The Quantum Linear Problem and its Quantum Algorithmic Solutions

SUBMITTED
BY

CHEE CHONG HIAN

DIVISION OF PHYSICS & APPLIED PHYSICS
SCHOOL OF PHYSICAL AND MATHEMATICAL SCIENCES

A final year project report
presented to
Nanyang Technological University
in partial fulfilment of the
requirements for the
Bachelor of Science (Hons) in Physics
Nanyang Technological University

April 2020

The Quantum Linear Problem and its Quantum Algorithmic Solutions

Chee Chong Hian[1]

School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore 637371,
Singapore

8th March 2020

[1]Email: chee0122@ntu.edu.sg

Abstract

In this final year project, we will attempt to quantum mechanically solve $\vec{x}$ for the system of linear equations problem, $A\vec{x} = \vec{b}$, given a problem matrix A and vector $\vec{b}$, by treating it as a quantum linear problem, $A|x\rangle = |b\rangle$. However, solving for $|x\rangle$ that involves hermitian operators, $A = M$, requires careful treatment. Intuitively, the solution state, $|x\rangle = \frac{1}{d}M^{-1}|b\rangle$, can be interpreted as a normalised pre-collapse quantum state of the measurement of input quantum state $|b\rangle$ by the hermitian operator M^{-1} with normalisation factor d. By employing three ideas that were implicitly developed by Aram Harrow, Avinatan Hassidim and Seth Lloyd (HHL), it is possible to construct a quantum linear solver (QLS) algorithm to obtain the solution state $|x\rangle$ probabilistically, but not without implementation problems. To mitigate this, two promising version of QLS, a simplified and a full version, will be proposed and thoroughly analysed for the 2 and 4 dimension, with an extension to the 2^n dimension. Several worked examples will be suggested and implemented by constructing and simulating its quantum circuit on a IBM Qiskit quantum simulator. The theoretical results suggest that predicting the probability of obtaining the solution state $|x\rangle$ requires knowledge of the eigenvalues of M. In addition, constructing any quantum circuit version of HHL QLS algorithm requires the possession of multiple copies of controlled-$\exp(2\pi i 2^{-t}M)$ quantum gate or its gate decomposition, where t is the number of qubits needed to express the eigenvalues of M in qubit binary representation up to a desired precision. Otherwise, its implementation would be impossible.

Contents

Abstract i

Acknowledgements iv

List of Figures viii

List of Tables ix

1 Project Introduction 1

I Project Background 3

2 Quantum Mechanics 4
2.1 Pure Quantum States . . . 5
2.2 Unitary Transformation and Operators . . . 7
2.3 Measurements and Hermitian Operators . . . 8
2.4 Multiple Quantum Systems and Operations . . . 10

3 Quantum Computation 11
3.1 Quantum Circuits Model . . . 11
3.2 Qubit Binary Representation of Quantum Information . . . 12
3.3 Quantum Gates . . . 13
3.4 Quantum Gate Decompositions Techniques . . . 15
3.5 Measurement Gates . . . 17

4 Quantum Algorithms and Subroutines 18
4.1 Amplitude Amplification . . . 18
4.2 Quantum Fourier Transform . . . 18
4.3 Quantum Phase Estimation . . . 21

II Final Year Project 24

5 Quantum Linear Problem 25
5.1 System of Linear equations . . . 25
5.2 Quantum Linear Problem . . . 26

5.3 Algorithmic Solution Idea Proposition . . . 28

6 Original Quantum Linear Solver 34
6.1 Known Applications . . . 36
6.2 Physical Implementations . . . 37
6.3 The Ancillary Rotation Problem . . . 37
6.4 Algorithm Adaptation Proposal . . . 40

7 Simplified Quantum Linear Solver 41
7.1 2D Case . . . 42
7.2 4D Case . . . 52
7.3 2^nD Case . . . 62

8 Full Quantum Linear Solver 65
8.1 Quantum Subroutines Simplifications . . . 67
8.2 1^{st} Order Ancillary Rotation Approximation . . . 71
8.3 Post-Selection Measurement . . . 74
8.4 Approximated Solution State Fidelity . . . 74

9 Full Quantum Linear Solver Implementation 75
9.1 2D Case . . . 75
9.2 4D Case . . . 80
9.3 2^nD Case . . . 86

III Project Conclusion 89

10 Discussion of Results 90
10.1 On the Theoretical Results . . . 90
10.2 On the Experimental Results . . . 91
10.3 Complications . . . 92

11 Future Research 94
11.1 Algorithm Extensions . . . 94
11.2 Alternate Quantum Linear Solvers . . . 95

12 Conclusion and Summary 96

13 Appendix 99
13.1 The Experiment codes . . . 99

Bibliography 100

Acknowledgements

I would like to express my sincere gratitude and appreciation to my supervisor Assistant Professor Gu Mile for his patience and guidance throughout the course of this final year project. I am also deeply grateful to Dr Jayne Thompson of Centre for Quantum Technologies (CQT) at the National University of Singapore (NUS) for her valuable advice and encouragement in pursuing a quantum computing research career. Finally, I would like to convey my gratefulness to my parents and friends for their kind assistance and moral support throughout the course of my final year project.

Chee Chong Hian
April 2020

List of Figures

2.0.1 The basic structure of classical mechanics, in the computational context. 5
2.0.2 The basic structure of quantum mechanics, in the computational context. 5

3.0.1 A basic comparison of the flow of classical and quantum information. 11
3.1.1 An generic quantum circuit diagram with 3 initialized qubits $|0\rangle_a |0\rangle_b |0\rangle_c$ on the left, a multi-qubit unitary quantum gate U in the middle and a set of measurements on the qubits, represented by a meter symbol on the right. 12
3.2.1 A qubit register $|x\rangle$ represented as a collection of qubits arranged in a quantum circuit model from top to bottom. 13
3.3.1 Common gates symbols, from left to right: Pauli-Z, Pauli-X, Pauli-Y, Hadamard H, Phase S and $\frac{\pi}{8}$ T. 13
3.3.2 Rotation gates symbols, from left to right: Rotate-X, $R_x(\theta)$, Rotate-Y, $R_y(\theta)$, Rotate-Z, $R_z(\theta)$. 14
3.3.3 Two qubit gate: Swap gate S_{wap}. 14
3.3.4 Single target single control controlled gates, from top to bottom: CNOT, C_x, Controlled-Z, C_z. 14
3.3.5 Quantum circuit representation of the controlled quantum gate U on the condition of the control qubit being $|0\rangle$ and its decomposition. 15
3.3.6 Controlled gates: from left to right, Toffoli gate, CC_x, Fredkin gate, C_{swap}. 15
3.4.1 ABC decomposition of an arbitrary 2D unitary gate U. 16
3.4.2 ABC decomposition of a controlled-U gate $C(U)$. 16
3.4.3 The quantum circuit of a n-control controlled arbitrary quantum gate U and its decomposition with the use of a work qubit register, that is to be discarded after computation. 17

4.2.1 Quantum circuit for QFT acting on $|x_n \ldots x_1\rangle$. 21
4.3.1 Quantum circuit for Quantum Phase Estimation. 23
4.3.2 Quantum circuit for inverse Quantum Fourier Transform. 23

6.0.1 The quantum circuit for the original QLS algorithm, where QPE and AR refers to Quantum Phase Estimation and Ancillary Rotation subroutine. The set of symbols in the quantum circuit, after the AR gate, represents the ancillary measurement. 34

7.1.1 An outline structure of the simplified 2D QLS quantum circuit. The single qubit registers from top to bottom are main, eigenvalue and ancillary register. Note that compared to the original QLS circuit in figure 6.0.1, the main difference is that EM replaces QPE. . . 43
7.1.2 State preparation of qubit registers starting from standard initial state $|0\rangle_m |0\rangle_e |0\rangle_a$ to $|b\rangle_m |0\rangle_e |0\rangle_a$ . . . 43
7.1.3 A quantum circuit for the Eigenvalue Marking (EM) subroutine, which is a simplified QPE algorithm that involves one qubit, where $U = \exp\left(2\pi i 2^{t-1} M\right)$. . . . 44
7.1.4 A quantum circuit for Eigenvalue Marking step decomposed using common quantum gates and gate R which represents an arbitrary 2D unitary operator, such that $M = R\Lambda R^{\dagger}$. 46
7.1.5 Ancillary Rotation (AR) quantum circuit using controlled-$R_y(\theta_k)$ with the purpose to introduce the amplitude coefficients $\frac{C}{\lambda_k}$ to the respective eigenstates $|u_k\rangle_m$ in the main register by using the eigenvalue marking qubits $|k_1\rangle_e$ in the eigenvalue register. 47
7.1.6 A quantum circuit for inverse Eigenvalue Marking step where $U = \exp\left(2\pi i 2^{t-1} M\right)$. Note that $H = H^{\dagger}$. . . . 47
7.1.7 The post selection of $|1\rangle_a$ ancilla qubit measurement in the ancillary register, with the later discard of both eigenvalue and ancillary register while keeping only the main register. . . . 48
7.1.8 The quantum circuit of the 2D simplified quantum linear solver. . . . 48
7.1.9 The quantum circuit of the simplified QLS for the 2D worked example where $\theta_0 = 2\arcsin\left(\frac{4}{3\cdot 2^r}\right)$ and $\theta_1 = 2\arcsin\left(\frac{8}{7\cdot 2^r}\right)$. . . . 50
7.1.10 A graph of the probability of success $P(|1\rangle_a)$ against the parameter $r = -\log_2(C)$ for both experiment and theory using the simplified QLS for the 2D worked example. . . . 52
7.2.1 An outline structure of the simplified 4D QLS quantum circuit. The qubit registers from top to bottom are main, eigenvalue and ancillary register. . . . 54
7.2.2 The quantum circuit of the eigenvalue marking step of the 4D case, where $U = \exp\left(2\pi i 2^{t-2} M\right)$ and $R_j = \begin{bmatrix} 1 & 0 \\ 0 & \exp\left(2\pi i 2^{-j}\right) \end{bmatrix}$. . . . 54
7.2.3 A quantum circuit of the decomposition of controlled-U^{2^0} and controlled-U^{2^1} gates using common quantum gates, Pauli-Z and Phase S, where R is a 2D arbitrary unitary quantum gate. . . . 56
7.2.4 A quantum circuit of the decomposition of the 4D Ancillary Rotation step using controlled-$R_y(\theta_k)$ gates, with parameter θ_k set such that $\sin\left(\frac{\theta_k}{2}\right) = \frac{C}{\lambda_k}$ for some real parameter C. . . . 57
7.2.5 The quantum circuit of the 4D simplified QLS. Note that R represents an arbitrary 2D unitary operator acting on main register, such that $M = R\Lambda R^{\dagger}$, $R_j = \begin{bmatrix} 1 & 0 \\ 0 & \exp\left(2\pi i 2^{-j}\right) \end{bmatrix}$ are phase rotation gates acting on eigenvalue register and $R_y(\theta_k)$ are controlled Rotate-Y quantum gates acting on the ancillary qubit. . . . 58
7.2.6 The quantum circuit of the simplified QLS for the 4D worked example where $\theta_0 = 2\arcsin\left(\frac{8}{4\cdot 2^r}\right)$, $\theta_1 = 2\arcsin\left(\frac{8}{5\cdot 2^r}\right)$, $\theta_2 = 2\arcsin\left(\frac{8}{6\cdot 2^r}\right)$ and $\theta_3 = 2\arcsin\left(\frac{8}{7\cdot 2^r}\right)$. . . . 60
7.2.7 A graph of the probability of success $P(|1\rangle_a)$ against the parameter $r = -\log_2(C)$ for both experiment and theory using the simplified QLS for the 4D worked example. . . . 61

