to outline, in a concise way yet indicating the essential elements and the underlying theory, a representative practical resolution of this problem on the commercially available (since recently) quantum computer, $\mathrm{IBM\ Q}$, involving:

- The fermionic formulation of quantum chemistry systems;
- The connection between fermionic Hamiltonians and the quantum circuits;
- The state preparation, running of the algorithm and the evaluation of the results.

The calculated results were obtained the quantum device simulator and comprise the total energy as a function of bond length (i.e. the dissociation curve), also under an applied stationary electric field. We also evaluated the shift of the molecule's energy at a fixed d (equal to the equilibrium interatomic distance) with the electric field, i.e. the stationary electronic Stark effect, supposedly quadratic in $\mathbb{E}$ and small for the non-polar H_2 molecule but containing the linear term, and the much stronger in case of the polar LiH molecule. Summing up, our case study seems to provide evidence for the feasibility of the use of this quantum computer for small molecules, with a reasonable number of iterations performed.

5

A LOGIC FOR THE QASM PROGRAMMING LANGUAGE

> I would like to make a confession which may seem immoral: I do not believe absolutely in Hilbert space anymore.
>
> John von Neumann, letter to Garrett Birkhoff, 1935.

In the beginning of the twentieth century, there was a line of thought on the derivation of a comprehensive theory, able to serve as a foundation for mathematics, out of logical principles and in a formal way: the so-called *logicism*. This endeavour aimed at replacing *ad-hoc* theories of mathematics, as it is the case of the Zermelo-Frankel set theory.

Eventually, this endeavour lost interest, due mainly to Gödel incompleteness theorems, however, understanding logical structures behind scientific theories in order to have a deeper understanding of them is, still today, a widely used approach across many fields of science. In this regard, quantum theory is a very good example, as it possesses a very specific and insightful notion of logic, as firstly discussed by *Von Neumman* and *Birkhoff* in their seminal work of 1936 [67], where quantum logic was introduced for the first time.

This logic is centred around the notion of *observable* properties, which organize themselves into an algebraic structure which is clearly not a Boolean algebra, due to the *non-commutativity* of some of them. The proper understanding of these notions originated an extensive research work, and it was particularly important in the understanding of quantum theory itself, for instance, by helping in the axiomatization of quantum theory in the formalism of Hilbert spaces. Furthermore, several approaches to the obtention of a quantum theory of gravity follow a *somewhat* logicist approach to physics, by attempting on defining the axiomatics for a potential unified theory, where, of course, the insights of quantum logic have played an important role.

Nowadays, the establishment of proper logical structures for quantum theory is also driven by the interest in automated reason methods and tools to deal with quantum protocols and algorithms, which has become particularly relevant in the recent years. In this regard many logical frameworks have been proposed coming

from different origins, from purely algebraic terms and category theory, to modal and linear logic. In this chapter we revisit the relevant background to understand quantum logic, in sections 5.1 and 5.2, and propose a logic able to deal with the QASM programming language, a quantum programming language involving quantum and classical procedures. An early version of this work, presented in sections 5.3, 5.4 and 5.5, appeared in the *Dali 2019* workshop [342].

5.1 Standard quantum logics

In 1936 *Von Neumman* and *Birkhoff*, published the seminal paper [67] on what would become quantum logic, focusing on the study the algebraic properties of quantum propositions. They consider that the latter correspond to the tests that can be performed experimentally, i.e. the so-called *testable properties*. Mathematically, such properties are *closed linear subspaces* of Hilbert spaces, which can be defined as follows:

Definition 5.1.1. (Orthogonality and Orthocomplement). Let $\mathcal{H}$ be a complex Hilbert space. Two vectors $x, y \in \mathcal{H}$ are said to be orthogonal, denoted $x \perp y$, iff $\langle x|y\rangle = 0$. Given a subset $X \subseteq \mathcal{H}$, the orthocomplement of X, denoted by $\sim X$, is given as:

$$\sim X = \{y \in \mathcal{H} |\ y \perp x \text{ for all } x \in X\}\,.$$

Definition 5.1.2. (Closed Linear Space) Let P be a subspace of a complex Hilbert space $\mathcal{H}$. P is a closed linear subspace if

$$\sim\sim P = P.$$

Taking testable properties as propositions, the logic develops as expected. Negation is given by the *orthocomplement* of a proposition, denoted $\sim \phi$, where ϕ is a proposition, and conjunction, denoted $\phi \wedge \psi$, is given by the intersection of the two closed linear subspaces. The definition of disjunction, slightly different from the classical one, is given by the quantum join: the closed linear subspace defined by the classical union of two propositions.

Definition 5.1.3. (Quantum Join) Let $\mathcal{H}$ be a complex Hilbert space and let $K, L \subseteq \mathcal{H}$ be two subsets. The quantum join of K and L, denoted $k \sqcup L$ is given by

$$K \sqcup L = \sim\sim (K \cup L)\,. \tag{138}$$

A minimalistic quantum logic builds on these operations, according to the following syntax:

$$\phi ::= \bot |\ \sim \phi | \phi \wedge \phi | \phi \sqcup \phi\,,$$

where $\perp$ is the property *false*, corresponding to the empty closed Hilbert space. Valid propositions, as usual in logic, possess an order relationship, based on the fact that some propositions can be *embedded* on each other. For instance, if $A \wedge B$ is true, one necessarily concludes that A is true, and hence one states that $A \wedge B \leq A$, i.e. A is more general and less restrictive. In fact, the whole set of propositions can be arranged in a bounded pre-order structure, denominated *lattice* (see definition 5.1.4). The most distinctive property of quantum lattices of propositions, is that the distributivity law, one of the *cornerstones* of classical logic, does not hold,

$$P \wedge (Q \sqcup R) \neq (P \wedge Q) \sqcup (P \wedge R)\,,$$

due to the non-commutative nature of observable properties of quantum mechanics, i.e. the order in which observation are made matters, as presented in example 5.1.1.

Example 5.1.1. Due to the Heinsenberg's uncertainty principle it is well known that it not possible to have, simultaneously, definite values for momentum and position. From this, it is possible to build an example where the distributivity law fails. Consider the following propositions,

- P - $|p| \leq 1$ (units are irrelevant),
- Q - $x \leq 0$ and
- R - $x \geq 0$

where x regards the position of a particle and p its momentum. From quantum mechanics, it can be stated that if P is true then

$$(P \wedge Q) \text{ is false, } (P \wedge R) \text{ is false, } (P \wedge Q) \sqcup (P \wedge R) \text{ is also false,}$$
$$(Q \sqcup R) \text{ is true, and } P \wedge (Q \sqcup R) \text{ is also true.}$$

Hence, $P \wedge (Q \sqcup R) \neq (P \wedge Q) \sqcup (P \wedge R)$ and in this particular case the distributivity law does not hold.

This fact has important implications in the way one can build inference rules for this logic, i.e. the traditional *modus ponens* is not valid for the most generalized version in infinite dimensions of quantum logic, however *weaker* forms of the *distributive* law are possible in certain situations. In particular, the *modular law*, has been shown to hold for finite dimensional Hilbert spaces

$$\text{If } A \leq B \text{ then } A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)\,, \tag{139}$$

so as the *orthodomular law* for Hilbert spaces of arbitrary dimension, as shown by Husimi [201],

$$\text{If } A \leq B \text{ and } \sim A \leq C \text{ then } \sim A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)\,. \tag{140}$$

These observations suggest that quantum logic is substantially harder for infinite dimensions. From the algebraic perspective, quantum logic presents itself in a *ortholattice* structure (see definitions 5.1.4 and 5.1.5), where the *commutative* law does dot hold, which prevents the distributive law to hold, as it would do in a Boolean algebra (definition 5.1.6).

Definition 5.1.4. (Lattice)
A lattice is a structure $\mathcal{L} = (\wedge, \vee)$ where the following laws hold:

1. $A \wedge A = A; A \vee A = A$ (idempotence)
2. $A \wedge B = B \wedge A; A \vee B = B \vee A$ (commutativity)
3. $A \wedge (B \wedge C) = (A \wedge B) \wedge C; A \vee (B \vee C) = (A \vee B) \vee C$ (associativity)
4. $A \wedge (A \vee B) = A; A \vee (A \wedge B) = A$ (absorption)

Definition 5.1.5. (Ortholattice)
An ortholattice is a lattice $\mathcal{L} = (\sim, \wedge, \vee, 0, 1)$ where the following laws hold:

1. Has a least element (0) and greatest element (1), thus called bounded;
2. Complemented, every element A has a orthocomplement $\sim A$, $(A \vee \sim A = 1)$, $(A \wedge \sim A = 0)$,
3. which is an involution $(\sim\sim A = A)$.

Definition 5.1.6. (Boolean algebra)
A Boolean algebra is an ortholattice $\mathcal{L} = (\sim, \wedge, \vee, 0, 1)$, where the distributivity laws hold:

1. $A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$;
2. $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$

Given this, *ortholattices* seem to cope with the needs quantum logic, however, the fact that a weaker form of the modular law to hold in general, leads to a more appropriate algebraic structure - that of a orthomodular lattices (definition 5.1.7) - which, arguably, seems to be the right structure for quantum propositions [217].

Definition 5.1.7. (Orthomodular lattice)
An orthomodular lattice is an ortholattice $\mathcal{L} = (\sim, \wedge, \vee, 0, 1)$, where the orthomodular law (eq (140)) holds:

$$\text{If } A \leq B \text{ then } A \vee (A \wedge \sim A) \text{ (Weak modularity)}. \tag{141}$$

Yet, while *orthomodular lattices* capture accurately the structure of quantum tests represented in *Hilbert spaces*, there is still a major issue preventing them to be equivalent to Hilbert spaces as a foundational structure of quantum mechanics: the inexistence of a *tensor product* [300]. This led to extensive research work to find the "right" logical structure of quantum mechanics, from which relevant lines of work have arisen, as for instance *orthoalgebras*, introduced by Randall and Foulis [159], for which a tensor structure exists [160], which, however, possess other issues [173]. Relevant lines of work are given, for instance, by *effect algebras* [158, 122], and partial Boolean algebras, introduced by Kochen-Speckter [231, 230].

Excellent reviews of these developments are available, for instance, in [157, 123], but so far no algebraic structure, based on the algebraic properties of quantum propositions, is able to cope completely with all the features of quantum mechanics, obtained, for instance, in Hilbert spaces.

One of the applications of quantum logic was on the *axiomatization* of quantum mechanics, from which a better characterization of the Hilbert spaces suitable for quantum mechanics was obtained. The work started by Piron in 1964 [288], where five axioms of quantum mechanics in Hilbert spaces were established and was completed many years later thanks to the work of Mayan and Soler [333, 16].

Further, there is also some work attempting to follow a similar approach to find the quantum theory of gravity out of logical principles see, in particular, work reported by Isham [204], or Hartle [166].

5.2 Dynamic aspects in Quantum Logic

The logical *apparatus* discussed in the previous sections aims exclusively at reasoning over the *static* perspective of quantum systems. However, to reason about quantum evolution, which possess quantum programming as a subcase, one is mostly interested in the dynamic perspective. To the best of our knowledge, no dynamical logical system based on the notion of logic given by *observable properties*, is able to deal with quantum systems of arbitrary dimension, in a complete way. The reasons for this may have to do exactly with the issues raised in previous section: the lack of a satisfactory notion of *tensor* for *orthomodular lattices* and the issues existent in the structures that do possess tensors.

However, there are complete reasoning systems, particularly those akin to the formalisms of quantum mechanics, Hilbert spaces and density operators, where the notions of tensor and compositionality come naturally, as it happens for instance in the extensive field of categorical quantum mechanics (CQM). Such formalism is focused on the *abstract* algebraic properties of such spaces and offer a wide range of tools to validate quantum programs and protocols [12, 109].

The notions acquired from standard quantum logics have also influenced a wide range of logical systems to deal with quantum programs and several associated tools originate in different contexts, including modal, linear logics and some forms of non-classical logics. Here one shall highlight, for instance, the works of Mateus et al. [262, 91], Adams et al. [13], based on the so-called *effectus theory* [104], or the works of Ying [379], a Floyd-Hoare logic [200] based on the notion of weakest-precondition semantics of D'Hondt and Panangaden [134]. The latter has originated several model checkers and theorem provers [246, 380]. Still in the Hoare logic line of work, there is the work of Unruh [345], and also several attempts of building Hoare type systems [331], in order to obtain automatic program validation, right on compile time.

The line of work which mostly influenced the work undertaken in this thesis is the one of quantum dynamic logic. Dynamic logic is a system often used for program verification introduced by Pratt [295, 190], and is essentially a modal logic, where modalities are labelled by program components [68]. On the quantum side, a modal perspective on quantum logic, has been explored, for instance, in the works of Dalla Chiara et al. [121] or Goldblatt [175], and the first dynamic logic for quantum programming was introduced by Baltag and Smets [39], a path that was also explored by Brunet et al. [85].

5.2.1 *Quantum dynamic logic and its semantics*

A dynamic logic for quantum programs was introduced in 2004 by Baltag and Smets [39, 40], and since then it has been extensively developed, as documented, for instance, in [42, 43], or [44, 41], focusing on different perspectives, e.g. capturing entanglement in quantum systems, or the probabilistic properties of quantum programs. More recently, akin to this kind of work, Bergfeld [58] proposed a probabilistic dynamic logic for quantum systems, focusing on the calculation of probabilities [57], which also influenced our own work.

There are still however many relevant lines of work to pursue in the field of dynamic logics, for instance, the design of logics to deal with more realistic quantum programming languages, particularly the ones that involve classical control instructions and measurements, which as, discussed in section 2.4.4, constitutes a large family of languages of increasing complexity.

Nonetheless, standard quantum logic can be expressed in dynamic logic, i.e. using a structure akin to a *labelled transition system* [198], where the state space is given by the set of valid states in an Hilbert space. Tests, rather than corresponding a *closed linear subspaces*, are given by valid transitions in the state space, in a relational way, such that they correspond to their actual *effects*. A structure that captures these premises, and supplies notions compatible with quantum logic, is that of a *quantum dynamic frame*, which is defined as follows:

Definition 5.2.1. (Quantum Dynamic Frame) A quantum dynamic frame $\mathcal{F}$ is a tuple $(\Sigma, \mathcal{L}, \{\xrightarrow{P?}\}_{P\in\mathcal{L}})$ such that Σ is a set, $\mathcal{L} \subseteq 2^{\Sigma}$, $\xrightarrow{P?} \subseteq \Sigma \times \Sigma$ for each $P \in \mathcal{L}$, that satisfies the following conditions, where $\rightarrow = \bigcup_{P\in\mathcal{L}} \xrightarrow{P?}$:

1. $\mathcal{L}$ is closed under arbitrary intersection.
2. $\mathcal{L}$ is closed under orthocomplement, where the orthocomplement of $A \subseteq \Sigma$ is

$$\sim A := \{s \in \Sigma | s \nrightarrow t \text{ for all } t \in A\} \tag{142}$$

3. Atomicity: For any $s \in \Sigma, \{s\} \in \mathcal{L}$.
4. Adequacy: For any $s \in \Sigma$ and $P \in \mathcal{L}$, if $s \to t$, then $s \xrightarrow{P?} s$.
5. Repeatability: For any $s, t \in \Sigma$ and $P \in \mathcal{L}$ if $s \xrightarrow{P?}$ then $t \in P$
6. Self-Adjointness: For any $s, t, u \in \Sigma$ and $P \in \mathcal{L}$, if $s \xrightarrow{P?} t \to u$, then there is a $v \in \Sigma$ such that $u \xrightarrow{P?} v \to s$, for all $v \in P$.
7. Covering Property: Suppose $s \xrightarrow{P?} t$, for $s, t \in \Sigma$ and $P \in \mathcal{L}$, then for any $u \in P$, if $u \neq t$, then $u \to v \nrightarrow s$ for some $v \in P$; or, contrapositively, $u = t$ if $u \to v$ implies $v \to s$ for all $v \in P$.
8. Proper superposition: For any $s, t \in \Sigma$, there is a $u \in \Sigma$ such that $s \to u \to t$.

There are other structures that have been shown to be equivalent to this one, such as, along these lines, quantum Kripke frames introduced by Zhong [385]. An interesting slightly different approach to the semantics of quantum logics, is the one intruduced by Bergfeld [57], where quantum systems consist of a series of *independent* systems, represented by *ordered basis*, the N-PQM model. In this approach, the notion of entanglement is a free connective, part of the syntax of the logic, which greatly simplifies some of the probabilistic calculations.

One advantage of using dynamic structures to represent quantum logic is that the corresponding logic can easily be transformed into a logic of programs, as done by Smet and Baltag's in [39]. The semantics is given by *quantum frames*, which is just an extension of *quantum dynamic frames* allowing unitary transitions, which can be typed as a tuple reading as:

$$\Sigma(\mathcal{H}) := \left(\Sigma, \{\xrightarrow{P?}\}_{P \in \mathcal{L}}, \{\xrightarrow{U}\}_{U \in \mathcal{U}}\right) \tag{143}$$

where, similarly to quantum dynamic frames, Σ corresponds to the set of possible states of an Hilbert space of finite dimension, $P \in \mathcal{L}$ are *testable* properties (i.e. the closed linear subspaces), $P?$ the actual *experimental processes* of measuring testable properties, and $U \in \mathcal{U}$ to unitary transitions. Naturally, quantum frames support all rules presented in in definition 5.2.1, and in addition the theorems 5.2.1 and 5.2.2, which concern the *reversibility* and adjointness of unitary operators, respectively.

Theorem 5.2.1. ([39]) In every quantum frame $\Sigma(\mathcal{H})$ the following properties for unitary transformations (stated for all $U, U^\dagger \in \mathcal{U}$) hold:

1. Functionality: For every state $s \in \Sigma$ we have $\exists t : s \xrightarrow{U} t$
2. Inverse-adjoint (bijectivity): $s \xrightarrow{U} t \xrightarrow{U^\dagger} w$ implies $s = w$. Similarly, $s \xrightarrow{U^\dagger} t \xrightarrow{U} w$

Theorem 5.2.2. (Adjointness [39]) Let F be a linear transformation and let $s, w, t \in \Sigma$ be states: If $s \xrightarrow{F} w \to t$ then there exists some state $v \in \Sigma$ such that $t \xrightarrow{F^\dagger} v \to s$.

The quantum frames are used in a logic for quantum programs explored, for instance, in [39, 45]. Such logic encompasses two layers, the layer of programs, (π) and the layer of propositions, (ϕ), with the following syntax:

$$\phi ::= p|\phi \wedge \phi|\neg\phi|[\pi]\phi|P^{\geq r}\phi$$
$$\pi ::= u|\phi?|\pi;\pi|\pi \cup \pi$$

On the propositional layer, one founds the usual Boolean connectives, $\wedge$ and $\neg$, with similar meaning as they do in classical logic, modal operator $[\pi]\phi$ with the *meaning* "if π terminates, then ϕ is necessarily true", the probabilistic operator $P^{\geq r}\phi$, meaning that "ϕ is true with a probability greater than r". The programs *layer* π contains the atomic programs and the operator for sequential composition of programs, *nondeterministic choice*, and *tests*, which correspond to the *quantum tests*, with the *destructive* effects that characterize them. The main innovation of this logic is to provide a concrete way to reason about quantum programs.

On the decidability side, the finite dimensional standard quantum logic, has been shown to be decidable by Dunn et al. [139], using a proof strategy based on a reduction to the *first order theory of the reals*, known to be decidable by an important result of Tarski [341]. Following to this result, many other logics, including the dynamic ones discussed in this section, have shown to be decidable using the same type of strategy, as shown by Baltag et al. in [45].

Furthermore, it is also shown in Dunn et al. [139], that whenever another qubit is added to the register, the correspondent logic is *different*, $QL(\mathbb{C}^{2^n}) \neq QL(\mathbb{C}^{2^{n+1}})$. Due to the inexistence of tensor operator defined between two such logics, the latter cannot be built from the former, just by *tensoring* it with a *single-qubit* logic. For infinite dimensions, standard quantum logic is known to be undecidable [163], so as, presumably, all the logics that handle programs with infinite dimensions that can be reduced to it.

5.3 A dynamic logic for QASM programming language (LQASM)

The Quantum Assembly Language (QASM language) [116], is the quantum circuit specification language in use in the commercially available quantum hardware supplied by IBM, the IBM Q platform [2]. The platform includes a visual editor for quantum circuits (a simple example for a quantum coin is presented in figure 22).

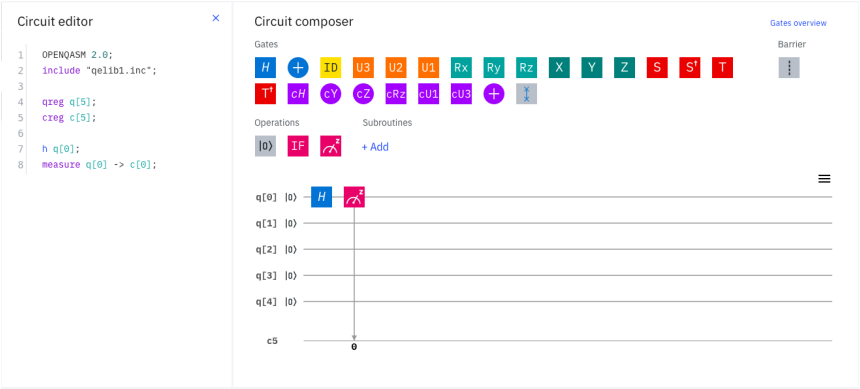


Figure 22: Example of the definition of a circuit in the QASM language. On the right side the visual definition of the circuit and on the left side the correspondent QASM code.

QASM is not a purely *quantum* language as it involves some form of *classical control*, i.e. it encompasses *classical variables*, *measurements*, which possess a probabilistic nature, and classical flow instructions (*if statements*), based on the (*probabilistic*) results of measurements. Therefore, a logic targeted to this kind of programs must be able to deal with these probabilistic instructions which may affect both propositions and programs, a challenge we intend to address in this work. More on this on section 5.3.3. Our point of departure is the extensive line of work on dynamic logics for quantum programs, initiated by Baltag and Smets [39, 45] and the work of Kozen on the dynamic logics for probabilistic programs [234, 233]. In the next few sections we discuss both the fragment of QASM to be handled by the logic, as well as the logic, its syntax and semantics.

5.3.1 *The QASM programming language*

The QASM language is a circuit specification language, allowing for the construction of a wide range of circuits, making use of quantum gates, expressed in rotations on up to *three* axis:

$$U(\phi) \qquad U_2(\phi,\theta) \qquad U_3(\phi,\theta,\lambda)$$

where ϕ, θ, λ correspond to angles in the x, y and z axis. This set of gates is clearly universal, as any set of gates can interpreted as a *multi-angle* rotation. Furthermore, the language also allows the definition of *customized* quantum gates, based on arbitrary compositions of these *rotational gates*, and the angles

themselves can be derived from a wide set of *complex-valued* constructible formulas, at *compile* time. Moreover, the language also encompasses a set of *classical control instructions*: creation of classical and quantum registers, resetting qubits, measurements and conditional statements. However, the language does not include any *while* cycles. The complete syntax of the language is available in [116], but in this thesis we explore only a fragment, as represented in figure 5.1.

$$
\begin{aligned}
\langle \pi_q \rangle \quad & ::= \text{x qreg\_id [index]} \mid \text{z qreg\_id [index]} \mid \text{h qreg\_id [index]} \\
& \mid \text{cx qreg\_id [index}_1\text{], qreg\_id [index}_2\text{]} \\
& \mid \text{measure qreg\_id [index]} \rightarrow \text{creg\_id [index]} \\
& \mid \pi_q ; \pi_q \\
\langle \pi \rangle \quad & ::= \text{creg id [size]} \mid \text{qreg id [size]} \\
& \mid \text{if } \langle \text{test} \rangle \text{ then } \pi_q \\
& \mid \pi ; \pi
\end{aligned}
$$

Figure 5.1: A fragment of the QASM programming language.

The instructions in this fragment have the following meaning:

- **Unitary gates**
  - x qreg_id [index] - Application of the X gate to the qubit $index$ of register qreg_id;
  - z qreg_id [index] - Application of the Z gate to the qubit $index$ of register qreg_id;
  - h qreg_id [index] - Application of the H gate to the qubit $index$ of register qreg_id;
  - cx qreg_id [index$_1$], qreg_id [index$_2$] - Application of the CNOT gate on the qubits index$_1$ and index$_2$, of register qreg_id.
- **Non-unitary instructions:**
  - creg id [size] - Creation of a register, denominated id, with a number of classical bits given by size;
  - qreg id [size] - Creation of a register, denominated id, with a number of quantum bits given by size;
  - measure qreg_id [index] → creg_id - Measurement of the qubit index of register qreg_id to the classical register creg_id.
- **Flow control instructions;**
  - if <test> then π_q - Conditional instruction, on the success of test, program π_q is executed.
  - $\pi;\pi$ - Composition of programs.

5.3.2 *Syntax for the dynamic logic*

As usual in dynamic logic, the syntax, defined in figure 5.2, is divided into two layers: one for *programs* and for *formulas*.

$$
\begin{array}{ll}
\langle \varphi_q? \rangle & ::= id_i^q == \text{a, where } a \in \{1,0\} \\
\langle \varphi_c? \rangle & ::= id_i^c == \text{a, where } a \in \{1,0\} \\
\langle \varphi? \rangle & ::= \varphi_c? \mid \varphi_q? \\
\langle \pi_q \rangle & ::= \text{x } id_i^q \mid \text{z } id_i^q \mid \text{h } id_i^q \\
 & \quad \mid \text{cx } id_i^q, id_i^q \\
 & \quad \mid \text{meas } id_i^q \text{ to } id_i^c \\
 & \quad \mid \pi_q ; \pi_q \\
\langle \pi \rangle & ::= \text{creg id [size]} \mid \text{qreg id [size]} \\
 & \quad \mid \text{if } \varphi_c? \text{ then } \pi_q \\
 & \quad \mid \pi ; \pi \\
\langle p \rangle & ::= i_{r_i}^c \mid i_{r_i}^q \text{ with } i \in \{0,1\} \\
\langle \varphi \rangle & ::= \text{p} \mid \top \mid P^{\geq r}\varphi \mid \mathcal{A}^{=\lambda}\varphi \text{ with } \lambda \in \mathbb{C} \mid [\pi]\ \varphi \mid \neg\varphi \mid \varphi \wedge \varphi
\end{array}
$$

Figure 5.2: The formulas layer and programs layer.

The program's layer encompasses a fragment of the QASM language, defined in figure 5.1, however with a simplified syntax, to make proofs more understandable and shorter. The major changes consist of the replacement of statements like $id[index]$, by id_{index}, where id is the *register* identifier and *index* the position of the bit/qubit on the vector. Furthermore, the bits and qubits are *typed*, i.e. qubits are expressed as $i_{r_i}^q$, and classical bits as $i_{r_i}^c$ with $i \in \{0,1\}$. The syntax of the measurement instruction was changed to **meas** id_i^q to id_i^c, where, from left to right, the first id_i defines a quantum bit to be measured, and the second on, the classical bit of destination of the measurement. The if statement, **if** $\varphi_c?$ **then** π_q, has the same in the QASM language: if the test is true, then the program is executed, otherwise it is not. On the formula side, atomic propositions correspond to propositions over qubit states: 0, 1 and $\bot$, the latters referring to propositions where tests 0, 1, result with 1 as probability, and the former $\top$ to the truth. Expressions p_{r_i}m refer to the narrowing of a proposition range to a specific register and qubit, as for instance 0_{q_0}, which means that qubit 0 of register q has value 0. The $P^{\geq r}\varphi$ modality establishes restrictions to the probability of propositions for instance $P^{=0.5}p$, while the $\mathcal{A}^{=\lambda}\varphi$ modality, refers to the quantum *amplitude* of a proposition on a state, i.e. the result of the internal product operator in a *state*, i.e.

$\langle\varphi|s\rangle$, where s is a state. The $[\pi]$ has the usual meaning of "the proposition φ *necessarily* holds upon the execution of program π, program π halts" and the usual *minimal* set of Boolean connectives is included. Finally, the following abbreviations of table 10 are valid.

$\bot ::= \varphi \wedge \neg\varphi$	$\bot := \neg\top$	$\langle\pi\rangle\varphi := \neg[\pi]\neg\varphi$
$\diamond\varphi := \langle\varphi?\rangle\top$	$\Box\varphi := \neg \diamond \neg\varphi$	$E\varphi := \diamond\diamond\varphi$
$A\varphi := \neg E\neg\varphi$	$\varphi \vee \psi := \neg(\neg\varphi \wedge \neg\psi)$	$\varphi \rightarrow \psi := \neg(\varphi \wedge \neg\psi)$
$p^{<R}\varphi := \neg p^{\geq r}\varphi$	$p^{\geq r}\varphi := \neg p^{\leq R}\varphi$	

Table 10: Allowed abbreviations

5.3.3 *Discussion*

The logic presented in this chapter (LQASM) targets the QASM language, which to the best of our knowledge, has not been addressed directly in literature before. The challenge offered by the QASM language revolve around the need for the coexistence of classical and quantum information, so as of purely unitary programs and classical ones. More precisely, the challenges of the design of LQASM translate into the following concrete requirements:

1. Existence of classical and purely unitary operations, which must behave as expected when inciding over quantum bits;
2. Coexistence of classical and quantum tests of bits and qubits, respectively, where the former ones conserve the state of the bit being tested, while the latter yield a *collapse* effect;
3. Existence of quantum measurements, which are interpreted as a *classical* probabilistic combination of the effects of two complementary quantum tests, hence preserving the probabilistic nature of the measurement results;
4. Existence of *if statements* with probabilistic control variables, as a consequence of requirement 3).
5. More generally, existence of two types of information, classical and quantum, where the latter can copied, but does not support quantum phenomena such as *interference* and *entanglement*, and the former behaves exactly as opposite.

From a semantic perspective these requirements can naturally be accommodated in a *density operator* setting, which supports, by definition, all possible types of physically sound operators, which includes unitary operators and all sorts of measurements. A Floyd-Hoare logic for quantum programs based on this semantics has been already developed (see Ying [379]), however, there is a slight difference between the such logic

and LQASM, which has to do with requirement 2), as non-destructive tests are not directly supported in a *density operator* setting, and also, in the logic of Ying [379].

However, the logic under development here is closer to the quantum dynamic logic for quantum programs developed in Baltag et al. [39, 45]. The semantic models presented in such logics, do not support naturally the requirements 1) to 5), as both the states and transitions are purely quantum, however as quantum information can be used to simulate the classical one, it provides an excellent starting point. The contribution of this chapter goes along these lines, i.e. of extending the model introduced in [39], to support requirements 1) to 5).

In terms of expressibility the advantages of this logic are to preserve the probabilistic nature of quantum results, which can be useful in a wide range of quantum protocols, such as the leader election, and further it provides a more flexible model to accommodate logics involving non-trivial classical programs, rather than simple if statements. Along the next few sections the solutions for these problems are discussed.

5.3.4 *Semantics*

The semantics of this logic is given in terms of a *labelled transition system* [198], defined by a *tuple*:

$$M = (S, [\![.]\!]_p : \mathcal{A}_p \rightarrow 2^S, [\![.]\!]_\pi : \mathcal{A}_\pi \rightarrow 2^{S \times S}) \tag{144}$$

where S is a set of states and $[\![.]\!]_p$ and $[\![.]\!]_\pi$ are *meaning* functions. The former function gives meaning to propositions, i.e. it is typed as a function from the set of well-formed syntactic expressions of propositions ($\mathcal{A}_p$) to the *powerset* of states, and the latter one does the same for programs, typing as a function from the set of well-formed syntactic expressions of programs ($\mathcal{A}_\pi$), to the powerset of the pairs of sets of states. Hence programs are deterministic. For simplification reasons we omit the identifiers of meaning functions, p and π, which shall be infered from context.

5.3.5 *The state space*

A state of a program in the *QASM* language is defined by its classical and quantum components. Each such component is divided into one or many *independent* registers, each one composed of a set of quantum or classical bits. The set of possible states for a *quantum bit* is given, as described in section 2.4, by all the *normalized* states in a *Hilbert space* of dimension 2, denoted here as $\mathcal{H}_i^2$ where i is the qubit index, but the actual state space of n qubits is given by all *normalized* states of $\mathcal{H}^{2n}$, which encompass, all *non-separable* states existent in the n qubit space. Normalization of quantum states comes from the *Born* rule and the space state can be expressed as:

$$\mathcal{H}^{2n} \equiv \sum_{i \in \{0,1\}^n} \lambda_i |i\rangle \text{ where } \sum_i \lambda_i * \lambda_i^\dagger = 1\,. \tag{145}$$

Due to the existence of *stochastic* instructions, i.e. *measurements*, we provide a *classical* probabilistic semantics to *classical bits*. According to Kozen et al. [234], the semantics of probabilistic programs is founded in the realm of *measurable* functions, where, for instance, the states of such programs are given by a measurable function, on the possible tests[1] over the program variables. In the case of a probabilistic *bit*, the state s is defined by a function typed as

$$\mu_s : \{0,1\} \to [0,1]\,.$$

Here and in an equivalent way, we represent a *stochastic bit* by an Hilbert space of dimension two over complex numbers, of basis $\{0,1\}$, where $(0) = \begin{pmatrix}1\\0\end{pmatrix}$ and $(1) = \begin{pmatrix}1\\0\end{pmatrix}$. Valid vectors are *normalized* linear combinations $\alpha(0) + \beta(1)$ with $\alpha, \beta \in \mathbb{C}^2$, i.e. elements of the form $\begin{pmatrix}\alpha\\\beta\end{pmatrix}$, where the condition $\alpha^2 + \beta^2 = 1$ holds, which we denote as C_i^2. This results in the following global state space, gathering several quantum and classical registers:

$$\underbrace{\underbrace{\mathcal{H}^{2n} \times \mathcal{H}^{2n}}_{\text{quantum registers}} \times \ldots \times \underbrace{C_i^2 \times \ldots \times C_i^2}_{\text{classical register}} \times \ldots}_{S}\,. \tag{146}$$

It is pretty straightforward that Cartesian product in classical registers, guarantee that states are always separable (convex), contrary to what happens in qubits. In conclusion the state space of a QASM program is given by the Cartesian product of the possible states of the independent quantum and classical registers, called Registers, where in the former the set of states is given by the *tensor product* of quantum bits, and in the latter by the possible distributions definable over the configurations of the classical *bits*. The number of qubits and bits available in each register is defined by reg_size. More concretely, the state space can be written as

$$S \equiv \prod_{\text{quantum register} \in Registers} \mathcal{H}^{2*reg\_size} \times \prod_{\text{classical register} \in Registers} \prod_{i \in \text{reg\_size}} C_i^2$$

For expressability purposes, the state space is represented by two families of functions $\pi^q_{r_{i_1}, r_{i_2}, \ldots, r_{i_n}} : S \to \mathcal{H}_{2n}$, which yield the current state of the list of qubits given by $r^q_{i_1}, r^q_{i_2}, \ldots, r^q_{i_n}$ and $\pi^c_{r_{i_1}, r_{i_2}, \ldots, r_{i_n}} : S \to$

1 Tests correspond to the σ-algebra over the valuation set $C = \{0,1\}$ of a single classical bit, the set of possible states corresponds to the distributions definable on the tests. For valuations with a discrete domain, it corresponds to the powerset 2^C. Tests form a Boolean algebra.

$\prod_i^n C_2^i$, which behaves similarly for the classical bits. A particular application of this are the functions $\pi_{r_i}^q : S \to \mathcal{H}^2$ and $\pi_{r_i}^c : S \to C^2$, which yield the state of a particular quantum and classical bits.

5.3.6 *Propositions*

The approach proposed in this work, based on keeping two different types of information, classical and quantum in the logical system, may be slightly problematic in what concerns the semantics of propositions. As usual, the semantics of a proposition corresponds to the set of states where it holds, hence

$$p : 2^S .$$

However, the different types of information influence the way propositions are interpreted.

Definition 5.3.1. Semantics for proposition constructors.

The definition of the primitive connectives is similar to the one used in modal logics, corresponding to set operations as follows:

1. $[\![p]\!] \subseteq 2^S$
2. $[\![\top]\!] = S$
3. $[\![i_{r_i}^q]\!]$ with $i \in \{0,1\} = \{s \in S | \pi_{r_i}^q(s) = |i\rangle\}$
4. $[\![0_{r_i}^c]\!] = \{s \in S | \pi_{r_i}^c(s) = (0)\}$, i.e. the classical 0 is a distribution, where 0 has probability 1.
5. $[\![1_{r_i}^c]\!] = \{s \in S | \pi_{r_i}^c(s) = (1)\}$, i.e. the classical 1 is a distribution, where 1 has probability 1.
6. $[\![\varphi_1 \wedge \varphi_2]\!] = [\![\varphi_1]\!] \cap [\![\varphi_2]\!]$
7. $[\![\neg\varphi]\!] = \{s | s \notin [\![\varphi]\!]\} = S - [\![\varphi]\!]$
   where $S - [\![\varphi]\!]$ stands for S except $[\![\varphi]\!]$, i.e. the complement of $[\![\varphi]\!]$ in S.
8. $[\![[\pi]\varphi]\!] = \{s | \forall u : (s,u) \in [\![\pi]\!] \Rightarrow u \in [\![\varphi]\!]\}$
   The set of states where the proposition φ holds upon the execution of program π (The semantics of programs π is given in section 5.3.7).
9. $[\![P^{\geq r}\varphi]\!] = \{s | \langle s \mid \varphi\rangle\langle\varphi \mid s\rangle \geq r\}$.
   The set of states where quantum proposition component φ holds with probability greater than r.

10. $[[\mathcal{A}^{=\lambda}\varphi]] = \{s|\langle\varphi \mid s\rangle = \lambda\}$. The states where the amplitude of a certain proposition is equal to a given constant λ.

5.3.7 *Program semantics*

The semantics of programs is given by a function from the set of well-formed programs to the power set of pairs of states, which entails the *accessibility relation*, i.e. the set of valid transitions between pairs of states (source to target) under the action of programs:

$$[[.]] : \mathcal{A}_\pi \to 2^{S\times S}. \tag{147}$$

A particular type of programs of this language are unitary programs, u, whose meaning reads as

- $[[u]] = \{(s,t) \in S \times S | t \in [[u(s)]] \wedge s \in [[u^{-1}(t)]]\}$

where $u(s)$, is the application of the unitary operator u to a state s, for instance, the Hadamard gate acting over state $|0\rangle$, i.e. $H.|0\rangle$. These rules apply to unitary operators h_i, z_i,x_i or $cnot_{ij}$ which correspond to the *quantum gates* H, Z, X, and $CNOT$, whose meaning was discussed in section 2.4.2. Indexes i,j correspond to the qubit indexes in the qubit array. It can be assumed that the unitary operator can be extended to the unaffected qubits by tensoring it with the unity operators, e.g. the semantics of a h_0 operator just affecting the first qubit in a system of two qubits, 0 and 1, reads as: $H \otimes I$.

The language also allows the existence of non-unitary operations, such as the creation of registers, classical and quantum, measurements of qubits, as well as if statements. Consider first the definition of quantum and classical registers, whose semantics read as follows:

- $[[\text{creg r [size]}]] = \{(s,t) \in S \times S | t \in [[\bigwedge_{i\in \text{size}} 0^c_{r_i}]]\}$ (Classical registry creation)
- $[[\text{qreg r [size]}]] = \{(s,t) \in S \times S | t \in [[\bigwedge_{i\in \text{size}} 0^q_{r_i}]]\}$ (Quantum registry creation)

Tests, both *quantum* and *classical*, possess a significant diference between each other: the former has a *destructive* effect on the state (the quantum bit it incides), while the latter conserves the state (the classical bit it insides). Their semantics reads as follows:

- $[[id^q_i == a]] = \{(s,t) \in S \times S | t \in \|a^q_{id_i}\rangle\langle a^q_{id_i}|(s)/\langle s|a^q_{id_i}\rangle\langle a^q_{id_i}|s\rangle\}$ (quantum test)
- $[[id^c_i == a]] = \{(s,t) \in S \times S | s \in [[P^{=x}a_{id^c_i}]] \wedge t \in [[P^{=x}a_{id^c_i}]]\}$ (classical test)

where $a \in 0,1$. The semantics of the measurement of quantum bit, can be understood as the branching of state into "alternative worlds", one where the test correspoding to 0 succeeded, with the probability of success of the test stored in the classical bit and the same for 1, which is expressed as a union of two sets as follows:

- $[\![\text{measure } q_i \rightarrow c_i]\!] = \{(s,t) \in S \times S | (s,t) \in [\![id_i^q \texttt{==} 0]\!] \wedge t \in [\![P^{=e_1} 0_{c_i}^c]\!]$ where $e_1 = \langle s|0_{id_i}^q\rangle\langle 0_{id_i}^q|s\rangle\} \cup \{(s,t) \in [\![q_i^q \texttt{==} 1]\!] \wedge [\![P^{=e_2} 1_{c_i}^c]\!]$ where $e_2 = \langle s|1_{id_i}^q\rangle\langle 1_{id_i}^q|s\rangle\}$ (Measurement)

If statements also consist of two alternative programs: one where, simultaneously, the classical test and the program effect holds, and the one where the classical test does not hold and the state remains unaltered, as follows:

- $[\![$ if id_i^c == b $\pi]\!] = \{(s,t) \in S \times S | (s,t) \in [\![|b\rangle_{id_i^c}\langle b|_{id_i^c}; \pi]\!] \wedge (s,t) \in [\![id_i^c == b]\!]\} \cup \{(s,t) \in [\![|\neg b\rangle_{id_i^c}\langle \neg b|_{id_i^c}; skip]\!] \wedge (s,t) \notin [\![id_i^c == b]\!]\}$, where $b \in \{0,1\}$ and skip is the identity operator.

Finally, the language also allows the sequencing (composition) of programs, for that the *accessibility relationship*, reads as follows:

- $[\![\pi_1; \pi_2]\!] = \{(s,u) \in S \times S | \exists t \in S : (s,t) \in [\![\pi_1]\!] t \wedge (t,u) \in [\![\pi_2]\!]\}$

5.4 Some valid rules and examples

In this section, we illustrate the semantics defined in the previous sections, by the proof of *soundness* of several rules and validities, as well as the correctness of both a coin tossing program and the teleportation protocol, expressed in the fragment of the QASM programming language. The proof strategy of showing the *validity* of a formula φ consists in verifying it is satisfied in the state-based model proposed in previous section, denoted as M, in every possible state: $\forall s \in S : M, s \models \varphi$ ($\models$ means that property φ is true in state s), or, equivalently, $[\![\varphi]\!] = [\![\top]\!]$.

5.4.1 *States, amplitudes and probabilities*

States can be defined by the amplitude operator for all elements of the basis of a quantum state, for instance, for a single qubit,

$$[\![\mathcal{A}^{\lambda_1} 0_{r_i}^q \wedge \mathcal{A}^{\lambda_2} 1_{r_i}^q]\!] = \{s \in S | \pi_{r_i}(s) = \lambda_1 |0\rangle + \lambda_2 |1\rangle\}\,.$$

The proof of this goes as follows, and makes use of the Born rule:

Proof: $[\![\mathcal{A}^{\lambda_1} 0_{r_i}^q \wedge \mathcal{A}^{\lambda_2} 1_{r_i}^q]\!] = \{s \in S | \pi_{r_i}(s) = \lambda_1 |0\rangle + \lambda_2 |1\rangle\}$

$\Leftrightarrow$

$[\![\mathcal{A}^{\lambda_1} 0_{r_i}^q]\!] \cap [\![\mathcal{A}^{\lambda_2} 1_{r_i}^q]\!] = \{s \in S | \pi_{r_i}(s) = \lambda_1 |0\rangle + \lambda_2 |1\rangle\}$

$\Leftrightarrow$ (by the Born rule, see equation 145)
$\{s \in S | \pi^q_{r_i} \sum_i \alpha_i |n\rangle \text{ with } \sum_i \alpha_i * \alpha_i^\dagger = 1\} \cap \{s \in S | \pi^q_{r_i}(s) = \lambda_1 |0\rangle\} \cap \{s | \pi^q_{r_i}(s) = \lambda_2 |1\rangle\} = \{s | \pi_{r_i}(s) = \lambda_1 |0\rangle + \lambda_2 |1\rangle\}$
$\Leftrightarrow$
$\{s \in S | \pi_{r_i}(s) = \lambda_1 |0\rangle + \lambda_2 |1\rangle\} = \{s \in S | \pi_{r_i}(s) = \lambda_1 |0\rangle + \lambda_2 |1\rangle\}$
□

This fact can also be generalized to systems of arbitrary dimension. Also an interesting observation that can be, comes from the need of normalization of bits/qubits:

$$P^{=x} a_{r_i} \rightarrow P^{=1-x} \neg a_{r_i}, \text{ where } a \in \{0, 1\}. \tag{148}$$

This also leads to the conclusion that

$$P^{=x} a_{r_i} \vee P^{=1-x} \neg a_{r_i} = P^{=x} a_{r_i} \wedge P^{=1-x} \neg a_{r_i} \tag{149}$$

,and the proof reads as follows:

Proof: $P^{=x} a_{r_i} \vee P^{=x} a_{r_i} = P^{=x} a_{r_i} \wedge P^{=1-x} \neg a_{r_i}$
$\Leftrightarrow$ (by the condition of equation (148))
$P^{=x} a_{r_i} \wedge P^{=1-x} \neg a_{r_i} \vee P^{=1-x} \neg a_{r_i} \wedge P^{=x} a_{r_i} = P^{=x} a_{r_i} \wedge P^{=1-x} \neg a_{r_i}$
$\Leftrightarrow$
$P^{=x} a_{r_i} \wedge P^{=1-x} \neg a_{r_i} = P^{=x} a_{r_i} \wedge P^{=1-x} \neg a_{r_i}$ □

Furthermore, the probability of a test inciding over a classical variable can be applied to the whole state:

$$(P^{=x} a^c_{r_i} \wedge \varphi) \rightarrow P^{=x}(a^c_{r_i} \wedge \varphi) \tag{150}$$

The veracity of this statement comes from the fact that the tests over a classical variable are always *compatible* with any other test that can be made in the system and hence the normal classical probability laws apply in these cases.

5.4.2 *Creation of registers*

The creation of registers, quantum or classical, is one of the possible actions of the QASM language, which sets them immediately to 0 in both cases. Therefore, the following rules hold:

- [creg r [size]] $\left(\bigwedge_{i \in \text{size}} 0^c_{r_i}\right)$
- [qreg r [size]] $\left(\bigwedge_{i \in \text{size}} 0^q_{r_i}\right)$

In both cases r corresponds to an arbitrary register description. We now give the proof of these rules, which are quite similar from the semantics defined in previous section.

Proof: Proof of rule **creg**
$[\![$[creg r [size]]$\left(\bigwedge_{i\in \text{size}} 0^c_{r_i}\right)]\!]$
= (by the definition of the dynamic operator $[\pi]\phi$)
$\{s \in S | \forall t : (s,t) \in [\![$[creg r [size]]$]\!] \Rightarrow t \in [\![\left(\bigwedge_{i\in \text{size}} 0^c_{r_i}\right)]\!]\}$
= (by the definition of the creation of classical register)
$\{s \in S | \forall t : t \in [\![\left(\bigwedge_{i\in \text{size}} 0^c_{r_i}\right)]\!]^{(1)} \Rightarrow t \in [\![\left(\bigwedge_{i\in \text{size}} 0^c_{r_i}\right)]\!]^{(2)}\}$
= (it can be easily verified that sets referred by $^{(1)}$,$^{(2)}$ are non-empty)
$\{s \in S | true\} = S = [\![\top]\!]$
□

Proof of rule **qreg** is exactly the same of **creg**, it is just necessary to use the definition of the creation of quantum register rather than classical register.

5.4.3 *Unitary gates*

In essence quantum programs are unitary gates, which, however while involving most of the times, one or two qubits. The proofs for these instructions are made in the semantic setting. More a demonstrative example we show the following validity $0_{q_0} \rightarrow [hq_0](\mathcal{A}^{\frac{1}{\sqrt{2}}} 0_{q_0} \wedge \mathcal{A}^{\frac{1}{\sqrt{2}}} 1_{q_0})$, which we denote **h0**.

Proof: Proof of **h0**
$[\![0^q_{q_0} \rightarrow [hq_0](\mathcal{A}^{=\frac{1}{\sqrt{2}}} 0^q_{q_0} \wedge \mathcal{A}^{=\frac{1}{\sqrt{2}}} 1^q_{q_0})]\!]$
= (by the definition of implication)
$[\![\neg\left(0^q_{q_0} \wedge \neg([hq_0](\mathcal{A}^{\frac{1}{\sqrt{2}}} 0^q_{q_0} \wedge \mathcal{A}^{\frac{1}{\sqrt{2}}} 1^q_{q_0}))\right)]\!]$
= (by the modal negation $\neg[\pi]\phi = \neg(\neg\langle\pi\rangle\neg\phi)$)
$[\![\neg\left(0^q_{q_0} \wedge (\langle hq_0\rangle\neg(\mathcal{A}^{\frac{1}{\sqrt{2}}} 0^q_{q_0} \wedge \mathcal{A}^{\frac{1}{\sqrt{2}}} 1^q_{q_0}))\right)]\!]$
= (by the definition of the dynamic operator $\langle\pi\rangle\phi$)
$S - ([\![0^q_{q_0}]\!] \cap \{s \in S | \exists t : (s,t) \in [\![$h $q_0]\!] \Rightarrow t \notin [\![(\mathcal{A}^{\frac{1}{\sqrt{2}}} 0^q_{q_0} \wedge \mathcal{A}^{\frac{1}{\sqrt{2}}} 1^q_{q_0})]\!]\})$
=
$S - \{s \in S | \exists t : s \in [\![0^q_{q_0}]\!] \wedge (s,t) \in [\![$h $q_0]\!] \Rightarrow t \notin [\![(\mathcal{A}^{\frac{1}{\sqrt{2}}} 0^q_{q_0} \wedge \mathcal{A}^{\frac{1}{\sqrt{2}}} 1^q_{q_0})]\!]\}$
= (by the definition of the h instruction and the semantics of state s)
$S - \{s \in S | \exists t : \pi_{q_0}(s) = |0\rangle \wedge \pi_{q_0}(t) = H|0\rangle \Rightarrow t \notin [\![(\mathcal{A}^{\frac{1}{\sqrt{2}}} 0^q_{q_0} \wedge \mathcal{A}^{\frac{1}{\sqrt{2}}} 1^q_{q_0})]\!]\}$
= (the action of the h instruction results in the state $|+\rangle$)

$S - \{s \in S | \exists t : s \in \pi_{q_0}(s) = |0\rangle \wedge \pi_{q_0}(t) = \mathcal{A}^{\frac{1}{\sqrt{2}}}(|0\rangle + |1\rangle) \Rightarrow t \notin [\![(\mathcal{A}^{\frac{1}{\sqrt{2}}} 0^q_{q_0} \wedge \mathcal{A}^{\frac{1}{\sqrt{2}}} 1^q_{q_0})]\!]\}$
$=$ (by the definition of the amplitude operator on the $|+\rangle$ state)
$S - \{s \in S | \exists t : s \in [\![0^q_{q_0}]\!]^{(1)} \wedge t \in [\![(\mathcal{A}^{\frac{1}{\sqrt{2}}} 0^q_{q_0} \wedge \mathcal{A}^{\frac{1}{\sqrt{2}}} 1^q_{q_0}]\!]^{(2)} \Rightarrow t \notin [\![\mathcal{A}^{\frac{1}{\sqrt{2}}} 0^q_{q_0} \wedge \mathcal{A}^{\frac{1}{\sqrt{2}}} 1^q_{q_0}]\!]^{(2)}\}$
$=$ (it can be easily verified that sets referred by $^{(1)}, ^{(2)}$ are non-empty)
$[\![\neg\bot]\!] = [\![\top]\!]$ □

5.4.4 *Measurements*

Measurements are also an important part, and in this sequel, only single qubit measurements are allowed, causing the *ramification* of into two consistent worlds each with the probability of obtaining one and zero, causing the *transference* of probability distribution from the qubit to the classical bit. An example of this is given by the validity $(\mathcal{A}^{\frac{1}{\sqrt{2}}} 0_{q_0} \wedge \mathcal{A}^{\frac{1}{\sqrt{2}}} 1_{q_0}) \rightarrow [\text{meas } q_0 \text{ to } c_0](P^{=0.5} 0_{c_0} \wedge P^{=0.5} 1_{c_0})$, which we denote **m1**.

Proof: Proof of validity **m1**
$[\![(\mathcal{A}^{\frac{1}{\sqrt{2}}} 0^q_{q_0} \wedge \mathcal{A}^{\frac{1}{\sqrt{2}}} 1^q_{q_0}) \rightarrow [\text{meas } q_0 \text{ to } c_0](P^{=0.5} 0^c_{c_0} \wedge P^{=0.5} 1^c_{c_0})]\!]$
$=$ (by the definition of implication)
$S - ([\![(\mathcal{A}^{\frac{1}{\sqrt{2}}} 0^q_{q_0} \wedge \mathcal{A}^{\frac{1}{\sqrt{2}}} 1^q_{q_0})]\!] \cap [\![\neg[\text{meas } q_0 \text{ to } c_0](P^{=0.5} 0^c_{c_0} \wedge P^{=0.5} 1^c_{c_0})]\!])$
$=$ (by the modal negation $\neg[\pi]\phi = \neg(\neg\langle\pi\rangle\neg\phi)$)
$S - ([\![(\mathcal{A}^{\frac{1}{\sqrt{2}}} 0^q_{q_0} \wedge \mathcal{A}^{\frac{1}{\sqrt{2}}} 1^q_{q_0})]\!] \cap [\![\langle\text{meas } q_0 \text{ to } c_0\rangle\neg(P^{=0.5} 0^c_{c_0} \wedge P^{=0.5} 1^c_{c_0})]\!])$
$=$ (by the definition of the dynamic operator $\langle\pi\rangle\phi$)
$S - \{s \in S | \exists t : s \in [\![(\mathcal{A}^{\frac{1}{\sqrt{2}}} 0^q_{q_0} \wedge \mathcal{A}^{\frac{1}{\sqrt{2}}} 1^q_{q_0})]\!] \wedge (s,t) \in [\![[\text{meas } q_0 \text{ to } c_0]]\!] \Rightarrow t \notin [\![(P^{=0.5} 0^c_{c_0} \wedge P^{=0.5} 1^c_{c_0})]\!]\}$
$=$ (by the definition of measurement)
$S - (\{s \in S | \exists t : s \in [\![(\mathcal{A}^{\frac{1}{\sqrt{2}}} 0^q_{q_0} \wedge \mathcal{A}^{\frac{1}{\sqrt{2}}} 1^q_{q_0})]\!] \wedge (s,t) \in (\{(s,t) | t \in [\![q^q_0 == 0]\!] \wedge t \in [\![P^{=0.5} 0^c_{c_0}]\!]\} \cup \{(s,t) | t \in [\![q^q_0 == 1]\!] \wedge t \in [\![P^{0.5} 1^c_{c_0}]\!]\}) \Rightarrow t \notin [\![(P^{=0.5} 0^c_{c_0} \wedge P^{=0.5} 1^c_{c_0})]\!]\})$
$=$
$S - (\{s \in S | \exists t : s \in [\![(\mathcal{A}^{\frac{1}{\sqrt{2}}} 0^q_{q_0} \wedge \mathcal{A}^{\frac{1}{\sqrt{2}}} 1^q_{q_0})]\!] \wedge t \in [\![P^{=0.5} 0^c_{c_0} \vee P^{=0.5} 1^c_{c_0}]\!] \Rightarrow t \notin [\![(P^{=0.5} 0^c_{c_0} \wedge P^{=0.5} 1^c_{c_0})]\!]\})$
$=$ (as exposed in equation (149))
$S - (\{s \in S | \exists t : s \in [\![(\mathcal{A}^{\frac{1}{\sqrt{2}}} 0^q_{q_0} \wedge \mathcal{A}^{\frac{1}{\sqrt{2}}} 1^q_{q_0})]\!]^{(1)} \wedge t \in [\![P^{=0.5} 0^c_{c_0} \wedge P^{=0.5} 1^c_{c_0}]\!]^{(2)} \Rightarrow t \notin [\![(P^{=0.5} 0^c_{c_0} \wedge P^{=0.5} 1^c_{c_0})]\!]^{(2)}\})$
$=$(it can be easily verified that sets referred by $^{(1)}, ^{(2)}$ are non-empty)
$[\![\neg\bot]\!]$

$=$
$[\![\top]\!]$
$\square$

5.4.5 *A Hoare style sequence rule*

Hoare logic is a formal system to deal with simple *while* languages, introduced by Hoare in 1969 [200], which constitutes an important landmark in computer science. One of the central rules of such logic is the sequence rule, which reads as

$$\frac{\{P\}\pi_1\{I\} \quad \{I\}\pi_2\{R\}}{\{P\}\pi_1;\pi_2\{R\}}$$

where P and R are *pre* and *post* conditions for programs, π_1 and π_2 are programs, and $\pi_1;\pi_2$ is a sequence of programs π_1 and π_2, i.e. the rule defines how programs compose, given the proofs of the individual programs. It is also well-known that Hoare logic can be retrieved from dynamic logic: the so-called *Hoare* triple $\{P\}\pi\{R\}$, can expressed as an expression $p \to [\pi]r$, which has the interpretation p *being true implies that it is necessary that* r *is true upon the execution of program* π. Given this, the sequence rule of Hoare can be translated into the following expression:

$$\frac{P \to [\pi_1]I \quad I \to [\pi_2]R}{P \to [\pi_1;\pi_2]R}$$

which also corresponds to the following expression $P \to [\pi_1]I \wedge I \to [\pi_2]R \leftrightarrow P \to [\pi_1;\pi_2]R$. We show this rule also holds here. Firstly, it can be easily shown that $[\pi_1;\pi_2]\phi \leftrightarrow [\pi_1][\pi_2]\phi$ holds:

Proof: $[\pi_1;\pi_2]\phi \leftrightarrow [\pi_1][\pi_2]\phi$
$\equiv$
$\{s \in S | \forall u : (s,u) \in [\![\pi_1;\pi_2]\!] \Rightarrow u \in [\![\phi]\!]\} = [\![[\pi_1][\pi_2]\phi]\!]$
$\equiv$(by definition of composition)
$\{s \in S | \forall u : (s,u) \in \{(s,u) | \exists t : (s,t) \in [\![\pi_1]\!] \wedge (t,u) \in [\![\pi_2]\!]\} \Rightarrow u \in [\![\phi]\!]\} = [\![[\pi_1][[\pi_2]\phi]]\!]$
$\equiv$
$\{s \in S | \forall u \forall t : (s,t) \in [\![\pi_1]\!] \wedge (t,u) \in [\![\pi_2]\!] \Rightarrow u \in [\![\phi]\!]\} = [\![[\pi_1][\pi_2]\phi]\!]$
$\equiv$
$\{s \in S | \forall t : (s,t) \in [\![\pi_1]\!] \Rightarrow t \in \{t | \forall (t,u) \in [\![[\pi_2]]\!] \Rightarrow u \in [\![\phi]\!]\}\} = [\![[\pi_1][\pi_2]\phi]\!]$
$\equiv$
$\{s \in S | \forall t : (s,t) \in [\![\pi_1]\!] \Rightarrow t \in [\![[\pi_2]\phi]\!]\} = [\![[\pi_1][\pi_2]\phi]\!]$

$\equiv$

$[\pi_1][\pi_2]\phi \leftrightarrow [\pi_1][\pi_2]\phi$ □

Thus,

$P \rightarrow [\pi_1]I \wedge I \rightarrow [\pi_2]R \leftrightarrow P \rightarrow [\pi_1;\pi_2]R$

$\equiv$ (by the substitution of I for $[\pi_2]R$)

$P \rightarrow [\pi_1][\pi_2]R \leftrightarrow P \rightarrow [\pi_1;\pi_2]R$

$\equiv$

$P \rightarrow [\pi_1;\pi_2]R \leftrightarrow P \rightarrow [\pi_1;\pi_2]R$

5.4.6 *Putting it all together: A quantum coin tossing program*

In this section, using the validities and rules of previous sections, we illustrate the use of the logic through the proof of correctness of a *simple* quantum program for *quantum coin tossing* (prepare a qubit in a superposition state and measure it, obtaining 0 or 1 with equal probability), which translates into the following QASM program:

```
OPENQASM 2.0;
include "qelib1.inc";
qreg q[1];
creg c[1];
h q[0];
measure q[0] -> c[0];
```

The correctness of such program implies the following post-condition:

$$[\text{qreg q [1]; creg c [1]; } h\ q_0\text{; meas } q_0 \text{ to } c_0]\ \left(P^{=0.5}0_{c_0} \wedge P^{=0.5}1_{c_0}\right)$$

Proof: by making use of **sequence** rule, as well as **creg** and **qreg** validities, the following inference is true:

$$\frac{[\text{qreg q [1]}](0^q_{q_0}) \quad [\text{creg c[1]}](0^c_{c_0})}{[\text{qreg q[1]; creg c[1]}](0^q_{q_0} \wedge 0^c_{c_0})}\ (1)$$

The following inference is true, through the application of validity **h0** and the **sequence** rule

$$\frac{(1)\quad 0_{q_0} \rightarrow [h\ q_0](\mathcal{A}^{\frac{1}{\sqrt{2}}} 0^q_{q_0} \wedge \mathcal{A}^{\frac{1}{\sqrt{2}}} 1^q_{q_0})}{[\text{qreg q[1]; creg c[1]; } h\ q_0](\mathcal{A}^{\frac{1}{\sqrt{2}}} 0_{q_0} \wedge \mathcal{A}^{\frac{1}{\sqrt{2}}} 1_{q_0} \wedge 0_{c_0})}\ (2)$$

And finally the post-condition is show to be true by validity **m1** and the **sequence** rule, completing the proof

$$\frac{(2)\quad (\mathcal{A}^{\frac{1}{\sqrt{2}}} 0^q_{q_0} \wedge \mathcal{A}^{\frac{1}{\sqrt{2}}} 1^q_{q_0}) \rightarrow [\text{meas } q_0 \text{ to } c_0](P^{=0.5} 0^c_{c_0} \wedge P^{=0.5} 1^c_{c_0})}{[\text{qreg q[1]; creg c[1]; } h\ q_0\text{;meas } q_0 \text{ to } c_0](P^{=0.5} 0^c_{c_0} \wedge P^{=0.5} 1^c_{c_0})}\ (3)$$

□

5.4.7 *The teleporting protocol*

The teleportation protocol was introduced by Bennet et al. [55] in 1993, and it is the cornerstone of many quantum communication protocols. It allows the replication of the state of a qubit onto a second qubit, where each of the qubits are held by two different parties, physically separated. The protocol works, regardless of the distance between parties, through the use of a *Bell* pair shared between both parties. The protocol, besides its enormous applicability, is also one of the most fundamental and paradigmatic examples of the employment of *quantum entanglement*, and it is a natural *testbed* for any new quantum formal method. In this section, we show that it is also possible to prove the correctness of the teleportation protocol in the logic developed in this chapter.

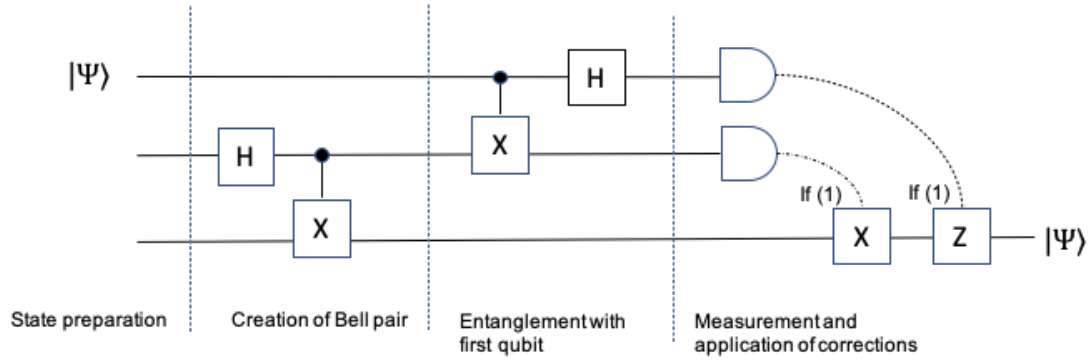


Figure 23: Circuit of the quantum teleportation protocol as proposed in Bennet et al. [55]

The protocol can be expressed by a quantum circuit involving three qubits, as depicted in figure 23, where the following parts can be identified:

- State preparation, which involves the preparation of the first qubit and the creation of the *Bell pair* between the second and the third qubits;

- Association of the Bell pair with the first qubit;
- Measurement of the first qubits and application of the results for post-correction on the third qubit, which requires the exchange of classical information between the holders of the first qubit and of the third one.

In the real-world scenario, qubit 3 is physically separated from qubits 1 and 2, and the former and the latters are controlled by two distinct parties, which have to possess access to a classical channel to convey classical information. Such information is obtained from the measurement of qubits 1 and 2 by the first party, which then conveys the results of the readings to party 2, which uses them to apply post corrections on qubit 3, hence retrieving the original state of qubit 0, held by party one. This process can be translated into the following QASM program (qubit 0 can be mapped to the upper line in the circuit of figure 23, qubit 1 to the middle line, and qubit 2 to the bottom one):

```
OPENQASM 2.0;
include "qelib1.inc";
qreg q[3];
creg c[2];
p0(α) q0;
h q [1];
cx q[1], q[2];
cx q[0], q[1];
h q [0];
measure q[0] → c[0];
measure q[1] → c[1];
if (c [1] == 1) x q[2];
if (c [0] == 1) z q[2];
```

The correctness of the teleportation protocol translates into the following statement:

$$[\text{qreg q[3]; creg c[2]; } p_0(\alpha)\ q_0\text{; h } q_1^q\text{; cx } q_1^q, q_2^q\text{; cx } q_0^q, q_1^q\text{; h } q_0^q\text{;}\\ \text{meas } q_0^q \text{ to } c_0^c\text{; meas } q_1^q \text{ to } c_1^c\text{; if (c [0] == 1) x } q_2^q\text{; if (c [1] == 1) z } q_2^q]\ \left(P^{=\alpha_1}0_2^q \wedge P^{=1-\alpha}1_2^q\right)\ ,$$

where $P^{=\alpha}0_2^q \wedge P^{=1-\alpha}1_2^q$ can be interpreted of the post-condition of the protocol, determining the target probability distribution on qubit 2: α for obtaining 0 upon measurement, and $1-\alpha$ for 1. Such probability

distribution is the one set up in qubit 0, by operator $p_0(\alpha)$, during the preparation stage. In a real-world scenario, the preparation stage is the one by two separate and independent parties, i.e. both the state preparation of qubit 0 and the existence of the Bell pair are pre-conditions of the actual teleportation.

The state preparation stage

The state preparation is the subprogram [qreg q [3];creg c[2]; $p_0(\alpha)$;h q_1^q; cx q_1^q, q_2^q], also denoted PREP, which includes the declaration of classical and quantum registers and the preparation of the first qubit according to the desired observational properties, probability α for obtention of 0 upon measurement and $1-\alpha$, for 1, as well as the creation of the Bell pair. Making use of **sequence** rule, as well as **creg** and **qreg** validities, the following inference is true:

$$\frac{\text{[qreg q [3]]}(0^q_{q_0} \wedge 0^q_{q_1} \wedge 0^q_{q_2}) \quad \text{[creg c[2]]}(0^c_{c_0} \wedge 0^c_{c_1})}{\text{[qreg q [3];creg c[2]]}(0^q_{q_0} \wedge 0^q_{q_1} \wedge 0^q_{q_2} \wedge 0^c_{c_0} \wedge 0^c_{c_1})} \; (i)$$

One can also assume the existence of an operator $p_0(\alpha)$, that can prepare a qubit on the state with the desired probability distribution, i.e. α for 0, and $1-\alpha$ for 1, through setting up the amplitudes λ_1 and λ_2,

$$0^q_{q_0} \rightarrow [p_0(\alpha)q_0](\mathcal{A}^{=\lambda_1}0_{q_0} \wedge \mathcal{A}^{=\lambda_2}1_{q_0}),$$

and that

$$\mathcal{A}^{\lambda_1}0^q_{x_x} \wedge \mathcal{A}^{\lambda_2}1^q_{x_x} \rightarrow P^{=\alpha}0^q_{x_x} \wedge P^{=1-\alpha}1^q_{x_x}. \tag{151}$$

Hence, the following inference is true

$$\frac{(i) \quad 0^q_{q_0} \rightarrow [p_0(\alpha)]\mathcal{A}^{=\lambda_1}0_{q_0} \wedge \mathcal{A}^{=\lambda_2}1_{q_0})}{\text{[qreg q [3];creg c[2]; } p_0(\alpha)\ q_0](\mathcal{A}^{=\lambda_1}0_{q_0} \wedge \mathcal{A}^{=\lambda_2}1_{q_0}) \wedge 0^q_{q_1} \wedge 0^q_{q_2} \wedge 0^c_{c_0} \wedge 0^c_{c_1}} \; (ii)$$

The following steps are the creation of the *BELL* pair between qubits 1 and 2, which is performed by program h q [1]; cx q [1], q [2]], which we also denote $\mathbf{Bell}_{12}$. The following inference is true for the application of the Hadamard operator on qubit 1, i.e. by program h q [1], (see proof **h2**),

$$\frac{\text{(ii)} \quad 0_{q_1} \rightarrow \text{[h q [1]]}\left(\mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1}) \wedge \mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1})\right)}{\text{[PREP; h } q_1](\mathcal{A}^{=\frac{1}{\sqrt{2}}}0_{q_1} \wedge \mathcal{A}^{=\frac{1}{\sqrt{2}}}(1_{q_1}) \wedge 0_{q_2} \wedge 0_{c_0} \wedge 0^c_{c_1})} \; (iii),$$

and when composed with the operator CNOT, it is possible to conclude the following (see proof **cnot1**):

$$\frac{\text{(iii)}\quad \left(\mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1}) \wedge \mathcal{A}^{=\frac{1}{\sqrt{2}}}(1^q_{q_1}) \wedge 0^q_{q_2}\right) \rightarrow [CNOT_{12}]\left(\mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=\frac{1}{\sqrt{2}}}(1^q_{q_1} \wedge 1^q_{q_2})\right)}{[\text{PREP; h } q^q_1\text{; cx } q^q_1, q^q_2]((\mathcal{A}^{=\lambda_1}(0_{q_0}) \wedge \mathcal{A}^{=\lambda_2}(1^q_{q_0})) \wedge \left(\mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=\frac{1}{\sqrt{2}}}(1^q_{q_1} \wedge 1^q_{q_2})\right))}\ (iv)\ .$$

This defines the state of the system upon system preparation, i.e. of program PREP;$\mathbf{Bell}_{12}$.