7.3.1 A quantum circuit of the decomposition of the truncated Quantum Phase Estimation in the Eigenvalue Marking step for 2^n dimensions. WH and QFT represents Walsh-Hadamard and Quantum Fourier Transform subroutines respectively (*See section 4.3 for more details*). 63
7.3.2 A quantum circuit of the decomposition of the Ancillary Rotation step using controlled-$R_y(\theta_k)$ gates, with parameter θ_k set such that $\sin\left(\frac{\theta_k}{2}\right) = \frac{C}{\lambda_k}$ for some real parameter C, for 2^n dimensions. 64

8.0.1 The quantum circuit structure of the full QLS algorithm. The qubit registers from top to bottom are main, eigenvalue and ancillary register. Legend: Quantum Phase Estimation (QPE), Eigenvalue Inversion (EI), Ancillary Rotation (AR). 67
8.1.1 Decomposition of controlled-$U^{2^{t-j}}$ gate. 69
8.1.2 Decomposition of controlled-U^{2^x} gate into 2^x number of controlled-U gates for any unitary U. For the example given above, $U = \exp(2\pi i 2^{-t}\Lambda)$. 69
8.1.3 Simplified quantum circuit of the Hamiltonian Simulation subroutine, where $U = \exp(2\pi i 2^{-t}\Lambda)$. Note the notation change: $\lambda_k \overset{\text{in binary}}{\Longrightarrow} \lambda_{tk}\ldots\lambda_{1k} \overset{\text{drop k}}{\Longrightarrow} \lambda_t\ldots\lambda_1$, so as to avoid cumbersome writing. 69
8.1.4 The quantum circuit of the Eigenvalue Inversion subroutine of powers of two with the use of an inverted register. Note the notation change: $\lambda_k \overset{\text{in binary}}{\Longrightarrow} \lambda_{tk}\ldots\lambda_{1k} \overset{\text{drop k}}{\Longrightarrow} \lambda_t\ldots\lambda_1$, so as to avoid cumbersome writing. 71
8.1.5 The simplified quantum circuit of the Eigenvalue Inversion subroutine without the use of an inverted register, assuming eigenvalues are powers of two. Note the notation change: $\lambda_k \overset{\text{in binary}}{\Longrightarrow} \lambda_{tk}\ldots\lambda_{1k} \overset{\text{drop k}}{\Longrightarrow} \lambda_t\ldots\lambda_1$, so as to avoid cumbersome writing. 71
8.2.1 The quantum circuit of the 1^{st} order Taylor approximation Ancillary Rotation subroutine, where real parameter $C = \frac{\pi}{2^r}$ has been substituted, for some real parameter $r > 0$. 74

9.1.1 The quantum circuit of the 2D full QLS algorithm where EI(s) refers to the Eigenvalue Inversion subroutine (Powers of Two). 76
9.1.2 Decomposition of 2D $\exp(2\pi i 2^{-t}\Lambda)$ into phase and rotate-Z gates. 76
9.1.3 Decomposition of 2D controlled-$\exp(2\pi i 2^{-t}\Lambda)$ into controlled phase and controlled rotate-Z gates. 77
9.1.4 A combined graph of probability of successes $P\left(\left|\tilde{1}\right\rangle_a\right)$, $P(|1\rangle_a)$ and fidelity of 1st order approximated solution state $F(|x\rangle, |\tilde{x}\rangle)$ against parameter r, using the full QLS for the 2D worked example. 80
9.2.1 The quantum circuit of the 4D full QLS algorithm, where EI(s) refers to the Eigenvalue Inversion Subroutine (Powers of Two). 81
9.2.2 Decomposition of 4D $\exp(2\pi i 2^{-t}\Lambda)$ into parametrised quantum gates. 82
9.2.3 Decomposition of 4D $C[\exp(2\pi i 2^{-t}\Lambda)]$ into parametrised quantum gates. 83
9.2.4 A combined graph of probability of successes $P\left(\left|\tilde{1}\right\rangle_a\right)$, $P(|1\rangle_a)$ and fidelity of 1st order approximated solution state $F(|x\rangle, |\tilde{x}\rangle)$ against parameter r, using the full QLS for the 4D worked example. 86

9.3.1 The quantum circuit of the 2^nD full quantum linear solver, where EI(s) refers to the Eigenvalue Inversion Subroutine (Simplified Version). 87

List of Tables

2.3.1 Classical post-measurement result of an unknown bit φ. 8
2.3.2 Quantum post-measurement of a generic qubit $|\varphi\rangle = a|m_1\rangle + b|m_2\rangle$ with measurement operator M. 9

5.3.1 Quantum post-measurement results of $A|b\rangle_m|0\rangle_a$, where the measurement is done on the ancillary qubit using a Pauli-Z operator. 29

6.3.1 A summary table detailing the experiments involving QLS algorithm. Note that the group leader optimization algorithm is a classical algorithm that finds a quantum circuit decomposition of the Hamiltonian Simulation subroutine using common and rotation quantum gates [DK10], it will not be used in this thesis. Legend: $U_B = R_x(\theta)R_y(\phi)$. . . 39

7.1.1 A summary of the resource breakdown of the quantum circuit of the simplified QLS algorithm for the 2D worked example. 51
7.2.1 A summary of the resource breakdown of the quantum circuit of the simplified QLS algorithm for the 4D worked example. 60
7.3.1 A table of the quantum gates $U^{2^{n-1-j}}$ for 2^n dimension truncated Quantum Phase Estimation subroutine for the simplified quantum linear solver for $n = 1,2,3,4$. Note that the 2^n unitary quantum gates R and $R^\dagger$ are dropped to avoid cumbersome writing. . . 63

8.0.1 A summary table showing the enhancement and assumptions used for simplifying implementation of the full QLS algorithm. Legend: Hamiltonian Simulation (HS), Quantum Phase Estimation (QPE), Eigenvalue Inversion (EI), Ancillary Rotation (AR). . 65

9.1.1 A summary of the resource breakdown of the quantum circuit of the full QLS algorithm for the 2D worked example. 79
9.2.1 A summary of the resource breakdown of the quantum circuit of the full QLS algorithm for the 4D worked example. 85

11.1.1A summary table showing the potential extensions to QLS algorithm. 94

12.0.1A summary of the features, assumptions and advantages of the original, simplified and full QLS algorithms. AR refers to the Ancillary Rotation quantum subroutine. 97
12.0.2A summary of the disadvantages of the original, simplified and full QLS algorithms. AR refers to the Ancillary Rotation quantum subroutine. 98

Chapter 1

Project Introduction

Project Aim and Motivation

Algorithms are sets of instructions that could be carried out solve a problem. It is often associated with classical computer programs that perform binary calculations and data processing to achieve a particular outcome. Its importance could not be stressed enough due to its immense contributions to improve our livelihoods. As being one of the digital natives of the millennial generation, I deeply appreciate the pioneering work of early computer scientists and engineers who had laid the groundwork in programming algorithms to solve everyday problems. However, computer algorithms themselves are soon facing an imminent crisis. As the performance of classical computers starts to plateau, due to inherent limitation of shrinking transistors in computer electrical circuits down to sub-molecular size, will it run algorithms with ever-increasing complexity fast enough to satisfy our ever-growing and changing needs? One of the more interesting propositions was to enlist the power quantum computing to potentially run algorithms faster than a classical computer. Quantum computing, in essence, is computing within the quantum world, bounded by quantum mechanical rules and laws. There have been promising results from the decades of theoretical research since its birth in the 1980s, when the first ever quantum computer model, Quantum Turing Machine, was invented by Paul Benioff [Ben82]. Shor's Factoring [Sho94] and Grover's Search [Gro96] are the two pioneering quantum algorithms that, under special circumstances, have theoretically proven exponential and polynomial speedup in computational time, respectively, over its classical counterparts. While these two quantum algorithms are prime candidates to succeed the classical ones, their potential applications are highly specialised and its impact in problem solving is limited in terms of its flexibility.

In 2008, a team of three researchers, Aram Harrow, Avinatan Hassidim and Seth Lloyd (HHL) [HHL09] invented a new quantum algorithm that attempts to solve the system of linear equation using quantum mechanics with an exponential speedup in computational, under special cases. This quantum linear solver has immense potential as it may be adapted for applications in assisting in solving classical system of linear equations. The importance in finding an efficient algorithm that solve system of linear equations could not be overstated as it forms a critical backbone of many algorithms that deal with optimizations, numerical simulations and signal processing. Despite its potential, the HHL's quantum linear solver are often poorly understood by undergraduate students as it is often neglected in most introductory quantum computing textbook. The original paper that describes the QLS algorithm gave minimal details on construction of the quantum algorithm, as it choose to focus on the complexity

side so as to rigorously prove its speedup limits and error bounds. Interested students may also find themselves confused with the overwhelming number of adaptations and modifications of QLS algorithms currently available in literature. Current pedagogical examples of QLS algorithm uses over-simplified versions which may fail to adequately explain and elaborate the ideas behind the inner workings of QLS algorithm. Therefore, this final year project aims to properly introduce and thoroughly analyse the quantum linear problem and its quantum linear solver proposed by HHL, so as to achieve a better understanding with hopes of extending it with further applications in other research fields. In addition, two adaptions versions of the QLS algorithms will be proposed to mitigate the inherent implementation problems posed by the original algorithm and several worked examples will be suggested and implemented using a quantum simulator programmed on a classical computer.

Project Objectives

The final year project has five main objectives to meet its aim:

1. Describe the quantum linear problem.
2. List and elaborate on the implicit ideas of QLS algorithmic solution.
3. Explain the implementation problems posed by the original QLS algorithm.
4. Propose and elaborate two adapted versions QLS algorithms that mitigate the implementation problems posed by the original one.
5. Suggest worked examples and conduct experiments by constructing its quantum circuit using an IBM Qiskit quantum simulator.

Project Overview

There are three parts to this project report. Part I is the project background that contains three chapters of introductory review on quantum mechanics, quantum computing and quantum algorithms respectively. Advanced readers may skim through part I. However, all readers are advised to pay special attention to subsections 2.3, 3.2, 3.4 and chapter 4 as the contents may differ from convention and will be used throughout the report without explicit mention. Part II is the final year project that contains five chapters which details the explanation of the quantum linear problem, the analysis and implementations of the original and its adapted versions of quantum linear solver algorithm. Part III concludes the project with 3 chapters on a discussion of results, future research and a project summary respectively. Finally, the appendix provides web links to the source code of the quantum experiments used in this final year project.

Part I

Project Background

Chapter 2

Quantum Mechanics

Quantum mechanics is a theory that attempts to explain the non-classical behaviours of quantum systems, by modelling quantum systems and operations using mathematical concepts found in linear algebra and probability. For example, in the Stern-Gerlach experiment, quantum mechanics can explain the discrete deflection of a continuous stream of hot sliver atoms observed on the screen, when it is passed through a field of constant inhomogeneous magnetic field [SB+16]. In the computational context, quantum mechanics is analogous to classical mechanics. Classical mechanics involves classical operations acting on classical states while quantum mechanics involves quantum operations acting on quantum states. Specifically, classical operations are algebraic arithmetic and measurements that seems to mirror the behaviour of quantum operations, which are unitary transformation and quantum measurements, mathematically represented by unitary and hermitian matrix respectively. Classical states are collections of n-level classical systems mathematically represented as bits or digits, defined under a binary or a decimal number system. On the other hand, a quantum state is a collection of n-level quantum systems that is mathematically represented as tensor products of n-level quantum systems. The summary of the basic structure of classical mechanics and quantum mechanics in the computation context is given in figures 2.0.1 and 2.0.2 respectively.

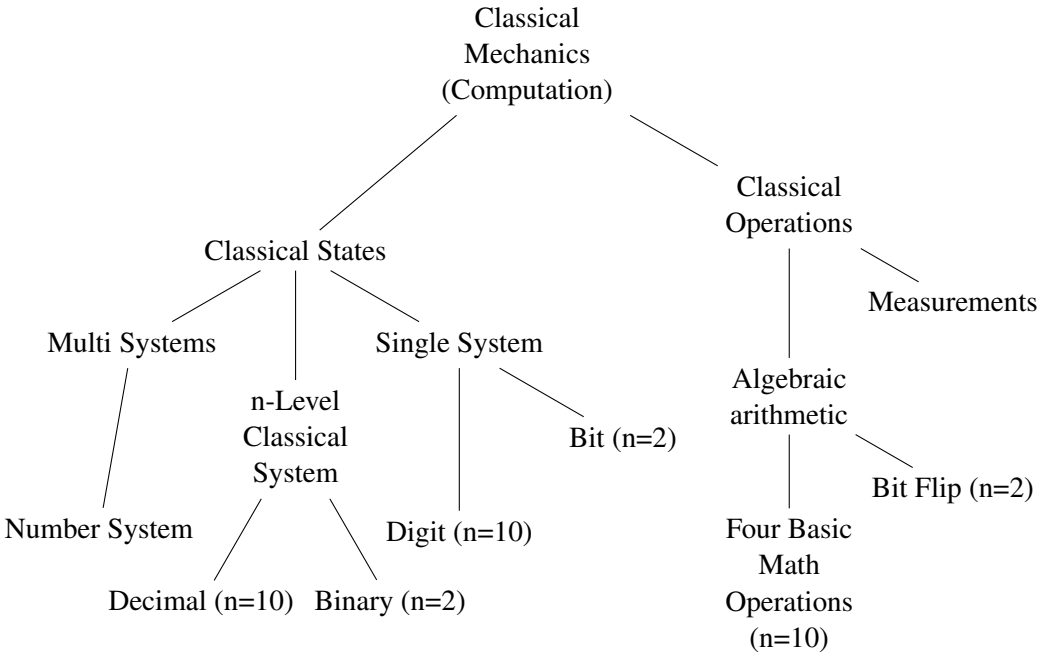

Figure 2.0.1: The basic structure of classical mechanics, in the computational context.

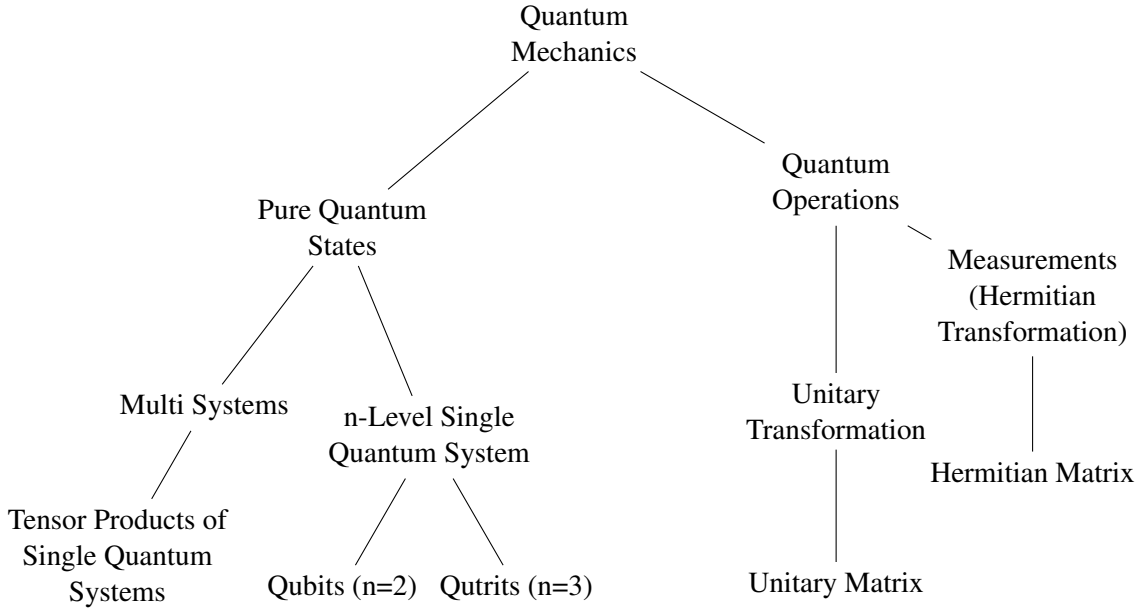

Figure 2.0.2: The basic structure of quantum mechanics, in the computational context.

2.1 Pure Quantum States

The simplest quantum system in quantum mechanics is a two-level quantum system, called a qubit, which is a quantum analogue of a classical bit. A classical bit $\varphi = \{0,1\}$, upon measurement, will reveal itself to be either 0 or 1. A general qubit $|\varphi\rangle$ is linear complex superposition of up state $|0\rangle$ and

down state $|1\rangle$ with complex coefficients α and β respectively, that is,

$$
\begin{aligned}
|\varphi\rangle &= \alpha|0\rangle + \beta|1\rangle && (2.1.1)\\
&| \quad \text{In vector notation from linear algebra,} && (2.1.2)\\
&= \alpha\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} && (2.1.3)
\end{aligned}
$$

We can also define the hermitian conjugate (dagger †: conjugate transpose) of the qubit as $\langle\varphi| = |\varphi\rangle^\dagger$, that is,

$$
\begin{aligned}
\langle\varphi| &= \alpha^*\langle 0| + \beta^*\langle 1| && (2.1.4)\\
&= \begin{pmatrix} \alpha^* & \beta^* \end{pmatrix} && (2.1.5)
\end{aligned}
$$

where $*$ represents complex conjugate. The qubit or vector pair $\{|0\rangle, |1\rangle\}$ is a 2D orthonormal basis set, called the computational basis. It spans the 2D Hilbert space $\mathscr{H}_2$, which is where the qubit is residing or is an element of. In general, there are infinite number of possible 2D orthonormal basis sets that spans $\mathscr{H}_2$. That is, any set $\{|x\rangle, |y\rangle\}$ can be an orthonormal basis set that spans $\mathscr{H}_2$ if and only if all the basis in the set satisfies the following orthogonality and normalization properties:

$$
\begin{aligned}
\text{Orthogonality}: \quad \langle x|y\rangle &= 0 \quad | \quad \text{where } |x\rangle \neq |y\rangle. && (2.1.6)\\
&| \quad \text{In vector notation from linear algebra,}\\
&| \quad \text{this usually means a dot product of two vectors.}\\
&| \quad \text{Let } x_i \text{ and } y_i \text{ be the } i^{th} \text{ element of vector } \vec{x} \text{ and } \vec{y} \text{ respectively.}\\
\sum_{i=1}^{2} x_i^* y_i &= 0 && (2.1.7)
\end{aligned}
$$

$$
\begin{aligned}
\text{Normalization:} \quad \langle x|x\rangle &= \langle y|y\rangle = 1 \quad | \quad \text{where } |x\rangle \neq |y\rangle. && (2.1.8)\\
&| \quad \text{In vector notation from linear algebra,}\\
\sum_{i=1}^{2} x_i^* x_i &= \sum_{i=1}^{2} y_i^* y_i = 1 && (2.1.9)
\end{aligned}
$$

Thus, any qubit $|\varphi\rangle$ that resides in $\mathscr{H}_2$, spanned by $\{|x\rangle, |y\rangle\}$, has the following normalization property:

$$
\begin{aligned}
\langle\varphi|\varphi\rangle &= [\alpha^*\langle x| + \beta^*\langle y|][\alpha|x\rangle + \beta|y\rangle] && (2.1.10)\\
&= |\alpha|^2\langle x|x\rangle + \alpha^*\beta\langle x|y\rangle + \beta^*\alpha\langle y|x\rangle + |\beta|^2\langle y|y\rangle && (2.1.11)\\
&| \quad \text{Using basis orthogonality and normalization properties}\\
&| \quad \text{from equation 2.1.6 and 2.1.8 respectively.}\\
&= |\alpha|^2 + |\beta|^2 = 1 && (2.1.12)
\end{aligned}
$$

According to one of the postulates of quantum mechanics, the Born's Rule, the squared amplitudes $|\alpha|^2$ and $|\beta|^2$ represent the probability of the qubit being in the $|x\rangle$ and $|y\rangle$ states respectively. That is $\text{Prob}(|x\rangle) = |\alpha|^2$ and $\text{Prob}(|y\rangle) = |\beta|^2$. As a result of equation 2.1.12, it must be that $\langle\varphi|\varphi\rangle =$

$\text{Prob}(|x\rangle) + \text{Prob}(|y\rangle) = 1$. Therefore, the normalization property of the qubit $|\varphi\rangle$ is interpreted as the unit sum of all probabilities of the states $|x\rangle, |y\rangle$ that qubit might be in [SN17].

2.2 Unitary Transformation and Operators

Classically, manipulating a bit involves a bit flip. This is an example of a classical operation that changes the value of the bit.

$$\varphi = \{0,1\} \quad \overset{\text{Bit Flip}}{\longrightarrow} \quad \{1,0\}$$

Quantum mechanically, manipulating a qubit involves a unitary transformation. It changes the probability of a qubit being in either orthogonal state $|x\rangle, |y\rangle$, while preserving the unit sum of the probabilities. A unitary transformation is physically represented as an unitary operator U. The defining property of a unitary U is that the inverse of U must be equal to the hermitian conjugate of U, that is $U^{-1} = U^{\dagger}$. For a general arbitrary 2D unitary operator U in the computational basis $|0\rangle, |1\rangle$, it can be written as

$$\begin{aligned} U(\theta,\phi,\lambda,\delta) &= e^{i\delta}\left[\cos\left(\frac{\theta}{2}\right)|0\rangle\langle 0| - e^{i\lambda}\sin\left(\frac{\theta}{2}\right)|0\rangle\langle 1| \right. \\ &\quad \left. + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle\langle 0| + e^{i(\phi+\lambda)}\sin\left(\frac{\theta}{2}\right)|1\rangle\langle 1|\right] \qquad (2.2.1) \\ &\mid \text{In matrix notation from linear algebra,} \\ &\mid \text{where} \quad \theta,\phi,\lambda,\delta \in [0,2\pi]. \\ &= e^{i\delta}\begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -e^{i\lambda}\sin\left(\frac{\theta}{2}\right) \\ e^{i\phi}\sin\left(\frac{\theta}{2}\right) & e^{i(\phi+\lambda)}\cos\left(\frac{\theta}{2}\right) \end{pmatrix} \qquad (2.2.2) \end{aligned}$$

The factor $e^{i\delta}$ of U is called a global phase factor that is a non-physical, non-observed property of the transformation. Therefore it is often ignored in calculations. Hence, dropping the phase factor $e^{i\delta}$, we have the general 2D unitary operator up to a global phase $U(\theta,\phi,\lambda)$,

$$U(\theta,\phi,\lambda) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -e^{i\lambda}\sin\left(\frac{\theta}{2}\right) \\ e^{i\phi}\sin\left(\frac{\theta}{2}\right) & e^{i(\phi+\lambda)}\cos\left(\frac{\theta}{2}\right) \end{pmatrix} \qquad (2.2.3)$$

When the unitary operator U is applied on the qubit $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$, the qubit transforms into another unique quantum qubit $|\varphi'\rangle = \gamma|0\rangle + \delta|1\rangle$ with different probabilities $|\gamma|^2$ and $|\delta|^2$, that is $|\varphi\rangle \overset{U}{\to} |\varphi'\rangle$.