Proofs of the state preparation stage

Proof: (**h2**)
$[\![0_{q_1} \rightarrow [\text{h q [1]}]\left(\mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1}) \wedge \mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1})\right)]\!]$
= (by the definition of the h instruction and the semantics of state s)
$S - \{s|\exists t : \pi_{q_0}(s) = |0\rangle \wedge \pi_{q_0}(t) = H|0\rangle \Rightarrow t \notin [\![\left(\mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1}) \wedge \mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1})\right)]\!]\}$
= (the action of the h instruction results in the state $|+\rangle$)
$S - \{s|\exists t : s \in \pi_{q_0}(s) = |0\rangle \wedge \pi_{q_0}(t) = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \Rightarrow t \notin [\![\left(\mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1}) \wedge \mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1})\right)]\!]\}$
= (by the definition of the amplitude operator on the $|+\rangle$ state)
$S - \{s|\exists t : s \in [\![0^q_{q_0}]\!]^{(1)} \wedge t \in [\![\left(\mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1}) \wedge \mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1})\right)]\!]^{(2)} \Rightarrow$
$t \notin [\![\left(\mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1}) \wedge \mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1})\right)]\!]^{(2)}\}$
=(it can be easily verified that sets referred by $^{(1)}$,$^{(2)}$ are non-empty)
$[\![\top]\!]$
□

Proof: (**cnot1**)
$[\![\left(\mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1}) \wedge \mathcal{A}^{=\frac{1}{\sqrt{2}}}(1^q_{q_1}) \wedge 0^q_{q_2}\right) \rightarrow [\text{cx } q^q_1, q^q_2]\left(\mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=\frac{1}{\sqrt{2}}}(1^q_{q_1} \wedge 1^q_{q_2})\right)]\!]$
= (by the definition of the CNOT instruction and the semantics of state s)
$S - \{s|\exists t : \pi_{q_1,q_2}(s) = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \wedge \pi_{q_1,q_2}(t) = CNOT.\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \Rightarrow t \notin [\![\mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=\frac{1}{\sqrt{2}}}(1^q_{q_1} \wedge 1^q_{q_2})]\!]\}$
= (the action of the h instruction results in the state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$)
$S - \{s|\exists t : \pi_{q_1,q_2}(s) = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \wedge \pi_{q_1,q_2}(t) = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \Rightarrow t \notin [\![\mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=\frac{1}{\sqrt{2}}}(1^q_{q_1} \wedge 1^q_{q_2})]\!]\}$
= (by the definition of the amplitude operator one can express the semantics of the state as follows)

$S - \{s|\exists t : s \in [\![\left(\mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1}) \wedge \mathcal{A}^{=\frac{1}{\sqrt{2}}}(1^q_{q_1}) \wedge 0^q_{q_2} \right)]\!]^{(1)} \wedge$
$t \in [\![\left(\mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=\frac{1}{\sqrt{2}}}(1^q_{q_1} \wedge 1^q_{q_2}) \right)]\!]^{(2)} \Rightarrow$
$t \notin [\![(\mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=\frac{1}{\sqrt{2}}}(1^q_{q_1} \wedge 1^q_{q_2})]\!]^{(2)}\}$
$=$ (it can be easily verified that sets referred by $^{(1)}$,$^{(2)}$ are non-empty)
$[\![\top]\!]$
□

The association with the first qubit

The next step of the protocol, and indeed the actual step in the real-world scenario is to associate qubit 0 and the Bell state of qubits 1 and 2, which is made by the CNOT operator, i.e. through the program cx q^q_0, q^q_1, also denoted CNOT_{01}. From the sequence rule and the semantics of the CNOT operator (see proof **cnot2**), one can infer the following

$$\frac{\begin{array}{ll} & (\mathcal{A}^{=\lambda_1}(0^q_{q_0}) \wedge \mathcal{A}^{=(\lambda_2)}(1^q_{q_0})) \wedge (\mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{\frac{1}{\sqrt{2}}}(1^q_{q_1} \wedge 1^q_{q_2})) \\ \text{(iv)} & \rightarrow [\mathsf{CNOT}_{01}] \begin{array}{l} ((\mathcal{A}^{=(\frac{1}{\sqrt{2}}*\lambda_1)}(0^q_{q_0} \wedge 0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=(\frac{1}{\sqrt{2}}*(\lambda_2)}(1^q_{q_0} \wedge 0^q_{q_1} \wedge 0^q_{q_2}) \\ \wedge \mathcal{A}^{=(\frac{1}{\sqrt{2}}*\lambda_1)}(0^q_{q_0} \wedge 1^q_{q_1} \wedge 1^q_{q_2}) \wedge \mathcal{A}^{=\frac{1}{\sqrt{2}}*(\lambda_2)}(1^q_{q_0} \wedge 0^q_{q_1} \wedge 1^q_{q_2})) \end{array} \end{array}}{\begin{array}{ll} [\mathsf{PREP}; \mathbf{Bell}_{12}; \mathsf{CNOT}_{01}] & (\mathcal{A}^{=(\frac{1}{\sqrt{2}}*\lambda_1)}(0^q_{q_0} \wedge 0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=(\frac{1}{\sqrt{2}}*(\lambda_2)}(1^q_{q_0} \wedge 0^q_{q_1} \wedge 0^q_{q_2}) \\ & \wedge \mathcal{A}^{=(\frac{1}{\sqrt{2}}*\lambda_1)}(0^q_{q_0} \wedge 1^q_{q_1} \wedge 1^q_{q_2}) \wedge \mathcal{A}^{=\frac{1}{\sqrt{2}}*(\lambda_2)}(1^q_{q_0} \wedge 0^q_{q_1} \wedge 1^q_{q_2})) \end{array}} (v)$$

which characterizes the resultant state of such operations. We denote the program PREP; $\mathbf{Bell}_{12}$;CNOT_{01} as ENT, for a matter of syntax simplification.

Proof: (**cnot2**) Let p denote the following expression

$$p = ((\mathcal{A}^{=(\frac{1}{\sqrt{2}}*\lambda_1)}(0^q_{q_0} \wedge 0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=(\frac{1}{\sqrt{2}}*(\lambda_2)}(1^q_{q_0} \wedge 0^q_{q_1} \wedge 0^q_{q_2}) \\ \wedge \mathcal{A}^{=(\frac{1}{\sqrt{2}}*\lambda_1)}(0^q_{q_0} \wedge 1^q_{q_1} \wedge 1^q_{q_2}) \wedge \mathcal{A}^{=\frac{1}{\sqrt{2}}*(\lambda_2)}(1^q_{q_0} \wedge 0^q_{q_1} \wedge 1^q_{q_2}))\,.$$

The proof goes as follows:
$[\![(\mathcal{A}^{=\lambda_1}(0^q_{q_0}) \wedge \mathcal{A}^{=(\lambda_2)}(1^q_{q_0})) \wedge (\mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{\frac{1}{\sqrt{2}}}(1^q_{q_1} \wedge 1^q_{q_2})) \rightarrow [\mathsf{CNOT}_{01}]p]\!]$
$=$ (by the definition of the h instruction and the semantics of state s)
$S - \{s|\exists t : \pi_{q_0,q_1,q_2}(s) = (\lambda_1 * \frac{1}{\sqrt{2}}|000\rangle + \lambda_1 * \frac{1}{\sqrt{2}}|011\rangle + \lambda_2 * \frac{1}{\sqrt{2}}|100\rangle + \lambda_2 * \frac{1}{\sqrt{2}}|111\rangle) \wedge$

$\pi_{q_0,q_1,q_2}(t) = (CNOT \otimes I).\pi_{q_0,q_1,q_2}(s) \Rightarrow t \notin [[p]]\}$
= (the action of the h instruction results in the state $|+\rangle$)
$S - \{s|\exists t : \pi_{q_0,q_1,q_2}(s) = (\lambda_1 * \frac{1}{\sqrt{2}}|000\rangle + \lambda_1 * \frac{1}{\sqrt{2}}|011\rangle + \lambda_2 * \frac{1}{\sqrt{2}}|100\rangle + \lambda_2 * \frac{1}{\sqrt{2}}|111\rangle) \wedge$
$\pi_{q_1,q_2}(t) = (\lambda_1 * \frac{1}{\sqrt{2}}|000\rangle + \lambda_1 * \frac{1}{\sqrt{2}}|011\rangle + \lambda_2 * \frac{1}{\sqrt{2}}|110\rangle + \lambda_2 * \frac{1}{\sqrt{2}}|101\rangle) \Rightarrow t \notin [[p]]\}$
= (by the definition of the probabilistic operator on the $|+\rangle$ state)
$S - \{s|\exists t : s \in [[(\mathcal{A}^{=\lambda_1}(0^q_{q_0}) \wedge \mathcal{A}^{=(\lambda_2)}(1^q_{q_0})) \wedge (\mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{\frac{1}{\sqrt{2}}}(1^q_{q_1} \wedge 1^q_{q_2}))]]^1 \wedge t \in$
$[[p]]^2 \Rightarrow t \notin [[p]]^2\}$
= (it can be easily verified that sets referred by $^{(1)}$,$^{(2)}$ are non-empty)
$[[\top]]$
□

Measuring qubits and classical information

The resulting state of program ENT holds a superposition of all the possible Bell states with the original amplitudes λ_1 and λ_2. All that is necessary is to eliminate the Bell states from the global state, in order to retrieve back the original state on the third qubit. This is done by the program h q_0;meas q^q_0 to c^c_0; meas q^q_1 to c^c_1; if (c [0] == 1) x q_2; if (c [1] == 1) z q_2, which measures the first two qubits, in two different axis each (two measures are enough to identify the Bell state) and uses the results of the measurements in two independent *if statements* to eliminate the Bell states. The first step of the measurement is the application is the application of the operator H over qubit 0, in order to be able to measure in *Hadamard basis*, i.e. $|+\rangle, |-\rangle$ rather than $|0\rangle, |1\rangle$ (see proof **h3**), leading to:

$$\text{(v)}\quad \begin{array}{l} (\mathcal{A}^{=(\frac{1}{\sqrt{2}}*\lambda_1)}(0^q_{q_0} \wedge 0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=(\frac{1}{\sqrt{2}}*(\lambda_2)}(1^q_{q_0} \wedge 0^q_{q_1} \wedge 0^q_{q_2}) \\ \wedge \mathcal{A}^{=(\frac{1}{\sqrt{2}}*\lambda_1)}(0^q_{q_0} \wedge 1^q_{q_1} \wedge 1^q_{q_2}) \wedge \mathcal{A}^{=\frac{1}{\sqrt{2}}*(\lambda_2)}(1^q_{q_0} \wedge 0^q_{q_1} \wedge 1^q_{q_2})) \\ \rightarrow [hq_0] \begin{array}{l} \mathcal{A}^{=(\frac{1}{2}*\lambda_1)}(0^q_{q_0} \wedge 0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=(\frac{1}{2}*\lambda_2)}(0^q_{q_0} \wedge 0^q_{q_1} \wedge 1^q_{q_2}) \\ \wedge \mathcal{A}^{=(\frac{1}{2}*\lambda_2)}(0^q_{q_0} \wedge 1^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=(\frac{1}{2}*\lambda_1)}(0^q_{q_0} \wedge 1^q_{q_1} \wedge 1^q_{q_2}) \\ \wedge \mathcal{A}^{=(\frac{1}{2}*\lambda_1)}(1^q_{q_0} \wedge 0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=(-\frac{1}{2}*\lambda_2)}(1^q_{q_0} \wedge 0^q_{q_1} \wedge 1^q_{q_2}) \\ \wedge \mathcal{A}^{=(-\frac{1}{2}*\lambda_2)}(1^q_{q_0} \wedge 1^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=(\frac{1}{2}*\lambda_1)}(1^q_{q_0} \wedge 1^q_{q_1} \wedge 1^q_{q_2}) \end{array} \end{array} \quad (vi)$$

$$\top \rightarrow [\text{ENT;h } q_0] \quad \begin{array}{l} \mathcal{A}^{=(\frac{1}{2}*\lambda_1)}(0^q_{q_0} \wedge 0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=(\frac{1}{2}*\lambda_2)}(0^q_{q_0} \wedge 0^q_{q_1} \wedge 1^q_{q_2}) \\ \wedge \mathcal{A}^{=(\frac{1}{2}*\lambda_2)}(0^q_{q_0} \wedge 1^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=(\frac{1}{2}*\lambda_1)}(0^q_{q_0} \wedge 1^q_{q_1} \wedge 1^q_{q_2}) \\ \wedge \mathcal{A}^{=(\frac{1}{2}*\lambda_1)}(1^q_{q_0} \wedge 0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=(-\frac{1}{2}*\lambda_2)}(1^q_{q_0} \wedge 0^q_{q_1} \wedge 1^q_{q_2}) \\ \wedge \mathcal{A}^{=(-\frac{1}{2}*\lambda_2)}(1^q_{q_0} \wedge 1^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=(-\frac{1}{2}*\lambda_1)}(1^q_{q_0} \wedge 1^q_{q_1} \wedge 1^q_{q_2}) \end{array}$$

The following step is the actual measurement of qubits 0 and 1, storing the results in the classical bits. This can be done by programs meas q_0^q to c_0^c and meas q_1^q to c_1^c, which we denote as M1 and M2, respectively. The effect of the measurements gives origin to the following inference (see proof **meast**):

$$\frac{\begin{array}{l}\mathcal{A}^{=\left(\frac{1}{2}*\lambda_1\right)}(0^q_{q_0}\wedge 0^q_{q_1}\wedge 0^q_{q_2})\wedge\mathcal{A}^{=\left(\frac{1}{2}*\lambda_2\right)}(0^q_{q_0}\wedge 0^q_{q_1}\vee 1^q_{q_2})\\ \wedge\mathcal{A}^{=\left(\frac{1}{2}*\lambda_2\right)}(0^q_{q_0}\wedge 1^q_{q_1}\wedge 0^q_{q_2})\wedge\mathcal{A}^{=\left(\frac{1}{2}*\lambda_1\right)}(0^q_{q_0}\wedge 1^q_{q_1}\vee 1^q_{q_2})\\ \wedge\mathcal{A}^{=\left(\frac{1}{2}*\lambda_1\right)}(1^q_{q_0}\wedge 0^q_{q_1}\wedge 0^q_{q_2})\wedge\mathcal{A}^{=\left(-\frac{1}{2}*\lambda_2\right)}(1^q_{q_0}\wedge 0^q_{q_1}\vee 1^q_{q_2})\\ \wedge\mathcal{A}^{=\left(-\frac{1}{2}*\lambda_2\right)}(1^q_{q_0}\wedge 1^q_{q_1}\wedge 0^q_{q_2})\wedge\mathcal{A}^{=\left(-\frac{1}{2}*\lambda_1\right)}(1^q_{q_0}\wedge 1^q_{q_1}\wedge 1^q_{q_2})\end{array}}{\begin{array}{l}P^{=0.25}(0^c_{c_0}\wedge 0^q_{q_0}\wedge 0^c_{c_1}\wedge 0^q_{q_1}\wedge\mathcal{A}^{=\lambda_1}(0^q_{q_2})\vee\mathcal{A}^{=(\lambda_2)}(1^q_{q_2}))\vee\\ P^{=0.25}(0^c_{c_0}\wedge 0^q_{q_0}\wedge 1^c_{c_1}\wedge 1^q_{q_1}\wedge\mathcal{A}^{=\lambda_2}(0^q_{q_2})\vee\mathcal{A}^{=(\lambda_1)}(1^q_{q_2}))\vee\\ P^{=0.25}(1^c_{c_0}\wedge 1^q_{q_0}\wedge 0^c_{c_1}\wedge 0^q_{q_1}\wedge\mathcal{A}^{=\lambda_1}(0^q_{q_2})\vee\mathcal{A}^{=(-\lambda_2)}(1^q_{q_2}))\vee\\ P^{=0.25}(1^c_{c_0}\wedge 1^q_{q_0}\wedge 1^c_{c_1}\wedge 1^q_{q_1}\wedge\mathcal{A}^{=\lambda_2}(0^q_{q_2})\vee\mathcal{A}^{=(-\lambda_1)}(1^q_{q_2}))\end{array}}\ [M1;M2]\ (vi)$$

$$\frac{\begin{array}{l}P^{=0.25}(0^c_{c_0}\wedge 0^q_{q_0}\wedge 0^c_{c_1}\wedge 0^q_{q_1}\wedge\mathcal{A}^{=\lambda_1}(0^q_{q_2})\vee\mathcal{A}^{=(\lambda_2)}(1^q_{q_2}))\vee\\ P^{=0.25}(0^c_{c_0}\wedge 0^q_{q_0}\wedge 1^c_{c_1}\wedge 1^q_{q_1}\wedge\mathcal{A}^{=\lambda_2}(0^q_{q_2})\vee\mathcal{A}^{=(\lambda_1)}(1^q_{q_2}))\vee\\ P^{=0.25}(1^c_{c_0}\wedge 1^q_{q_0}\wedge 0^c_{c_1}\wedge 0^q_{q_1}\wedge\mathcal{A}^{=\lambda_1}(0^q_{q_2})\vee\mathcal{A}^{=(-\lambda_2)}(1^q_{q_2}))\vee\\ P^{=0.25}(1^c_{c_0}\wedge 1^q_{q_0}\wedge 1^c_{c_1}\wedge 1^q_{q_1}\wedge\mathcal{A}^{=\lambda_2}(0^q_{q_2})\vee\mathcal{A}^{=(-\lambda_1)}(1^q_{q_2}))\end{array}}{\begin{array}{l}P^{=0.25}(0^c_{c_0}\wedge 0^q_{q_0}\wedge 0^c_{c_1}\wedge 0^q_{q_1}\wedge\mathcal{A}^{=\lambda_1}(0^q_{q_2})\wedge\mathcal{A}^{=(\lambda_2)}(1^q_{q_2}))\vee\\ P^{=0.25}(0^c_{c_0}\wedge 0^q_{q_0}\wedge 1^c_{c_1}\wedge 1^q_{q_1}\wedge\mathcal{A}^{=\lambda_2}(0^q_{q_2})\wedge\mathcal{A}^{=(\lambda_1)}(1^q_{q_2}))\vee\\ P^{=0.25}(1^c_{c_0}\wedge 1^q_{q_0}\wedge 0^c_{c_1}\wedge 0^q_{q_1}\wedge\mathcal{A}^{=\lambda_1}(0^q_{q_2})\wedge\mathcal{A}^{=-\lambda_2}(1^q_{q_2}))\vee\\ P^{=0.25}(1^c_{c_0}\wedge 1^q_{q_0}\wedge 1^c_{c_1}\wedge 1^q_{q_1}\wedge\mathcal{A}^{=\lambda_2}(0^q_{q_2})\wedge\mathcal{A}^{=(-\lambda_1)}(1^q_{q_2}))\end{array}}\ [\text{ENT;h } q_0\text{;M1;M2}]\ (vii)$$

Finally, following the measurements of the first qubits, all that remains is to apply the error corrections, which come from the programs if (c [0] == 1) z q_2 and if (c [0] == 1) z q_2, which we name as IF1 and IF2, respectively. The following inference is possible upon the execution of program IF1 (see proof **if1**),

$$\frac{\begin{array}{l}P^{=0.25}(0^c_{c_0}\wedge 0^q_{q_0}\wedge 0^c_{c_1}\wedge 0^q_{q_1}\wedge\mathcal{A}^{=\lambda_1}(0^q_{q_2})\vee\mathcal{A}^{=(\lambda_2)}(1^q_{q_2}))\vee\\ P^{=0.25}(0^c_{c_0}\wedge 0^q_{q_0}\wedge 1^c_{c_1}\wedge 1^q_{q_1}\wedge\mathcal{A}^{=\lambda_2}(0^q_{q_2})\vee\mathcal{A}^{=(\lambda_1)}(1^q_{q_2}))\vee\\ P^{=0.25}(1^c_{c_0}\wedge 1^q_{q_0}\wedge 0^c_{c_1}\wedge 0^q_{q_1}\wedge\mathcal{A}^{=\lambda_1}(0^q_{q_2})\vee\mathcal{A}^{=-\lambda_2}(1^q_{q_2}))\vee\\ P^{=0.25}(1^c_{c_0}\wedge 1^q_{q_0}\wedge 1^c_{c_1}\wedge 1^q_{q_1}\wedge\mathcal{A}^{=\lambda_2}(0^q_{q_2})\vee\mathcal{A}^{=(-\lambda_1)}(1^q_{q_2}))\end{array}}{\begin{array}{l}P^{=0.5}((0^c_{c_1}\wedge 0^q_{q_1}\wedge\mathcal{A}^{=\lambda_1}(0^q_{q_2})\wedge\mathcal{A}^{=(\lambda_2)}(1^q_{q_2}))\vee\\ P^{=0.5}(1^c_{c_1}\wedge 1^q_{q_1}\wedge\mathcal{A}^{=\lambda_2}(0^q_{q_2})\wedge\mathcal{A}^{=(\lambda_1)}(1^q_{q_2}))\end{array}}\ [IF1]\ (viii)$$

$$\frac{\begin{array}{l}P^{=0.5}(0^c_{c_1}\wedge 0^q_{q_1}\wedge\mathcal{A}^{=\lambda_1}(0^q_{q_2})\wedge\mathcal{A}^{=(\lambda_2)}(1^q_{q_2}))\vee\\ P^{=0.5}(1^c_{c_1}\wedge 1^q_{q_1}\wedge\mathcal{A}^{=\lambda_2}(0^q_{q_2})\wedge\mathcal{A}^{=(\lambda_1)}(1^q_{q_2}))\end{array}}{(\mathcal{A}^{=\lambda_1}(0^q_{q_2})\wedge\mathcal{A}^{=(\lambda_2)}(1^q_{q_2}))}\ [\text{ENT;h } q_0\text{;M1;M2;IF1}]\ (ix)$$

and after the program IF2 the following post-condition holds (see proof **if2**)

$$\frac{\begin{array}{l}P^{=0.5}(0^c_{c_1}\wedge 0^q_{q_1}\wedge\mathcal{A}^{=\lambda_1}(0^q_{q_2})\wedge\mathcal{A}^{=(\lambda_2)}(1^q_{q_2}))\vee\\ P^{=0.5}(1^c_{c_1}\wedge 1^q_{q_1}\wedge\mathcal{A}^{=\lambda_2}(0^q_{q_2})\wedge\mathcal{A}^{=(\lambda_1)}(1^q_{q_2}))\end{array}}{(\mathcal{A}^{=\lambda_1}(0^q_{q_2})\wedge\mathcal{A}^{=(\lambda_2)}(1^q_{q_2}))}\ [IF2]\ (viii)$$

$$\frac{}{(\mathcal{A}^{=\lambda_1}(0^q_{q_2})\wedge\mathcal{A}^{=(\lambda_2)}(1^q_{q_2}))}\ [\text{ENT;h } q_0\text{;M1;M2;IF1;IF2}]\ (ix)$$

.

which due to the assertion of (151), results in [ENT;h q_0;M1;M2;IF1;IF2]($P^{=\alpha}(0^q_{q_2}) \wedge P^{=1-\alpha}(1^q_{q_2})$) being true, completing the proof.

Proofs of the measurement and post-error correction

Proof: **(h3)**

Let p, q denote the following expressions

$$p = ((\mathcal{A}^{=(\frac{1}{\sqrt{2}}*\lambda_1)}(0^q_{q_0} \wedge 0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=(\frac{1}{\sqrt{2}}*(\lambda_2)}(1^q_{q_0} \wedge 0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=(\frac{1}{\sqrt{2}}*\lambda_1)}(0^q_{q_0} \wedge 1^q_{q_1} \wedge 1^q_{q_2}) \wedge \mathcal{A}^{=\frac{1}{\sqrt{2}}*(\lambda_2)}(1^q_{q_0} \wedge 0^q_{q_1} \wedge 1^q_{q_2}))\,,$$

and

$$\begin{aligned} q = {} & \mathcal{A}^{=\left(\frac{1}{2}*\lambda_1\right)}(0^q_{q_0} \wedge 0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=\left(\frac{1}{2}*\lambda_2\right)}(0^q_{q_0} \wedge 0^q_{q_1} \wedge 1^q_{q_2}) \\ & \wedge \mathcal{A}^{=\left(\frac{1}{2}*\lambda_2\right)}(0^q_{q_0} \wedge 1^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=\left(\frac{1}{2}*\lambda_1\right)}(0^q_{q_0} \wedge 1^q_{q_1} \wedge 1^q_{q_2}) \\ & \wedge \mathcal{A}^{=\left(\frac{1}{2}*\lambda_1\right)}(1^q_{q_0} \wedge 0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=\left(-\frac{1}{2}*\lambda_2\right)}(1^q_{q_0} \wedge 0^q_{q_1} \wedge 1^q_{q_2}) \\ & \wedge \mathcal{A}^{=\left(-\frac{1}{2}*\lambda_2\right)}(1^q_{q_0} \wedge 1^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=\left(-\frac{1}{2}*\lambda_1\right)}(1^q_{q_0} \wedge 1^q_{q_1} \wedge 1^q_{q_2})\,. \end{aligned}$$

The proof of the expression $p \rightarrow [H]q$ goes as follows:

$[\![p \rightarrow [H]q]\!]$

$=$ (by the definition of the h instruction and the semantics of state s)

$S - \{s|\exists t : \pi_{q_0,q_1,q_2}(s) = (\lambda_1 * \frac{1}{\sqrt{2}}|000\rangle + \lambda_1 * \frac{1}{\sqrt{2}}|011\rangle + \lambda_2 * \frac{1}{\sqrt{2}}|110\rangle + \lambda_2 * \frac{1}{\sqrt{2}}|101\rangle) \wedge \pi_{q_0,q_1,q_2}(t) = (H \otimes I \otimes I).\pi_{q_0,q_1,q_2}(s) \Rightarrow t \notin [\![q]\!]\}$

$=$ (the action of the h instruction results in the state

$S - \{s|\exists t : \pi_{q_0,q_1,q_2}(s) = (\lambda_1 * \frac{1}{\sqrt{2}}|000\rangle + \lambda_1 * \frac{1}{\sqrt{2}}|011\rangle + \lambda_2 * \frac{1}{\sqrt{2}}|110\rangle + \lambda_2 * \frac{1}{\sqrt{2}}|101\rangle) \wedge \pi_{q_0,q_1,q_2}(t) = (\lambda_1 * \frac{1}{2}|000\rangle + \lambda_2 * \frac{1}{2}|001\rangle + \lambda_2 * \frac{1}{2}|010\rangle + \lambda_1 * \frac{1}{2}|100\rangle + -\lambda_2 * \frac{1}{2}|101\rangle + -\lambda_2 * \frac{1}{2}|110\rangle + \lambda_1 * \frac{1}{2}|111\rangle) \Rightarrow t \notin [\![p]\!]\}$

$=$ (by the definition of the amplitude operator on the state)

$S - \{s|\exists t : s \in [\![(\mathcal{A}^{=\lambda_1}(0^q_{q_0}) \wedge \mathcal{A}^{=(\lambda_2)}(1^q_{q_0})) \wedge (\mathcal{A}^{=\frac{1}{\sqrt{2}}}(0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{\frac{1}{\sqrt{2}}}(1^q_{q_1} \wedge 1^q_{q_2}))]\!]^{(1)} \wedge t \in [\![p]\!]^{(2)} \Rightarrow t \notin [\![p]\!]^{(2)}\}$

$=$ (it can be easily verified that sets referred by $^{(1)}$,$^{(2)}$ are non-empty)

$[\![\top]\!]$

□

Proof: (**meast**)

Let p and q denote the following expressions:

$$
\begin{aligned}
p = & \mathcal{A}^{=\left(\frac{1}{2}*\lambda_1\right)}(0^q_{q_0} \wedge 0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=\left(\frac{1}{2}*\lambda_2\right)}(0^q_{q_0} \wedge 0^q_{q_1} \wedge 1^q_{q_2}) \\
& \wedge \mathcal{A}^{=\left(\frac{1}{2}*\lambda_2\right)}(0^q_{q_0} \wedge 1^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=\left(\frac{1}{2}*\lambda_1\right)}(0^q_{q_0} \wedge 1^q_{q_1} \wedge 1^q_{q_2}) \\
& \wedge \mathcal{A}^{=\left(\frac{1}{2}*\lambda_1\right)}(1^q_{q_0} \wedge 0^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=\left(-\frac{1}{2}*\lambda_2\right)}(1^q_{q_0} \wedge 0^q_{q_1} \wedge 1^q_{q_2}) \\
& \wedge \mathcal{A}^{=\left(-\frac{1}{2}*\lambda_2\right)}(1^q_{q_0} \wedge 1^q_{q_1} \wedge 0^q_{q_2}) \wedge \mathcal{A}^{=\left(-\frac{1}{2}*\lambda_1\right)}(1^q_{q_0} \wedge 1^q_{q_1} \wedge 1^q_{q_2}), \\
q = & P^{=0.25}(0^c_{c_0} \wedge 0^q_{q_0} \wedge 0^c_{c_1} \wedge 0^q_{q_1} \wedge \mathcal{A}^{=\lambda_1}(0^q_{q_2}) \wedge \mathcal{A}^{=(\lambda_2)}(1^q_{q_2})) \vee \\
& P^{=0.25}(0^c_{c_0} \wedge 0^q_{q_0} \wedge 1^c_{c_1} \wedge 1^q_{q_1} \wedge \mathcal{A}^{=\lambda_2}(0^q_{q_2}) \wedge \mathcal{A}^{=(\lambda_1)}(1^q_{q_2})) \vee \\
& P^{=0.25}(1^c_{c_0} \wedge 1^q_{q_0} \wedge 0^c_{c_1} \wedge 0^q_{q_1} \wedge \mathcal{A}^{=\lambda_1}(0^q_{q_2}) \wedge \mathcal{A}^{=(-\lambda_2)}(1^q_{q_2})) \vee \\
& P^{=0.25}(1^c_{c_0} \wedge 1^q_{q_0} \wedge 1^c_{c_1} \wedge 1^q_{q_1} \wedge \mathcal{A}^{=\lambda_2}(0^q_{q_2}) \wedge \mathcal{A}^{=(-\lambda_1)}(1^q_{q_2})).
\end{aligned}
$$

The proof of the expression $p \rightarrow [M1; M2]q$ reads as:

$[\![p \rightarrow [M1; M2]q]\!]$

$=$ (by the definition of the dynamic operator $\langle\pi\rangle\phi$)

$S - \{s|\exists t : s \in [\![p]\!] \wedge (s,t) \in [\![[\text{meas } q_0 \text{ to } c_0; \text{meas } q_1 \text{ to } c_1]]\!] \Rightarrow t \notin [\![q]\!]\}$

$\equiv$ (by the definition of measurement)

$S - \{s|\exists u \exists t : s \in [\![p]\!] \wedge (s,u) \in (\{(s,u)|u \in [\![q^q_0 == 0]\!](s) \wedge u \in [\![P^{e_1}0^c_{c_0}]\!]$ where $e_1 = \left\langle s\middle|0^q_{q_0}\right\rangle\left\langle 0^q_{q_0}\middle|s\right\rangle\} \cup \{(s,u)|u \in [\![q^q_0 == 1]\!] \wedge u \in [\![P^{e_2}1^c_{c_0}]\!]$ where $e_2 = \left\langle s\middle|1^q_{q_0}\right\rangle\left\langle 1^q_{q_0}\middle|s\right\rangle\}) \wedge (u,t) \in [\![\text{meas } q_1 \text{ to } c_1]\!] \Rightarrow t \notin [\![q]\!]\}$

$=$ (by the definltion of measurement and combination)

$$
\begin{aligned}
S - \{s|\exists t : s \in [\![p]\!] \wedge & \\
& (s,t) \in (\{(s,t)|t \in [\![q^q_1 == 0]\!] \wedge t \in [\![0^q_{q_0} \wedge P^{=e_1}0^c_{c_0} \wedge P^{=e_2}0^c_{c_1}]\!]\} \cup \\
& \{(s,t)|t \in [\![q^q_1 == 1]\!] \wedge t \in [\![0^q_{q_0} \wedge P^{=e_1}0^c_{c_0} \wedge P^{=e_3}1^c_{c_1}]\!]\} \cup \\
& \{(s,t)|t \in [\![q^q_1 == 0]\!] \wedge t \in [\![1^q_{q_0} \wedge P^{=e_4}1^c_{c_0} \wedge P^{=e_2}0^c_{c_1}]\!]\} \cup \\
& \{(s,t)|t \in [\![q^q_1 == 1]\!] \wedge t \in [\![1^q_{q_0} \wedge P^{=e_4}1^c_{c_0} \wedge P^{=e_3}1^c_{c_1}]\!]\} \Rightarrow t \notin [\![q]\!]\}
\end{aligned}
$$

where $e_1 = \left\langle s\middle|0^q_{q_0}\right\rangle\left\langle 0^q_{q_0}\middle|s\right\rangle, e_2 = \left\langle s\middle|0^q_{q_1}\right\rangle\left\langle 0^q_{q_1}\middle|s\right\rangle, e_3 = \left\langle s\middle|1^q_{q_1}\right\rangle\left\langle 1^q_{q_1}\middle|s\right\rangle$ and $e_4 = \left\langle s\middle|1^q_{q_0}\right\rangle\left\langle 1^q_{q_0}\middle|s\right\rangle)$

$=$ (the actual application of quantum tests, lead to the following state)

$$
\begin{aligned}
&S-\{s|\exists t: s\in[\![p]\!](1)\wedge\\
&\quad (s,t)\in(\{(s,t)|t\in[\![(0^q_{q_1}\wedge 0^q_{q_0}\wedge 0^c_{c_0}\wedge 0^c_{c_1}\wedge \mathcal{A}^{=\lambda_1}(0^q_{q_2})\wedge \mathcal{A}^{=(\lambda_2)}(1^q_{q_2}))]\!]\}\cup\\
&\quad \{(s,t)|t\in[\![P^{=0.5*0.5}(1^q_{q_1}\wedge 0^q_{q_0}\wedge 0^c_{c_0}\wedge 1^c_{c_1}\wedge \mathcal{A}^{=\lambda_2}(0^q_{q_2})\wedge \mathcal{A}^{=(\lambda_1)}(1^q_{q_2}))]\!]\}\cup\\
&\quad \{(s,t)|t\in[\![P^{=0.5*0.5}(0^q_{q_1}\wedge 1^q_{q_0}\wedge 1^c_{c_0}\wedge 0^c_{c_1}\wedge \mathcal{A}^{=\lambda_1}(0^q_{q_2})\wedge \mathcal{A}^{=-\lambda_2}(1^q_{q_2}))]\!]\}\cup\\
&\quad \{(s,t)|t\in[\![P^{=0.5*0.5}(1^q_{q_1}\wedge 1^q_{q_0}\wedge 1^c_{c_0}\wedge 1^c_{c_1}]\!]\wedge \mathcal{A}^{=\lambda_2}(0^q_{q_2})\wedge \mathcal{A}^{=(-\lambda_1)}(1^q_{q_2}))]\!]\})^{(2)}\Rightarrow t\notin[\![q]\!]^{(2)}\}
\end{aligned}
$$

$=$

$S-\{s|\exists t: s\in[\![p]\!]^{(1)}\wedge t\in[\![q]\!]^{(2)}\Rightarrow$

$t\notin[\![q]\!]^{(2)})\}$

$=$ (it can be easily verified that sets referred by $^{(1)}$,$^{(2)}$ are non-empty)

$[\![\neg\bot]\!]$

$=$

$[\![\top]\!]$ □

Proof: (**if1**) Let p, q denote the following expressions:

$$
\begin{aligned}
p=&(P^{=0.25}(0^c_{c_0}\wedge 0^q_{q_0}\wedge 0^c_{c_1}\wedge 0^q_{q_1}\wedge \mathcal{A}^{=\lambda_1}(0^q_{q_2})\wedge \mathcal{A}^{=(\lambda_2)}(1^q_{q_2}))\vee\\
&P^{=0.25}(0^c_{c_0}\wedge 0^q_{q_0}\wedge 1^c_{c_1}\wedge 1^q_{q_1}\wedge \mathcal{A}^{=\lambda_2}(0^q_{q_2})\wedge \mathcal{A}^{=(\lambda_1)}(1^q_{q_2}))\vee\\
&P^{=0.25}(1^c_{c_0}\wedge 1^q_{q_0}\wedge 0^c_{c_1}\wedge 0^q_{q_1}\wedge \mathcal{A}^{=\lambda_1}(0^q_{q_2})\wedge \mathcal{A}^{=-\lambda_2}(1^q_{q_2}))\vee\\
&P^{=0.25}(1^c_{c_0}\wedge 1^q_{q_0}\wedge 1^c_{c_1}\wedge 1^q_{q_1}\wedge \mathcal{A}^{=\lambda_2}(0^q_{q_2})\wedge \mathcal{A}^{=(-\lambda_1)}(1^q_{q_2}))\,,\\
q=&P^{=0.5}((0^c_{c_1}\wedge 0^q_{q_1}\wedge \mathcal{A}^{=\lambda_1}(0^q_{q_2})\wedge \mathcal{A}^{=(\lambda_2)}(1^q_{q_2}))\wedge\\
&P^{=0.5}(1^c_{c_1}\wedge 1^q_{q_1}\wedge \mathcal{A}^{=\lambda_2}(0^q_{q_2})\wedge \mathcal{A}^{=(\lambda_1)}(1^q_{q_2}))\,.
\end{aligned}
$$

The proof of statement $p\rightarrow[IF1]q$ reads as:

$[\![p\rightarrow[IF1]q]\!]$

$=$ (by the definition of the dynamic operator $\langle\pi\rangle\phi$)

$S-\{s|\exists t: s\in[\![p]\!]\wedge(s,t)\in\{[\![[\text{if } (c_0==1) \text{ z } q_3]]\!]\}\Rightarrow t\notin[\![q]\!]\}$

$=$ (by the semantics of the if statement)

$S-\{s|\exists t: s\in[\![p]\!]\wedge(s,t)\in([\![c_0==1]\!]\cap[\![|1_{c_0}\rangle\langle 1_{c_0}|; \text{z } q_3]\!]\}\cup\{[\![c_0==0]\!]\cap[\![|0_{c_0}\rangle\langle 0_{c_0}|; \text{skip}]\!])\Rightarrow$

$t\notin[\![q]\!]\}$

$=$ (by the definition of the if statement, only two worlds, the ones where test c_0 == 1 holds, will be affected by the phase flip operator)

$$S - \{s|\exists t : s \in [[p]] \wedge t \in ([[P^{=0.25}(0^c_{c_0} \wedge 0^q_{q_0} \wedge 0^c_{c_1} \wedge 0^q_{q_1} \wedge \mathcal{A}^{=\lambda_1}(0^q_{q_2}) \wedge \mathcal{A}^{=(\lambda_2)}(1^q_{q_2}))]] \cup$$
$$[[P^{=0.25}(0^c_{c_0} \wedge 0^q_{q_0} \wedge 1^c_{c_1} \wedge 1^q_{q_1} \wedge \mathcal{A}^{=\lambda_2}(0^q_{q_2}) \wedge \mathcal{A}^{=(\lambda_1)}(1^q_{q_2}))]] \cup$$
$$[[P^{=0.25}(1^c_{c_0} \wedge 1^q_{q_0} \wedge 0^c_{c_1} \wedge 0^q_{q_1} \wedge \mathcal{A}^{=\lambda_1}(0^q_{q_2}) \wedge \mathcal{A}^{=\lambda_2}(1^q_{q_2}))]] \cup$$
$$[[P^{=0.25}(1^c_{c_0} \wedge 1^q_{q_0} \wedge 1^c_{c_1} \wedge 1^q_{q_1} \wedge \mathcal{A}^{=\lambda_2}(0^q_{q_2}) \wedge \mathcal{A}^{=(\lambda_1)}(1^q_{q_2}))]]) \Rightarrow t \notin [[q]]\}$$

$=$ (upon the application of the phase flip correction, the two pairs of possible worlds become equivalent, hence the possible worlds are narrowed to 2)

$$S - \{s|\exists t : s \in [[p]]^{(1)} \wedge t \in ([[P^{=0.5}(1^c_{c_0} \wedge 1^q_{q_0} \wedge 0^c_{c_1} \wedge 0^q_{q_1} \wedge \mathcal{A}^{=\lambda_1}(0^q_{q_2}) \wedge \mathcal{A}^{=\lambda_2}(1^q_{q_2}))]] \cup$$
$$[[P^{=0.5}(1^c_{c_0} \wedge 1^q_{q_0} \wedge 1^c_{c_1} \wedge 1^q_{q_1} \wedge \mathcal{A}^{=\lambda_2}(0^q_{q_2}) \wedge \mathcal{A}^{=(\lambda_1)}(1^q_{q_2}))]])^{(2)}$$
$$\Rightarrow t \notin [[q]]^{(2)}\}$$

$=$ (it can be easily verified that sets referred by $^{(1)}$,$^{(2)}$ are non-empty)
$[[\neg\bot]]$
$=$
$[[\top]]$ □

Proof: (**if2**) Let p, q denote the following expressions:

$$p = P^{=0.5}(0^c_{c_1} \wedge 0^q_{q_1} \wedge \mathcal{A}^{=\lambda_1}(0^q_{q_2}) \wedge \mathcal{A}^{=(\lambda_2)}(1^q_{q_2})) \wedge P^{=0.5}(1^c_{c_1} \wedge 1^q_{q_1} \wedge \mathcal{A}^{=\lambda_2}(0^q_{q_2}) \wedge \mathcal{A}^{=(\lambda_1)}(1^q_{q_2}))\,,$$
$$q = (P^{=\alpha}(0^q_{q_2}) \wedge P^{=1-\alpha}(1^q_{q_2}))\,.$$

The proof of the statement $p \rightarrow [IF2]q$, read as
$[[p \rightarrow [IF2]q]]$
$=$ (by the definition of the dynamic operator $\langle\pi\rangle\phi$)
$S-\{s|\exists t : s \in [[p]] \wedge (s,t) \in ([[c_1 == 1]] \cap [[|1_{c_1}\rangle\langle 1_{c_1}|; \mathrm{x}\, q_3]]\} \cup \{[[c_1 == 0]] \cap [[|1_{c_0}\rangle\langle 1_{c_1}|; \mathrm{skip}]]) \Rightarrow$
$t \notin [[q]]\}$
$=$ (by the semantics of the if statement, i.e. one *possible world* is affected by operator X, and the other is not)
$S - \{s|\exists t : s \in [[p]] \wedge t \in ([[P^{=0.5}(1^c_{c_0} \wedge 1^q_{q_0} \wedge 0^c_{c_1} \wedge 0^q_{q_1} \wedge \mathcal{A}^{=\lambda_1}(0^q_{q_2}) \wedge \mathcal{A}^{=\lambda_2}(1^q_{q_2}))]] \cup$
$[[P^{=0.5}(1^c_{c_0} \wedge 1^q_{q_0} \wedge 1^c_{c_1} \wedge 1^q_{q_1} \wedge A^{=(\lambda_1)}(0^q_{q_2}) \wedge \mathcal{A}^{=\lambda_2}(1^q_{q_2}))]]) \Rightarrow t \notin [[q]]\}$
$=$ (the two resulting worlds are equivalent and hence, merge into one)
$S - \{s|\exists t : s \in [[p]] \wedge t \in ([[(\mathcal{A}^{=\lambda_1}(0^q_{q_2}) \wedge \mathcal{A}^{=\lambda_2}(1^q_{q_2}))]] \Rightarrow t \notin [[q]]\}$

$=$ (by assumption of equation 151)

$S - \{s | \exists t : s \in [\![p]\!]^{(1)} \wedge t \in ([\![P^{=\alpha} 0^q_{q_2} \wedge P^{=1-\alpha} 1^q_{q_2}]\!]^{(2)} \Rightarrow t \notin [\![(P^{=\alpha}(0^q_{q_2}) \wedge P^{=1-\alpha}(1^q_{q_2}))]\!]^{(2)}\}$

$=$ (it can be easily verified that sets referred by $^{(1)}, ^{(2)}$ are non-empty)

$[\![\neg \bot]\!] = [\![\top]\!]$ □

5.5 Proof of decidability

In this section we show that the logic presented in this chapter is decidable. The method for this is based on the reduction of the logic to the theory of the first order language of complex numbers, i.e. first order theory of the algebraically closed fields[2], known to be decidable by an important result of Tarski [341].

Definition 5.5.1. The first order theory of complex numbers, denoted by $\mathcal{T}_{\mathcal{L}_{\mathbb{C}}}$, is the theory of $\mathbb{C}$ in the language $\mathcal{L}_{\mathbb{C}}$, which corresponds to $(\mathbb{C}, *, +, ., 0, 1)$, where $*$ is the conjugate transpose operation.

This strategy of proving the decidability of quantum logic has been defined for the standard quantum finite dimensional logic by Dunn et al. [139], and a more general recipe for a wide range of quantum logics, including modal and dynamic ones, has been discussed in Baltag et al. in [45]. The latter work is the main inspiration for this section, from which the main ideas and notation were borrowed. We revisit the main ingredients of the proof recipe through the next sections and the decidability is stated and proved in theorem 5.5.2.

5.5.1 *The main idea of the proof*

The cornerstone of the proof presented in [139] is based on the equivalence between quantum propositions, *closed linear sub-spaces* of an *Hilbert* space, and *kernels* of matrices, i.e.

$$[[p]] \Leftrightarrow \overline{[[p]]} \Leftrightarrow \{v|\hat{p}.v = 0\}\,, \tag{152}$$

where p is a proposition, $\overline{[[p]]}$ is a closed linear subspace of an Hilbert space, and $\hat{p} \in \mathbb{C}^{n\times n}$ is a matrix and $\{v|\hat{p}.v = 0\}$ corresponds to its kernel. This equivalence makes possible the entire translation of standard quantum logic into closed statements of the first order theory of complex numbers, as the *kernel* of the matrix $\hat{p} = (p_{11}, \ldots, p_{ij}, \ldots, p_{nn})$, can efficiently be reduced to an expression of this theory as

$$\begin{aligned}\mathbb{C} \models \forall v\, \hat{p}.v = 0 \Leftrightarrow C \models & p_{11}.v_1 + p_{12}.v_2 + \ldots + p_{1n}.v_n = 0 \wedge \\ & p_{21}.v_1 + p_{22}.v_2 + \ldots + p_{2n}.v_n = 0 \wedge \\ & \ldots \wedge \\ & p_{n1}.v_1 + p_{n2}.v_2 + \ldots + p_{nn}.v_n = 0\,.\end{aligned}$$

where p_{11} to p_{nn} are a set of variables with valuations in $\mathbb{C}$. From here one can extract useful notions of logic: $v \in \overline{[[p]]}$ as

$$\mathbb{C} \models \hat{p}.v = 0 \tag{153}$$

2 Theory composed of axioms of algebraically closed fields, equality operators, and first order quantifiers.

and the notion of satisfaction $\mathcal{H} \models p$, meaning that p is true independently of the matrixial representation, $\hat{p}$, chosen:

$$\mathcal{H} \models p \Leftrightarrow \mathbb{C} \models \forall \hat{p} \forall v \, \hat{p}.v = 0 \,. \tag{154}$$

Furthermore, it is also possible to express unitary operators with an additional condition enforcing their unitarity,

$$\mathcal{H} \models [u]p \Leftrightarrow \mathbb{C} \models \forall \, \hat{u} \, \hat{p} \, U_n(\hat{u}) \rightarrow \forall v \hat{p}(\hat{u}v) = 0, \text{where } U_n(\hat{u}) = \hat{u}.\hat{u}^* = I \,,$$

completing the necessary *machinery* to capture the *static* and *dynamic* aspects of quantum logic. In a more general way every quantum dynamic logic expression can be translated into expressions of type

$$\mathcal{H} \models \varphi \Leftrightarrow \mathbb{C} \models \forall \vec{x} (\varphi^p(\vec{x}) \rightarrow \forall v . \varphi^\delta(v, \vec{x})) \,, \tag{155}$$

where both $\varphi^\delta(v, \vec{x})$ and $\varphi^p(\vec{x})$ are functions, v is a vector space, and $\vec{x}$ are lists of variables, with valuations in $\mathbb{C}$, which corresponds to the matrixial representation of φ, $\hat{\varphi}$. The former function defines $[[\varphi]] \in \mathcal{H}$, i.e. by making the $\vec{x}$ correspond to the assignment $[[.]]$, and the latter, $\varphi^p(\vec{x})$, defines the range of good values for $\vec{x}$. Hence, as all the statements falling into this type of expressions are inherently *well-formed* in the first order theory of complex numbers, and hence implicitly decidable, and the proof of decidability somehow reduces to showing that such functions can be obtained by an *effective* method.

5.5.2 *Application to the Probabilistic logic of quantum programs*

These ideas are applicable to the probabilistic logic of quantum programs (PLQP), introduced in [45]. Recalling, its syntax reads as:

$$\begin{aligned} \phi &::= p | \phi \wedge \phi | \neg \phi | [\pi] \phi | P^{\geq r} \phi \\ \pi &::= u | \phi ? | \pi ; \pi | \pi \cup \pi \end{aligned}$$

The first element of the proof of decidability of the logic comes from the fact that all models over a Hilbert space of this logic, $(\Sigma, [[.]])$, are isomorphic to a $\mathbb{C}^n$-interpretation (see definition 5.5.2), $\mathcal{H} \cong \mathbb{C}^n$, i.e. an interpretation over the *closed subspaces* of an Hilbert space.

Definition 5.5.2. Given a language $\mathcal{L}$, by a $\mathbb{C}^n$-interpretation we mean a map $[[.]]$, that assigns to each sentence of $\mathcal{L}$ a subspace of $\mathbb{C}^n$. Also given any class $\mathcal{F}$ of $\mathbb{C}^n$ interpretations of $\mathcal{L}$, we say that a sentence φ is $\mathcal{F}$-valued and write $\mathcal{F} \models \varphi$, if $\overline{[[\varphi]]} = \mathbb{C}^n$ for all $[[.]] \in \mathcal{F}$

Based on the equivalence of equation (154) and (155), one can conclude that for each assignment $[[.]]$, there must be a function $\alpha : var(\mathcal{L}_{\mathbb{C}}) \to \mathbb{C}$, where $var(\mathcal{L}_{\mathbb{C}})$ corresponds to a list of variables $\vec{x}$, i.e. one that makes $\varphi^{\delta}(v, \vec{x}))$ be equivalent to some $[[.]]$. For instance, In the case of the PLQP language, to an *atomic* proposition p, defined by a tuple $\hat{x} = (x_{11}, \ldots, x_{nn})$ of $n \times n$ variables, the function α, given by $\alpha(\hat{x}) = (\alpha(x_{11}), \ldots, \alpha(x_{nn}))$, is such that $\varphi^{\delta}(v, \alpha(\vec{x})))$ is equivalent to $[[p]]$. There is a wide range of functions α for each assignment $[[.]]$, however not all them are valid. In the case of the PLQP language, such pairs, i.e. valid pairs, must respect the following conditions ($\mathcal{A}_{\mathcal{U}}$ is the set of well-formed unitary operators):

- for each $u \in \mathcal{A}_{\mathcal{U}}$, $\mathbb{C} \models U_n[\alpha(\hat{u})]$, that is $\alpha(\hat{u})$ is an unitary matrix;
- for each $p \in \mathcal{A}_{\mathcal{F}}$, $[[p]]$ is the kernel of the matrix ($\alpha(\hat{p})$ and
- for each $u \in \mathcal{A}_{\mathcal{U}}$, $[[u]]$ is the unitary transformation, given by the unitary matrix $\alpha(\hat{u})$.

The structure that maps the valid assignments, $[[.]]$ and α-functions is denominated a $\mathbb{C}$-coding (see definition 5.5.3), which is nothing more than a relational structure, containing all $[[.]] - \alpha$ function pairs.

Definition 5.5.3. Given any class $\mathcal{F}$ of $\mathbb{C}^n$-interpretations, a $\mathbb{C}$ coding of $\mathcal{F}$ is a partial function R from $\mathbb{C}^{var(\mathcal{L}_{\mathbb{C}})}$ onto $\mathcal{F}$ such that, for every finite list of variables $\vec{x} = (x_1, \ldots, x_m) \subseteq var(\mathcal{L}_{\mathbb{C}})$ of $\mathcal{L}_{\mathbb{C}}$ there is an $m-$ary formula $p_{\vec{x}}(\vec{y})$ of $\mathcal{L}_{\mathbb{C}}$ defining the set $\{\alpha(\vec{x}) \in \mathbb{C}^m | a \in dom(R))\}$. Moreover, we say that a $\mathbb{C}$-coding is effective if there is an effective procedure of computing such $p_{\vec{x}}(\vec{y})$ for any given finite $\vec{x}$.

An *effective* $\mathbb{C}-$coding is one where the function $p_{\vec{x}}(\vec{y})$, which determines the valid assignments for the lists $\vec{x}$, can be determined *effectively*. There is a close relationship between $\mathbb{C}$-codings and function pairs ($\varphi^{p}(\vec{x})$ and $\varphi^{\delta}(v, \vec{x})$), which translates as follows:

$$R(\alpha, [[.]]) \text{ entails } [[\varphi]] = \varphi^{\delta}(\mathbb{C}, \alpha(\vec{x}))\,, \qquad \varphi^{p}(\mathbb{C}) = p_{\vec{x}}(\mathbb{C})\,. \tag{156}$$

Hence, the decidability of a logic, which corresponds to the existence of an *effective* method to the generation of functions $\varphi^{\delta}(v, \vec{c})$ and $\varphi^{p}(\vec{x})$, can also be reduced to the *r-translatability* (see definition 5.5.4) of expressions in an *effective* $\mathbb{C}$-coding, as stated in theorem 5.5.1.

Definition 5.5.4. ([45]) Fix a finite-dimensional Hilbert space $\mathcal{H} \cong \mathbb{C}^n$, any class $\mathcal{F}$ of $\mathbb{C}^n-$interpretation of a language $\mathcal{L}$, and any effective $\mathbb{C}-$coding R of $\mathcal{F}$. Then for any sentence φ of $\mathcal{L}$, we say that a formula $\varphi^{\delta}(v_1, \ldots, v_n, \hat{x})$ of $\mathcal{L}_{\mathbb{C}}$ (with a specific tuple of variables $\hat{x}$) translates φ in R, or is an R-translation of φ, if conditions of equation (156)

Theorem 5.5.1. ([45]) Fix a finite-dimensional Hilbert space $\mathcal{H} \cong \mathbb{C}^n$, any class $\mathcal{F}$ of $\mathbb{C}^n$-interpretations of language $\mathcal{L}$, and any effective $\mathbb{C}$-coding R of $\mathcal{F}$. Suppose a sentence φ of $\mathcal{L}$ has an R-translation. Then it is decidable whether $\mathcal{F} \models \varphi$ or not.

5.5.3 *The decidability proof of the QASM logic*

We now discuss the actual proof of decidability for the logic developed in this chapter, the logic for the QASM language (LQASM), essentially by showing that the conditions presented in lemma 5.5.2 also apply to it. The model proposed for LQASM in section 5.3.4, is quite similar to the PLQP ones, and it can be straightforwardly verified that the fundamental assumptions to the proof of decidability of latter also hold in the former:

- The model of LQASM isomorphic to a $\mathbb{C}^n$-interpretation;
- Propositions correspond to closed linear subspaces.

The main differences between the LQASM and PLPQ lie on the type of programs allowed, i.e. the former allows unitary operators and classical control instructions and includes the connective $\mathcal{A}^{=\lambda}$. The theorem 5.5.1, presented in previous section, establishes the conditions for the general *proof recipe* of decidability, presented in lemma 5.5.2.

Lemma 5.5.2. ([45]) For every finite-dimensional Hilbert space $\mathcal{H} \cong \mathbb{C}^n$, and any class $\mathcal{F}$ of $\mathbb{C}^n$-intepretations of a language $\mathcal{L}$, the logic of $\mathcal{F}$ is decidable if

1. The theory is recursively enumerable;
2. There is an effective $\mathbb{C}$-coding of R;
3. There is an effective procedure that yields R-translation to a given atomic sentence of $\mathcal{L}$;
4. Each (n-ary) connective of $\mathcal{L}$, R-translation is preserved.

In theorem 5.5.3, proven and discussed in [45], this recipe is followed to show the decidability of the PLQP logic by showing that each individual condition holds in it.

Theorem 5.5.3. ([45]) Let $\mathcal{L}_{PLQP}$ be a language of PLQP. For any $n \in \mathbb{N}$ of any Hilbert space $\mathcal{H} \cong \mathbb{C}^n$, the language PLQP $(\mathcal{H}) = \{\varphi \in \mathcal{L}_{PLQP} | \mathcal{H} \models \varphi\}$ is decidable.

We also follow this recipe to show the decidability of LQASM. The functions $\varphi^{\delta}(v, \vec{x})$ are defined in table 11 and functions $par(\varphi)$ correspond to the set of $\mathbb{C}$-valued variables $\hat{p}$ that are part of a proposition φ. We are ready to show the decidability of LQASM in theorem 5.5.4.

Static propositions		
$p^\delta(v, par(p))$	=	$\hat{p}v = 0$
$(\varphi \wedge \psi)^\delta(v, par(\varphi \wedge \psi))$	=	$\varphi^\delta(v, par(\varphi)) \wedge \varphi^\delta(v, par(\psi))$
$(\varphi \vee \psi)^\delta(v, par(\varphi \vee \psi))$	=	$\varphi^\delta(v, par(\varphi)) \vee \varphi^\delta(v, par(\psi))$
$(\neg\varphi)^\delta(v, par(\neg\varphi))$	=	$\varphi^\delta(v, par(\varphi)) \to v = 0$
$\sim F(v, \vec{x})$	=	$\forall w(F(w, \vec{x})) \to \langle v, w\rangle = 0$
$(P^{\geq r}\varphi)^\delta(v, par(P^{\geq r}))$	=	$\exists w(w \neq 0 \wedge \sim\sim \varphi^\delta(w, par(\varphi)) \wedge |\langle v, w\rangle|^2 \geq r\|v\|^2\|w\|^2$
$(\mathcal{A}^{=\lambda}\varphi)^\delta(v, par(\mathcal{A}^{=\lambda}\varphi))$	=	$\exists w(w \neq 0 \wedge \sim\sim \varphi^\delta(w, par(\varphi)) \wedge \langle v, w\rangle = \lambda$
Programs and unitaries		
$([\pi]\varphi)^\delta(v, par([\pi]\varphi))$	=	$([\pi]\varphi)^\delta(\hat{\pi}v, par(\varphi))$
$([\pi;\pi]\varphi))^\delta(v, par([\pi;\pi]\varphi))$	=	$\varphi^\delta(\hat{\pi}v, [\pi]\varphi)$
$([\mathrm{u}]\varphi)^\delta(v, par([\mathrm{u}]\varphi))$	=	$\varphi^\delta(\hat{u}v, par(\varphi))$
Creation of registers		
$([\text{creg id [size]}]\varphi)^\delta(v, [\text{creg id [size]}]\varphi)$	=	$\varphi^\delta(\text{creg id [size]v}, par(\varphi))$ = $\varphi^\delta(v, par(\varphi)) \wedge \varphi^\delta(v, par(\hat{0}))$ where $\hat{0}$ is the assignment to variables corresponding to the semantics of creg (see section 5.3.7)
$([\text{qreg id [size]}]\varphi)^\delta(v, [\text{qreg id [size]}]\varphi)$	=	$\varphi^\delta(\text{qreg id [size]v}, par(\varphi))$ = $\varphi^\delta(v, par(\varphi)) \wedge \varphi^\delta(v, par(\hat{0}))$ where $\hat{0}$ is the assignment to variables corresponding to the semantics of qreg (see section 5.3.7)
Measurements and tests		
$([id_i^q == \mathrm{a}]\varphi)^\delta(v, par([id_i^q == \mathrm{a}]\varphi))$	=	$\varphi^\delta(\hat{m}v, par(\varphi))$ where $\hat{m}$ = $\left\|id_i^q\right\rangle\left\langle id_i^q\right\| / \sqrt{v^* \left\|id_i^q\right\rangle\left\langle id_i^q\right\| v}$
For m = measure qreg_id [index] → creg_id [index]]		
$([m]\varphi)^\delta(v, par([m]\varphi))$	=	$\varphi^\delta(v, par([id_i^q == 0]\varphi)) \wedge (P^{=r_1}(id_i^q == 0))^\delta(v, par(P^{=r_1}(1_i^c)) \vee \varphi^\delta(v, par([id_i^q == 1]\varphi)) \wedge (P^{=r_2}(id_i^q == 1))^\delta(v, par(P^{=r_2}(1_i^c))$ where $r_1 = v^* \left\|0_i^q\right\rangle\left\langle 0_i^q\right\| v$ and $r_2 = v^* \left\|1_i^q\right\rangle\left\langle 1_i^q\right\| v$
Classical tests and if statements		
$([id_i^c == \mathrm{a}]\varphi)^\delta(v, par([id_i^c == \mathrm{a}]\varphi))$	=	$(P^{=r_1}1_{id_i^c})(v, par(P^{=r_1}1_{id_i^c})) \to \varphi^\delta(w, par(\varphi)) \wedge (P^{=r_1}1_{id_i^c})(v, par(P^{=r_1}1_{id_i^c}))$
$([\text{if } \varphi_c\text{? then}]\ \pi_q]\varphi)^\delta(v, par([\text{if } \varphi_c\text{? then}]\ \pi_q]\varphi))$	=	$([id_i^c == \mathrm{a}]\varphi)^\delta(v, par([id_i^c == \mathrm{a}]\varphi)) \wedge (\pi_t)^\delta(w, par([\pi_t]\varphi)) \vee (\neg[id_i^c == \mathrm{a}])^\delta(v, par(\neg[id_i^c == \mathrm{a}])) \wedge (\pi_f)^\delta(w, par([\pi_f]\varphi))$ where $a \in 0, 1$, $\pi_t = \|a_{id_i^c}\rangle\langle a_{id_i^c}\|; \pi$ and $pi_f = (\neg a)_{id_i^c}\rangle\langle(\neg a)_{id_i^c}\|; skip$ and $skip$ is the identity operator

Table 11: Recursive definition of the $\varphi^\delta(v, par(\varphi))$ of φ

Theorem 5.5.4. Let $\mathcal{L}_{QASM}$ be a language of the QASM logic. For any $n \in \mathbb{N}$ of any Hilbert space $\mathcal{H} \cong \mathbb{C}^n$, the language QASM $(\mathcal{H}) = \{\varphi \in \mathcal{L}_{QASM} | \mathcal{H} \models \varphi\}$ is decidable.

Proof: The first condition of lemma 5.5.2 is accomplished, since the logic is clearly *recursively enumerable*. The second condition , of having an *efficient* $\mathbb{C}$-coding, is also accomplished, as the set of $\varphi^p(v)$ is defined as follows:

- $\varphi^p(v) = u.u^* = I$, for the case of $\varphi^\delta(v, [u]\varphi)$;
- $\varphi^p(v) = 0 = 0$ (tautology), for all other cases.

The next condition necessary to hold in lemma 5.5.2 is that all propositions and connectives are *efficiently r-translatable*. The case of all atomic propositions of $\mathcal{L}_{QASM}$ the fuction $par(\varphi)$ is equal to $\hat{p}$ and the construction $\varphi^\delta(v, \hat{p})$ is equivalent to $[[p]]$, according to equation (152), if the kernel of the matrix $\alpha(\hat{p})$ is equivalent to $[[p]]$. There efficient algorithms to solve this algebraic problem of calculating $\varphi^\delta(v, \hat{p})$ given $[[p]]$ and hence:

$$\varphi^\delta(v, p) \text{ can be efficiently generated for any } p \text{ atomic and is r-translatable.} \tag{157}$$

The $\wedge$ connective also preserves *effectiveness*. According to the semantics of LQASM, $[[\varphi_1 \wedge \varphi_2]] = [[\varphi_1]] \cap [[\varphi_1]]$, and, therefore, the *r-translatability* implies the following equivalence:

$$(\varphi \wedge \psi)^\delta(v, par(\varphi \wedge \psi)) = \varphi^\delta(v, par(\varphi)) \cap \varphi^\delta(v, par(\psi)) = [[\varphi]] \cap [[\psi]]\,. \tag{158}$$

Due to this equivalence, one can conclude that there is an effective method to generate $(\varphi \wedge \psi)^\delta(v, par(\varphi \wedge \psi))$ if $(\varphi)^\delta(v, par(\varphi))$, $(\psi)^\delta(v, par(\psi))$, and the set intersection of two finite sets, the $\cap$ operation, are *effectively* generatable, which clearly are. A similar reasoning can also be taken to the $\vee$ connective, as $[[\cup]]$ is also effective and a similar equivalence exists for this operation:

$$(\varphi \vee \psi)^\delta(v, par(\varphi \vee \psi)) = \varphi^\delta(v, par(\varphi)) \cup \varphi^\delta(v, par(\psi)) = [[\varphi]] \cup [[\psi]]\,. \tag{159}$$

The negation operator can also be, clearly, effectively generated, as its corresponding expression in the first-order theory of complex numbers, can be trivially generated, from $\varphi^\delta(v, par(\varphi))$:

$$(\neg\varphi)^\delta(v, par(\neg\varphi)) = \varphi^\delta(v, par(\varphi)) \rightarrow v = 0\,. \tag{160}$$

The same applies for the construction of the expression in first order theory of the complex numbers for the complement operator $\sim$, as it is clearly observable from its definition:

$$\sim F(v,\vec{x}) = \forall w(F(w,\vec{x})) \rightarrow \langle v,w\rangle = 0\,, \tag{161}$$

where F is a any valid function, where $\langle v,w\rangle$ is the first order theory of complex numbers equivalent to the internal product operation. The fact presented in (161), also makes the construction of the probabilistic operators effective, as the corresponding expression can be obtained by

$$(P^{\geq r}\varphi)(v,par(P^{\geq r})) = \exists w(w \neq 0 \wedge \sim\sim \varphi^{\delta}(w,par(\varphi)) \wedge |\langle v,w\rangle|^2 \geq r||v||^2||w||^2\,, \tag{162}$$

and the same applies to the amplitude operator

$$(\mathcal{A}^{=\lambda}\varphi)^{\delta}(v,par(\mathcal{A}^{=\lambda}\varphi)) = \exists w(w \neq 0 \wedge \sim\sim \varphi^{\delta}(w,par(\varphi)) \wedge \langle v,w\rangle = \lambda\,. \tag{163}$$

As a result, it can be concluded that the expressions in first order theory of complex numbers, for both the *complement*, *probabilistic* and *amplitude operatorss* can be effectively generated, and, hence, such expressions are *r-translatable*. A wider class of connectives is given by programs, whose definitions encompass general expressions such as

$$([\pi]\varphi)^{\delta}(v,par([\pi]\varphi)) = ([\pi]\varphi)^{\delta}(\hat{\pi}v,par(\varphi))\,. \tag{164}$$

We show that for every possible program π, function $([\pi]\varphi)^{\delta}(v,par([\pi]\varphi))$ is *r-translatable*. For unitary operators

$$([\mathrm{u}]\varphi)^{\delta}(v,[\mathrm{u}]\varphi) = \varphi^{\delta}(\hat{u}v,par(\varphi)) \tag{165}$$

the function can be effectively built, as the operator $\hat{u}.v$ corresponds to a set of sums and multiplications, i.e. it is similar to a matrix multiplication, for which efficient computations exist and it can also be efficiently expressed in the first order theory of the reals. Furthermore, for the **creg** program, whose definition reads as

$$\varphi^{\delta}(v,[\textbf{creg}\ \mathrm{id}\ [\mathrm{size}]]\varphi) = \varphi^{\delta}(v,\varphi) \wedge \varphi^{\delta}(v,\hat{0})\,,$$

one can conclude that it is also computable, because $\hat{0}$ is an atomic statement and the connective $\wedge$ preserves R-translatability. The same applies for $\varphi^{\delta}(v,[\textbf{qreg}\ \mathrm{id}\ [\mathrm{size}]]\varphi)$. The definition of quantum tests reads as

$$([id_i^q == \mathsf{a}]\varphi)^{\delta}(v, par([id_i^q == \mathsf{a}]\varphi)) = \varphi^{\delta}(\hat{m}v, par(\varphi)) \text{ where } \hat{m} = \left|id_i^q\right\rangle \left\langle id_i^q\right| / \sqrt{v^* \left|id_i^q\right\rangle \left\langle id_i^q\right| v}\,. \tag{166}$$

On can easily observe that the expression can be effectively generated as, matrix m is a *well-formed* statement in the first order of complex numbers, i.e. only valid operations in this theory are necessary to generate it. Given this, one can quickly verify from their definitions that the valid first order theory of reals expressions for the remaining operations, namely measurements, classical tests and if statements, can be effectively generated as they rely on constructions that have been shown to be decidable throughout this proof. Finally, the program sequential composition, defined by

$$([\pi;\pi]\varphi))^{\delta}(v, par([\pi;\pi]\varphi)) = \varphi^{\delta}(\hat{\pi}v, [\pi]\varphi)\,, \tag{167}$$

also preserves *R-translatability*, due to the fact that all possible programs available preserve it *R-translatability*. Hence all possible statements in LQASM preserve R-translatability, which makes it decidable, finishing the proof.

□

5.6 Summary

In this chapter, it were reviewed the main concepts of standard quantum logics and their derivations, namely, the family of quantum dynamic logics proposed by A. Baltag and S. Smets. Upon these results, we built a new logic targeted to a fragment of the *QASM* language, able to deal, explicitly, with probabilistic classical and quantum information. We provide a semantics for the logic, based on labelled transition systems proposed to quantum dynamic logics. We exercise the logic with a simple example of a coin toss program, which involves classical and quantum instructions, as well as with the teleportation protocol, showing it can deal *effectively* with *entanglement*. Furthermore, we show that the logic is decidable.

6

CONCLUSIONS AND FUTURE WORK

The objective of this work was to contribute towards a mathematical framework to aid in the conception of new algorithms, which is an endeavour that must cope with two historically different fields, the ones of algorithm performance and algorithm correction. For the former, a more pragmatic approach was followed, based on the study of the structures behind algorithms that allow for quantum advantage, both for the case of *efficient* algorithms that fall under the class BQP, and for the techniques employed in problems with industrial, but that, most likely will not be efficient even on quantum computers. We also made two case studies of the techniques studied, for distinct examples in the fields of biology and chemistry, both of them targeted and executed in a quantum computer, the IBM Q. For the latter, a more formal approach was followed, based on the conception of a logical system to a more realistic quantum programming language, the *QASM* language.

The major aim of chapter 2 was to revisit the theoretical foundations of quantum computation: quantum mechanics in the formalism of Hilbert spaces and density operators; the most well-known quantum computer models; the theory of quantum programming languages and how they map into physical features; the main sources of quantum advantage and a brief study on the most well-known results of quantum complexity.

Chapter 3 was devoted to the exploration of the structures behind *efficient* quantum algorithms, which included the *weak simulation* of local and d-sparse Hamiltonians, as well as the algorithms based on Fourier transform. The latter includes the Shor algorithm and the hidden subgroup problem. Moreover, the Harrow, Hassidim and Lloyd algorithm, targeted at solving linear equations, may be considered a composition of an efficient soft simulation on an Hamiltonian system representing the system of equations, and the execution of the Fourier transform, hence being dependent on the existence of an efficient algorithm to compute the former. Furthermore, it was possible to make the following observations:

- All sampling problems that can be reduced to the simulation of local, sparse or d-sparse Hamiltonian possess efficient quantum algorithms;
- All search problems that *cannot* be reduced to finding a ground-state, involving only local components with less than dimension 2, does not have an efficient quantum algorithm.

From this analysis, it was also proposed a digital simulation of the *non-radiative* energy transfer taking place in photosynthesis, which involves the *weak simulation* of a local Hamiltonian, along with environmental interaction. This way it was possible to evaluate the importance of the environmental effects in photosynthesis and obtain further insight about quantum mechanics in an open regime.

Chapter 4 aimed at exploring problems that have no quantum efficient algorithm, but for which a *polynomial* advantage is expected, once regarding them as optimization problems. There is a natural relationship between some processes in physics, such as the convergence to equilibrium of certain physical systems, and optimization problems, which make the former a good model to the latter. This idea is the basis of quantum annealing and adiabatic optimization, where a quantum advantage is expected in a wide range of problems, across a wide range of fields. Furthermore, in the so-called short-term devices method a mention should be made to the Variational method, an hybrid *classical-quantum* method, which possess a lot of applications.

One of those applications is on the field of quantum chemistry, and we explore a case study of the application of the method to find the total *ground-state* energy of the H_2 and LiH molecules, subject to a stationary electric field in a quantum computer. This is a non-trivial process, and we particularly explored:

- The fermionic formulation of quantum chemistry systems;
- The connection between fermionic Hamiltonians and the quantum circuits;
- The state preparation, running of the algorithm and the evaluation of the results.

The calculated results comprise the total energy as a function of bond length (i.e. the dissociation curve), also under an applied stationary electric field. We also evaluated the shift of the molecule's energy at a fixed d (equal to the equilibrium interatomic distance) with the electric field, i.e. the stationary electronic Stark effect. In total, our case study seems to provide evidence for the feasibility of the use of this quantum computer for small molecules, with a reasonable number of iterations performed.

In chapter 5, we briefly explore the logic induced by quantum mechanics in Hilbert spaces, i.e. the so-called quantum logic, and the path from there to logics able to deal with quantum programs, and the issues involved, particularly the inexistence of a tensor operator, focusing on the work on quantum dynamic logics introduced by Baltag and Smets. We proposed a logic able to deal with classical and quantum information simultaneously, and that, furthermore, involves measurements, to which a stochastic semantics is provided. We provide some valid rules for the logic, based on such semantics, and provide proof for a quantum coin toss program.

6.1 Future work

This work, in its course of the understanding of the sources of quantum advantage and how to use them to obtain new algorithms and applications of quantum computation, ended touching many different fields related to quantum computation. Many state-of the-art problems and lines of research to pursue were identified and explored. On the other hand, some of the ideas explored during the course of this work, were abandoned.

On the field of efficient algorithms, progress has been slow, with no big advancements in the last few years, worth of notice. In the early 2000's a huge research effort has been done on trying to extend the algorithm for the hidden subgroup problem (HSP), maintaining the exponential advantage, from Abelian groups to non-Abelian ones, such as the *dihedral* or the *symmetric group*. Efficient algorithms for the HSP in these groups would have important industrial impact, such as breaking *lattice-based cryptography*, the *cornerstone* of *post-quantum cryptography*, or solving the *graph isomorphism* problem, however, the efforts to build efficient algorithms to do so, have fallen short. However, while solving the HSP for non-Abelian groups seems to be a hard computational task, an interesting line of research is to explore applications where the exponential advantage of quantum Fourier transforms can be relevant, which may the case, for instance, in fields such as computer vision, statistics or machine learning. Furthermore, the exploration of problems that can expressed as simulations of local, or sparse, Hamiltonians can very fruitful in finding new applications for quantum algorithms.