Explicitly, it does the following,

$$
\begin{aligned}
& U|\varphi\rangle && (2.2.4)\\
= \; & \left[\cos\left(\frac{\theta}{2}\right)|0\rangle\langle 0| - e^{i\lambda}\sin\left(\frac{\theta}{2}\right)|0\rangle\langle 1| \right. \\
& \left. + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle\langle 0| + e^{i(\phi+\lambda)}\cos\left(\frac{\theta}{2}\right)|1\rangle\langle 1|\right][\alpha|0\rangle + \beta|1\rangle] && (2.2.5)\\
| \; & \text{Using the orthonormalisation properties of the computational basis.} \\
= \; & \alpha\cos\left(\frac{\theta}{2}\right)|0\rangle\langle 0|0\rangle - \alpha e^{i\lambda}\sin\left(\frac{\theta}{2}\right)\cancelto{0}{|0\rangle\langle 1|0\rangle} \\
& + \alpha e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle\langle 0|0\rangle + \alpha e^{i(\phi+\lambda)}\cos\left(\frac{\theta}{2}\right)\cancelto{0}{|1\rangle\langle 1|0\rangle} \\
& + \beta\cos\left(\frac{\theta}{2}\right)\cancelto{0}{|0\rangle\langle 0|1\rangle} - \beta e^{i\lambda}\sin\left(\frac{\theta}{2}\right)|0\rangle\langle 1|1\rangle \\
& + \beta e^{i\phi}\sin\left(\frac{\theta}{2}\right)\cancelto{0}{|1\rangle\langle 0|1\rangle} + \beta e^{i(\phi+\lambda)}\cos\left(\frac{\theta}{2}\right)|1\rangle\langle 1|1\rangle && (2.2.6)\\
= \; & \left[\alpha\cos\left(\frac{\theta}{2}\right) - \beta e^{i\lambda}\sin\left(\frac{\theta}{2}\right)\right]|0\rangle + \left[\alpha e^{i\phi}\sin\left(\frac{\theta}{2}\right) + \beta e^{i(\phi+\lambda)}\cos\left(\frac{\theta}{2}\right)\right]|1\rangle && (2.2.7)\\
| \; & \text{In matrix notation from linear algebra,} \\
= \; & \begin{pmatrix} \alpha\cos\left(\frac{\theta}{2}\right) - \beta e^{i\lambda}\sin\left(\frac{\theta}{2}\right) \\ \alpha e^{i\phi}\sin\left(\frac{\theta}{2}\right) + \beta e^{i(\phi+\lambda)}\cos\left(\frac{\theta}{2}\right) \end{pmatrix} \overset{\text{Let this be}}{=} \begin{pmatrix} \gamma \\ \delta \end{pmatrix} && (2.2.8)\\
| \; & \text{Note that } |\gamma| + |\delta|^2 = 1. \\
= \; & \gamma|0\rangle + \delta|1\rangle = |\varphi'\rangle && (2.2.9)
\end{aligned}
$$

2.3 Measurements and Hermitian Operators

Classical measurement is another classical operation and measuring an unknown classical bit is trivial. Suppose a measurement on an unknown bit φ, that has outcomes 1 and -1 for bit states 0 and 1 respectively, is conducted. Then, two things will happen: the revelation of the measurement outcomes and the final bit state.

Final Bit State	Measurement Outcomes
$\varphi = 0$	1
$\varphi = 1$	-1

Table 2.3.1: Classical post-measurement result of an unknown bit φ.

Similarly, a quantum measurement is another kind of a quantum operation and the measurement of a qubit involves a 2D hermitian operator M that has the defining property that the hermitian conjugate of M is equal to itself, $M^\dagger = M$. As a consequence of the hermiticity, the eigenvalues of M are always real, called it m, and they are interpreted as the measurement outcomes. However, unlike classical measurements, the post-measured qubit will permanently transform into one of the ortho-normalized

eigenstates of M, called it $|m\rangle$. This quantum transformation is commonly phrased as the *collapse of a measured quantum state*, because the pre-measured qubit cannot be recovered back by any inverse hermitian transformation on the measured qubit.

To see this collapse explicitly, let m_1, m_2 be eigenvalues that corresponds to $|m_1\rangle, |m_2\rangle$ the ortho-normalized eigenstates of M, where $m_1 \neq m_2$ and $|m_1\rangle \neq |m_2\rangle$. Decompose the qubit $|\varphi\rangle$ into linear combinations of the ortho-normalized eigenstates $|m_1\rangle, |m_2\rangle$ with corresponding complex coefficients a, b as follows,

$$\begin{aligned} |\varphi\rangle &= a|m_1\rangle + b|m_2\rangle && (2.3.1) \\ &| \quad \text{Apply measurement operator } M. \\ M|\varphi\rangle &= aM|m_1\rangle + bM|m_2\rangle && (2.3.2) \\ &| \quad \text{Using eigenequation } M|m_j\rangle = m_j|m_j\rangle. \\ &= am_1|m_1\rangle + bm_2|m_2\rangle && (2.3.3) \end{aligned}$$

Note that since $|am_1|^2 + |bm_2|^2 \neq 1$ in general, $M|\varphi\rangle$ is not a valid qubit or a quantum state as it does not satisfy normalization property in equation 2.1.12. By the measurement postulate of quantum mechanics [SN17], such a state $M|\varphi\rangle$ will therefore collapse to either eigenstate $|m_1\rangle$ or $|m_2\rangle$, with probability $|a|^2$, $|b|^2$ and with measurement outcomes m_1, m_2 respectively.

Final Qubit State	Measurement Outcomes	Probability
$\|m_1\rangle$	m_1	$\|a\|^2$
$\|m_2\rangle$	m_2	$\|b\|^2$

Table 2.3.2: Quantum post-measurement of a generic qubit $|\varphi\rangle = a|m_1\rangle + b|m_2\rangle$ with measurement operator M.

This property of quantum measurement has far reaching consequences. Unlike a classical measurement, quantum measurement does not reveal any information about the pre-measured qubit. Therefore, a single measurement on an unknown qubit is insufficient to gain knowledge of the pre-measured qubit. Worse, repeated measurements on the same qubit would not yield any meaningful results, as it would lead to the same outcomes as revealed in the first measurement. One workaround is to ensemble a collection of equivalent unknown qubits and perform repeated single measurements on these unknown qubits individually. By doing a statistical count of the measurement outcomes, the probabilities $|a|^2$, $|b|^2$ can be calculated and thus be able to gain some, though not complete, information about the pre-measured qubit using equation 2.3.1.

Often, one is usually interested in finding out the average value of the measurement outcomes given an unknown qubit $|\varphi\rangle$, called expectation value of the measurement operator M with respect to the qubit $|\varphi\rangle$, symbolically represented as $\langle M\rangle = \langle\varphi|M|\varphi\rangle$. We can show this by theoretically calculating the

expectation value by applying a hermitian conjugate quantum state $\langle\varphi| = |\varphi\rangle^\dagger$ to $M|\varphi\rangle$.

$$\begin{aligned}
\langle\varphi|M|\varphi\rangle &= [a^*\langle m_1| + b^*\langle m_2|][am_1|m_1\rangle + bm_2|m_2\rangle] && (2.3.4)\\
&= |a|^2 m_1\langle m_1|m_1\rangle + a^*bm_2\langle m_1|m_2\rangle + b^*am_1\langle m_2|m_1\rangle + |b|^2 m_2\langle m_2|m_2\rangle && (2.3.5)\\
&| \quad \text{Apply orthonormality } \langle m_i|m_j\rangle = \delta_{ij}.\\
&= |a|^2 m_1 + |b|^2 m_2 && (2.3.6)\\
&= \text{Prob}(m_1)\cdot m_1 + \text{Prob}(m_2)\cdot m_2 && (2.3.7)\\
&| \quad \text{By statistics, this is the definition of average using probablities.}\\
&= \langle M\rangle && (2.3.8)
\end{aligned}$$

The expectation value is always a value between the measurement outcomes, $m_1 \leq \langle M\rangle \leq m_2$, if $m_1 < m_2$. Hence, it is a useful quantity that can be used as a rough gauge to determine how 'close' is the unknown qubit to either of the ortho-normalized eigenvectors.

2.4 Multiple Quantum Systems and Operations

To represent more information classically, two or more bits are needed and ordered in a pre-defined way such as a binary system. Quantum mechanically, the act of putting two or more qubits together mathematically represents a linear tensor product operation $\otimes$, a symbol that is usually omitted for clarity. Consequently, quantum information can be represented as tensor products of multiple qubits. Let $|\varphi_1\rangle$ and $|\varphi_2\rangle$ be two qubits, then,

$$\begin{aligned}
|\varphi_1\rangle\otimes|\varphi_2\rangle &= |\varphi_1\rangle|\varphi_2\rangle \quad | \quad \text{omitted } \otimes \text{ for clarity.} && (2.4.1)\\
&| \quad \text{Using orthonormal basis set } \{|0\rangle, |1\rangle\}.\\
&= [\alpha_1|0_1\rangle + \beta_1|1_1\rangle][\alpha_2|0_2\rangle + \beta_2|1_2\rangle] && (2.4.2)\\
&| \quad \text{Dropping all indices in } |\,\rangle \text{ and combining them.}\\
&= \alpha_1\alpha_2|00\rangle + \alpha_1\beta_2|01\rangle + \beta_1\alpha_2|10\rangle + \beta_1\beta_2|10\rangle && (2.4.3)
\end{aligned}$$

Similarly, multi-qubit unitary transformation and measurements can be constructed from tensor products of the single qubit versions. For example, applying a multi-qubit unitary operation $A\otimes B$ on $|\varphi_1\rangle\otimes|\varphi_2\rangle$ where A, B are single qubit unitary operators yields,

$$\begin{aligned}
(A\otimes B)|\varphi_1\rangle\otimes|\varphi_2\rangle &= A|\varphi_1\rangle B|\varphi_2\rangle && (2.4.4)\\
&= \alpha_1\alpha_2 A|0\rangle B|0\rangle + \alpha_1\beta_2 A|0\rangle B|1\rangle\\
&\quad + \beta_1\alpha_2 A|1\rangle B|0\rangle + \beta_1\beta_2 A|1\rangle B|1\rangle && (2.4.5)
\end{aligned}$$

However, tensor products of the single qubit quantum operations are just a subset of the possible multi-qubit operations. There are multi-qubits operation that cannot be represented using tensor products of single qubit quantum operators. we shall investigate more multi-qubits operations the later sections.

Chapter 3

Quantum Computation

Classical computing uses a string of bits s and manipulates it by doing an ordered sequence of bit flips before measurement to get an output s'. Similarly, in quantum computing, a set of qubits $|s\rangle$, called a qubit register, is initialized and unitary transformed by unitary operators to output a quantum state $|s'\rangle$. A measurement is then performed on the output state $|s'\rangle$ by a hermitian operator M to get the measurement outcomes m and the collapsed quantum state $|m\rangle$. A simple diagram showing the flow of information for each kind of computing is given in figure 3.0.1.