On the side of quantum optimization, there is an *infinitude* of possible industrial and academic applications that can be explored with the current short-term devices. A particular important branch of optimization problems is the one of mixed-integer, for which the first quantum techniques are starting to appear, mostly, resorting to interaction between classical and quantum solvers. An example of these problems is the *unit commitment*, i.e. the optimization of the production schedule of individual energy stations in power grids, a problem that usually involves a large set of variables, discrete and continuous, which constitutes a completely unexplored realm for quantum computation.

On the logical side the major development would be the improvement of the *calculus* for the logic and the proof of its completeness. Furthermore, it would be also be useful to explore the formalization of the logic in a proof system, or alternatively, by the conception of a model checker.

The approach taken in the work, specially in the algorithmic part, was somehow pragmatical, however, the interest on pursuing a unified mathematical theory to deal with both complexity and correctness remains. Research paths on this direction can be given by the study of *compositional* mathematical theories behind the advantage in quantum algorithms, as well as of characteristic quantum programming languages of complexity classes. Concerning the former, a particular interesting line of research is one the application of finite model theory to quantum algorithms, where models are given by combinatory structures, rather than algebraic ones, and may possess interesting notions of logic.

BIBLIOGRAPHY

[1] Open quantum systems. 22.51 course notes, chapter 8, 2012. URL https://ocw.mit.edu/courses/nuclear-engineering/22-51-quantum-theory-of-radiation-interactions-fall-2012/lecture-notes/MIT22_51F12_Ch8.pdf. MIT OpenCourseWare.

[2] Ibm q - quantum computing, Jun 2018. Available in: https://www.research.ibm.com/ibm-q/.

[3] Scott Aaronson. Guest column: Np-complete problems and physical reality. *ACM Sigact News*, 36 (1):30–52, 2005.

[4] Scott Aaronson. Quantum computing, postselection, and probabilistic polynomial-time. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 461(2063):3473–3482, 2005.

[5] Scott Aaronson. BQP and the polynomial hierarchy. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 141–150. ACM, 2010.

[6] Scott Aaronson. Why philosophers should care about computational complexity. In *In Computability: Gödel, Turing, Church, and beyond (eds*. Citeseer, 2012.

[7] Scott Aaronson. The equivalence of sampling and searching. *Theory of Computing Systems*, 55(2): 281–298, 2014.

[8] Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70(5):052328, 2004.

[9] Scott Aaronson and John Watrous. Closed timelike curves make quantum and classical computing equivalent. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 465(2102):631–647, 2008.

[10] Daniel S Abrams and Seth Lloyd. Nonlinear quantum mechanics implies polynomial-time solution for np-complete and# p problems. *Physical Review Letters*, 81(18):3992, 1998.

[11] Daniel S Abrams and Seth Lloyd. Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors. *Physical Review Letters*, 83(24):5162, 1999.

[12] Samson Abramsky and Bob Coecke. Categorical quantum mechanics. *Handbook of quantum logic and quantum structures: quantum logic*, pages 261–324, 2009.

[13] R Adams, G Alpàr, JM Balasch Masoliver, Lejla Batina, Łukasz Chmielewski, Louiza Papachristodoulou, Peter Schwabe, Michael Tunstall, Lejla Batina, Jens Hermans, et al. Qpel: Quantum program and effect language. In *QPL 2014: Proceedings 11th workshop on Quantum Physics and Logic*, volume 1, pages 236–252. EPTCS, 2014.

[14] Julia Adolphs and Thomas Renger. How proteins trigger excitation energy transfer in the fmo complex of green sulfur bacteria. *Biophysical Journal*, 91(8):2778–2797, 2006.

[15] Diederik Aerts. Quantum axiomatics. *Handbook of quantum logic and quantum structures quantum logic*, page 79, 2009.

[16] Diederik Aerts and Bart Van Steirteghem. Quantum axiomatics and a theorem of mp soler. *International Journal of Theoretical Physics*, 39(3):497–502, 2000.

[17] Dorit Aharonov and Amnon Ta-Shma. Adiabatic quantum state generation and statistical zero knowledge. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 20–29, 2003.

[18] Dorit Aharonov, Wim Van Dam, Julia Kempe, Zeph Landau, Seth Lloyd, and Oded Regev. Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM review*, 50(4):755–787, 2008.

[19] Yakir Aharonov, Eliahu Cohen, Fabrizio Colombo, Tomer Landsberger, Irene Sabadini, Daniele C Struppa, and Jeff Tollaksen. Finally making sense of the double-slit experiment. *Proceedings of the National Academy of Sciences*, 114(25):6480–6485, 2017.

[20] Qing Ai, Tzu-Chi Yen, Bih-Yaw Jin, and Yuan-Chung Cheng. Clustered geometries exploiting quantum coherence effects for efficient energy transfer in light harvesting. *The Journal of Physical Chemistry Letters*, 4(15):2577–2584, 2013.

[21] Tameem Albash and Daniel A Lidar. Adiabatic quantum computation. *Reviews of Modern Physics*, 90(1):015002, 2018.

[22] Thorsten Altenkirch, Jonathan Grattage, Juliana K Vizzotto, and Amr Sabry. An algebra of pure quantum programming. *Electronic Notes in Theoretical Computer Science*, 170:23–47, 2007.

[23] Luigi Amico, Rosario Fazio, Andreas Osterloh, and Vlatko Vedral. Entanglement in many-body systems. *Reviews of modern physics*, 80(2):517, 2008.

[24] Matthew Amy, Andrew N Glaudell, and Neil J Ross. Number-theoretic characterizations of some restricted clifford+ t circuits. *Quantum*, 4:252, 2020.

[25] Hajnal Andréka, István Németi, and Gergely Székely. Closed timelike curves in relativistic computation. *Parallel Processing Letters*, 22(03):1240010, 2012.

[26] Eric Anschuetz, Jonathan Olson, Alán Aspuru-Guzik, and Yudong Cao. Variational quantum factoring. In *International Workshop on Quantum Technology and Optimization Problems*, pages 74–85. Springer, 2019.

[27] Bruno Apolloni, C Carvalho, and Diego De Falco. Quantum stochastic optimization. *Stochastic Processes and their Applications*, 33(2):233–244, 1989.

[28] Mateus Araújo, Philippe Allard Guérin, and Ämin Baumeler. Quantum computation with indefinite causal structures. *Physical Review A*, 96(5):052315, 2017.

[29] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[30] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[31] Alain Aspect, Philippe Grangier, and Gérard Roger. Experimental tests of realistic local theories via bell's theorem. *Physical review letters*, 47(7):460, 1981.

[32] 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.

[33] Ryan Babbush, Jarrod McClean, Dave Wecker, Alán Aspuru-Guzik, and Nathan Wiebe. Chemical basis of trotter-suzuki errors in quantum chemistry simulation. *Physical Review A*, 91(2):022311, 2015.

[34] Ryan Babbush, Dominic W Berry, Ian D Kivlichan, Annie Y Wei, Peter J Love, and Alán Aspuru-Guzik. Exponentially more precise quantum simulation of fermions in second quantization. *New Journal of Physics*, 18(3):033032, 2016.

[35] Ryan Babbush, Dominic W Berry, Yuval R Sanders, Ian D Kivlichan, Artur Scherer, Annie Y Wei, Peter J Love, and Alán Aspuru-Guzik. Exponentially more precise quantum simulation of fermions in the configuration interaction representation. *Quantum Science and Technology*, 3(1):015006, 2017.

[36] Dave Bacon. Quantum computational complexity in the presence of closed timelike curves. *Physical Review A*, 70(3):032309, 2004.

[37] Dave Bacon, Andrew M Childs, and Wim van Dam. From optimal measurement to efficient quantum algorithms for the hidden subgroup problem over semidirect product groups. In *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*, pages 469–478. IEEE, 2005.

[38] Costin Ba□descu and Prakash Panangaden. Quantum alternation: Prospects and problems. *arXiv preprint arXiv:1511.01567*, 2015.

[39] Alexandru Baltag and Sonja Smets. The logic of quantum programs. *Proc. QPL*, pages 39–56, 2004.

[40] Alexandru Baltag and Sonja Smets. Complete axiomatizations for quantum actions. *International Journal of Theoretical Physics*, 44(12):2267–2282, 2005.

[41] Alexandru Baltag and Sonja Smets. Correlated information: a logic for multi-partite quantum systems. *Electronic Notes in Theoretical Computer Science*, 270(2):3–14, 2011.

[42] Alexandru Baltag and Sonja Smets. Quantum logic as a dynamic logic. *Synthese*, 179(2):285–306, 2011.

[43] Alexandru Baltag and Sonja Smets. The dynamic turn in quantum logic. *Synthese*, 186(3):753–773, 2012.

[44] Alexandru Baltag, Jort M Bergfeld, Kohei Kishida, Joshua Sack, Sonja JL Smets, and Shengyang Zhong. Quantum probabilistic dyadic second-order logic. In *International Workshop on Logic, Language, Information, and Computation*, pages 64–80. Springer, 2013.

[45] Alexandru Baltag, Jort Bergfeld, Kohei Kishida, Joshua Sack, Sonja Smets, and Shengyang Zhong. Plqp & company: Decidable logics for quantum algorithms. *International Journal of Theoretical Physics*, 53(10):3628–3647, 2014.

[46] Francisco Barahona. On the computational complexity of ising spin glass models. *Journal of Physics A: Mathematical and General*, 15(10):3241, 1982.

[47] 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.

[48] William P Baritompa, David W Bulger, and Graham R Wood. Grover's quantum algorithm applied to global optimization. *SIAM Journal on Optimization*, 15(4):1170–1184, 2005.

[49] Panagiotis Kl Barkoutsos, Jerome F Gonthier, Igor Sokolov, Nikolaj Moll, Gian Salis, Andreas Fuhrer, Marc Ganzhorn, Daniel J Egger, Matthias Troyer, Antonio Mezzacapo, et al. Quantum algorithms for electronic structure calculations: Particle-hole hamiltonian and optimized wave-function expansions. *Physical Review A*, 98(2):022322, 2018.

[50] Robert Beals. Quantum computation of fourier transforms over symmetric groups. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 48–53. ACM, 1997.

[51] John S Bell. On the einstein podolsky rosen paradox. *Physics Physique Fizika*, 1(3):195, 1964.

[52] John S Bell. On the problem of hidden variables in quantum mechanics. *Reviews of Modern Physics*, 38(3):447, 1966.

[53] Paul Benioff. Quantum mechanical hamiltonian models of turing machines. *Journal of Statistical Physics*, 29(3):515–546, 1982.

[54] Paul Benioff. Quantum mechanical models of turing machines that dissipate no energy. *Physical Review Letters*, 48(23):1581, 1982.

[55] 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. *Physical review letters*, 70(13):1895, 1993.

[56] Charles H Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM journal on Computing*, 26(5):1510–1523, 1997.

[57] Jort Martinus Bergfeld and Joshua Sack. Deriving the correctness of quantum protocols in the probabilistic logic for quantum programs. *Soft Computing*, 21(6):1421–1441, 2017.

[58] Jort Martinus Bergfeld et al. Quantum logics for expressing and proving the correctness of quantum programs. 2019.

[59] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 11–20. ACM, 1993.

[60] Dominic W Berry. High-order quantum algorithm for solving linear differential equations. *Journal of Physics A: Mathematical and Theoretical*, 47(10):105301, 2014.

[61] Dominic W Berry, Andrew M Childs, Richard Cleve, Robin Kothari, and Rolando D Somma. Simulating hamiltonian dynamics with a truncated taylor series. *Physical review letters*, 114(9):090502, 2015.

[62] Dominic W Berry, Andrew M Childs, and Robin Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 792–809. IEEE, 2015.

[63] Dominic W Berry, Andrew M Childs, Aaron Ostrander, and Guoming Wang. Quantum algorithm for linear differential equations with exponentially improved dependence on precision. *Communications in Mathematical Physics*, 356(3):1057–1081, 2017.

[64] Dimitri P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3): 334–334, 1997.

[65] Shalabh Bhatnagar, HL Prasad, and LA Prashanth. *Stochastic recursive algorithms for optimization: simultaneous perturbation methods*, volume 434. Springer, 2012.

[66] Iwo Bialynicki-Birula and Jerzy Mycielski. Nonlinear wave mechanics. *Annals of Physics*, 100(1-2): 62–93, 1976.

[67] Garrett Birkhoff and John Von Neumann. The logic of quantum mechanics. *Annals of mathematics*, pages 823–843, 1936.

[68] Patrick Blackburn, Johan FAK van Benthem, and Frank Wolter. *Handbook of modal logic*. Elsevier, 2006.

[69] Immanuel Bloch, Jean Dalibard, and Wilhelm Zwerger. Many-body physics with ultracold gases. *Reviews of modern physics*, 80(3):885, 2008.

[70] David Bohm. A suggested interpretation of the quantum theory in terms of" hidden" variables. i. *Physical Review*, 85(2):166, 1952.

[71] Niels Bohr. I. on the constitution of atoms and molecules. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 26(151):1–25, 1913.

[72] Niels Bohr. The quantum postulate and the recent development of atomic theory 1, 1928.

[73] Sergio Boixo, Sergei V Isakov, Vadim N Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J Bremner, John M Martinis, and Hartmut Neven. Characterizing quantum supremacy in near-term devices. *Nature Physics*, 14(6):595–600, 2018.

[74] Dan Boneh and Richard J Lipton. Quantum cryptanalysis of hidden linear functions. In *Advances in Cryptology—CRYPTO'95*, pages 424–437. Springer, 1995.

[75] Dan Boneh et al. Twenty years of attacks on the rsa cryptosystem. *Notices of the AMS*, 46(2): 203–213, 1999.

[76] Adam D Bookatz. Qma-complete problems. *arXiv preprint arXiv:1212.6312*, 2012.

[77] Adam D Bookatz. Qma-complete problems. *Quantum Information & Computation*, 14(5&6):361–383, 2014.

[78] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[79] Fernando GSL Brandao and Krysta M Svore. Quantum speed-ups for solving semidefinite programs. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 415–426. IEEE, 2017.

[80] Daniel Braun and Bertrand Georgeot. Quantitative measure of interference. *Physical Review A*, 73 (2):022314, 2006.

[81] Samuel L Braunstein and Hoi-Kwong Lo. *Scalable quantum computers: Paving the way to realization*. 2001.

[82] Heinz-Peter Breuer, Francesco Petruccione, et al. *The theory of open quantum systems*. Oxford University Press on Demand, 2002.

[83] HJ Briegel, DE Browne, W Dür, R Raussendorf, and Maarten Van den Nest. Measurement-based quantum computation. *Nature Physics*, 5(1):19–26, 2009.

[84] Todd A Brun, Jim Harrington, and Mark M Wilde. Localized closed timelike curves can perfectly distinguish quantum states. *Physical Review Letters*, 102(21):210402, 2009.

[85] Olivier Brunet and Philippe Jorrand. Dynamic quantum logic for quantum programs. *International Journal of Quantum Information*, 2(01):45–54, 2004.

[86] Iulia Buluta and Franco Nori. Quantum simulators. *Science*, 326(5949):108–111, 2009.

[87] Cristian S Calude and Elena Calude. The road to quantum computational supremacy. In *Jonathan M. Borwein Commemorative Conference*, pages 349–367. Springer, 2017.

[88] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. D. Sawaya, S. Sim, L. Veis, and A. Aspuru-Guzik. Quantum chemistry in the age of quantum computing. *Chemical Reviews*, 119(19):10856 – 10915, 2019.

[89] Yudong Cao, Jonathan Romero, Jonathan P Olson, Matthias Degroote, Peter D Johnson, Mária Kieferová, Ian D Kivlichan, Tim Menke, Borja Peropadre, Nicolas PD Sawaya, et al. Quantum chemistry in the age of quantum computing. *Chemical reviews*, 119(19):10856–10915, 2019.

[90] Davide Castelvecchi. Quantum computers ready to leap out of the lab in 2017. *Nature News*, 541 (7635):9, 2017.

[91] Rohit Chadha, Paulo Mateus, and Amílcar Sernadas. Reasoning about imperative quantum programs. *Electronic Notes in Theoretical Computer Science*, 158:19–39, 2006.

[92] Shouvanik Chakrabarti, Andrew M Childs, Tongyang Li, and Xiaodi Wu. Quantum algorithms and lower bounds for convex optimization. *Quantum*, 4:221, 2020.

[93] YC Cheng and Robert J Silbey. Coherence in the b800 ring of purple bacteria lh2. *Physical Review Letters*, 96(2):028103, 2006.

[94] Yuan-Chung Cheng, Gregory S Engel, and Graham R Fleming. Elucidation of population and coherence dynamics using cross-peaks in two-dimensional electronic spectroscopy. *Chemical Physics*, 341(1-3): 285–295, 2007.

[95] Andrew M Childs. On the relationship between continuous-and discrete-time quantum walk. *Communications in Mathematical Physics*, 294(2):581–603, 2010.

[96] Andrew M Childs and Jeffrey Goldstone. Spatial search by quantum walk. *Physical Review A*, 70(2): 022314, 2004.

[97] Andrew M Childs and Robin Kothari. Limitations on the simulation of non-sparse hamiltonians. *arXiv preprint arXiv:0908.4398*, 2009.

[98] Andrew M Childs and Wim van Dam. Quantum algorithm for a generalized hidden shift problem. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1225–1232. Society for Industrial and Applied Mathematics, 2007.

[99] Andrew M Childs and Wim Van Dam. Quantum algorithms for algebraic problems. *Reviews of Modern Physics*, 82(1):1, 2010.

[100] Andrew M Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A Spielman. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 59–68, 2003.

[101] Andrew M Childs, Richard Cleve, Stephen P Jordan, and David Yonge-Mallo. Discrete-query quantum algorithm for nand trees. *arXiv preprint quant-ph/0702160*, 2007.

[102] Giulio Chiribella, Giacomo Mauro D'Ariano, Paolo Perinotti, and Benoit Valiron. Quantum computations without definite causal structure. *Physical Review A*, 88(2):022318, 2013.

[103] Kenta Cho. Semantics for a quantum programming language by operator algebras. *New Generation Computing*, 34(1-2):25–68, 2016.

[104] Kenta Cho, Bart Jacobs, Bas Westerbaan, and Abraham Westerbaan. An introduction to effectus theory. *arXiv preprint arXiv:1512.05813*, 2015.

[105] Barry A Cipra. The ising model is np-complete. *SIAM News*, 33(6):1–3, 2000.

[106] Pierre Clairambault, Marc De Visme, and Glynn Winskel. Game semantics for quantum programming. *Proceedings of the ACM on Programming Languages*, 3(POPL):32, 2019.

[107] Richard Cleve, Artur Ekert, Chiara Macchiavello, and Michele Mosca. Quantum algorithms revisited. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1969):339–354, 1998.

[108] Bob Coecke and Aleks Kissinger. *Picturing quantum processes*. Cambridge University Press, 2017.

[109] Bob Coecke and Aleks Kissinger. Picturing quantum processes. In *International Conference on Theory and Application of Diagrams*, pages 28–31. Springer, 2018.

[110] J. I. Colless, V. V. Ramasesh, D. Dahlen, M. S. Blok, M. E. Kimchi-Schwartz, J. R. McClean, J. Carter, W. A. de Jong, and I. Siddiqi. Computation of molecular spectra on a quantum processor with an error-resilient algorithm. *Physical Review X*, 8:011021, 2018.

[111] Elisabetta Collini, Cathy Y Wong, Krystyna E Wilk, Paul MG Curmi, Paul Brumer, and Gregory D Scholes. Coherently wired light-harvesting in photosynthetic marine algae at ambient temperature. *Nature*, 463(7281):644–647, 2010.

[112] Arthur H Compton. A quantum theory of the scattering of X-rays by light elements. *Physical review*, 21(5):483, 1923.

[113] Stephen A Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, 1971.

[114] 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.

[115] Andrew Cross. The ibm q experience and qiskit open-source quantum computing software. *Bulletin of the American Physical Society*, 2018.

[116] Andrew W Cross, Lev S Bishop, John A Smolin, and Jay M Gambetta. Open quantum assembly language. *arXiv preprint arXiv:1707.03429*, 2017.

[117] Toby Cubitt and Ashley Montanaro. Complexity classification of local hamiltonian problems. *SIAM Journal on Computing*, 45(2):268–316, 2016.

[118] George Cybenko. Reducing quantum computations to elementary unitary operations. *Computing in science & engineering*, 3(2):27–32, 2001.

[119] B Dakić, Tomasz Paterek, and Č Brukner. Density cubes and higher-order interference theories. *New Journal of Physics*, 16(2):023028, 2014.

[120] Ugo Dal Lago, Andrea Masini, and Margherita Zorzi. Quantum implicit computational complexity. *Theoretical Computer Science*, 411(2):377–409, 2010.

[121] Maria Luisa Dalla Chiara. Quantum logic and physical modalities. *Journal of Philosophical Logic*, 6 (1):391–404, 1977.

[122] Maria Luisa Dalla Chiara and Roberto Giuntini. The logics of orthoalgebras. *Studia Logica*, 55(1): 3–22, 1995.

[123] Maria Luisa Dalla Chiara, Roberto Giuntini, and Richard Greechie. *Reasoning in quantum theory: sharp and unsharp quantum logics*, volume 22. Springer Science & Business Media, 2013.

[124] GM D'Ariano, C Macchiavello, and MF Sacchi. On the general problem of quantum phase estimation. *Physics Letters A*, 248(2):103–108, 1998.

[125] Arnab Das and Bikas K Chakrabarti. Colloquium: Quantum annealing and analog quantum computation. *Reviews of Modern Physics*, 80(3):1061, 2008.

[126] Louis De Broglie. *Recherches sur la théorie des quanta*. PhD thesis, Migration-université en cours d'affectation, 1924.

[127] Johann Deisenhofer and Hartmut Michel. The photosynthetic reaction center from the purple bacterium rhodopseudomonas viridis. *Science*, 245(4925):1463–1473, 1989.

[128] John S Denker. A review of adiabatic computing. In *Low power electronics, 1994. Digest of technical papers., IEEE symposium*, pages 94–97. IEEE, 1994.

[129] David Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 400, pages 97–117. The Royal Society, 1985.

[130] David Deutsch. Quantum computational networks. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 425, pages 73–90. The Royal Society, 1989.

[131] David Deutsch. Quantum mechanics near closed timelike lines. *Physical Review D*, 44(10):3197, 1991.

[132] David Deutsch. *The beginning of infinity: Explanations that transform the world*. Penguin UK, 2011.

[133] David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 439, pages 553–558. The Royal Society, 1992.

[134] Ellie D'hondt and Prakash Panangaden. Quantum weakest preconditions. *Mathematical Structures in Computer Science*, 16(3):429–451, 2006.

[135] Yongcheng Ding, Xi Chen, Lucas Lamata, Enrique Solano, and Mikel Sanz. Logistic network design with a d-wave quantum annealer. *arXiv preprint arXiv:1906.10074*, 2019.

[136] Paul Adrien Maurice Dirac. A new notation for quantum mechanics. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 35, pages 416–418. Cambridge University Press, 1939.

[137] Paul Adrien Maurice Dirac. *The principles of quantum mechanics*. Number 27. Oxford university press, 1981.

[138] David P DiVincenzo. Two-bit gates are universal for quantum computation. *Physical Review A*, 51(2): 1015, 1995.

[139] J Michael Dunn, Tobias J Hagge, Lawrence S Moss, and Zhenghan Wang. Quantum logic as motivated by quantum computing. *The Journal of Symbolic Logic*, 70(2):353–359, 2005.

[140] Albert Einstein. The photoelectric effect. *Ann. Phys*, 17:132, 1905.

[141] Albert Einstein, Boris Podolsky, and Nathan Rosen. Can quantum-mechanical description of physical reality be considered complete? *Physical review*, 47(10):777, 1935.

[142] Jens Eisert and Hans J Briegel. Schmidt measure as a tool for quantifying multiparticle entanglement. *Physical Review A*, 64(2):022306, 2001.

[143] Gregory S Engel, Tessa R Calhoun, Elizabeth L Read, Tae-Kyu Ahn, Tomáš Mančal, Yuan-Chung Cheng, Robert E Blankenship, and Graham R Fleming. Evidence for wavelike energy transfer through quantum coherence in photosynthetic systems. *Nature*, 446(7137):782–786, 2007.

[144] Mark Ettinger and Peter Hoyer. A quantum observable for the graph isomorphism problem. *arXiv preprint quant-ph/9901029*, 1999.

[145] Mark Ettinger, Peter Hoyer, and Emanuel Knill. Hidden subgroup states are almost orthogonal. *arXiv preprint quant-ph/9901034*, 1999.

[146] Edward Farhi and Aram W Harrow. Quantum supremacy through the quantum approximate optimization algorithm. *arXiv preprint arXiv:1602.07674*, 2016.

[147] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106*, 2000.

[148] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, Joshua Lapan, Andrew Lundgren, and Daniel Preda. A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem. *Science*, 292(5516):472–475, 2001.

[149] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[150] Richard P Feynman. Simulating physics with computers. *International journal of theoretical physics*, 21(6/7):467–488, 1982.

[151] Richard Phillips Feynman and FL Vernon Jr. The theory of a general quantum system interacting with a linear dissipative system. *Annals of Physics*, 24:118–173, 1963.

[152] Aleta Berk Finnila, MA Gomez, C Sebenik, Catherine Stenson, and Jimmie D Doll. Quantum annealing: a new method for minimizing multidimensional functions. *Chemical physics letters*, 219 (5-6):343–348, 1994.

[153] Graham R Fleming and Rienk Van Grondelle. The primary steps of photosynthesis. *Physics Today*, 47(2):48–57, 1994.

[154] Lance Fortnow. One complexity theorist's view of quantum computing. *Theoretical Computer Science*, 292(3):597–610, 2003.

[155] Lance Fortnow. The status of the p versus np problem. *Communications of the ACM*, 52(9):78–86, 2009.

[156] Lance Fortnow and John Rogers. Complexity limitations on quantum computation. In *Computational Complexity, 1998. Proceedings. Thirteenth Annual IEEE Conference on*, pages 202–209. IEEE, 1998.

[157] David J Foulis. A half-century of quantum logic what have we learned? In *Quantum structures and the nature of reality*, pages 1–36. Springer, 1999.

[158] David J Foulis and Mary Katherine Bennett. Effect algebras and unsharp quantum logics. *Foundations of physics*, 24(10):1331–1352, 1994.

[159] David James Foulis and Charles Hamilton Randall. Empirical logic and tensor products. In *Interpretations and foundations of quantum theory*. 1981.

[160] DJ Foulis and MK Bennett. Tensor products of orthoalgebras. *Order*, 10(3):271–282, 1993.

[161] Michael Freedman, Alexei Kitaev, Michael Larsen, and Zhenghan Wang. Topological quantum computation. *Bulletin of the American Mathematical Society*, 40(1):31–38, 2003.

[162] Katalin Friedl, Gábor Ivanyos, Frédéric Magniez, Miklos Santha, and Pranab Sen. Hidden translation and translating coset in quantum computing. *SIAM Journal on Computing*, 43(1):1–24, 2014.

[163] Tobias Fritz. Quantum logic is undecidable. *Archive for Mathematical Logic*, 60(3):329–341, 2021.

[164] Claudio Gambella and Andrea Simonetto. Multiblock admm heuristics for mixed-binary optimization on classical and quantum computers. *IEEE Transactions on Quantum Engineering*, 1:1–22, 2020.

[165] Silvano Garnerone, Paolo Zanardi, and Daniel A Lidar. Adiabatic quantum algorithm for search engine ranking. *Physical review letters*, 108(23):230506, 2012.

[166] Murray Gell-Mann and James B Hartle. Quantum mechanics in the light of quantum cosmology. In *Foundations Of Quantum Mechanics In The Light Of New Technology: Selected Papers from the Proceedings of the First through Fourth International Symposia on Foundations of Quantum Mechanics*, pages 347–369. World Scientific, 1996.

[167] IM Georgescu, S Ashhab, and Franco Nori. Quantum simulation. *Reviews of Modern Physics*, 86(1): 153, 2014.

[168] Stefan Gerlich, Sandra Eibenberger, Mathias Tomandl, Stefan Nimmrichter, Klaus Hornberger, Paul J Fagan, Jens Tüxen, Marcel Mayor, and Markus Arndt. Quantum interference of large organic molecules. *Nature communications*, 2(1):1–5, 2011.

[169] Brett Giles and Peter Selinger. Exact synthesis of multiqubit clifford+ t circuits. *Physical Review A*, 87 (3):032332, 2013.

[170] John Gill. Computational complexity of probabilistic Turing machines. *SIAM Journal on Computing*, 6 (4):675–695, 1977.

[171] Austin Gilliam, Stefan Woerner, and Constantin Gonciulea. Grover adaptive search for constrained polynomial binary optimization. *Quantum*, 5:428, 2021.

[172] András Gilyén, Srinivasan Arunachalam, and Nathan Wiebe. Optimizing quantum optimization algorithms via faster quantum gradient computation. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1425–1444. SIAM, 2019.

[173] Roberto Giuntini and Heinz Greuling. Toward a formal language for unsharp properties. *Foundations of Physics*, 19(7):931–945, 1989.

[174] Andrew M Gleason. Measures on the closed subspaces of a hilbert space. *Journal of mathematics and mechanics*, pages 885–893, 1957.

[175] Robert Goldblatt. Orthomodularity is not elementary 1. *The Journal of Symbolic Logic*, 49(2):401–404, 1984.

[176] Daniel Gottesman. The heisenberg representation of quantum computers. *arXiv preprint quant-ph/9807006*, 1998.

[177] Markus Greiner, Olaf Mandel, Tilman Esslinger, Theodor W Hänsch, and Immanuel Bloch. Quantum phase transition from a superfluid to a mott insulator in a gas of ultracold atoms. *nature*, 415(6867): 39–44, 2002.

[178] 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.

[179] José Diogo Guimarães, Carlos Tavares, Luís Soares Barbosa, and Mikhail I. Vasilevskiy. Simulation of nonradiative energy transfer in photosynthetic systems using a quantum computer. *Complexity*, 2020:1–12, September 2020. doi: 10.1155/2020/3510676. URL https://doi.org/10.1155/2020/3510676.

[180] Osman Güler. *Foundations of optimization*, volume 258. Springer Science & Business Media, 2010.

[181] N. D. Gurav, S. P. Gejji, and R. K. Pathak. Electronic Stark effect for a single molecule: Theoretical UV response. *Computational and Theoretical Chemistry*, 1138:23, 2018.

[182] Devens Gust, Thomas A Moore, and Ana L Moore. Mimicking photosynthetic solar energy transduction. *Accounts of Chemical Research*, 34(1):40–48, 2001.

[183] Devens Gust, Thomas A Moore, and Ana L Moore. Solar fuels via artificial photosynthesis. *Accounts of Chemical Research*, 42(12):1890–1898, 2009.

[184] Hermann Haken and Gert Strobl. An exactly solvable model for coherent and incoherent exciton motion. *Zeitschrift für Physik A Hadrons and nuclei*, 262(2):135–148, 1973.

[185] Sean Hallgren. Fast quantum algorithms for computing the unit group and class group of a number field. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 468–474. ACM, 2005.

[186] Sean Hallgren. Polynomial-time quantum algorithms for pell's equation and the principal ideal problem. *Journal of the ACM (JACM)*, 54(1):4, 2007.

[187] Sean Hallgren, Alexander Russell, and Amnon Ta-Shma. Normal subgroup reconstruction and quantum computation using group representations. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 627–635. ACM, 2000.

[188] Sean Hallgren, Cristopher Moore, Martin Rötteler, Alexander Russell, and Pranab Sen. Limitations of quantum coset states for graph isomorphism. *Journal of the ACM (JACM)*, 57(6):34, 2010.

[189] Lucien Hardy. Quantum gravity computers: On the theory of computation with indefinite causal structure. In *Quantum reality, relativistic causality, and closing the epistemic circle*, pages 379–401. Springer, 2009.

[190] David Harel, Dexter Kozen, and Jerzy Tiuryn. Dynamic logic. In *Handbook of philosophical logic*, pages 99–217. Springer, 2001.

[191] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.

[192] Matthew B Hastings, Dave Wecker, Bela Bauer, and Matthias Troyer. Improving quantum algorithms for quantum chemistry. *arXiv preprint arXiv:1403.1539*, 2014.

[193] Ichiro Hasuo and Naohiko Hoshino. Semantics of higher-order quantum computation via geometry of interaction. *Annals of Pure and Applied Logic*, 168(2):404–469, 2017.

[194] Stephen W Hawking. Chronology protection conjecture. *Physical Review D*, 46(2):603, 1992.

[195] Warren J Hehre, Robert F Stewart, and John A Pople. Self-consistent molecular-orbital methods. i. use of gaussian expansions of slater-type atomic orbitals. *The Journal of Chemical Physics*, 51(6): 2657–2664, 1969.

[196] Werner Heisenberg. Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik. *Zeitschrift für Physik*, 43(3-4):172–198, 1927.

[197] Raymond Hemmecke, Matthias Köppe, Jon Lee, and Robert Weismantel. Nonlinear integer programming. In *50 Years of Integer Programming 1958-2008*, pages 561–618. Springer, 2010.

[198] Matthew Hennessy and Robin Milner. On observing nondeterminism and concurrency. In *International Colloquium on Automata, Languages, and Programming*, pages 299–309. Springer, 1980.

[199] Richard Hildner, Daan Brinks, Jana B Nieder, Richard J Cogdell, and Niek F van Hulst. Quantum coherent energy transfer over varying pathways in single light-harvesting complexes. *Science*, 340 (6139):1448–1451, 2013.

[200] Charles Antony Richard Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):576–580, 1969.

[201] Kodi Husimi. Studies on the foundation of quantum mechanics. i. *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series*, 19:766–789, 1937.

[202] Kazuki Ikeda, Yuma Nakamura, and Travis S Humble. Application of quantum annealing to nurse scheduling problem. *Scientific reports*, 9(1):1–10, 2019.

[203] Yoshifumi Inui and François Le Gall. An efficient algorithm for the hidden subgroup problem over a class of semi-direct product groups. Technical report, 2004.

[204] Christopher J Isham. Quantum logic and the histories approach to quantum theory. *Journal of Mathematical Physics*, 35(5):2157–2185, 1994.

[205] Akihito Ishizaki and Yoshitaka Tanimura. Nonperturbative non-markovian quantum master equation: Validity and limitation to calculate nonlinear response functions. *Chemical Physics*, 347(1-3):185–193, 2008.

[206] Gábor Ivanyos, Frédéric Magniez, and Miklos Santha. Efficient quantum algorithms for some instances of the non-abelian hidden subgroup problem. *International Journal of Foundations of Computer Science*, 14(05):723–739, 2003.

[207] Rahul Jain, Zhengfeng Ji, Sarvagya Upadhyay, and John Watrous. Qip= pspace. *Communications of the ACM*, 53(12):102–109, 2010.

[208] JM Jauch and C Piron. Helv. physica acta 36, 827 (1963); d. finkelstein, jm jauch, s. schiminovich and d. speiser,'. *J. Math. Phys*, 3:207, 1962.

[209] Jan Jeske, David J Ing, Martin B Plenio, Susana F Huelga, and Jared H Cole. Bloch-redfield equations for modeling light-harvesting complexes. *The Journal of Chemical Physics*, 142(6):064104, 2015.

[210] Bogumil Jeziorski and Hendrik J Monkhorst. Coupled-cluster method for multideterminantal reference states. *Physical Review A*, 24(4):1668, 1981.

[211] J Robert Johansson, Paul D Nation, and Franco Nori. Qutip: An open-source python framework for the dynamics of open quantum systems. *Computer Physics Communications*, 183(8):1760–1772, 2012.

[212] Stephen Jordan. Quantum Algorithm Zoo, 2015. http://math.nist.gov/quantum/zoo/.

[213] Richard Jozsa. On the simulation of quantum circuits. *arXiv preprint quant-ph/0603163*, 2006.

[214] Richard Jozsa and Noah Linden. On the role of entanglement in quantum-computational speed-up. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 459(2036):2011–2032, 2003.

[215] Tadashi Kadowaki and Hidetoshi Nishimori. Quantum annealing in the transverse ising model. *Physical Review E*, 58(5):5355, 1998.

[216] Amir Kalev, Tongyang Li, Cedric Yen-Yu Lin, Krysta M Svore, Xiaodi Wu, et al. Quantum sdp solvers: Large speed-ups, optimality, and applications to quantum learning. *Leibniz international proceedings in informatics*, 2019.

[217] Gudrun Kalmbach. *Orthomodular lattices*. Number 18. Academic Press, 1983.

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

[219] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[220] Ivan Kassal and Alán Aspuru-Guzik. Quantum algorithm for molecular properties and geometry optimization. *The Journal of chemical physics*, 131(22):224102, 2009.

[221] Ivan Kassal, Stephen P Jordan, Peter J Love, Masoud Mohseni, and Alán Aspuru-Guzik. Polynomial-time quantum algorithm for the simulation of chemical dynamics. *Proceedings of the National Academy of Sciences*, pages pnas–0808245105, 2008.

[222] Tosio Kato. On the adiabatic theorem of quantum mechanics. *Journal of the Physical Society of Japan*, 5(6):435–439, 1950.

[223] Julia Kempe, Alexei Kitaev, and Oded Regev. The complexity of the local hamiltonian problem. *SIAM Journal on Computing*, 35(5):1070–1097, 2006.

[224] Anatole Kenfack and Karol Z□yczkowski. Negativity of the wigner function as an indicator of non-classicality. *Journal of Optics B: Quantum and Semiclassical Optics*, 6(10):396, 2004.

[225] A Yu Kitaev. Quantum measurements and the Abelian stabilizer problem. *arXiv preprint quant-ph/9511026*, 1995.

[226] Alexei Yu Kitaev, Alexander Shen, and Mikhail N Vyalyi. *Classical and quantum computation*, volume 47. American Mathematical Society Providence, 2002.

[227] Morten Kjaergaard, Mollie E Schwartz, Jochen Braumüller, Philip Krantz, Joel I-J Wang, Simon Gustavsson, and William D Oliver. Superconducting qubits: Current state of play. *Annual Review of Condensed Matter Physics*, 11:369–395, 2020.

[228] Vadym Kliuchnikov, Dmitri Maslov, and Michele Mosca. Fast and efficient exact synthesis of single qubit unitaries generated by clifford and t gates. *arXiv preprint arXiv:1206.5236*, 2012.

[229] Emmanuel Knill. Conventions for quantum pseudocode. Technical report, Los Alamos National Lab., NM (United States), 1996.

[230] Simon Kochen and Ernst P Specker. The calculus of partial propositional functions. In *Ernst Specker Selecta*, pages 222–234. Springer, 1990.

[231] Simon Kochen and Ernst P Specker. Logical structures arising in quantum theory. In *The theory of models*, pages 177–189. Elsevier, 2014.

[232] Gary Kochenberger, Jin-Kao Hao, Fred Glover, Mark Lewis, Zhipeng Lü, Haibo Wang, and Yang Wang. The unconstrained binary quadratic programming problem: a survey. *Journal of Combinatorial Optimization*, 28(1):58–81, 2014.

[233] Dexter Kozen. Semantics of probabilistic programs. *Journal of computer and system sciences*, 22 (3):328–350, 1981.

[234] Dexter Kozen. A probabilistic pdl. *Journal of Computer and System Sciences*, 30(2):162–178, 1985.

[235] Philip Krantz, Morten Kjaergaard, Fei Yan, Terry P Orlando, Simon Gustavsson, and William D Oliver. A quantum engineer's guide to superconducting qubits. *Applied Physics Reviews*, 6(2):021318, 2019.

[236] Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM Journal on Computing*, 35(1):170–188, 2005.

[237] Neill Lambert, Yueh-Nan Chen, Yuan-Chung Cheng, Che-Ming Li, Guang-Yin Chen, and Franco Nori. Quantum biology. *Nature Physics*, 9(1):10–18, 2013.

[238] Benjamin P Lanyon, James D Whitfield, Geoff G Gillett, Michael E Goggin, Marcelo P Almeida, Ivan Kassal, Jacob D Biamonte, Masoud Mohseni, Ben J Powell, Marco Barbieri, et al. Towards quantum chemistry on a quantum computer. *Nature chemistry*, 2(2):106, 2010.

[239] Ciarán M Lee and John H Selby. Higher-order interference in extensions of quantum theory. *Foundations of Physics*, 47(1):89–112, 2017.

[240] Hohjai Lee, Yuan-Chung Cheng, and Graham R Fleming. Coherence dynamics in photosynthesis: protein protection of excitonic coherence. *Science*, 316(5830):1462–1465, 2007.

[241] Jan A Leegwater. Coherent versus incoherent energy transfer and trapping in photosynthetic antenna complexes. *The Journal of Physical Chemistry*, 100(34):14403–14409, 1996.

[242] I.N. Levine. *Quantum Chemistry*. Pearson advanced chemistry series. Pearson, 2014. ISBN 9780321890603. URL https://books.google.pt/books?id=ht6jMQEACAAJ.

[243] Nathan S Lewis. Research opportunities to advance solar energy utilization. *Science*, 351(6271), 2016.

[244] Chi-Kwong Li, Rebecca Roberts, and Xiaoyan Yin. Decomposition of unitary matrices and quantum gates. *International Journal of Quantum Information*, 11(01):1350015, 2013.

[245] Daniel A Lidar and Haobin Wang. Calculating the thermal rate constant with exponential speedup on a quantum computer. *Physical Review E*, 59(2):2429, 1999.

[246] Junyi Liu, Bohua Zhan, Shuling Wang, Shenggang Ying, Tao Liu, Yangjia Li, Mingsheng Ying, and Naijun Zhan. Formal verification of quantum algorithms using quantum hoare logic. In *International conference on computer aided verification*, pages 187–207. Springer, 2019.

[247] Seth Lloyd. Almost any quantum logic gate is universal. *Physical Review Letters*, 75(2):346, 1995.

[248] Seth Lloyd. Universal quantum simulators. *Science*, pages 1073–1078, 1996.

[249] Seth Lloyd. Quantum search without entanglement. *Physical Review A*, 61(1):010301, 1999.

[250] Seth Lloyd and Samuel L Braunstein. Quantum computation over continuous variables. In *Quantum information with continuous variables*, pages 9–17. Springer, 1999.

[251] Seth Lloyd, Lorenzo Maccone, Raul Garcia-Patron, Vittorio Giovannetti, Yutaka Shikano, Stefano Pirandola, Lee A Rozema, Ardavan Darabi, Yasaman Soudagar, Lynden K Shalm, et al. Closed timelike curves via postselection: theory and experimental test of consistency. *Physical review letters*, 106(4):040403, 2011.

[252] Neil B Lovett, Sally Cooper, Matthew Everitt, Matthew Trevers, and Viv Kendon. Universal quantum computation using the discrete-time quantum walk. *Physical Review A*, 81(4):042330, 2010.

[253] Guang Hao Low. Hamiltonian simulation with nearly optimal dependence on spectral norm. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 491–502, 2019.

[254] Guang Hao Low and Isaac L Chuang. Optimal hamiltonian simulation by quantum signal processing. *Physical review letters*, 118(1):010501, 2017.

[255] Guang Hao Low and Isaac L Chuang. Hamiltonian simulation by qubitization. *Quantum*, 3:163, 2019.

[256] Guang Hao Low and Nathan Wiebe. Hamiltonian simulation in the interaction picture. *arXiv preprint arXiv:1805.00675*, 2018.

[257] Andrew Lucas. Ising formulations of many np problems. *Frontiers in Physics*, 2:5, 2014.

[258] Frédéric Magniez, Ashwin Nayak, Peter C Richter, and Miklos Santha. On the hitting times of quantum versus random walks. *Algorithmica*, 63(1):91–116, 2012.

[259] Octavio Malherbe. Categorical models of computation: partially traced categories and presheaf models of quantum computation. *arXiv preprint arXiv:1301.5087*, 2013.

[260] Andrea Mari and Jens Eisert. Positive wigner functions render classical simulation of quantum computation efficient. *Physical review letters*, 109(23):230503, 2012.

[261] Igor L Markov and Yaoyun Shi. Simulating quantum computation by contracting tensor networks. *SIAM Journal on Computing*, 38(3):963–981, 2008.

[262] Paulo Mateus and Amílcar Sernadas. Exogenous quantum logic. In *Proceedings of CombLog*, volume 4, pages 141–149, 2004.

[263] Ken Matsumoto and Kazuyuki Amano. Representation of quantum circuits with clifford and

$$\pi i/8$$

gates. *arXiv preprint arXiv:0806.3834*, 2008.

[264] 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.

[265] Nicolas C Menicucci, Peter Van Loock, Mile Gu, Christian Weedbrook, Timothy C Ralph, and Michael A Nielsen. Universal quantum computation with continuous-variable cluster states. *Physical review letters*, 97(11):110501, 2006.

[266] Masoud Mohseni, Yasser Omar, Gregory S Engel, and Martin B Plenio. *Quantum effects in biology*. Cambridge University Press, 2014.

[267] Nikolaj Moll, Panagiotis Barkoutsos, Lev S Bishop, Jerry M Chow, Andrew Cross, Daniel J Egger, Stefan Filipp, Andreas Fuhrer, Jay M Gambetta, Marc Ganzhorn, et al. Quantum optimization using variational algorithms on near-term quantum devices. *arXiv preprint arXiv:1710.01022*, 2017.

[268] Nikolaj Moll, Panagiotis Barkoutsos, Lev S Bishop, Jerry M Chow, Andrew Cross, Daniel J Egger, Stefan Filipp, Andreas Fuhrer, Jay M Gambetta, Marc Ganzhorn, et al. Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Science and Technology*, 3(3): 030503, 2018.

[269] Mauro ES Morales, Timur Tlyachev, and Jacob Biamonte. Variational learning of grover's quantum search algorithm. *Physical Review A*, 98(6):062333, 2018.

[270] Michele Mosca. Abelian hidden subgroup problem. *Encyclopedia of Algorithms*, pages 1–99, 2008.

[271] Michele Mosca. Quantum algorithms. *arXiv preprint arXiv:0808.0369*, 2008.

[272] Sarah Mostame, Patrick Rebentrost, Alexander Eisfeld, Andrew J Kerman, Dimitris I Tsomokos, and Alan Aspuru-Guzik. Quantum simulator of an open quantum system using superconducting qubits: exciton transport in photosynthetic complexes. *New Journal of Physics*, 14(10):105013, 2012.

[273] R Muller. Pyquante-python quantum chemistry. *URL http://pyquante. sourceforge. net*, 2017.

[274] Chetan Nayak, Steven H Simon, Ady Stern, Michael Freedman, and Sankar Das Sarma. Non-abelian anyons and topological quantum computation. *Reviews of Modern Physics*, 80(3):1083, 2008.

[275] John Von Neumann. *Mathematical foundations of quantum mechanics*. Number 2. Princeton university press, 1955.

[276] Michael A Nielsen and Isaac Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2010.

[277] Gerardo Ortiz, JE Gubernatis, Emanuel Knill, and Raymond Laflamme. Quantum algorithms for fermionic simulations. *Physical Review A*, 64(2):022319, 2001.

[278] Román Orús, Samuel Mugel, and Enrique Lizaso. Forecasting financial crashes with quantum computing. *Physical Review A*, 99(6):060301, 2019.

[279] Roman Orus, Samuel Mugel, and Enrique Lizaso. Quantum computing for finance: overview and prospects. *Reviews in Physics*, 4:100028, 2019.

[280] Stefano Paesani, Andreas A Gentile, Raffaele Santagati, Jianwei Wang, Nathan Wiebe, David P Tew, Jeremy L O'Brien, and Mark G Thompson. Experimental bayesian quantum phase estimation on a silicon photonic chip. *Physical review letters*, 118(10):100503, 2017.

[281] Michele Pagani, Peter Selinger, and Benoît Valiron. Applying quantitative semantics to higher-order quantum computing. In *ACM SIGPLAN Notices*, volume 49, pages 647–658. ACM, 2014.

[282] Gitt Panitchayangkoon, Dugan Hayes, Kelly A Fransted, Justin R Caram, Elad Harel, Jianzhong Wen, Robert E Blankenship, and Gregory S Engel. Long-lived quantum coherence in photosynthetic complexes at physiological temperature. *Proceedings of the National Academy of Sciences*, 107(29): 12766–12770, 2010.

[283] Christos H Papadimitriou. Computational complexity,(1994). *Reading: Addison-Wesley*.

[284] Edwin Pednault, John A Gunnels, Giacomo Nannicini, Lior Horesh, and Robert Wisnieff. Leveraging secondary storage to simulate deep 54-qubit sycamore circuits. *arXiv preprint arXiv:1910.09534*, 2019.

[285] Alejandro Perdomo-Ortiz, Neil Dickson, Marshall Drew-Brook, Geordie Rose, and Alán Aspuru-Guzik. Finding low-energy conformations of lattice protein models by quantum annealing. *Scientific reports*, 2:571, 2012.

[286] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5:4213, 2014.

[287] Olivier Pfister. Continuous-variable quantum computing in the quantum optical frequency comb. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 53(1):012001, 2019.

[288] Constantin Piron. Axiomatique quantique. *Helvetica physica acta*, 37(4-5):439, 1964.

[289] Max Planck. On the theory of the energy distribution law of the normal spectrum. *Verh. Deut. Phys. Ges*, 2:237–245, 1900.

[290] Martin B Plenio and Shashank S Virmani. An introduction to entanglement theory. *Quantum information and coherence*, pages 173–209, 2014.

[291] Renato Portugal. *Quantum walks and search algorithms*. Springer, 2018.

[292] Anton Potočnik, Arno Bargerbos, Florian AYN Schröder, Saeed A Khan, Michele C Collodo, Simone Gasparinetti, Yves Salathé, Celestino Creatore, Christopher Eichler, Hakan E Türeci, et al. Studying light-harvesting models with superconducting circuits. *Nature Communications*, 9(1):1–7, 2018.

[293] David Poulin, Matthew B Hastings, Dave Wecker, Nathan Wiebe, Andrew C Doherty, and Matthias Troyer. The trotter step size required for accurate quantum simulation of quantum chemistry. *arXiv preprint arXiv:1406.4920*, 2014.

[294] Michael JD Powell. A view of algorithms for optimization without derivatives. *Mathematics Today-Bulletin of the Institute of Mathematics and its Applications*, 43(5):170–174, 2007.

[295] Vaughan R Pratt. Semantical consideration on floyo-hoare logic. In *17th Annual Symposium on Foundations of Computer Science (SFCS 1976)*, pages 109–121. IEEE, 1976.

[296] John Preskill. Quantum computing and the entanglement frontier. *arXiv preprint arXiv:1203.5813*, 2012.

[297] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[298] Lorenzo M Procopio, Amir Moqanaki, Mateus Araújo, Fabio Costa, Irati Alonso Calafell, Emma G Dowd, Deny R Hamel, Lee A Rozema, Časlav Brukner, and Philip Walther. Experimental superposition of orders of quantum gates. *Nature communications*, 6(1):1–6, 2015.

[299] Google AI Quantum et al. Hartree-fock on a superconducting qubit quantum computer. *Science*, 369 (6507):1084–1089, 2020.

[300] Charles Randall and David Foulis. Tensor products of quantum logics do not exist. *Notices Amer. Math. Soc*, 26(6), 1979.

[301] Robert Raussendorf, Daniel E Browne, and Hans J Briegel. Measurement-based quantum computation on cluster states. *Physical review A*, 68(2):022312, 2003.

[302] Patrick Rebentrost, Masoud Mohseni, Ivan Kassal, Seth Lloyd, and Alán Aspuru-Guzik. Environment-assisted quantum transport. *New Journal of Physics*, 11(3):033003, 2009.

[303] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Physical review letters*, 113(13):130503, 2014.

[304] Patrick Rebentrost, Adrian Steffens, Iman Marvian, and Seth Lloyd. Quantum singular-value decomposition of nonsparse low-rank matrices. *Physical review A*, 97(1):012327, 2018.

[305] Patrick Rebentrost, Maria Schuld, Leonard Wossnig, Francesco Petruccione, and Seth Lloyd. Quantum gradient descent and newton's method for constrained polynomial optimization. *New Journal of Physics*, 21(7):073023, 2019.

[306] O. Regev. Quantum computation and lattice problems. *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, 33(3):1–20, 2002. ISSN 0272-5428. doi: 10.1109/SFCS.2002.1181976.

[307] David R Reichman and Robert J Silbey. On the relaxation of a two-level system: Beyond the weak-coupling approximation. *The Journal of Chemical Physics*, 104(4):1506–1518, 1996.

[308] Markus Reiher, Nathan Wiebe, Krysta M Svore, Dave Wecker, and Matthias Troyer. Elucidating reaction mechanisms on quantum computers. *Proceedings of the National Academy of Sciences*, 114(29):7555–7560, 2017.

[309] Eleanor G Rieffel, Davide Venturelli, Bryan O'Gorman, Minh B Do, Elicia M Prystay, and Vadim N Smelyanskiy. A case study in programming a quantum annealer for hard operational planning problems. *Quantum Information Processing*, 14(1):1–36, 2015.

[310] Martin Roetteler and Thomas Beth. Polynomial-time solution to the hidden subgroup problem for a class of non-abelian groups. *arXiv preprint quant-ph/9812070*, 1998.

[311] Jonathan Romero, Jonathan P Olson, and Alan Aspuru-Guzik. Quantum autoencoders for efficient compression of quantum data. *Quantum Science and Technology*, 2(4):045001, 2017.

[312] Gili Rosenberg, Poya Haghnegahdar, Phil Goddard, Peter Carr, Kesheng Wu, and Marcos López De Prado. Solving the optimal trading trajectory problem using a quantum annealer. *IEEE Journal of Selected Topics in Signal Processing*, 10(6):1053–1060, 2016.

[313] Amr Sabry, Benoît Valiron, and Juliana Kaizer Vizzotto. From symmetric pattern-matching to quantum control. In *International Conference on Foundations of Software Science and Computation Structures*, pages 348–364. Springer, 2018.

[314] Simon Saunders, Jonathan Barrett, Adrian Kent, and David Wallace. *Many worlds?: Everett, quantum theory, & reality*. Oxford University Press, 2010.

[315] Irmi Sax, Sebastian Feld, Sebastian Zielinski, Thomas Gabor, Claudia Linnhoff-Popien, and Wolfgang Mauerer. Approximate approximation on a quantum annealer. In *Proceedings of the 17th ACM International Conference on Computing Frontiers*, pages 108–117, 2020.

[316] James Schneeloch, Christopher C Tison, Michael L Fanto, Paul M Alsing, and Gregory A Howland. Quantifying entanglement in a 68-billion-dimensional quantum state space. *Nature communications*, 10(1):1–7, 2019.

[317] E Schrödinger. On the Relation of the Heisenberg-Born-Jordan Quantum Mechanics and Mine. *Annalen der Physik*, 79:734–756, 1926.

[318] Erwin Schrödinger. An undulatory theory of the mechanics of atoms and molecules. *Physical Review*, 28(6):1049, 1926.

[319] Erwin Schrödinger. Discussion of probability relations between separated systems. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 31, pages 555–563. Cambridge University Press, 1935.

[320] Wolf-Dieter Schubert, Olaf Klukas, Norbert Krauß, Wolfram Saenger, Petra Fromme, and Horst Tobias Witt. Photosystem i of synechococcus elongatus at 4 å resolution: comprehensive structure analysis. *Journal of Molecular Biology*, 272(5):741–769, 1997.

[321] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. Prediction by linear regression on a quantum computer. *Physical Review A*, 94(2):022342, 2016.

[322] Peter Selinger. Towards a semantics for higher-order quantum computation. In *Proceedings of the 2nd International Workshop on Quantum Programming Languages, TUCS General Publication*, volume 33, pages 127–143. Citeseer, 2004.

[323] Peter Selinger. Towards a quantum programming language. *Mathematical Structures in Computer Science*, 14(4):527–586, 2004.

[324] Peter Selinger and Benoit Valiron. A lambda calculus for quantum computation with classical control. *Mathematical Structures in Computer Science*, 16(03):527–552, 2006.

[325] Yangchao Shen, Xiang Zhang, Shuaining Zhang, Jing-Ning Zhang, Man-Hong Yung, and Kihwan Kim. Quantum implementation of the unitary coupled cluster for simulating molecular electronic structure. *Physical Review A*, 95(2):020501, 2017.

[326] Neil Shenvi, Julia Kempe, and K Birgitta Whaley. Quantum random-walk search algorithm. *Physical Review A*, 67(5):052307, 2003.

[327] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994.

[328] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM journal on computing*, 26(5):1484–1509, 1997.

[329] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[330] Daniel R Simon. On the power of quantum computation. *SIAM journal on computing*, 26(5): 1474–1483, 1997.

[331] Kartik Singhal. Quantum hoare type theory. *arXiv preprint arXiv:2012.02154*, 2020.

[332] Steven S Skiena. *The algorithm design manual*. Springer International Publishing, 2020.

[333] Maria Pia Soler. Characterization of hilbert spaces by orthomodular spaces. *Communications in Algebra*, 23(1):219–243, 1995.

[334] Rafael D Sorkin. Quantum mechanics as quantum measure theory. *Modern Physics Letters A*, 9(33): 3119–3127, 1994.

[335] Dan Stahlke. Quantum interference as a resource for quantum speedup. *Physical Review A*, 90(2): 022302, 2014.

[336] Qiming Sun, Timothy C Berkelbach, Nick S Blunt, George H Booth, Sheng Guo, Zhendong Li, Junzi Liu, James D McClain, Elvira R Sayfutyarova, Sandeep Sharma, et al. Pyscf: the python-based simulations of chemistry framework. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 8(1): e1340, 2018.

[337] Attila Szabo and Neil S Ostlund. *Modern quantum chemistry: introduction to advanced electronic structure theory*. Courier Corporation, 2012.

[338] Yoshitaka Tanimura. Stochastic liouville, langevin, fokker–planck, and master equation approaches to quantum dissipative systems. *Journal of the Physical Society of Japan*, 75(8):082001, 2006.

[339] Yoshitaka Tanimura and Ryogo Kubo. Time evolution of a quantum system in contact with a nearly gaussian-markoffian noise bath. *Journal of the Physical Society of Japan*, 58(1):101–114, 1989.

[340] Ming-Jie Tao, Ming Hua, Qing Ai, and Fu-Guo Deng. Quantum simulation of clustered photosynthetic light harvesting in a superconducting quantum circuit. *arXiv preprint arXiv:1810.05825*, 2018.

[341] Alfred Tarski. A decision method for elementary algebra and geometry. In *Quantifier elimination and cylindrical algebraic decomposition*, pages 24–84. Springer, 1998.

[342] Carlos Tavares. A dynamic logic for qasm programs. In Luís Soares Barbosa and Alexandru Baltag, editors, *Dynamic Logic. New Trends and Applications*, pages 209–217, Cham, 2020. Springer International Publishing. ISBN 978-3-030-38808-9.

[343] Carlos Tavares, Sofia Oliveira, Vitor Fernandes, Andrei Postnikov, and Mikhail I Vasilevskiy. Quantum simulation of the ground-state stark effect in small molecules: a case study using ibm q. *Soft Computing*, pages 1–24, 2021.

[344] Valentin Torggler, Philipp Aumann, Helmut Ritsch, and Wolfgang Lechner. A quantum n-queens solver. *Quantum*, 3:149, 2019.

[345] Dominique Unruh. Quantum relational hoare logic. *Proceedings of the ACM on Programming Languages*, 3(POPL):1–31, 2019.

[346] William Unruh and Ralf Schützhold. *Quantum analogues: from phase transitions to black holes and cosmology*, volume 718. Springer, 2007.

[347] LG Valiant. Quantum computers that can be simulated classically in polynomial time. in?, editor. In *Proceedings of the 33rd Annual ACM Symposium on the Theory of Computation (STOC'01)*, 2001.

[348] Joran van Apeldoorn and András Gilyén. Improvements in quantum sdp-solving with applications. *arXiv preprint arXiv:1804.05058*, 2018.

[349] Joran Van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf. Quantum sdp-solvers: Better upper and lower bounds. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 403–414. IEEE, 2017.

[350] Joran van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf. Convex optimization using quantum oracles. *Quantum*, 4:220, 2020.

[351] Wim Van Dam, Sean Hallgren, and Lawrence Ip. Quantum algorithms for some hidden shift problems. *SIAM Journal on Computing*, 36(3):763–778, 2006.

[352] Juha J Vartiainen, Mikko Möttönen, and Martti M Salomaa. Efficient decomposition of quantum gates. *Physical review letters*, 92(17):177902, 2004.

[353] Vlatko Vedral. Quantifying entanglement in macroscopic systems. *Nature*, 453(7198):1004–1007, 2008.

[354] Vlatko Vedral and Martin B Plenio. Entanglement measures and purification procedures. *Physical Review A*, 57(3):1619, 1998.

[355] Vlatko Vedral et al. *Introduction to quantum information science*. Oxford University Press on Demand, 2006.

[356] Victor Veitch, Nathan Wiebe, Christopher Ferrie, and Joseph Emerson. Efficient simulation scheme for a class of quantum optics experiments with non-negative wigner representation. *New Journal of Physics*, 15(1):013037, 2013.

[357] Davide Venturelli, Dominic JJ Marchand, and Galo Rojo. Quantum annealing implementation of job-shop scheduling. *arXiv preprint arXiv:1506.08479*, 2015.

[358] Guifré Vidal. Efficient classical simulation of slightly entangled quantum computations. *Physical review letters*, 91(14):147902, 2003.

[359] John Von Neumann. Wahrscheinlichkeitstheoretischer aufbau der quantenmechanik. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 1927: 245–272, 1927.

[360] John Von Neumann. Wahrscheinlichkeitstheoretischer aufbau der quantenmechanik. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 1927: 245–272, 1927.

[361] S. Vuckovic, L. O. Wagner, A. Mirtschink, and P. Gori-Giorgi. Hydrogen molecule dissociation curve with functionals based on the strictly correlated regime. *Journal of Chemical Theory and Computations*, 11:3153, 2015.

[362] Bi-Xue Wang, Ming-Jie Tao, Qing Ai, Tao Xin, Neill Lambert, Dong Ruan, Yuan-Chung Cheng, Franco Nori, Fu-Guo Deng, and Gui-Lu Long. Efficient quantum simulation of photosynthetic light harvesting. *NPJ Quantum Information*, 4(1):1–6, 2018.

[363] Chunhao Wang and Leonard Wossnig. A quantum algorithm for simulating non-sparse hamiltonians. *arXiv preprint arXiv:1803.08273*, 2018.

[364] Hefeng Wang, Sahel Ashhab, and Franco Nori. Quantum algorithm for simulating the dynamics of an open quantum system. *Physical Review A*, 83(6):062317, 2011.

[365] John Watrous. PSPACE has constant-round quantum interactive proof systems. *Theoretical Computer Science*, 292(3):575–588, 2003.

[366] Dave Wecker, Bela Bauer, Bryan K Clark, Matthew B Hastings, and Matthias Troyer. Gate-count estimates for performing quantum chemistry on small quantum computers. *Physical Review A*, 90 (2):022305, 2014.

[367] Christian Weedbrook, Stefano Pirandola, Raúl García-Patrón, Nicolas J Cerf, Timothy C Ralph, Jeffrey H Shapiro, and Seth Lloyd. Gaussian quantum information. *Reviews of Modern Physics*, 84(2):621, 2012.

[368] Steven Weinberg. Precision tests of quantum mechanics. *Physical Review Letters*, 62(5):485, 1989.

[369] Steven Weinberg. Testing quantum mechanics. *Annals of Physics*, 194(2):336–386, 1989.

[370] James D Whitfield, Jacob Biamonte, and Alán Aspuru-Guzik. Simulation of electronic structure hamiltonians using quantum computers. *Molecular Physics*, 109(5):735–750, 2011.

[371] Nathan Wiebe, Dominic Berry, Peter Høyer, and Barry C Sanders. Higher order decompositions of ordered operator exponentials. *Journal of Physics A: Mathematical and Theoretical*, 43(6):065203, 2010.

[372] Nathan Wiebe, Daniel Braun, and Seth Lloyd. Quantum algorithm for data fitting. *Physical review letters*, 109(5):050505, 2012.

[373] Stephen Wiesner. Simulations of many-body quantum systems by a quantum computer. Technical report, 1996.

[374] Eugene P Wigner. The problem of measurement. *American Journal of Physics*, 31(1):6–15, 1963.

[375] Stefan Woerner and Daniel J Egger. Quantum risk analysis. *npj Quantum Information*, 5(1):1–8, 2019.

[376] William K Wootters and Wojciech H Zurek. A single quantum cannot be cloned. *Nature*, 299(5886): 802–803, 1982.

[377] Lei Xu, ZR Gong, Ming-Jie Tao, Qing Ai, et al. Artificial light harvesting by dimerized möbius ring. *Physical Review E*, 97(4):042124, 2018.

[378] A Chi-Chih Yao. Quantum circuit complexity. In *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*, pages 352–361. IEEE, 1993.

[379] Mingsheng Ying. Floyd–hoare logic for quantum programs. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 33(6):19, 2011.

[380] Mingsheng Ying and Yuan Feng. Model checking quantum systems—a survey. *arXiv preprint arXiv:1807.09466*, 2018.

[381] Mingsheng Ying, Nengkun Yu, and Yuan Feng. Alternation in quantum programming: from superposition of data to superposition of programs. *arXiv preprint arXiv:1402.5172*, 2014.

[382] Christof Zalka. Simulating quantum systems on a quantum computer. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1969):313–322, 1998.

[383] H Dieter Zeh. On the interpretation of measurement in quantum theory. *Foundations of Physics*, 1 (1):69–76, 1970.

[384] Han-Sen Zhong, Hui Wang, Yu-Hao Deng, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Jian Qin, Dian Wu, Xing Ding, Yi Hu, et al. Quantum computational advantage using photons. *Science*, 370 (6523):1460–1463, 2020.

[385] Shengyang Zhong et al. *Orthogonality and quantum geometry: Towards a relational reconstruction of quantum theory*. 9789064648946, 2015.

[386] Wojciech H Zurek. Decoherence and the transition from quantum to classical–revisited. *arXiv preprint quant-ph/0306072*, 2003.

A

APPENDIX A - CALCULATION OF THE MATRIX ELEMENTS

a.1 STO-LG wavefunctions

The STO-3G type combinations of Gaussian functions are used to calculate the matrix elements of various electronic interactions in the molecules under study. As the minimal basis of the H_2 molecule includes the s-type orbitals only, whereas that for LiH comprises both the s- and the p-type orbitals, by throughout covering the latter molecule we leave a possibility to fall back to the H_2 case by removing the factor of 3 (Li nucleus charge) in those matrix elements where it appears explicitly (namely, in Table 13 below). Also, the parameters of the STO-3G functions have to be chosen accordingly (see Table 12 below).

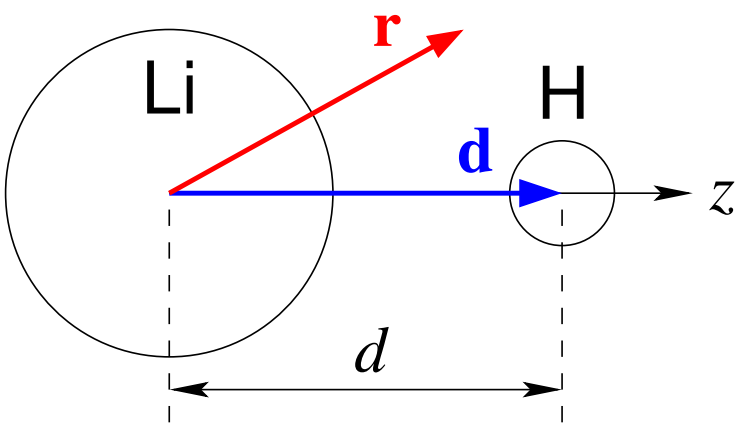


Figure 24: Geometry setting in calculations concerning the LiH molecule.

The minimal basis will include the following atomic orbitals: $1s$ for H; $1s$, $2s$ and $2p_z$ for Li. All of them will be approximated by the STO-3G type combinations of the following Gaussian functions [337]:

$$\psi_{1s}(\zeta) = \sum_{i=1}^{L} d_{i,1s} g_{1s}(\alpha_{i,1s}); \tag{168}$$

$$\psi_{2s}(\zeta) = \sum_{i=1}^{L} d_{i,2s} g_{1s}(\alpha_{i,2sp}); \tag{169}$$

$$\psi_{2p_z}(\zeta) = \sum_{i=1}^{L} d_{i,2p} g_{2p_z}(\alpha_{i,2sp}). \tag{170}$$

Here ζ is a parameter appearing in the Slater-type orbitals ($\zeta = 1.24$ for H and $\zeta = 2.69$ as the "recommended" value for Li$1s$); the coefficients d_i and α_i are fitted parameters and g are the normalized Gaussian functions:

$$g_s(\alpha) = c_\alpha^{(s)} e^{-\alpha r^2}, \qquad \text{with} \quad c_\alpha^{(s)} = \left(\frac{2\alpha}{\pi}\right)^{\frac{3}{4}}; \tag{171}$$

$$g_p(\beta) = c_\beta^{(p)} r \cos\theta e^{-\beta r^2}, \qquad \text{with} \quad c_\beta^{(p)} = \left(\frac{128\,\beta^5}{\pi^3}\right)^{\frac{1}{4}}. \tag{172}$$

The fitted Gaussian exponents and the corresponding coefficients d_i depend on the parameter ζ in the Slater orbital, also called "scaling factor", which is different for each atomic shell (e.g for $2s$ and $2p$ states of Li the recommended value is $\zeta = 0.75$). The exponents for $\zeta = 1$ are given in Table 3.7 of Szabo and Ostlund [337]; for $\zeta \neq 1$ they scale as $\alpha(\zeta) = \alpha(1) \cdot \zeta^2$, whereby the coefficients d are the same for each type of states in different atoms, – e.g $1s$ (H) and $1s$ (Li), – although α's are different. The parameters used by us are compiled in Table 12.

Table 12: Parameters of STO-3G orbitals for H and Li atoms used in the calculations.