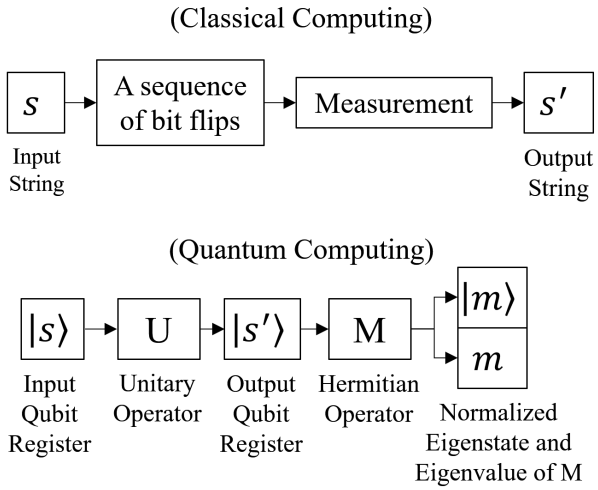

Figure 3.0.1: A basic comparison of the flow of classical and quantum information.

To formalize quantum computations using qubits, models of quantum computations are created to help describe the organization of quantum operations and information so that the performance of a quantum algorithm running on these models can be analysed and evaluated. The standard model of organizing quantum operations is the quantum circuits model as an quantum analogue of classical circuit gate model.

3.1 Quantum Circuits Model

Quantum Circuits is a standard model of quantum computation. Quantum circuits are diagrammatically represented a one way directed set of wires with no loops, cycles, join ins, or fan outs. The wires are laid out horizontally and ordered vertically. Each wire represents a single qubit or a qubit register, depending

on context. At the beginning of the quantum circuit, on the left, qubits are prepared and initialized at their respective positions. Quantum and measurement gates, representing unitary operations and measurements, are placed in the middle and at the end of a quantum circuit on the right, respectively. For example, figure 3.1.1 shows a quantum circuit with all qubits initialised in $|0\rangle$, then a series of quantum gates (unitary operators) are applied. Finally, multiple measurements (hermitian operators) are performed, in the computation basis $\{|0\rangle, |1\rangle\}$, on the respective qubits to obtain multiple measurement outcomes.

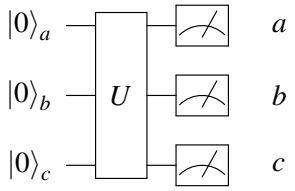

Figure 3.1.1: An generic quantum circuit diagram with 3 initialized qubits $|0\rangle_a |0\rangle_b |0\rangle_c$ on the left, a multi-qubit unitary quantum gate U in the middle and a set of measurements on the qubits, represented by a meter symbol on the right.

3.2 Qubit Binary Representation of Quantum Information

Conventionally, classical information is represented in binary bits. Analogously, quantum information is also represented in binary using the tensor product formalism of qubits. Recall, given a classical real decimal integer x, its n-binary representation can be written as,

$$\begin{aligned} x &= x_n \cdot 2^{n-1} + \ldots + x_j \cdot 2^{j-1} + \ldots + x_2 \cdot 2^1 + x_1 \cdot 2^0 && (3.2.1) \\ &\Rightarrow x_n \ldots x_2 x_1 && (3.2.2) \end{aligned}$$

where $n = 1 + \lfloor \log_2 x \rfloor$, $x_j \in \{0,1\}$, $j \in [1,n]$. Therefore, a quantum state $|x\rangle$ can be represented as

$$\begin{aligned} |x\rangle &\Rightarrow |x_n \ldots x_2 x_1\rangle && (3.2.3) \\ &= |x_n\rangle \otimes \ldots \otimes \left|x_j\right\rangle \otimes \ldots \otimes |x_2\rangle \otimes |x_1\rangle && (3.2.4) \end{aligned}$$

For example, consider a decimal real integer 23 and the analogous quantum state $|23\rangle$,

$$\begin{aligned} 23 &= 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 && (3.2.5) \\ &\Rightarrow 10111 && (3.2.6) \\ &\mid \text{Note: } n = 1 + \lfloor \log_2 23 \rfloor = 5. && (3.2.7) \\ |23\rangle &\Rightarrow |10111\rangle && (3.2.8) \end{aligned}$$

Such quantum information can be represented on a quantum circuit as a qubit register $|x\rangle$ in binary shown in figure 3.2.1.

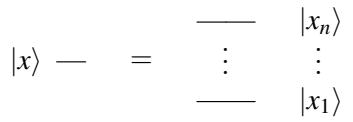

Figure 3.2.1: A qubit register $|x\rangle$ represented as a collection of qubits arranged in a quantum circuit model from top to bottom.

3.3 Quantum Gates

Quantum gates are quantum operators in the quantum circuits model. The following subsections will summarise the common set of single qubit and multi quantum gates and measurement gates.

3.3.1 Single Qubit Gates

Common Gates

There are many single qubit quantum gates which are derivatives of the general arbitrary 2D unitary operator U in equation 2.2.3. The common set of single qubit quantum gates are: Pauli-Z, Pauli-Z, Pauli-X, Pauli-Y, Hadamard H, Phase S and $\frac{\pi}{8}$-gate[1] T. A summary of operations on a general qubit $|\varphi\rangle$ is given in the equation set 3.3.1 and its symbols are shown in figure 3.3.1.

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle \begin{cases} \xrightarrow{Z} & \alpha|0\rangle - \beta|1\rangle \\ \xrightarrow{X} & \beta|0\rangle + \alpha|1\rangle \\ \xrightarrow{Y} & i(-\beta|0\rangle + \alpha|1\rangle) \\ \xrightarrow{H} & \alpha\frac{|0\rangle+|1\rangle}{\sqrt{2}} + \beta\frac{|0\rangle-|1\rangle}{\sqrt{2}} \\ \xrightarrow{S} & \alpha|0\rangle + \beta i|1\rangle \\ \xrightarrow{T} & e^{i\pi/8}\left(\alpha e^{-i\pi/8}|0\rangle + \beta e^{i\pi/8}|1\rangle\right) \end{cases} \tag{3.3.1}$$

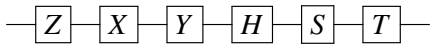

Figure 3.3.1: Common gates symbols, from left to right: Pauli-Z, Pauli-X, Pauli-Y, Hadamard H, Phase S and $\frac{\pi}{8}$ T.

Rotation Gates

Rotation gates are parametrized quantum gates with an angular parameter θ. A summary of these operations on a general qubit $|\varphi\rangle$ is given in the equation set 3.3.2 and its quantum circuit representation are shown in figure 3.3.2. Note that $\mathbb{I}$ represents an identity operation, $\mathbb{I}|\varphi\rangle = |\varphi\rangle$.

$$\begin{aligned} R_x(\theta) &= e^{-i\theta X/2} = cos(\theta/2)\mathbb{I} - isin(\theta/2)X \quad \text{(Rotate-X)} \\ R_y(\theta) &= e^{-i\theta Y/2} = cos(\theta/2)\mathbb{I} - isin(\theta/2)Y \quad \text{(Rotate-Y)} \\ R_z(\theta) &= e^{-i\theta Z/2} = cos(\theta/2)\mathbb{I} - isin(\theta/2)Z \quad \text{(Rotate-Z)} \end{aligned} \tag{3.3.2}$$

[1] An unfortunate naming of the square root of phase gate S [NC10].

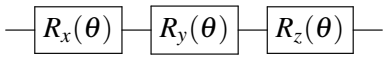

Figure 3.3.2: Rotation gates symbols, from left to right: Rotate-X, $R_x(\theta)$, Rotate-Y, $R_y(\theta)$, Rotate-Z, $R_z(\theta)$.

3.3.2 Two Qubit Gates

Swap Gate

Swap gate is a quantum gate that swaps the position of two qubits. Suppose the input qubit register $|x,y\rangle$ is acted upon by a swap gate S_{wap}, then the resulting qubit register is $|y,x\rangle$. This qubit swap operation and the swap gate symbol are respectively summarised in equation 3.3.3 and figure 3.3.3 below.

$$|x,y\rangle \xrightarrow{S_{wap}} |y,x\rangle \tag{3.3.3}$$

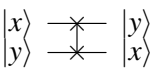

Figure 3.3.3: Two qubit gate: Swap gate S_{wap}.

CNOT and Controlled-Z Gates

Controlled gates are multi-qubit quantum gates that apply a unitary operation on a set of target qubits on the condition of the control qubit. The most simplest controlled gates are gates that apply a quantum gate on a single target qubit on the condition of a single control qubit. Two most common controlled gates are CNOT gate C_x and controlled-Z gate C_z. CNOT gate applies a Pauli-X gate on a single target qubit on the condition of the control qubit being $|1\rangle$. Controlled-Z gate is similar to a CNOT, but with a difference that it applies a Pauli-Z gate instead. Both controlled operations and the gate symbols are respectively summarised in equations 3.3.4, 3.3.5 and figure 3.3.4 below.

$$|x,y\rangle \xrightarrow{C_x} |x,y\oplus x\rangle \quad | \quad \text{(CNOT)} \tag{3.3.4}$$

$$|x,y\rangle \xrightarrow{C_z} (-1)^{xy}|x,y\rangle \quad | \quad \text{(Controlled-Z)} \tag{3.3.5}$$

$$| \quad \text{Note: } \oplus \text{ modulo 2}$$

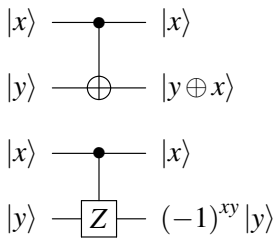

Figure 3.3.4: Single target single control controlled gates, from top to bottom: CNOT, C_x, Controlled-Z, C_z.

Note that there are controlled gates that apply an unitary operation U on the condition of the control qubit being $|0\rangle$. It can be decomposed using standard controlled gates with condition of the control qubit being $|1\rangle$ and Pauli-X gates as shown in the figure below 3.3.5.

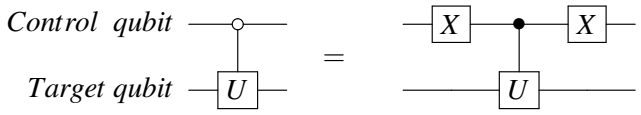

Figure 3.3.5: Quantum circuit representation of the controlled quantum gate U on the condition of the control qubit being $|0\rangle$ and its decomposition.

3.3.3 Three Qubit Gates

Toffoli and Fredkin Gates

Toffoli gate is a controlled-CNOT gate CC_x while Fredkin gate is a controlled-swap gate C_{swap}. The Toffoli CC_x and Fredkin C_{swap} gates apply a CNOT C_x and a Swap gate S_{wap} respectively on the target qubits on the condition of the control qubits being $|1\rangle$. Both controlled operations and the gate symbols are respectively summarised in equations 3.3.6, 3.3.7 and figure 3.3.6 below.

$$|x,y,z\rangle \xrightarrow{CC_x} |x,y,z\oplus(x\wedge y)\rangle \quad | \quad \text{(Toffoli)} \tag{3.3.6}$$

$$|x,y,z\rangle \xrightarrow{C_{swap}} \begin{cases} |0,y,z\rangle & x=0 \\ |1,z,y\rangle & x=1 \end{cases} \quad | \quad \text{(Fredkin)} \tag{3.3.7}$$

$$| \quad \text{Note: } \wedge \text{ Boolean AND}$$

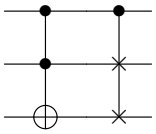

Figure 3.3.6: Controlled gates: from left to right, Toffoli gate, CC_x, Fredkin gate, C_{swap}.

3.4 Quantum Gate Decompositions Techniques

The quantum circuits model allows the ease of organizing quantum gates which greatly contributes to the development of several important quantum gates decomposition techniques. This allows users to decompose a complex multi-qubit quantum gates into simpler constructions using single or two qubit quantum gates, for ease of implementation. This section attempts to introduce a few basic quantum gate decomposition techniques, interested readers may wish to consult these textbooks and papers [Bar+95; NC10; SBM04; MV05] for more advance gate decompositions and proofs.

3.4.1 ABC Decomposition

ABC decomposition is a technique to decompose an arbitrary 2D unitary gate into products of gates from the common and rotation gates set. Notice that the arbitrary 2D unitary gate from equation 2.2.2 can be decomposed in terms of Rotate-Z $R_Z(\theta)$ and Rotate-Y $R_y(\theta)$ up to a global phase of $\exp\left(\frac{i(\lambda+\phi)}{2}\right)$ [Bar+95]. Then, by further breaking it up, yields the ABC decomposition of an arbitrary 2D unitary gate

as shown in figure 3.4.1.

$$
\begin{aligned}
U &= e^{i\delta}\begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -e^{i\lambda}\sin\left(\frac{\theta}{2}\right) \\ e^{i\phi}\sin\left(\frac{\theta}{2}\right) & e^{i(\phi+\lambda)}\cos\left(\frac{\theta}{2}\right) \end{pmatrix} && (3.4.1)\\
&= \exp\left(i\delta + \frac{i(\lambda+\phi)}{2}\right) R_z(\phi) R_y(\theta) R_z(\lambda) && (3.4.2)\\
&\quad | \quad \text{Let } \varepsilon = \delta + \frac{(\lambda+\phi)}{2} \\
&\quad | \quad A = R_z(\phi) R_y(\theta/2), \\
&\quad | \quad B = R_y(-\theta/2) R_z(-(\lambda-\phi)/2), \\
&\quad | \quad C = R_z((\lambda-\phi)/2). \\
&= \exp(i\varepsilon) AXBXC && (3.4.3)
\end{aligned}
$$

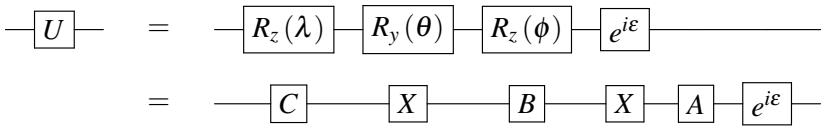

Figure 3.4.1: ABC decomposition of an arbitrary 2D unitary gate U.

Using the above ABC decomposition of an arbitrary 2D unitary gate, it is possible to make a single control qubit version of as shown in figure 3.4.2 below,

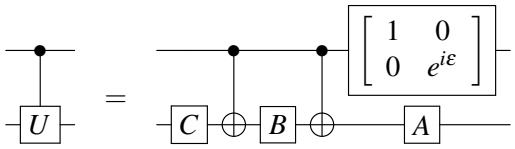

Figure 3.4.2: ABC decomposition of a controlled-U gate $C(U)$.

3.4.2 Single-Target Multi-Control Gate Decomposition

According to the quantum computing textbook [NC10], a n-control controlled arbitrary quantum gate U can be decomposed using $2(n-1)$ Toffoli gates, a single control quantum gate with an additional work qubit register with $n-1$ qubits, as shown in the figure 3.4.3 below. This decomposition is important and it will be frequently used without mention in this project when the actual quantum circuit for the quantum algorithm of interest is being built.

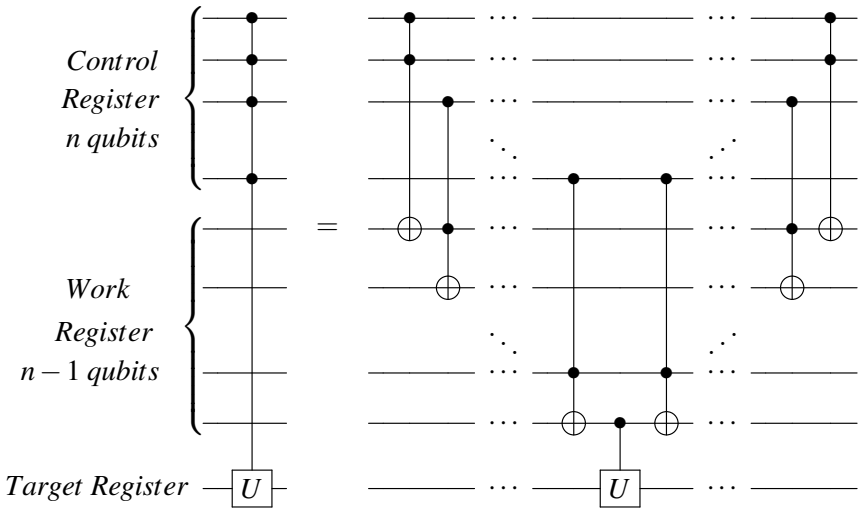

Figure 3.4.3: The quantum circuit of a n-control controlled arbitrary quantum gate U and its decomposition with the use of a work qubit register, that is to be discarded after computation.

3.5 Measurement Gates

Pauli quantum gates, introduced in subsection 3.3.1, are special gates as they are both unitary and hermitian. Therefore, Pauli gates can act either as unitary gates or measurement gates. Typically, in the quantum circuit model, measurements are done in the computational basis $\{|0\rangle, |1\rangle\}$ which uses Pauli-Z gates. The other two pauli gates, Pauli-X and Pauli-Y are rarely used in quantum computation as measurements as both measure in a different basis.

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad , \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad , \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \tag{3.5.1}$$

The Pauli gates have the following eigenvectors, eigenbras and eigenvalues.

Eigen	Vectors 1	Bra 1	Vectors 2	Bra 2
Z	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\|0\rangle$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\|1\rangle$
X	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$\|+\rangle$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$	$\|-\rangle$
Y	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ i \end{pmatrix}$	$\|\uparrow_y\rangle$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -i \end{pmatrix}$	$\|\downarrow_y\rangle$
Values	1		-1	

(3.5.2)

Chapter 4

Quantum Algorithms and Subroutines

Classical algorithms are sets of instructions that consist one or more classical operations acting on sets of classical information. Similarly, quantum algorithms are sequences of quantum operations applied on sets of quantum information. It may consist of several quantum subroutines which may have superior runtime complexity than its classical counterparts, thereby achieving a overall speedup in computation time. Two common quantum subroutines are Amplitude Amplification (AA), Quantum Fourier Transform (QFT). Each of these quantum subroutines serves as the backbone of the famous and popular quantum algorithms such as Grover's search algorithm [Gro96] and Shor's factoring algorithm [Sho94].

4.1 Amplitude Amplification

Amplitude amplification (AA) [Bra+00] is a quantum subroutine applied in Grover's search quantum algorithm [Gro96] that aims to increase the amplitude of one particular orthonormal basis while decreasing the rest in an arbitrary quantum state. Suppose given an arbitrary qubit given in linear combinations of orthonormal basis $\{|x\rangle, |y\rangle\}$ with real coefficients α, β respectively, where the amplitude of $|x\rangle$ basis is the target for amplification. Applying AA will yield a new qubit $|\varphi'\rangle$ whose where the new real coefficients γ, δ will be such that $\gamma \gg \delta$.

$$|\varphi\rangle = \alpha|x\rangle + \beta|y\rangle \quad | \quad \text{for some } \alpha, \beta \in \mathbb{R} \tag{4.1.1}$$

$$\xrightarrow{\text{AA}} \gamma|x\rangle + \delta|y\rangle \quad | \quad \text{where } \gamma \gg \delta \in \mathbb{R} \tag{4.1.2}$$

Amplitude amplification will not be not used in this project, any further analysis shall be postponed to section 11.1 on page 94 on the possible extensions of this project for future research. Interested readers may consult the relevant chapters in quantum computing textbook [NC10] for more detailed information.

4.2 Quantum Fourier Transform

The Quantum Fourier Transform (QFT) is simply a quantum analogue of the classical discrete Fourier transform. Suppose a 2^nD orthonormal basis $|x\rangle$ from the set $\{|x\rangle : x = 0, \ldots, 2^n - 1\}$, applying QFT on

the basis $|x\rangle$ yields a normalised quantum superposition of $\exp\left(\frac{2\pi ixy}{2^n}\right)|y\rangle$ states for $y = 0, \ldots, 2^n - 1$.

$$|x\rangle \quad \xrightarrow{QFT} \quad \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} \exp\left(\frac{2\pi ixy}{2^n}\right)|y\rangle \tag{4.2.1}$$

To implement QFT using quantum circuits, we need to know the binary representation of the QFT of $|x\rangle$. First, recall the binary notation x and express it in terms of coefficients of the binary bits x_l, that is $x = \sum_{l=1}^{n} x_l \cdot 2^{l-1} \Rightarrow x_n \ldots x_j \ldots x_2 x_1$. Consider dividing x by 2^k for some positive integer $k \leq n$, then $\frac{x}{2^k} = \sum_{l=1}^{n} x_l \cdot 2^{l-k-1} \Rightarrow x_n x_{n-1} \ldots x_{k+1}.x_k \ldots x_2 x_1$. If $k = n$, then the binary notation of $\frac{x}{2^n} = \sum_{l=1}^{n} x_l \cdot 2^{l-n-1} \Rightarrow 0.x_n x_{n-1} \ldots x_1$. With the binary representations of x, $\frac{x}{2^k}$ and $\frac{x}{2^n}$ defined, we can now express $QFT\,|x\rangle$ in qubit binary representation,

$$\begin{aligned}
QFT\,|x\rangle &= \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} \exp\left(\frac{2\pi ixy}{2^n}\right)|y\rangle && (4.2.2)\\
&\mid \text{Substitute } \frac{y}{2^n} = \sum_{l=1}^{n} y_l \cdot 2^{l-n-1} \\
&\mid \text{and express } \sum_{y=0}^{2^n-1} |y\rangle = \sum_{y_n=0}^{1} \ldots \sum_{y_1=0}^{1} |y_n \ldots y_2 y_1\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{y_n=0}^{1} \ldots \sum_{y_1=0}^{1} \exp\left[2\pi ix\left(\sum_{l=1}^{n} y_l 2^{l-n-1}\right)\right]|y_n \ldots y_2 y_1\rangle && (4.2.3)\\
&\mid \text{Expand in terms of tensor products.} \\
&= \frac{1}{\sqrt{2^n}} \sum_{y_n=0}^{1} \ldots \sum_{y_1=0}^{1} \bigotimes_{l=1}^{n} \exp\left[2\pi ixy_l 2^{l-n-1}\right]|y_l\rangle && (4.2.4)\\
&\mid \text{Since tensor product is an linear operation.} \\
&= \frac{1}{\sqrt{2^n}} \bigotimes_{l=1}^{n} \left[\sum_{y_l=0}^{1} \exp\left(2\pi ixy_l 2^{l-n-1}\right)|y_l\rangle\right] && (4.2.5)\\
&= \frac{1}{\sqrt{2^n}} \bigotimes_{l=1}^{n} \left[|0\rangle + \exp\left(2\pi ix2^{l-n-1}\right)|1\rangle\right] && (4.2.6)\\
&= \frac{1}{\sqrt{2^n}} \left[|0\rangle + \exp\left(2\pi ix2^{-n}\right)|1\rangle\right] \otimes \left[|0\rangle + \exp\left(2\pi ix2^{1-n}\right)|1\rangle\right] \otimes \ldots \\
&\quad \ldots \otimes \left[|0\rangle + \exp\left(2\pi ix2^{-2}\right)|1\rangle\right] \otimes \left[|0\rangle + \exp\left(2\pi ix2^{-1}\right)|1\rangle\right] && (4.2.7)\\
&\mid \text{Use binary notation of } \frac{x}{2^k}. \\
QFT\,|x_n \ldots x_2 x_1\rangle &= \frac{1}{\sqrt{2^n}} \left[|0\rangle + \exp\left(2\pi i0.x_n \ldots x_2 x_1\right)|1\rangle\right] \otimes \left[|0\rangle + \exp\left(2\pi i0.x_{n-1} \ldots x_2 x_1\right)|1\rangle\right] \otimes \\
&\quad \ldots \otimes \left[|0\rangle + \exp\left(2\pi i0.x_2 x_1\right)|1\rangle\right] \otimes \left[|0\rangle + \exp\left(2\pi i0.x_1\right)|1\rangle\right] && (4.2.8)
\end{aligned}$$