H		Li				
α_{1s} (ζ = 1.24)	d_{1s}	α_{1s} (ζ = 2.69)	d_{1s}	$\alpha_{2s,p}$ (ζ = 0.75)	d_{2s}	d_{2p}
3.425250914	0.1543289673	16.11957475	0.1543289673	0.6362897469	−0.09996722919	0.1559162750
0.6239137298	0.5353281423	2.936200663	0.5353281423	0.1478600533	0.3995128261	0.6076837186
0.1688554040	0.4446345422	0.7946504870	0.4446345422	0.04808867840	0.7001154689	0.3919573931

a.2 One-electron matrix elements

We shall use spherical coordinates with the origin at the Li atom, as shown in Figure 24. From now on, the Li atom will be denoted "B" and the H atom will be "A", and, according to the previous section, we shall consider the matrix elements between the following three functions:

$$|A\rangle = c_\alpha^{(s)} e^{-\alpha(\vec{r}-\vec{d})^2}\,; \qquad |B\rangle_{1s\ or\ 2s} = c_\beta^{(s)} e^{-\beta r^2}\,; \qquad |B\rangle_{2p} = c_\beta^{(p)} r\cos\theta e^{-\beta r^2}\,. \tag{173}$$

Nuclear Potential Energy Matrix Elements

To calculate the nuclear potential energy matrix elements, one needs to calculate the following integrals:

$$I_{ab}^{(s(1))} = \langle A|\frac{1}{r}|B\rangle_s = c_\alpha^{(s)} c_\beta^{(s)} \int e^{-\alpha(\vec{r}-\vec{d})^2-\beta r^2}\frac{1}{r}\,d\vec{r}\,; \tag{174}$$

$$I_{ab}^{(s(2))} = \langle A|\frac{1}{|\vec{r}-\vec{d}|}|B\rangle_s = c_\alpha^{(s)} c_\beta^{(s)} \int e^{-\alpha(\vec{r}-\vec{d})^2-\beta r^2}\frac{1}{|\vec{r}-\vec{d}|}\,d\vec{r}\,. \tag{175}$$

These integrals are the same as for the H_2 molecule, so we can use the result of Equation (A33) from Szabo and Ostlund [337]:

$$I_{ab}^{(s(1))} = c_\alpha^{(s)} c_\beta^{(s)} \frac{2\pi}{\alpha+\beta}\exp\left\{\left(-\frac{\alpha\beta}{\alpha+\beta}d^2\right)\right\}F_o\left(\frac{\beta^2}{\alpha+\beta}d^2\right)\,; \tag{176}$$

$$I_{ab}^{(s(2))} = c_\alpha^{(s)} c_\beta^{(s)} \frac{2\pi}{\alpha+\beta}\exp\left\{\left(-\frac{\alpha\beta}{\alpha+\beta}d^2\right)\right\}F_o\left(\frac{\alpha^2}{\alpha+\beta}d^2\right)\,, \tag{177}$$

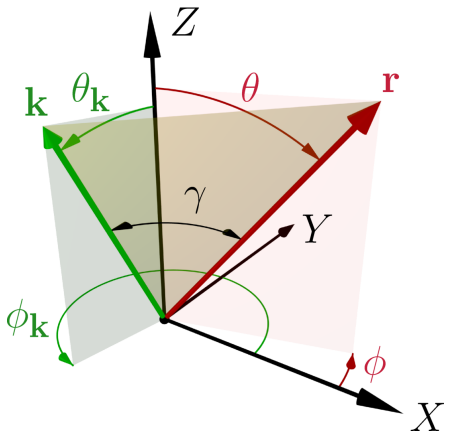


Figure 25: Coordinate system and angles used in the calculation of integrals. See text for details.

where $F_o(x)$ is expressed via the error function, $F_o(x) = \sqrt{\frac{\pi}{4x}}\text{erf}(\sqrt{x})$. The matrix elements involving the p-orbital are:

$$\begin{aligned} I_{ab}^{(p(1))} &= \langle A|\frac{1}{r}|B\rangle_p = c_\beta^{(p)} c_\alpha^{(s)} \int e^{-\alpha(\vec{r}-\vec{d})^2} r\cos\theta e^{-\beta r^2}\frac{1}{r}\,d\vec{r} \\ &= c_\beta^{(p)} c_\alpha^{(s)} \int e^{-\alpha(\vec{r}-\vec{d})^2}\cos\theta e^{-\beta r^2}\,d\vec{r} \\ &= c_\beta^{(p)} c_\alpha^{(s)} \int f_1(\vec{r}) f_2(\vec{r}) d\vec{r}\,, \end{aligned} \tag{178}$$

where $f_1(\vec{r}) = e^{-\alpha(\vec{r}-\vec{d})^2}$ and $f_2(\vec{r}) = \cos\theta e^{-\beta r^2}$. It is convenient to use the Fourier transform of these functions:

$$\begin{aligned} f_1(\vec{k}) &= \int f_1(\vec{r}) e^{-i\vec{k}\cdot\vec{r}} d\vec{r} \\ &= e^{-i\vec{k}\cdot\vec{d}} \int e^{-\alpha(\vec{r}-\vec{d})^2} e^{-i\vec{k}(\vec{r}-\vec{d})}\,d(\vec{r}-\vec{d}) \\ &= e^{-i\vec{k}\cdot\vec{d}} \left(\frac{\pi}{\alpha}\right)^{\frac{3}{2}} e^{-\frac{k^2}{4\alpha}}\,; \end{aligned} \tag{179}$$

$$f_2(\vec{k}) = \int \cos\theta e^{-\beta r^2} e^{-i\vec{k}\cdot\vec{r}} d\vec{r}\,. \tag{180}$$

For $f_2(\vec{k})$ we need to express $\cos\theta$ in terms of $\cos\gamma$, since $\vec{k}\cdot\vec{r} = kr\cos\gamma$. The vectors $\vec{k}$, $\vec{e_z}$ and $\vec{r}$, in general, do not lie in the same plane, so we need to consider the spherical triangle shown in the Figure 25. We can use the following formula relating the angles θ, θ_k and γ:

$$\cos\theta = \cos\theta_k\cos\gamma + \sin\theta_k\sin\gamma\cos(\phi - \phi_k) \tag{181}$$

Using (181), we obtain:

$$f_2(\vec{k}) = 2\pi \int_0^\infty r^2\,dr \int_{-1}^1 \cos\theta_k\cos\gamma e^{-\beta r^2 - ikr\cos\gamma} d\gamma$$

[notice that the integration over ϕ eliminated the second term in (181)]. The integral with respect to $\cos\gamma$ yields:

$$\int_{-1}^1 z e^{-ikrz} dz = \frac{2i}{kr}\left[\cos(kr) - \frac{\sin(kr)}{kr}\right] = -2ij_i(kr)\,,$$

where $j_i(x)$ is the spherical Bessel function. Then

$$f_2(\vec{k}) = 2\pi\cos\theta_k\left(\frac{2i}{k}\right)\int_0^\infty\left[\cos(kr)-\frac{\sin(kr)}{kr}\right]e^{-\beta r^2}r\,dr \tag{182}$$

$$= 2\pi i\cos\theta_k\frac{1}{k\beta}\left\{1-\left(\frac{2\beta}{k^2}+1\right)F_D\left(\frac{k}{2\sqrt{\beta}}\right)\right\}, \tag{183}$$

whereby

$$F_D(t) = \frac{\sqrt{\pi}}{2}e^{-t^2}\mathrm{erfi}(t)\,, \tag{184}$$

in which $\mathrm{erfi}(t) = -i\mathrm{erf}(t)$, is called the Dawson's function. Then

$$I_{ab}^{(p(1))} = c_\beta^{(p)}c_\alpha^{(s)}\int d\vec{r}\left\{\int f_1(\vec{k}_1)e^{i\vec{k}_1\vec{r}}\frac{d\vec{k}_1}{(2\pi)^3}\int f_2(\vec{k}_2)e^{i\vec{k}_2\vec{r}}\frac{d\vec{k}_2}{(2\pi)^3}\right\}$$

$$= c_\beta^{(p)}c_\alpha^{(s)}\int\frac{d\vec{k}}{(2\pi)^3}f_2(\vec{k})f_1(-\vec{k})\,. \tag{185}$$

The angular part of the integral in (185) is:

$$2\pi\int_{-1}^{1}e^{ikd\cos\theta_k}\cos\theta_k\,d\cos\theta_k = \frac{4\pi}{ikd}\left[\cos(kd)-\frac{\sin(kd)}{kd}\right],$$

and we have:

$$I_{ab}^{(p(1))}(\alpha,\beta;d) = c_\beta^{(p)}c_\alpha^{(s)}\left(\frac{\pi}{\alpha}\right)^{\frac{3}{2}}$$

$$\times\frac{1}{\pi\beta d}\int_0^\infty\left[1-\left(\frac{2\beta}{k^2}+1\right)F_D\left(\frac{k}{2\sqrt{\beta}}\right)\right]$$

$$\times\exp\left\{\left(-\frac{k^2}{4\alpha}\right)\right\}\left[\cos(kd)-\frac{\sin(kd)}{kd}\right]dk\,. \tag{186}$$

Another integral of this type, describing electrons interaction with the H atom, is:

$$I_{ab}^{(p(1))} = \langle A|\frac{1}{|\vec{r}-\vec{d}|}|B\rangle_p \tag{187}$$

$$= c_\beta^{(p)}c_\alpha^{(s)}\int r\cos\theta e^{-\beta r^2}\frac{1}{|\vec{r}-\vec{d}|}e^{-\alpha(\vec{r}-\vec{d})^2}\,d\vec{r}\,, \tag{188}$$

where $f_1(\vec{r}) = r\cos\theta e^{-\beta r^2}$ and $f_2(\vec{r}) = \dfrac{1}{|\vec{r}-\vec{d}|}e^{-\alpha(\vec{r}-\vec{d})^2}$. The Fourier transforms of these functions are:

$$\begin{aligned} f_1(\vec{k}) &= 2i\frac{\cos\theta_k}{k}2\pi\int_0^\infty r^2\left[\cos(kr) - \frac{\sin(kr)}{kr}\right]e^{-\beta r^2}\,dr \\ &= i\cos\theta_k\frac{\pi^{\frac{3}{2}}k}{2\beta^{\frac{5}{2}}}\exp\left\{\left(-\frac{k^2}{4\beta}\right)\right\}; \end{aligned} \tag{189}$$

$$\begin{aligned} f_2(\vec{k}) &= e^{-i\vec{k}\cdot\vec{d}}2\pi\int_0^\infty\int_{-1}^1 e^{-\alpha r^2 - ik\cos\gamma}r\,dr\,d\cos\gamma \\ &= \frac{4\pi}{k}e^{-i\vec{k}\cdot\vec{d}}\frac{1}{\sqrt{\alpha}}F_D\left(\frac{k}{2\sqrt{\alpha}}\right). \end{aligned} \tag{190}$$

With this,

$$\begin{aligned} I_{ab}^{(p(2))}(\alpha;\beta;d) &= c_\alpha^{(s)}c_\beta^{(p)}\frac{2\pi^{\frac{5}{2}}}{\sqrt{\alpha}\beta^{\frac{5}{2}}}i\int\frac{d\vec{k}}{(2\pi)^3}\cos\theta_k k e^{-i\vec{k}\cdot\vec{d}} \\ &\quad\times\exp\left\{\left(-\frac{k^2}{4\beta}\right)\right\}F_D\left(\frac{k}{2\sqrt{\alpha}}\right) \\ &= \frac{c_\alpha^{(s)}c_\beta^{(p)}\sqrt{\pi}}{\sqrt{\alpha}\beta^{\frac{5}{2}}d}\int_0^\infty\exp\left\{\left(-\frac{k^2}{4\beta}\right)\right\}F_D\left(\frac{k}{2\sqrt{\alpha}}\right)\left[\cos(kd) - \frac{\sin(kd)}{kd}\right]k\,dk \\ &= \frac{c_\alpha^{(s)}c_\beta^{(p)}\sqrt{\pi}}{\sqrt{\alpha}\beta^{\frac{5}{2}}d^3}\int_0^\infty(x\cos x - \sin x)e^{-b'x^2}F_D(a'x)\,dx, \end{aligned} \tag{191}$$

where $b' = \dfrac{1}{4\beta d^2}$ and $a' = \dfrac{1}{2\sqrt{\alpha}d}$, F_D is the Dawson's function (184). Note that the dimension of the normalization constants is $[c_\alpha^{(s)}] = L^{-\frac{3}{2}}$, $[c_\beta^{(p)}] = L^{-\frac{5}{2}}$, while $[\alpha] = [\beta] = L^2$, thus, overall dimension of (191) is L^{-1}, as it should be. The integral in (191) couldn't be evaluated analytically, so it has to be calculated numerically.

We still need matrix elements of r^{-1} diagonal in atomic index, which are as follows:

$$\begin{aligned} I_{aa}^{(s(1))}(\alpha,\beta) &= c_\alpha^{(s)} c_\beta^{(p)} \int \frac{1}{r} e^{-(\alpha+\beta)r^2} d\vec{r} \\ &= \frac{2\pi_\alpha^{(s)} c_\beta^{(p)}}{\alpha+\beta}; \end{aligned} \tag{192}$$

$$I_{bb}^{(s(1))}(\alpha,\beta) = I_{aa}^{(s(1))}(\alpha,\beta);$$

$$I_{bb}^{(ps(1))} = c_\alpha^{(s)} c_\beta^{(p)} \int \frac{1}{r} r\cos\theta e^{-(\alpha+\beta)r^2} d\vec{r} = 0\,; \tag{193}$$

$$\begin{aligned} I_{bb}^{(p(1))}(\alpha,\beta) &= c_\alpha^{(p)} c_\beta^{(p)} \int \frac{1}{r} r^2\cos^2\theta e^{-(\alpha+\beta)r^2} d\vec{r} \\ &= \frac{4\pi}{3} \frac{c_\alpha^{(p)} c_\beta^{(p)}}{(\alpha+\beta)^2}\,; \end{aligned} \tag{194}$$

$$\begin{aligned} I_{aa}^{(s(2))}(\alpha,\beta;d) &= c_\alpha^{(s)} c_\beta^{(s)} \int \frac{1}{r} e^{-(\alpha+\beta)(\vec{r}-\vec{d})^2} d\vec{r} \\ &= c_\alpha^{(s)} c_\beta^{(s)} \frac{2\pi^{\frac{3}{2}}}{(\alpha+\beta)^{\frac{3}{2}} d^2} \mathrm{erf}\left(\sqrt{\alpha+\beta}\, d\right)\,; \end{aligned} \tag{195}$$

$$I_{bb}^{(s(2))} = I_{aa}^{(s(2))}\,;$$

$$I_{bb}^{(ps(2))}(\alpha,\beta;d) = c_\alpha^{(s)} c_\beta^{(p)} \int \frac{1}{|\vec{r}-\vec{d}|} e^{-(\alpha+\beta)r^2} r\cos\theta\, d\vec{r}\,. \tag{196}$$

Here we use the following expansion:

$$\begin{aligned} \frac{1}{|\vec{r}-\vec{d}|} &= \frac{1}{\sqrt{r^2+d^2-2rd\cos\theta}} \\ &= \frac{1}{d}\sum_{l=0}^{\infty} P_l(\cos\theta) \times \begin{cases} x^l (x<1)\,, \\ x^{-l-1} (x>1)\,, \end{cases} \end{aligned} \tag{197}$$

where $x = \dfrac{r}{d}$; since $\cos\theta = P_1(\cos\theta)$ (P_l are the Legendre polynomials), the angular integration in (196) eliminates all the terms in the sum over l except $l = 1$. Therefore, we have:

$$\begin{aligned} I_{bb}^{(ps(2))}(\alpha,\beta;d) &= c_\alpha^{(s)} c_\beta^{(p)} \left\{ \frac{1}{d^2} \int_0^d e^{-(\alpha+\beta)r^2} r^4 \, dr \right. \\ &\qquad \left. + d \int_d^\infty e^{-(\alpha+\beta)r^2} r \, dr \right\} \times \frac{4\pi}{3} \\ &= \pi \frac{\sqrt{\pi}\,\mathrm{erf}(\sqrt{\alpha+\beta}\,d) - 2\sqrt{\alpha+\beta}\,d\,e^{-(\alpha+\beta)d^2}}{2(\alpha+\beta)^{\frac{5}{2}} d^2} c_\alpha^{(s)} c_\beta^{(p)} \,. \end{aligned} \tag{198}$$

Finally, the last integral of this type is:

$$I_{bb}^{(p(2))}(\alpha,\beta;d) = c_\alpha^{(p)} c_\beta^{(p)} \int \frac{1}{|\vec{r} - \vec{d}|} e^{-(\alpha+\beta)r^2} r^2 \cos^2\theta \, d\vec{r} \,. \tag{199}$$

Again, we use the formula (197) and the relation

$$z^2 = \frac{2}{3}\left[P_2(z) + \frac{1}{2} P_0(z) \right] . \tag{200}$$

Using (200), the angular integration in (199) yields:

$$\begin{aligned} & 2\pi \int_{-1}^{1} \cos^2\theta \sum_{l=1}^{\infty} P_l(\cos\theta) \times \left\{ \begin{array}{ll} \left(\dfrac{r}{d}\right)^l & (r < d) \\ \left(\dfrac{r}{d}\right)^{-l-1} & (r > d) \end{array} \right\} d\cos\theta \\ &= \frac{8\pi}{15} \left\{ \begin{array}{l} \left(\dfrac{r}{d}\right)^2 \\ \left(\dfrac{d}{r}\right)^3 \end{array} \right\} + \frac{4\pi}{3} \left\{ \begin{array}{l} 1 \\ \dfrac{d}{r} \end{array} \right\} . \end{aligned}$$

The result is:

$$\begin{aligned} I_{bb}^{(p(2))}(\alpha,\beta;d) = \pi \Bigg\{ & \frac{\left[1 + (\alpha+\beta)d^2\right] \sqrt{\pi}\,\mathrm{erf}(\sqrt{\alpha+\beta}\,d)}{2(\alpha+\beta)^{\frac{7}{2}} d^3} \\ & - \frac{\left[1 + (\alpha+\beta)d^2\right]}{(\alpha+\beta)^3 d^2} e^{-(\alpha+\beta)d^2} \Bigg\} . \end{aligned} \tag{201}$$

a.3 Kinetic Energy Matrix Elements

The calculation of the kinetic energy matrix elements involves the following integrals:

$$K_{ab}^{(s)} = \langle A|-\nabla^2|B\rangle_s = -c_\alpha^s c_\beta^s \int e^{-\alpha(\vec{r}-\vec{d})^2}\nabla^2 e^{-\beta r^2} d\vec{r} \tag{202}$$

$$= -c_\alpha^s c_\beta^s \int \frac{d\vec{k}}{(2\pi)^3} f_1\left(\vec{k}\right) f_2\left(-\vec{k}\right), \tag{203}$$

where $f_2(\vec{r}) = e^{-\alpha(\vec{r}-\vec{d})^2}$ and $f_1(\vec{r}) = \nabla^2 e^{-\beta r^2}$. Fourier transforms of these functions are:

$$f_1\left(\vec{k}\right) = \int d\vec{r}\, e^{-i\vec{k}\cdot\vec{r}}\nabla^2 e^{-\beta r^2} = -k^2\left(\frac{\pi}{\alpha}\right)^{\frac{3}{2}} \exp\left\{\left(-\frac{k^2}{4\beta}\right)\right\}. \tag{204}$$

$$f_2\left(\vec{k}\right) = e^{-i\vec{k}\cdot\vec{d}}\left(\frac{\pi}{\alpha}\right)^{\frac{3}{2}} \exp\left\{\left(-\frac{k^2}{4\alpha}\right)\right\}; \tag{205}$$

Then

$$K_{ab}^{(s)} = c_\alpha^s c_\beta^s \left(\frac{\pi^2}{\alpha\beta}\right)^{\frac{3}{2}} \int \frac{d\vec{k}}{(2\pi)^3} e^{-i\vec{k}\cdot\vec{d}} \exp\left\{\left(-\frac{\alpha+\beta}{4\alpha\beta}k^2\right)\right\}k^2$$

$$= c_\alpha^s c_\beta^s \frac{\pi}{2(\alpha\beta)^{\frac{3}{2}}} \int_0^\infty \frac{\sin kd}{kd} \exp\left\{\left(-\frac{\alpha+\beta}{4\alpha\beta}k^2\right)\right\}k^4 dk.$$

Denoting $x = kd$, we have the following integral, $\int_0^\infty \sin x e^{-bx^2} x^3\, dx$, where $b = \frac{\alpha+\beta}{4\alpha\beta d^2}$. The result of the integration reads:

$$K_{ab}^{(s)} = c_\alpha^s c_\beta^s \frac{4\pi^{\frac{3}{2}}(\alpha\beta)^2 d^2}{(\alpha+\beta)^{\frac{7}{2}}}\left[\frac{3(\alpha+\beta)}{2\alpha\beta d^2} - 1\right] \exp\left\{\left(-\frac{\alpha+\beta}{4\alpha\beta}d^2\right)\right\}. \tag{206}$$

The similar integral involving the s and p states:

$$\begin{aligned} K_{ab}^{(p)} &= -c_\alpha^s c_\beta^p \int r\cos\theta e^{-\beta r^2}\,\nabla^2 e^{-\alpha(\vec{r}-\vec{d})^2}\,d\vec{r} \\ &= -c_\alpha^s c_\beta^p \int \frac{d\vec{k}}{(2\pi)^3} f_1(\vec{k}) f_2(-\vec{k})\,, \qquad \text{with} \end{aligned} \tag{207}$$

$$\begin{aligned} f_2(\vec{k}) &= e^{-i\vec{k}\cdot\vec{d}} \int d(\vec{r}-\vec{d}) e^{-i\vec{k}\cdot(\vec{r}-\vec{d})}\,\nabla^2 e^{-\alpha(\vec{r}-\vec{d})^2} \\ &= e^{-i\vec{k}\cdot\vec{d}} k^2 \left(\frac{\pi}{\alpha}\right)^{\frac{3}{2}} \exp\left\{\left(-\frac{k^2}{4\alpha}\right)\right\} \qquad \text{and} \end{aligned} \tag{208}$$

$$\begin{aligned} f_1(\vec{k}) &= i\frac{4\pi}{k}\cos\theta_k \int_0^\infty \left[\cos(kr) - \frac{\sin(kr)}{kr}\right] e^{-\beta r^2} r^2\,dr \\ &= i\cos\theta_k \frac{\pi^{\frac{3}{2}} k}{2\beta^{\frac{5}{2}}} \exp\left\{\left(-\frac{k^2}{4\beta}\right)\right\}. \end{aligned} \tag{209}$$

Using (208) and (209),

$$\begin{aligned} K_{ab}^{(p)} &= c_\alpha^s c_\beta^p i \frac{\pi^3}{2\alpha^{\frac{3}{2}}\beta^{\frac{5}{2}}} \int \cos\theta_k e^{i(\vec{k}\cdot\vec{d})} k^3 \exp\left\{\left(-\frac{\alpha+\beta}{4\alpha\beta}k^2\right)\right\} \frac{d\vec{k}}{(2\pi)^3} \\ &= c_\alpha^s c_\beta^p \frac{\pi^2}{2\alpha^{\frac{3}{2}}\beta^{\frac{5}{2}} d^6} \int_0^\infty \left(\cos x - \frac{\sin x}{x}\right) x^4 e^{-bx^2} dx\,. \end{aligned}$$

The integral is calculated with the help of *Mathematica*, with the result:

$$\begin{aligned} K_{ab}^{(p)}(\alpha,\beta;d) = c_\alpha^s c_\beta^p \frac{\pi^2}{64\alpha^{\frac{3}{2}}\beta^{\frac{5}{2}} d^6} b^5 \times \Big\{ & 1 + 4b(3b-4) \\ & - [1 + 6b(4b^2+6b-3)] \frac{1}{\sqrt{b}} F_D\left(\frac{1}{2\sqrt{b}}\right)\Big\}, \end{aligned} \tag{210}$$

where $b = \dfrac{\alpha+\beta}{4\alpha\beta d^2}$ and F_D is the Dawson's function (184). The matrix elements diagonal in atomic index are as follows:

$$\begin{aligned}
K_{aa}^{(s)}(\alpha,\beta) &= c_\alpha^{(s)} c_\beta^{(s)} \left(\frac{\pi^2}{\alpha\beta}\right) \frac{1}{2\pi^2} \int_0^\infty \exp\left\{\left(-\frac{\alpha+\beta}{4\alpha\beta}k^2\right) k^4\, dk\right\} \\
&= c_\alpha^{(s)} c_\beta^{(s)} \frac{3\pi^{\frac{3}{2}}}{2} \frac{\alpha\beta}{(\alpha+\beta)^{\frac{5}{2}}}\,; \qquad (211) \\
K_{bb}^{(s)}(\alpha,\beta) &= K_{aa}^{(s)}(\alpha,\beta)\,; \\
K_{bb}^{(sp)}(\alpha,\beta) &= 0 \qquad \text{by symmetry}\,; \\
K_{bb}^{(p)}(\alpha,\beta) &= \langle B|_p - \nabla^2 |B\rangle_p \\
&= -c_\alpha^{(p)} c_\beta^{(p)} \int \frac{d\vec{k}}{(2\pi)^3} \exp\left\{\left(-\frac{\alpha+\beta}{4\alpha\beta}\right) k^2\right\} \left(-k^2 \cos^2\theta_k \frac{\pi^3 k^2}{2(\alpha\beta)^{\frac{5}{2}}}\right) \\
&= c_\alpha^{(p)} c_\beta^{(p)} \frac{5\pi^{\frac{3}{2}}\alpha\beta}{2(\alpha+\beta)^{\frac{7}{2}}}\,. \qquad (212)
\end{aligned}$$

Summary of one-electron Hamiltonian (for zero external field):

The one-electron Hamiltonian in the absence of external electric field is as follows:

$$H_1 = -\nabla^2 - \frac{3}{r} - \frac{1}{|\vec{r}-\vec{d}|}\,.$$

For convenience, the necessary integrals are presented in Table 13, and Table 14 indicates the reference of the corresponding equation.

Table 13: Matrix elements of one-electron interactions

	$\lvert A_{1s}\rangle$	$\lvert B_{1s}\rangle$	$\lvert B_{2s}\rangle$	$\lvert B_{2p}\rangle$
$\langle A_{1s}\rvert$	$K_{aa}^{(s)}(\alpha_{i,1s},\alpha_{j,1s})$ $-I_{aa}^{(s(1))}(\alpha_{i,1s},\alpha_{j,1s})$ $-3I_{aa}^{(s(2))}(\alpha_{i,1s},\alpha_{j,1s};d)$	$K_{ab}^{(s)}(\alpha_{i,1s},\alpha'_{j,1s};d)$ $-3I_{ab}^{(s(1))}(\alpha_{i,1s},\alpha'_{j,1s};d)$ $-I_{ab}^{(s(2))}(\alpha_{i,1s},\alpha'_{j,1s};d)$	$K_{ab}^{(s)}(\alpha_{i,1s},\alpha'_{j,2sp};d)$ $-3I_{ab}^{(s(1))}(\alpha_{i,1s},\alpha'_{j,2sp};d)$ $-I_{ab}^{(s(2))}(\alpha_{i,1s},\alpha'_{j,2sp};d)$	$K_{ab}^{(p)}(\alpha_{i,1s},\alpha'_{j,1s};d)$ $-3I_{ab}^{(p(1))}(\alpha_{i,1s},\alpha'_{j,2sp};d)$ $-I_{ab}^{(p(2))}(\alpha_{i,1s},\alpha'_{j,2sp};d)$
$\langle B_{1s}\rvert$		$K_{bb}^{(s)}(\alpha'_{i,1s},\alpha'_{j,1s})$ $-3I_{bb}^{(s(1))}(\alpha'_{i,1s},\alpha'_{j,1s})$ $-I_{bb}^{(s(2))}(\alpha'_{i,1s},\alpha'_{j,1s};d)$	$K_{bb}^{(s)}(\alpha'_{i,1s},\alpha_{j,2sp})$ $-3I_{bb}^{(s(1))}(\alpha'_{i,1s},\alpha_{j,2sp})$ $-I_{bb}^{(s(2))}(\alpha'_{i,1s},\alpha_{j,2sp};d)$	$-I_{bb}^{(ps(2))}(\alpha'_{i,1s},\alpha_{j,2sp};d)$
$\langle B_{2s}\rvert$			$K_{bb}^{(s)}(\alpha_{i,2sp},\alpha_{j,2sp})$ $-3I_{bb}^{(s(1))}(\alpha_{i,2sp},\alpha_{j,2sp})$ $-I_{bb}^{(s(2))}(\alpha_{i,2sp},\alpha_{j,2sp};d)$	$-I_{bb}^{(ps(2))}(\alpha_{i,2sp},\alpha_{j,2sp};d)$
$\langle B_{2p}\rvert$				$K_{bb}^{(p)}(\alpha_{i,2sp},\alpha_{j,2sp})$ $-3I_{bb}^{(p(1))}(\alpha_{i,2sp},\alpha_{j,2sp})$ $-I_{bb}^{(p(2))}(\alpha_{i,2sp},\alpha_{j,2sp};d)$

Table 14: Equations specifying matrix elements of Table 13.

Matrix elements	Eq.	Matrix elements	Eq.
$K_{aa}^{(s)}(\alpha,\beta)$	(211)	$I_{aa}^{(s(1))}(\alpha,\beta)$	(192)
$K_{ab}^{(s)}(\alpha,\beta;d)$	(206)	$I_{aa}^{(s(2))}(\alpha,\beta;d)$	(195)
$K_{ab}^{(p)}(\alpha,\beta;d)$	(210)	$I_{ab}^{(s(1))}(\alpha,\beta;d)$	(176)
$K_{bb}^{(s)}=K_{aa}^{(s)}$		$I_{ab}^{(s(2))}(\alpha,\beta;d)$	(177)
$K_{bb}^{(p)}(\alpha,\beta;d)$	(212)	$I_{ab}^{(p(1))}(\alpha,\beta;d)$	(186)
$I_{ab}^{(p(2))}(\alpha,\beta;d)$	(191)	$I_{bb}^{(s(1))}=I_{aa}^{(s(1))}$	
$I_{bb}^{(s(2))}=I_{aa}^{(s(2))}$		$I_{bb}^{(ps(2))}(\alpha,\beta;d)$	(198)
$I_{bb}^{(p(1))}(\alpha,\beta)$	(194)	$I_{bb}^{(p(2))}(\alpha,\beta;d)$	(201)

a.4 Matrix elements of the interaction with external electric field

We shall consider the field parallel to the z axis, so the interaction Hamiltonian reads:

$$H_S=-\mathbb{E}r\cos\theta\,.$$

We shall keep the same notation as for the kinetic energy matrix elements just changing $K \to J$. First, we have:

$$J_{aa} = \mathbb{E}d; \qquad J_{bb}^{(pp)} = J_{bb}^{(ss)} = 0\,, \tag{213}$$

because the diagonal matrix elements for any atom vanish for non-degenerate atomic states and J_{aa} is compensated by the energy of the proton at point $\vec{d}$ (see Fig. 24). For the matrix element between the s and p-orbitals of the Li atom we have:

$$\begin{aligned} J_{bb}^{(ps)} = J_{bb}^{(sp)} &= -c_\alpha^{(s)} c_\beta^{(p)} \mathbb{E} \int r^2 \cos^2\theta e^{-(\alpha+\beta)r^2}\, d\vec{r} \\ &= c_\alpha^{(s)} c_\beta^{(p)} \mathscr{E} \frac{\pi^{\frac{3}{2}}}{2(\alpha+\beta)^{\frac{5}{2}}}\,. \end{aligned} \tag{214}$$

The matrix elements $J_{ab}^{(s)}$ are the same as for H_2:

$$J_{ab}^{(s)}(\alpha,\beta;d) = -c_\alpha^{(s)} c_\beta^{(s)} \int \left(\vec{\mathscr{E}}\cdot\vec{r}\right) e^{-\alpha(\vec{r}-\vec{d})^2-\beta r^2} d\vec{r}\,. \tag{215}$$

We use the transformation:

$$e^{-\alpha(\vec{r}-\vec{d})^2-\beta r^2} = \exp\left(-\frac{\alpha\beta}{\alpha+\beta}\,d^2\right) e^{-p(\vec{r}-\vec{R}_P)}\,, \tag{216}$$

where $P = \alpha+\beta$ and $\vec{R}_P = \frac{1}{p}\left(\alpha\vec{R}_A + \beta\vec{R}_B\right) = \frac{\alpha}{p}\vec{d}$. Then

$$\begin{aligned} J_{ab}^{(s)} &= -c_\alpha^{(s)} c_\beta^{(s)} \exp\left\{\left(-\frac{\alpha\beta}{\alpha+\beta}d^2\right)\right\} \int [\vec{\mathscr{E}}\cdot(\vec{r}-\vec{R}_P) \\ &\qquad + \vec{\mathscr{E}}\cdot\vec{R}_P] e^{-p(\vec{r}-\vec{R}_P)^2} d\vec{r} \\ &= -c_\alpha^{(s)} c_\beta^{(s)} \exp\left\{\left(-\frac{\alpha\beta}{\alpha+\beta}d^2\right)\right\} (J_1 + J_2)\,, \quad \text{where} \\ J_1 &= \int (\vec{\mathscr{E}}\cdot\vec{r}') e^{-p(\vec{r}')^2} d\vec{r}' = 0\,, \quad \text{and} \\ J_2 &= (\vec{\mathscr{E}}\cdot\vec{R}_P)\int e^{-pr^2} d\vec{r} = \left(\frac{\pi}{\alpha+\beta}\right)^{\frac{3}{2}} \frac{\alpha}{\alpha+\beta}\mathscr{E}d\,. \end{aligned}$$

Thus, we have:

$$J_{ab}^{(s)}(\alpha,\beta;d) = -c_\alpha^{(s)} c_\beta^{(s)} \exp\left\{\left(-\frac{\alpha\beta}{\alpha+\beta}d^2\right)\right\} \frac{\pi^{\frac{3}{2}}\alpha}{(\alpha+\beta)^{\frac{5}{2}}}\mathscr{E}d\,. \tag{217}$$

Obviously, $J_{ab}^{(s)} = J_{ba}^{(s)}$. Now we shall calculate

$$\begin{aligned} J_{ab}^{(p)} &= -c_\alpha^{(s)} c_\beta^{(p)} \mathscr{E} \int r^2 \cos^2\theta e^{-\beta r^2} e^{-\alpha(\vec{r}-\vec{d})^2} d\vec{r} \\ &= -c_\alpha^{(s)} c_\beta^{(p)} \mathscr{E} \int f_1(\vec{r}) f_2(\vec{r}) d\vec{r}\,, \end{aligned} \tag{218}$$

where

$$f_1(\vec{r}) = r^2 \cos^2\theta e^{-\beta r^2} \qquad \text{and} \qquad f_2(\vec{r}) = e^{-\alpha(\vec{r}-\vec{d})^2}\,.$$

The Fourier transform of $f_1(\vec{r})$ is:

$$\begin{aligned} f_1(\vec{k}) &= \int f_1(\vec{r}) e^{-i\vec{k}\cdot\vec{r}} d\vec{r} \\ &= \int_0^\infty dr \int_{-1}^1 d\cos\gamma \int_0^{2\pi} d\phi \left[\cos\gamma\cos\theta_k + \sin\gamma\sin\theta_k \sin(\phi-\phi_k)\right]^2 \\ &\quad \times e^{-\beta r^2 - ikr\cos\gamma}\,, \end{aligned}$$

where we made use of (181). The term linear in $\sin(\phi-\phi_k)$ vanishes after integration over ϕ, while $\int_0^{2\pi} \sin^2(\phi-\phi_k) d\phi = \pi$. Therefore,

$$\begin{aligned} &\int_{-1}^1 d\cos\gamma \left[\cos^2\theta_k \cos^2\gamma + \frac{1}{2}(1-\cos^2\gamma)\sin^2\theta_k\right] e^{-ikr\cos\gamma} \\ &\quad = \frac{2}{(kr)^3}\Big\{(3\cos^2\theta k - 1)kr\cos kr \\ &\qquad + \left[(1-3cos^2\theta_k) + (kr)^2 cos^2(\theta_k)\right]\sin kr\Big\} \quad \equiv g(\vec{k}, r) \end{aligned}$$

and

$$\begin{aligned} f_1(\vec{k}) &= 2\pi \int_0^\infty r^4 e^{-\beta r^2} g(\vec{k}, r) dr \\ &= \left(\frac{k^2}{2\beta}\right)\frac{1}{k^2}\left[\frac{2\beta}{k^2} - \cos^2\theta_k\right]\left(\frac{\pi}{\beta}\right)^{\frac{3}{2}} \exp\left\{\left(-\frac{k^2}{4\beta}\right)\right\}; \end{aligned} \tag{219}$$

$$f_2(\vec{k}) = \int e^{-\alpha r^2 - i\vec{k}\cdot(\vec{r}+\vec{d})} d\vec{r} = e^{i\vec{k}\cdot\vec{d}}\left(\frac{\pi}{\alpha}\right)^{\frac{3}{2}} \exp\left\{\left(\frac{-k^2}{4\alpha}\right)\right\}, \tag{220}$$

The integral (218) is given by

$$J_{ab}^{(p)} = -c_\alpha^{(s)} c_\beta^{(p)} \mathscr{E} \int f_1(\vec{k}) f_2(-\vec{k}) \frac{d\vec{k}}{(2\pi)^3}\,. \tag{221}$$

In (221), the following angular integrals come about:

$$\int_0^{2\pi} d\phi_k \int_{-1}^{1} e^{ikd\cos\theta_k} d\cos\theta_k = 4\pi \frac{\sin kd}{kd} = 4\pi j_0(kd) \tag{222}$$

and

$$\begin{aligned}\int_0^{2\pi} d\phi_k \int_{-1}^{1} \cos^2\theta_k e^{ikd\cos\theta_k} d\cos\theta_k &= 4\pi \frac{2kd\cos kd + [(kd)^2 - 2]\sin kd}{(kd)^3} \\ &= -4\pi \left[j_2(kd) + \frac{j_1(kd)}{kd} \right] \equiv -4\pi Z(kd)\,. \end{aligned} \tag{223}$$

In (222) and (223), $j_l(c)$ are the spherical Bessel functions and $Z(x)$ is just a short-hand notation. With this, Eq. (221) reduces to:

$$\begin{aligned} J_{ab}^{(p)}(\alpha,\beta;d) &= -c_\alpha^{(s)} c_\beta^{(p)} \mathscr{E} \frac{\pi d}{\beta^2} \left(\frac{1}{4\alpha\beta d^4} \right)^{\frac{3}{2}} \\ &\quad \times \int_0^\infty [2\beta d^2 j_0(x) + x^2 Z(x)] e^{-bx^2} dx\,, \end{aligned} \tag{224}$$

where $b = \dfrac{\alpha+\beta}{4\alpha\beta d^2}$. The calculation of the integral in (224) yields:

$$\begin{aligned} &\sqrt{\pi} \frac{1 + 4\left(\dfrac{\alpha+\beta}{4\alpha\beta d^2}\right)}{4\left(\dfrac{\alpha+\beta}{4\alpha\beta d^2}\right)^{\frac{3}{2}}} \exp\left\{ \left(-\frac{1}{4\left(\dfrac{\alpha+\beta}{4\alpha\beta d^2}\right)} \right) \right\} \\ &\quad + \frac{2\beta d^2 - 2}{2} \pi \,\mathrm{erf}\left(\frac{1}{2\sqrt{\dfrac{\alpha+\beta}{4\alpha\beta d^2}}} \right) \\ &= \sqrt{\pi} \frac{1+4b}{4b^{\frac{3}{2}}} \exp\left\{ \left(-\frac{1}{4b} \right) \right\} + \frac{a-2}{2} \pi \,\mathrm{erf}\left(\frac{1}{2\sqrt{b}} \right), \end{aligned}$$

where $a = 2\beta d^2$.