As shown in equation 4.2.8, the idea behind the QFT algorithm subroutine is to transform the quantum state initial state $|x_n \ldots x_2 x_1\rangle$ to the desired state qubit by qubit, starting from first qubit $|x_n\rangle$. That is to transform $|x_n\rangle$ to $\frac{1}{\sqrt{2}}\left[|0\rangle + \exp\left(2\pi i0.x_n \ldots x_2 x_1\right)|1\rangle\right]$, then the second qubit $|x_{n-1}\rangle$ to $\frac{1}{\sqrt{2}}\left[|0\rangle + \exp\left(2\pi i0.x_{n-1} \ldots x_2 x_1\right)|1\rangle\right]$ so on and so forth till the last qubit $|x_1\rangle$ to $\frac{1}{\sqrt{2}}\left[|0\rangle + \exp\left(2\pi i0.x_1\right)|1\rangle\right]$. To transform the first qubit $|x_n\rangle$, consider applying a Hadamard gate H to

it,

$$H|x_n\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^{x_n}|1\rangle\right) \tag{4.2.9}$$

$$= \frac{1}{\sqrt{2}}\left(|0\rangle + \exp\left(\frac{2\pi i x_n}{2}\right)|1\rangle\right) \tag{4.2.10}$$

$$= \frac{1}{\sqrt{2}}\left[|0\rangle + \exp\left(2\pi i 0.x_n\right)|1\rangle\right] \tag{4.2.11}$$

Next, apply controlled-R_2 on $H|x_n\rangle$, conditioned on the second qubit $|x_{n-1}\rangle$, where $R_k = \begin{bmatrix} 1 & 0 \\ 0 & \exp\left(2\pi i 2^{-k}\right) \end{bmatrix}$,

$$H|x_n\rangle|x_{n-1}\rangle = \frac{1}{\sqrt{2}}\left[|0\rangle + \exp\left(2\pi i 0.x_n\right)|1\rangle\right]|x_{n-1}\rangle \tag{4.2.12}$$

$$\xrightarrow{C(R_2)} \frac{1}{\sqrt{2}}\left[|0\rangle + \exp\left(2\pi i 0.x_n\right)\exp\left(2\pi i x_{n-1}2^{-2}\right)|1\rangle\right]|x_{n-1}\rangle \tag{4.2.13}$$

$$= \frac{1}{\sqrt{2}}\left[|0\rangle + \exp\left(2\pi i 0.x_n x_{n-1}\right)|1\rangle\right]|x_{n-1}\rangle \tag{4.2.14}$$

Apply further repeated applications of controlled-R_k on transformed first qubit $\frac{1}{\sqrt{2}}\left[|0\rangle + \exp\left(2\pi i 0.x_n x_{n-1}\right)|1\rangle\right]$, conditioned on the subsequent qubits $|x_{n-k}\rangle$, for $k = 2$ to $n-1$ to yield the first desired qubit,

$$\underbrace{\frac{1}{\sqrt{2}}\left[|0\rangle + \exp\left(2\pi i 0.x_n x_{n-1}\ldots x_1\right)|1\rangle\right]}_{\text{first desired qubit}}\underbrace{|x_{n-1}\ldots x_1\rangle}_{\text{Rest of the qubits}} \tag{4.2.15}$$

To transform the second qubit $|x_{n-1}\rangle$, apply a Hadamard gate H and then repeated applications of controlled-R_k on the second qubit $|x_{n-1}\rangle$, conditioned on the subsequent qubits $|x_{n-k}\rangle$ for $k = 2$ to $n-1$. This will yield the second desired qubit,

$$\underbrace{\frac{1}{\sqrt{2}}\left[|0\rangle + \exp\left(2\pi i 0.x_n x_{n-1}\ldots x_1\right)|1\rangle\right]}_{\text{first desired qubit}}\underbrace{\frac{1}{\sqrt{2}}\left[|0\rangle + \exp\left(2\pi i 0.x_{n-1}\ldots x_1\right)|1\rangle\right]}_{\text{second desired qubit}}\underbrace{|x_{n-2}\ldots x_1\rangle}_{\text{Rest of the qubits}} \tag{4.2.16}$$

To obtain the rest transformed qubits, repeat the above steps for the rest of the qubits $|x_{n-j}\rangle$ for $j = 2$ to $n-1$ by applying a Hadamard gate and then repeated applications of controlled-R_k on qubit $|x_{n-j}\rangle$, conditioned on the subsequent qubits $|x_{n-k}\rangle$ for $k = j+1$ to $n-1$. Collectively, the transformed qubits result in a qubit binary representation of the QFT of $|x\rangle$ as given in equation line 4.2.8. The QFT quantum circuit is illustrated in figure 4.2.1 below. Note that the QFT algorithm subroutine given here is slightly different from the Nielsen and Chuang's textbook [NC10] due to the difference in the binary representation of numbers.

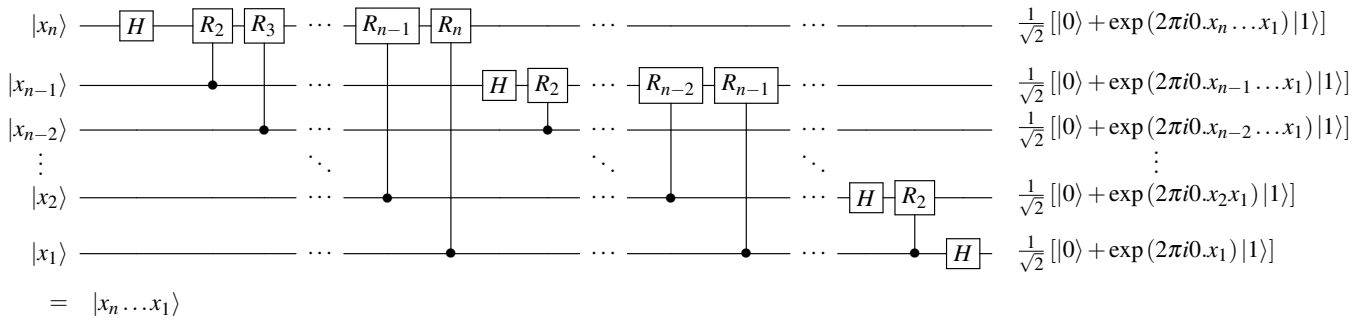

Figure 4.2.1: Quantum circuit for QFT acting on $|x_n \ldots x_1\rangle$.

4.3 Quantum Phase Estimation

Quantum Phase Estimation (QPE) is an important quantum subroutine that aims to solves the problem:

> Suppose an unitary operator U of dimension 2^n with unknown eigenvalues $\lambda_j = \exp\left(2\pi i \frac{\phi_j}{2^t}\right)$, where ϕ_j are unknown phase factors and t is the minimum number of bits needed to express the all phase factors ϕ_j up to a desired binary precision. Given an input quantum state $|x\rangle = \sum_{j=1}^{2^n} \alpha_j |u_j\rangle$, where α_j and $|u_j\rangle$ are unknown complex coefficients and eigenstates of U respectively, find and obtain the quantum state $|y\rangle = \sum_{j=1}^{2^n} \alpha_j |\phi_j\rangle$, where $|\phi_j\rangle$ is a quantum state that is in a qubit binary representation of eigenvalues up to the desired finite binary precision.

Breaking down the problem, it simply requires QPE to do the the following,

$$\text{(Simplest Case)} \quad |\mathbf{0}\rangle |u\rangle \xrightarrow{QPE} |\phi\rangle |u\rangle \tag{4.3.1}$$
$$| \quad \text{where } |u\rangle \text{ is one of the eigenstates of } U,$$
$$| \quad \text{and } |\phi\rangle \text{ qubit binary representation of } \phi.$$
$$| \quad \text{where } |\mathbf{0}\rangle \text{ refers to all qubits in } |0\rangle.$$
$$\text{(General Case)} \quad |\mathbf{0}\rangle |x\rangle \xrightarrow{QPE} \left(\sum_{j=1}^{2^n} \alpha_j |\phi_j\rangle\right) |x\rangle \tag{4.3.2}$$

To understand the QPE algorithm better, we shall consider simplest case of the problem, the input quantum state $|u\rangle$ would be one of the eigenstate of U and the output quantum state would be a qubit binary representation of the corresponding eigenvalue $|\phi\rangle$. The idea behind QPE is to first create a Fourier-transformed $|\phi\rangle$, then apply the inverse of QFT on it to obtain the desired output state $|\phi\rangle$. To create a Fourier-transformed $|\phi\rangle$, first consider initialising a eigenvalue register to $|\mathbf{0}\rangle_e$ and implement a Walsh-Hadamard (WH) transformation, followed by a Hamiltonian Simulation (HS) subroutine. The QPE algorithm is summarised below:

Inputs:

- A 2^n-dimensional unitary matrix U and a eigenstate $|u\rangle$, such that $U|u\rangle = \exp\left(2\pi i \frac{\phi}{2^t}\right)|u\rangle$.
- Prepare the main register $|\mathbf{0}\rangle_m$ with n qubits to store the eigenstate $|u\rangle_m$.
- Prepare the eigenvalue register $|\mathbf{0}\rangle_e$ with t qubits, where t is the number of qubits need to store the eigenvalues ϕ to a desired binary precision.

Outputs:

- A quantum state $|\phi\rangle_e$ that is a qubit binary representation of the eigenvalue ϕ.

Implementation steps:

1. Initialize quantum state $|\mathbf{0}\rangle_e |u\rangle_m$ from $|\mathbf{0}\rangle_e |\mathbf{0}\rangle_m$.

$$|\mathbf{0}\rangle_e |\mathbf{0}\rangle_m \xrightarrow{\text{Initialize}} |\mathbf{0}\rangle_e |u\rangle_m \tag{4.3.3}$$

2. (Walsh–Hadamard Transformation) Apply Hadamard gates to all qubits in the eigenvalue register to create superposition of all possible states.

$$\xrightarrow{\text{WS}} \bigotimes_{j=1}^{t} \left(\frac{|0_j\rangle_e |u\rangle_m + |1_j\rangle_e |u\rangle_m}{\sqrt{2}} \right)$$

3. (Hamiltonian Simulation) Apply a sequence of controlled-$U^{2^{t-j}}$ gates on the main register conditioned on qubit $|x_j\rangle$ to obtain the Fourier-transformed $|\phi\rangle_e$.