Summary of the perturbation operator

The matrix elements of the perturbation operator due to external electric field, H_S, are summarized in Table 15 and the corresponding equations are referred to in Table 16. Notice that the proton energy ($-Ed$)

Table 15: Matrix elements for the perturbation operator

	$\|A_{1s}\rangle$	$\|B_{1s}\rangle$	$\|B_{2s}\rangle$	$\|B_{2p_z}\rangle$
$\langle A_{1s}\|$	0	$J^{(s)}_{ab}(\alpha_{i,1s}, \alpha'_{j,1s}; d)$	$J^{(s)}_{ab}(\alpha_{i,1s}, \alpha_{j,2sp}; d)$	$J^{(p)}_{ab}(\alpha_{i,1s}, \alpha_{j,2sp}; d)$
$\langle B_{1s}\|$	$J^{(s)}_{ab}(\alpha_{i,1s}, \alpha'_{j,1s}; d)$	0	0	$J^{(ps)}_{bb}(\alpha'_{i,1s}, \alpha_{j,2sp}; d)$
$\langle B_{2s}\|$	$J^{(s)}_{ab}(\alpha_{i,1s}, \alpha_{j,2sp}; d)$	0	0	$J^{(ps)}_{bb}(\alpha_{i,2sp}, \alpha_{j,2sp}; d)$
$\langle B_{2p_z}\|$	$J^{(p)}_{ab}(\alpha_{i,1s}, \alpha_{j,2sp}; d)$	$J^{(ps)}_{bb}(\alpha'_{i,1s}, \alpha_{j,2sp}; d)$	$J^{(ps)}_{bb}(\alpha_{i,2sp}, \alpha_{j,2sp}; d)$	0

Table 16: Equations specifying matrix elements of Table 15

Matrix elements	Eq.	Matrix elements	Eq.	Matrix elements	Eq.
$J^{s}_{ab}(\alpha, \beta; d)$	(217)	$J^{p}_{ab}(\alpha, \beta; d)$	(224)	$J^{ps}_{bb}(\alpha, \beta; d)$	(214)

has been added to compensate J_{aa} and it is necessary to substitute $\alpha_{i,1s}$, $\alpha_{i,2sp}$ for α and β, respectively and $\alpha'_{i,1s}$ is for Li in the appropriate relations.

a.5 Two-electron matrix elements

Matrix elements of the electron–electron interaction, $r_{12}^{-1} = |\vec{r}_1 - \vec{r}_2|^{-1}$, in the "chemist's notation" are written in round brackets [337]:

$$(ij|r_{12}^{-1}|kl) = \int d\vec{r}_1 d\vec{r}_2 \psi_i^*(\vec{r}_1)\psi_j(\vec{r}_1) r_{12}^{-1} \psi_k^*(\vec{r}_2)\psi_l(\vec{r}_2)\,,$$

which is different from the physicist's notation for the same thing, $\langle ik|r_{12}^{-1}|jl\rangle$, which uses angular brackets and different order of orbitals. Here ψ_i denotes a molecular spatial orbital constructed as a linear combination of atomic orbitals, i.e. in our case

$$|\psi\rangle = c_1|A\rangle + c_2|B\rangle_{1s} + c_3|B\rangle_{2s} + c_4|B\rangle_{2p}\,. \tag{225}$$

The HF energy includes the so called Coulomb and exchange integrals:

$$J_{ij} = (ii|r_{12}^{-1}|jj) = \langle ij|r_{12}^{-1}|ij\rangle \qquad \text{(Coulomb)}\,; \tag{226}$$

$$K_{ij} = (ij|r_{12}^{-1}|ji) = \langle ij|r_{12}^{-1}|ji\rangle \qquad \text{(exchange)}\,. \tag{227}$$

Since $|i\rangle$ and $|j\rangle$ are linear combinations of $g_{1s}(\vec{r} - \vec{d})$, $g_{1s}(\vec{r})$ and $g_{2p}(\vec{r})$ functions with different coefficients in the exponent, several kinds of integrals occur in (226) and (227), namely: (*i*) four kinds

of one-center integrals; (*ii*) four kinds of two-center integrals. We proceed by elaborating on the first type (one-center) integrals, (*i*).

$$
\begin{aligned}
D_{aa}^{(ss)}(\alpha,\beta,\gamma,\delta) &= \int d\vec{r}_1 d\vec{r}_2 g_{1s}(\alpha,\vec{r}_1) g_{1s}(\beta,\vec{r}_1) r_{12}^{-1} g_{1s}(\gamma,\vec{r}_2) g_{1s}(\delta,\vec{r}_2) \\
&= c_\alpha^{(s)} c_\beta^{(s)} c_\gamma^{(s)} c_\delta^{(s)} \int d\vec{r}_1 d\vec{r}_2 \, e^{-(\alpha+\beta)r_1^2} \left(r_{12}^{-1}\right) e^{-(\gamma+\delta)r_2^2} \\
&= c_\alpha^{(s)} c_\beta^{(s)} c_\gamma^{(s)} c_\delta^{(s)} \int \frac{d\vec{k}}{(2\pi)^3} \Big[\Big(\frac{\pi}{\alpha+\beta} \Big)^{\frac{3}{2}} \exp\Big\{ \Big(-\frac{k^2}{4(\alpha+\beta)} \Big) \Big\} \Big] \\
&\quad \times \Big[\frac{4\pi}{k^2} \Big] \Big[\Big(\frac{\pi}{\gamma+\delta} \Big)^{\frac{3}{2}} \exp\Big\{ \Big(-\frac{k^2}{4(\gamma+\delta)} \Big) \Big\} \Big] \\
&= c_\alpha^{(s)} c_\beta^{(s)} c_\gamma^{(s)} c_\delta^{(s)} \frac{2\pi^2}{[(\alpha+\beta)(\gamma+\delta)]^{\frac{3}{2}}} \frac{\sqrt{\pi}}{2\Big[\frac{1}{4(\alpha+\beta)} + \frac{1}{4(\gamma+\delta)} \Big]^{\frac{1}{2}}} ; \\
D_{aa}^{(ss)}(\alpha,\beta,\gamma,\delta) &= 2\pi^{\frac{5}{2}} \frac{c_\alpha^{(s)} c_\beta^{(s)} c_\gamma^{(s)} c_\delta^{(s)}}{(\alpha+\beta)(\gamma+\delta)(\alpha+\beta+\gamma+\delta)^{\frac{1}{2}}} .
\end{aligned} \tag{228}
$$

The same expression applies to $D_{bb}^{(ss)}(\alpha,\beta,\gamma,\delta)$.

$$
\begin{aligned}
D_{bb}^{(pp)}(\alpha,\beta,\gamma,\delta) &= \int d\vec{r}_1 d\vec{r}_2 g_{2p}(\alpha,\vec{r}_1) g_{2p}(\beta,\vec{r}_1) r_{12}^{-1} g_{2p}(\gamma,\vec{r}_2) g_{2p}(\delta,\vec{r}_2) \\
&= c_\alpha^{(p)} c_\beta^{(p)} c_\gamma^{(p)} c_\delta^{(p)} \int d\vec{r}_1 d\vec{r}_2 \cos^2\theta_1 r_1^2 e^{-(\alpha+\beta)r_1^2} \left(r_{12}^{-1}\right) \cos^2\theta_2 r_2^2 e^{-(\gamma+\delta)r_2^2} \\
&= c_\alpha^{(p)} c_\beta^{(p)} c_\gamma^{(p)} c_\delta^{(p)} \int \frac{d\vec{k}}{(2\pi)^3} \left\{ \left[\frac{k}{2(\alpha+\beta)} \right]^2 \left[\frac{2(\alpha+\beta)}{k^2} - \cos^2\theta_k \right] \right. \\
&\quad \left. \times \left(\frac{\pi}{\alpha+\beta} \right)^{\frac{3}{2}} \exp\left\{ \left(-\frac{k^2}{4(\alpha+\beta)} \right) \right\} \right\} \times \left[\frac{4\pi}{k^2} \right] \\
&\quad \times \left\{ \left[\frac{k}{2(\gamma+\delta)} \right]^2 \left[\frac{2(\gamma+\delta)}{k^2} - \cos^2\theta_k \right] \left(\frac{\pi}{\gamma+\delta} \right)^{\frac{3}{2}} \exp\left\{ \left(-\frac{k^2}{4(\gamma+\delta)} \right) \right\} \right\} ,
\end{aligned}
$$

where we used the Fourier transform result (219). The calculation of such integrals finally yields:

$$
\begin{aligned}
D_{bb}^{(pp)}(\alpha,\beta,\gamma,\delta) &= c_\alpha^{(p)} c_\beta^{(p)} c_\gamma^{(p)} c_\delta^{(p)} \frac{\pi^{\frac{5}{2}}}{[(\alpha+\beta)(\gamma+\delta)]^2} \frac{1}{(\alpha+\beta+\gamma+\delta)^{\frac{1}{2}}} \\
&\quad \times \left\{ -\frac{1}{12} + \frac{6}{5} \frac{(\alpha+\beta)(\gamma+\delta)}{(\alpha+\beta+\gamma+\delta)^2} \right\} .
\end{aligned} \tag{229}
$$

Table 17: Matrix elements of two-electron interactions

$\lvert A_{1s}\rangle$	$\lvert B_{1s}\rangle$	$\lvert B_{2s}\rangle$	$\lvert B_{2p}\rangle$
$\langle A_{1s}\rvert\, D_{aa}^{(ss)}(\alpha_{i,1s},\alpha_{j,1s},\alpha_{k,1s},\alpha_{l,1s})$	$D_{ab}^{(ssE)}(\alpha_{i,1s},\alpha'_{j,1s},\alpha_{k,1s},\alpha'_{l,1s};d)$, $D_{ab}^{(ssC)}(\alpha_{i,1s},\alpha_{j,1s},\alpha'_{k,1s},\alpha'_{l,1s};d)$	$D_{ab}^{(ssE)}(\alpha_{i,1s},\alpha_{j,2sp},\alpha_{k,1s},\alpha_{l,2sp};d)$ $D_{ab}^{(ssC)}(\alpha_{i,1s},\alpha_{j,1s},\alpha_{k,2sp},\alpha_{l,1sp};d)$	$D_{ab}^{(spE)}(\alpha_{i,1s},\alpha_{j,2sp},\alpha_{k,1s},\alpha_{l,2sp};d)$, $D_{ab}^{(spC)}(\alpha_{i,1s},\alpha_{j,1s},\alpha'_{k,2sp},\alpha_{l,2sp};d)$
$\langle B_{1s}\rvert$	$D_{bb}^{(ss)}(\alpha'_{i,1s},\alpha'_{j,1s},\alpha'_{k,1s},\alpha'_{l,1s})$	$D_{bb}^{(ss)}(\alpha'_{i,1s},\alpha_{j,2sp},\alpha'_{k,1s},\alpha_{l,2sp};d)$, $D_{bb}^{(ss)}(\alpha'_{i,1s},\alpha'_{k,1s},\alpha_{j,2sp},\alpha'_{l,2sp};d)$	$D_{bb}^{(spE)}(\alpha'_{i,1s},\alpha_{j,2sp},\alpha'_{k,1s},\alpha_{l,2sp};d)$, $D_{bb}^{(spC)}(\alpha'_{i,1s},\alpha'_{j,1s},\alpha_{k,2sp},\alpha_{l,2sp};d)$
$\langle B_{2s}\rvert$		$D_{bb}^{(ss)}(\alpha_{i,2sp},\alpha_{j,2sp},\alpha_{k,2sp},\alpha_{l,2sp})$	$D_{bb}^{(spE)}(\alpha_{i,2sp},\alpha_{j,2sp},\alpha_{k,2sp},\alpha_{l,2sp};d)$, $D_{bb}^{(spC)}(\alpha_{i,2sp},\alpha_{j,2sp},\alpha_{k,2sp},\alpha_{l,2sp};d)$
$\langle B_{2p}\rvert$			$D_{bb}^{(pp)}(\alpha_{i,2sp},\alpha_{j,2sp},\alpha_{k,2sp},\alpha_{l,2sp})$

In the calculation of **exchange-type** integrals,

$$
\begin{aligned}
&D_{bb}^{(spE)}(\alpha,\beta,\gamma,\delta) = \\
&\quad = \int d\vec{r}_1 d\vec{r}_2 g_{1s}(\alpha,\vec{r}_1) g_{2p}(\beta,\vec{r}_1)\left(r_{12}^{-1}\right) g_{1s}(\gamma,\vec{r}_2) g_{2p}(\delta,\vec{r}_2) \\
&\quad = c_\alpha^{(s)} c_\beta^{(p)} c_\gamma^{(s)} c_\delta^{(p)} \int d\vec{r}_1 d\vec{r}_2 \cos\theta_1 r_1 e^{-(\alpha+\beta)r_1^2} \frac{1}{r_{12}} \cos\theta_2 r_2 e^{-(\gamma+\delta)r_2^2} \\
&\quad = c_\alpha^{(s)} c_\beta^{(p)} c_\gamma^{(s)} c_\delta^{(p)} \int \frac{d\vec{k}}{(2\pi)^3}\left[i\cos\theta_k \frac{k\pi^{\frac{3}{2}}}{2(\gamma+\delta)^{\frac{5}{2}}} \exp\left\{\left(-\frac{k^2}{4(\gamma+\delta)}\right)\right\}\right] \\
&\qquad \times \left[\frac{4\pi}{k^2}\right]\left[i\cos\theta_k \frac{k\pi^{\frac{3}{2}}}{2(\alpha+\beta)^{\frac{5}{2}}} \exp\left\{\left(-\frac{k^2}{4(\alpha+\beta)}\right)\right\}\right],
\end{aligned}
$$

where we used the Fourier transform (189). The calculation of the integral finally yields:

$$
D_{bb}^{(spE)}(\alpha,\beta,\gamma,\delta) = \pi^{\frac{5}{2}} \frac{c_\alpha^{(s)} c_\beta^{(p)} c_\gamma^{(s)} c_\delta^{(p)}}{3(\alpha+\beta)(\gamma+\delta)(\alpha+\beta+\gamma+\delta)^{\frac{3}{2}}}. \tag{230}
$$

For the **Coulomb-type** integrals,

$$
\begin{aligned}
D_{bb}^{(spC)}(\alpha,\beta,\gamma,\delta) &= \int d\vec{r}_1 d\vec{r}_2 g_{1s}(\alpha,\vec{r}_1) g_{1s}(\beta,\vec{r}_1) r_{12}^{-1} \\
&\quad \times g_{2p}(\gamma,\vec{r}_2) g_{2p}(\delta,\vec{r}_2) \\
&= c_\alpha^{(s)} c_\beta^{(s)} c_\gamma^{(p)} c_\delta^{(p)} \frac{\pi^{\frac{5}{2}}}{(\alpha+\beta)(\gamma+\delta)^2(\alpha+\beta+\gamma+\delta)^{\frac{1}{2}}} \\
&\quad \times\left[1 - \frac{2}{3}\frac{\alpha+\beta}{\alpha+\beta+\gamma+\delta}\right].
\end{aligned} \tag{231}
$$

Passing now to the discussion of **two-center integrals**, we begin with the **exchange-type** ones, involving the s functions on both centers:

$$
\begin{aligned}
D_{ab}^{(ssE)}(\alpha,\beta,\gamma,\delta;d) &= \int d\vec{r}_1\, d\vec{r}_2 g_{1s}(\alpha,\vec{r}_1-\vec{d}) g_{1s}(\beta,\vec{r}_1) \\
&\quad \times \frac{1}{|\vec{r}_1-\vec{r}_2|} g_{1s}(\gamma,\vec{r}_2-\vec{d}) g_{1s}(\delta,\vec{r}_2) \\
&= \int d\vec{r}_1\, d\vec{r}_2 f_1(\vec{r}_1) f_2(r_{12}) f_3(\vec{r}_2),
\end{aligned} \tag{232}
$$

where

$$f_1(\vec{r}_1) = g_{1s}(\alpha, \vec{r}_1 - \vec{d}) g_{1s}(\beta, \vec{r}_1)\,,$$
$$f_2(r_{12}) = \frac{1}{|\vec{r}_1 - \vec{r}_2|}\,,$$
$$f_3(\vec{r}_2) = g_{1s}(\gamma, \vec{r}_2 - \vec{d}) g_{1s}(\delta, \vec{r}_2)\,.$$

Following Szabo and Ostlund [337, Appendix A], we first express products of Gaussian functions occurring in $f_1(\vec{r}_1)$ and $f_3(\vec{r}_2)$ as other Gaussian. Normalization constants will be ignored at this step, they will be introduced in the final results. The integral in (232) becomes:

$$M \int d\vec{r}_1\, d\vec{r}_2 e^{-p(\vec{r} - \vec{R}_p)^2} \frac{1}{r_{12}} e^{-q(\vec{r} - \vec{R}_q)^2}\,, \tag{233}$$

where:

$$\vec{R}_p = \frac{\alpha}{p}\vec{d},\ \vec{R}_q = \frac{\gamma}{q}\vec{d},\ p = \alpha + \beta,\ q = \gamma + \delta\,, \text{ and}$$
$$M = \exp\left\{\left(-\left[\frac{\alpha\beta}{\alpha + \beta} + \frac{\gamma\delta}{\gamma + \delta}\right] d^2\right)\right\}. \tag{234}$$

Now we can use Fourier transform for each factor in the integral (233):

$$f_1(\vec{k}) = \left(\frac{\pi}{p}\right)^{\frac{3}{2}} \exp\left\{\left(-\frac{k^2}{4p} - i(\vec{k} \cdot \vec{R}_p)\right)\right\}; \qquad f_2(\vec{k}) = \frac{4\pi}{k^2};$$
$$f_3(\vec{k}) = \left(\frac{\pi}{q}\right)^{\frac{3}{2}} \exp\left\{\left(-\frac{k^2}{4q} - i(\vec{k} \cdot \vec{R}_q)\right)\right\}. \tag{235}$$

The integrals over $\vec{r}_1$ and $\vec{r}_2$ introduce two δ-functions of $\vec{k}$ and remove two integrations over different $\vec{k}$-vectors that appear after substituting the Fourier integrals into (233), so we obtain:

$$\begin{aligned} D_{ab}^{(ssE)}(\alpha,\beta,\gamma,\delta;d) &= c_\alpha^{(s)} c_\beta^{(s)} c_\gamma^{(s)} c_\delta^{(s)} \frac{\pi M}{2(pq)^{\frac{3}{2}}} \\ &\quad \times \int \frac{d\vec{k}}{k^2} \exp\left\{\left(-\frac{p+q}{4pq}k^2 + i\vec{k}\cdot(\vec{R}_p - \vec{R}_q)\right)\right\} \\ &= c_\alpha^{(s)} c_\beta^{(s)} c_\gamma^{(s)} c_\delta^{(s)} \frac{\pi M}{2(pq)^{\frac{3}{2}}} 4\pi \int_0^\infty j_0(kR_z) \exp\left\{\left(-\frac{p+q}{4pq}k^2\right)\right\} dk \\ &= c_\alpha^{(s)} c_\beta^{(s)} c_\gamma^{(s)} c_\delta^{(s)} \frac{\pi^3 M}{(pq)^{\frac{3}{2}}} \operatorname{erf}\left[\sqrt{\frac{pq}{p+q}} \cdot R_z\right], \end{aligned} \tag{236}$$

where $R_z = |R_p - R_q|$.

The **two-center** *s-s* **Coulomb-type** integrals read:

$$\begin{aligned} D_{ab}^{(ssC)}(\alpha,\beta,\gamma,\delta;d) &= \int d\vec{r}_1 d\vec{r}_2 g_{1s}(\alpha,\vec{r}_1 - \vec{d}) g_{1s}(\beta,\vec{r}_1 - \vec{d}) \\ &\quad \times \frac{1}{|\vec{r}_1 - \vec{r}_2|} g_{1s}(\gamma,\vec{r}_2) g_{1s}(\delta,\vec{r}_2)\,. \end{aligned} \tag{237}$$

We can use here the previous result with $\vec{R}_p = \vec{d}$, $\vec{R}_q = 0$ and $M \to \exp\left\{\left(-\frac{\alpha\beta}{\alpha+\beta}d^2\right)\right\}$. Explicitly, we have:

$$\begin{aligned} D_{ab}^{(ssC)}(\alpha,\beta,\gamma,\delta;d) &= c_\alpha^{(s)} c_\beta^{(s)} c_\gamma^{(s)} c_\delta^{(s)} \frac{\pi^3}{[(\alpha+\beta)(\gamma+\delta)]^{\frac{3}{2}}} \\ &\quad \times \exp\left\{\left(-\frac{\alpha\beta}{\alpha+\beta}d^2\right)\right\} \times \operatorname{erf}\left[\sqrt{\frac{(\alpha+\beta)(\gamma+\delta)}{\alpha+\beta+\gamma+\delta}}\, d\right]. \end{aligned} \tag{238}$$

The **two-center exchange-type** integrals involving s and p-functions are:

$$\begin{aligned} D_{ab}^{(spE)}(\alpha,\beta,\gamma,\delta;d) &= \int d\vec{r}_1 d\vec{r}_2 g_{1s}(\alpha,\vec{r}_1 - \vec{d}) g_{2p}(\beta,\vec{r}_1) \\ &\quad \times \frac{1}{|\vec{r}_1 - \vec{r}_2|} g_{1s}(\gamma,\vec{r}_2 - \vec{d}) g_{2p}(\delta,\vec{r}_2) \\ &= \int d\vec{r}_1 d\vec{r}_2 f_1(\vec{r}_1) f_2(r_{12}) f_3(\vec{r}_2)\,, \end{aligned} \tag{239}$$

where:

$$f_1(\vec{r}_1) = g_{1s}(\alpha, \vec{r}_1 - \vec{d}) g_{2p}(\beta, \vec{r}_1),$$
$$f_2(r_{12}) = \frac{1}{|\vec{r}_1 - \vec{r}_2|},$$
$$f_3(\vec{r}_2) = g_{1s}(\gamma, \vec{r}_2 - \vec{d}) g_{2p}(\delta, \vec{r}_2)\,.$$

Now we shall use Fourier transform in the integral (239):

$$\begin{aligned}
f_1(\vec{k}) &= \int r_1 \cos\theta_1 e^{-p(\vec{r}_1 - \vec{R}_p)^2 - i\vec{k}\cdot\vec{r}_1} d\vec{r}_1 \\
&= \frac{1}{R_p} e^{-i\vec{k}\cdot\vec{R}_p} \Bigg[\int [(\vec{r}_1 - \vec{R}_p)\cdot\vec{R}_p] e^{-p(\vec{r}_1 - \vec{R}_p)^2 - i\vec{k}\cdot(\vec{r}_1 - \vec{R}_p)} d\vec{r}_1 \\
&\qquad + R_p^2 \int e^{-p(\vec{r} - \vec{R}_p)^2 - i\vec{k}\cdot(\vec{r}_1 - \vec{R}_p)} d\vec{r}_1 \Bigg] \\
&= e^{-i\vec{k}\cdot\vec{R}_p} \Big[\int r\cos\theta e^{-pr^2 - i(\vec{k}\cdot\vec{r})} d\vec{r} + R_p \int e^{-pr^2} e^{-i\vec{k}\cdot\vec{r}} d\vec{r} \Big] \\
&= e^{-i\vec{k}\cdot\vec{R}_p} \left[i\cos\theta_k \frac{\pi^{\frac{3}{2}}}{2p^{\frac{5}{2}}} k \exp\left\{\left(-\frac{k^2}{4p}\right)\right\} + R_p \frac{\pi^{\frac{3}{2}}}{p^{\frac{3}{2}}} \exp\left\{\left(-\frac{k^2}{4p}\right)\right\} \right] \\
&\qquad \text{where we have used the result (209)} \\
&= e^{-i\vec{k}\cdot\vec{R}_p} \frac{\pi^{\frac{3}{2}}}{p^{\frac{3}{2}}} \left[i\cos\theta_k \left(\frac{k}{2p}\right) + R_p \right] \exp\left\{\left(-\frac{k^2}{4p}\right)\right\}. \qquad (240)
\end{aligned}$$

As before,

$$f_2(\vec{k}) = \frac{4\pi}{k^2}, \quad \text{and}$$
$$f_3(\vec{k}) = e^{-i\vec{k}\cdot\vec{R}_q} \frac{\pi^{\frac{3}{2}}}{q^{\frac{3}{2}}} \left[-i\cos\theta_k \left(\frac{k}{2q}\right) + R_q \right] \exp\left\{\left(-\frac{k^2}{4q}\right)\right\}. \qquad (241)$$

Substituting this into (239),

$$
\begin{aligned}
D_{ab}{}^{(spE)}(\alpha,\beta,\gamma,\delta;d) &= c_\alpha^{(s)} c_\beta^{(p)} c_\gamma^{(s)} c_\delta^{(p)} 4\pi\, M\, \frac{\pi^3}{(pq)^{\frac{3}{2}}} \\
&\quad \times \int \frac{d\vec{k}}{(2\pi)^3} \Big\{ \Big[R_p + i\cos\theta_k \left(\frac{k}{2p}\right) \Big] \\
&\quad \times \Big[R_q - i\cos\theta_k \left(\frac{k}{2q}\right) \Big] \frac{1}{k^2} \exp\left\{ \left[\left(-\frac{1}{4q} + \frac{1}{4p}\right) k^2 \right] \right\} \Big\} \qquad (242) \\
&= c_\alpha^{(s)} c_\beta^{(p)} c_\gamma^{(s)} c_\delta^{(p)} M \frac{\pi}{2(pq)^{\frac{3}{2}}} \\
&\quad \times \int \frac{d\vec{k}}{k^2} \Big[R_p R_q + i\Big(\frac{R_q}{p} - \frac{R_p}{q}\Big) \frac{k}{2} \cos\theta_k + \frac{k^2}{4pq} \cos^2\theta_k \Big] \\
&\quad \times \exp\left\{ \Big(-\frac{p+q}{4pq} k^2 + i\vec{k}\cdot(\vec{R}_p - \vec{R}_q)\Big) \right\} \\
&= c_\alpha^{(s)} c_\beta^{(p)} c_\gamma^{(s)} c_\delta^{(p)} M \frac{\pi}{2(pq)^{\frac{3}{2}}} \big(I_1 + I_2 + I_3 \big), \qquad (243)
\end{aligned}
$$

where I_1, I_2, I_3 are given by the following expressions:

$$
\begin{aligned}
I_1 &= R_p R_q \int_0^\infty \Big[\int_{-1}^{1} e^{ik(R_p - R_q)\cos\theta_k} d\cos\theta_k \Big] \exp\left\{ \left(-\frac{p+q}{4pq} k^2 \right) \right\} dk \\
&= R_p R_q \int_0^\infty j_0(kR_z) \exp\left\{ \left(-\frac{p+q}{4pq} k^2 \right) \right\} dk \\
&= \frac{R_p R_q}{R_z} \int_0^\infty j_0(x) e^{-sx^2} dx = \frac{\pi}{2} \frac{R_p R_q}{R_z} \operatorname{erf}\left[\frac{1}{2\sqrt{s}} \right], \qquad (244)
\end{aligned}
$$

with $R_z = |R_p - R_q|$ and $s = \dfrac{p+q}{4pqR_z^2}$;

$$
\begin{aligned}
I_2 &= \frac{1}{2} i \left(\frac{R_p}{p} - \frac{R_q}{q} \right) \int_0^\infty \Big[\int_{-1}^{1} d\cos\theta_k \, \cos\theta_k \, e^{ik(R_p - R_q)\cos\theta_k} \Big] k \\
&\qquad \times \exp\left\{ \left(-\frac{p+q}{4pq} k^2 \right) \right\} dk \\
&= \frac{R_q q - R_p p}{pqR_z^2} \int_0^\infty j_1(x) x e^{-sx^2} dx \\
&= \frac{R_q q - R_p p}{pqR_z^2} \left\{ \sqrt{\frac{\pi}{4s}} \exp\left\{ \left(-\frac{1}{4s} \right) \right\} - \frac{\pi}{2} \operatorname{erf}\left(\frac{1}{2\sqrt{s}} \right) \right\}; \qquad (245)
\end{aligned}
$$

$$
\begin{aligned}
I_3 &= \frac{1}{16pq}\int_0^\infty\Bigg[\int_{-1}^{1} d\cos\theta_k\ \cos^2\theta_k\, e^{ik(R_p-R_q)\cos\theta_k}\Bigg]k^2 \\
&\qquad\times \exp\left\{\left(-\frac{p+q}{4pq}k^2\right)\right\}dk \\
&= \frac{1}{8pqR_z^3}\int_0^\infty\left[-j_2(x)-\frac{j_1(x)}{x}\right]e^{-sx^2}x^2dx \\
&= \frac{\sqrt{\pi}}{8pqR_z^3}\left[\frac{1+4s}{4s^{\frac{3}{2}}}\exp\left\{\left(-\frac{1}{4s}\right)\right\}-\sqrt{\pi}\,\mathrm{erf}\left(\frac{1}{2\sqrt{s}}\right)\right]. \qquad (246)
\end{aligned}
$$

Thus, $D_{ab}^{(spE)}$ is given by (243) where M is given by (234), $p=\alpha+\beta$, $q=\gamma+\delta$; I_1, I_2 and I_3 are given by Eqs (244) – (246), $R_z=|R_p-R_q|$, $R_p=\frac{\alpha}{p}d$, $R_q=\frac{\gamma}{q}d$ and $s=\frac{p+q}{4pqR_z^2}$.
Finally, the evaluation of the **Coulomb-type** integrals between s and p functions at different sites proceeds as follows:

$$
\begin{aligned}
D_{ab}^{(spC)}(\alpha,\beta,\gamma,\delta;d) &= \int d\vec{r}_1 d\vec{r}_2 g_{1s}(\alpha,\vec{r}_1-\vec{d})g_{1s}(\beta,\vec{r}_1-\vec{d}) \\
&\times\frac{1}{|\vec{r}_1-\vec{r}_2|}g_{2p}(\gamma,\vec{r}_2)g_{2p}(\delta,\vec{r}_2) \qquad (247)\\
&= \int \vec{r}_1 d\vec{r}_2 f_1(\vec{r}_1)f_2(r_{12})f_3(\vec{r}_2)\,,
\end{aligned}
$$

where

$$
\begin{aligned}
f_1(\vec{r}_1) &= g_{1s}(\alpha,\vec{r}_1-\vec{d})g_{1s}(\beta,\vec{r}_1-\vec{d})\,,\\
f_2(r_{12}) &= \frac{1}{|\vec{r}_1-\vec{r}_2|}\,,\\
f_3(\vec{r}_2) &= g_{2p}(\gamma,\vec{r}_2)g_{2p}(\delta,\vec{r}_2)\,.
\end{aligned}
$$

The Fourier transforms of these functions are:

$$
\begin{aligned}
&f_1(\vec{k}) = \exp\left\{\left(-\frac{\alpha\beta}{\alpha+\beta}d^2\right)\right\}\left(\frac{\pi}{p}\right)^{\frac{3}{2}}\exp\left\{\left(-\frac{k^2}{4p}-i\vec{k}\cdot\vec{d}\right)\right\},\\
&f_2(\vec{k}) \text{ as in (50)},\\
&f_3(\vec{k}) = \int e^{i\vec{k}\cdot\vec{r}}r^2\cos^2\theta e^{-(\gamma+\delta)r^2}d\vec{r}\\
&= \left(\frac{k^2}{2q}\right)^2\frac{1}{k^2}\left[\frac{2q}{k^2}-\cos^2\theta_k\right]\left(\frac{\pi}{q}\right)^{\frac{3}{2}}\exp\left\{\left(-\frac{k^2}{4q}\right)\right\}, \qquad (248)
\end{aligned}
$$

as it has been calculated before, Eq. (219). Therefore:

$$D_{ab}^{(spC)} = c_\alpha^{(s)} c_\beta^{(s)} c_\gamma^{(p)} c_\delta^{(p)} \exp\left\{\left(-\frac{\alpha\beta}{\alpha+\beta}d^2\right)\right\} \int \frac{d\vec{k}}{(2\pi)^3} e^{i(\vec{k}\cdot\vec{d})} \left(\frac{\pi}{p}\right)^{\frac{3}{2}}$$
$$\times \exp\left\{\left(-\frac{k^2}{4p}\right)\right\} \times \frac{4\pi}{k^2}\left(\frac{\pi}{q}\right)^{\frac{3}{2}} \exp\left\{\left(-\frac{k^2}{4q}\right)\right\} \frac{1}{2q}\left[1 - \frac{k^2}{2q}\cos^2\theta_k\right]$$
$$= \frac{\pi^3}{(pq)^{\frac{3}{2}}} \frac{2\pi}{q} \frac{1}{(2\pi)^3} \exp\left\{\left(-\frac{\alpha\beta}{\alpha+\beta}d^2\right)\right\} [X_1 + X_2],$$

with

$$X_1 = \int \frac{d\vec{k}}{k^2} \exp\left\{\left(i(\vec{k}\cdot\vec{d}) - \frac{k^2}{4p} - \frac{k^2}{4q}\right)\right\} = \frac{2\pi^2}{d} \operatorname{erf}\left(\sqrt{\frac{pq}{p+q}}d\right),$$
$$X_2 = -\int \frac{d\vec{k}}{k^2} e^{i(\vec{k}\cdot\vec{d})} \frac{k^2}{2q}\cos^2\theta_k \exp\left\{\left(-\frac{p+q}{4pq}k^2\right)\right\}$$
$$= \frac{2\pi}{qd^3}\left[\sqrt{\pi}\frac{1+4s}{4s^{\frac{3}{2}}} \exp\left\{\left(-\frac{1}{4s}\right)\right\} - \pi \operatorname{erf}\left(\frac{1}{2\sqrt{s}}\right)\right],$$

where $s = \dfrac{p+q}{4pqd^2}$. Finally, we obtain:

$$D_{ab}^{(spC)} = c_\alpha^{(s)} c_\beta^{(s)} c_\gamma^{(p)} c_\delta^{(p)} \exp\left\{\left(-\frac{\alpha\beta}{\alpha+\beta}d^2\right)\right\}$$
$$\times \left[\frac{\pi^3}{2p^{\frac{3}{2}}q^{\frac{5}{2}}d}\left(1 - \frac{1}{qd^2}\right)\operatorname{erf}\left(\sqrt{\frac{pq}{p+q}}d^2\right)\right.$$
$$\left. + \frac{\pi^{\frac{5}{2}}}{(p+q)^{\frac{3}{2}}q^2}\left(1 + \frac{p+q}{pqd^2}\right)\exp\left\{\left(-\frac{pq}{p+q}d^2\right)\right\}\right]. \quad (249)$$

The equations according to which the matrix elements summarized in Table 17 are calculated are reminded in Table 18.

Table 18: Equations specifying matrix elements of Table 17.

Matrix elements	Eq.	Matrix elements	Eq.
$D_{aa}^{(ss)}$	(228)	$D_{ab}^{(ssE)}$	(236)
$D_{ab}^{(ssC)}$	(238)	$D_{ab}^{(spE)}$	(243) – (246)
$D_{ab}^{(spC)}$	(249)	$D_{bb}^{(spE)}$	(230)
$D_{bb}^{(spC)}$	(231)	$D_{bb}^{(pp)}$	(229)