$$\xrightarrow{\text{HS}} \bigotimes_{j=1}^{t} \left(\frac{|0_j\rangle_e |u\rangle_m + \left(\mathbb{I} \oplus U^{2^{t-j}}\right) |1_j\rangle_e |u\rangle_m}{\sqrt{2}} \right) \tag{4.3.4}$$

$$\Big| \quad \text{Using eigenequation } U|u\rangle = \exp\left(2\pi i \frac{\phi}{2^n}\right)|u\rangle .$$

$$\Big| \quad \text{Consider binary representation of } \phi \text{ using } t \text{ qubits : } \frac{\phi}{2^t} = \sum_{l=1}^{t} \phi_l \cdot 2^{l-1-t}$$

$$= \frac{1}{\sqrt{2^t}} \bigotimes_{j=1}^{t} \left(|0_j\rangle_e + \exp\left(2\pi i \left(\sum_{l=1}^{t} \phi_l \cdot 2^{l-j-1}\right)\right) |1_j\rangle_e \right) |u\rangle_m \tag{4.3.5}$$

$$\frac{1}{\sqrt{2^t}} \bigotimes_{j=1}^{t} \left(|0_j\rangle_e + \exp\left(2\pi i \left(\phi_t 2^{t-j-1} + \ldots + \phi_l \cdot 2^{l-j-1} + \ldots + \phi_1 \cdot 2^{-j}\right)\right) |1_j\rangle_e \right) |u\rangle_m \tag{4.3.6}$$

$$= \frac{1}{\sqrt{2^t}} \left(|0_j\rangle_e + \exp\left(2\pi i \left(\phi_t 2^{-1} + \ldots + \phi_l \cdot 2^{l-t-1} + \ldots + \phi_1 \cdot 2^{-t}\right)\right) |1_j\rangle_e \right) \otimes \ldots$$

$$\ldots \otimes \left(|0_1\rangle_e + \exp\left(2\pi i \left(\phi_t 2^{t-2} + \ldots + \phi_l \cdot 2^{l-2} + \ldots + \phi_1 \cdot 2^{-1}\right)\right) |1_1\rangle_e \right) |u\rangle_m \tag{4.3.7}$$

$$= \frac{1}{\sqrt{2^t}} \left(|0_j\rangle_e + \exp\left(2\pi i 0.\phi_t \ldots \phi_1\right) |1_j\rangle_e \right) \otimes \ldots \otimes \left(|0_1\rangle_e + \exp\left(2\pi i 0.\phi_1\right) |1_1\rangle_e \right) |u\rangle_m \tag{4.3.8}$$

$$= \text{Fourier-transformed } |\phi\rangle \tag{4.3.9}$$

4. (Inverse QFT) Apply inverse Quantum Fourier Transform $QFT^\dagger$ on Fourier-transformed $|\phi\rangle_e$ in the eigenvalue register to obtain the desired state $|\phi\rangle_e |u\rangle_m$.

$$\xrightarrow{\text{QFT}^\dagger} |\phi\rangle_e |u\rangle_m \tag{4.3.10}$$

The QPE quantum circuit is given in the figure 4.3.1 below.

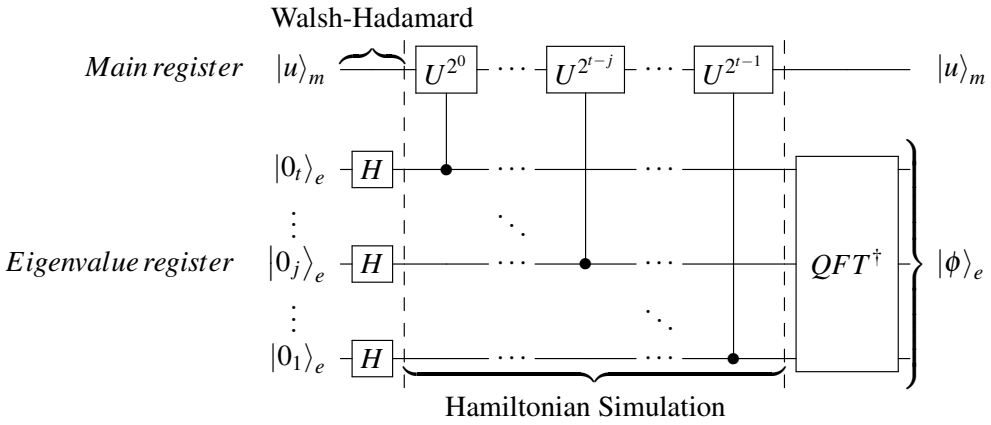

Figure 4.3.1: Quantum circuit for Quantum Phase Estimation.

where the quantum circuit for the inverse Quantum Fourier Transform is given in figure 4.3.2 below

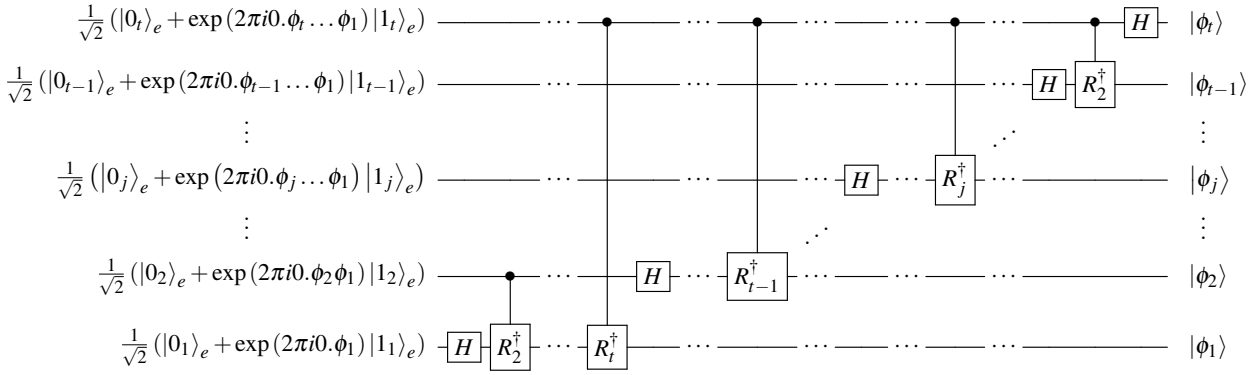

Figure 4.3.2: Quantum circuit for inverse Quantum Fourier Transform.

Historically, QPE was first developed by Alexey Kitaev to generalise and reformulate Peter Shor's quantum order-finding in his factoring algorithm [Sho94]. Kitaev showed that Shor reduction of factoring to an finding the order of an element is just one of the many Abelian stabilizer problems which QPE is designed to solve. Thus, QPE forms a crucial part in solving any Abelian stabilizer problems that involves finding the period of a restricted class of periodic functions [Kit95]. The Kitaev's QPE expressed in terms quantum circuit model was given by Richard Cleve et al [Cle+98].

Part II

Final Year Project

Chapter 5

Quantum Linear Problem

In this chapter, we shall start first by briefly describing the problem system of linear equation and its classical algorithm solver. Next, we will attempt to reformulate the system of linear equation based on the quantum mechanics formalism and resolve any potential complications that might unfold as a result. Thereafter, we will introduce the quantum linear solver, as first proposed by Aram Harrow, Avinatan Hassidim and Seth Lloyd (HHL). Due to the lack of explanation and elaboration given by the original authors, I will attempt to propose three implicit ideas behind the quantum linear solver that HHL might have considered during its creation, but not stated in their proposal.

5.1 System of Linear equations

A general system of linear equations is a set of m linear equations with n unknowns that can be written as,

$$\begin{aligned} b_1 &= a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \\ b_2 &= a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \\ &\vdots \\ b_m &= a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \end{aligned} \tag{5.1.1}$$

where $x_1, x_2, \ldots, x_n$ are the unknowns to be found, $a_{11}, a_{12}, \ldots, a_{mn}$ are the coefficients of the system and $b_1, b_2, \ldots, b_m$ are the constants of the systems. One can reformulate this linear problem as a vector equation, that is finding the linear combination of the given constant vector $\begin{pmatrix} b_1 & b_2 & \ldots & b_m \end{pmatrix}^T$ in terms of the system vectors $\begin{pmatrix} a_{1i} & a_{2i} & \ldots & a_{mi} \end{pmatrix}^T$ for $1 \leq i \leq n$, $i \in \mathbb{Z}$. Rewriting equation set 5.1.1 into $b_j = \sum_{i=1}^{n} x_i a_{ji}$ where $0 \leq j \leq m$, $j \in \mathbb{Z}$, yields the famous matrix linear equation,

$$\vec{b} = A\vec{x} \tag{5.1.2}$$

By this matrix reformulation of the system of linear equation, one can easily find its solution $\vec{x} = A^{-1}\vec{b}$, if A^{-1} exists. This also means that the problem for solving system of linear equations is reduced to just finding the inverse of the matrix A. Classical algorithms to find A^{-1} are numerous and depending on the given property of the matrix A, it may be possible to solve classically in a few steps. The

most commonly taught matrix inversion algorithm for a arbitrary unknown matrix is the Gauss-Jordan Elimination algorithm. Simply, the Gauss-Jordan Elimination transforms an augmented matrix $[A|\mathbb{I}]$ to $\left[\mathbb{I}|A^{-1}\right]$ by elementary row operations. It has a cubic computation time scaling with respect to the dimension of the matrix.

5.2 Quantum Linear Problem

In order to harness quantum computation, the problem of solving system of linear equations has to be reformulated according to the axioms in quantum mechanics. Normalized vectors $\frac{\vec{v}}{||\vec{v}||}$ can be used to represent pure quantum states $|v\rangle$. Unitary or hermitian matrices A can be used to represent quantum unitary operators U or quantum measurements M respectively.

$$\frac{\vec{v}}{||\vec{v}||} \longrightarrow |v\rangle \tag{5.2.1}$$

$$A \longrightarrow U \text{ or } M \tag{5.2.2}$$

By this reformulation, there are two possible analogous quantum linear problems:

> **Problem 1:** Given an unitary operator U and a pure quantum state $|b\rangle$, find the unknown quantum state $|x\rangle$ such that$|b\rangle = U|x\rangle$.
>
> **Problem 2:** Given a measurement operator M and a pure quantum state $|b\rangle$, find the unknown quantum state $|x\rangle$ such that$|b\rangle = M|x\rangle$.

In the first quantum linear problem, obtaining the solution state $|x\rangle$ is easy if one could find $U^{-1} = U^{\dagger}$ and apply $U^{\dagger}$ on $|b\rangle$. Thus, the solution state $|x\rangle$ can be easily stated as, $|x\rangle = U^{-1}|b\rangle = U^{\dagger}|b\rangle$. The sub-problem of finding $U^{\dagger}$or U^{-1}, given U, is trivial in the quantum circuit formalism as it is simply equivalent to a switch of the inputs and the outputs of quantum gate U. Similarly, for the second quantum linear problem, it seems that the solution state $|x\rangle$ can be easily stated as,

$$|x\rangle = M^{-1}|b\rangle \tag{5.2.3}$$

However, this equation 5.2.3 is not consistent with the property of a quantum measurement on a quantum state, resulting in several complications in defining such a quantum linear problem for hermitian operators:

- According to the defining property of hermiticity, an inverse hermitian operator is still an considered as a measurement as it is also an hermitian operator,

$$\text{By definition, } M = M^{\dagger} \quad | \quad \text{(Hermiticity)} \tag{5.2.4}$$

$$| \quad \text{Applying } \left(M^{-1}\right)^{\dagger} \text{ and } M^{-1} \text{ on the left and right for each side.}$$

$$\left(M^{-1}\right)^{\dagger} \cancelto{\mathbb{I}}{MM^{-1}} = \cancelto{\mathbb{I}}{\left(M^{-1}\right)^{\dagger} M^{\dagger}} M^{-1} \tag{5.2.5}$$

$$\left(M^{-1}\right)^{\dagger} = M^{-1} \tag{5.2.6}$$

Consequently, as introduced in subsection 2.3, applying hermitian operator M^{-1} on the quantum state $|b\rangle$ will collapse into one of the normalised eigenstates of M^{-1} with a measurement outcome

of an eigenvalue of M^{-1}. Therefore, equation 5.2.3 might be a physically impossible problem as the expected solution state $|x\rangle$ might not be any of the normalised eigenstates of M^{-1}.

- Even if the expected solution state $|x\rangle$ is to be any of the normalised eigenstates of M, the equation 5.2.3 still does not mathematically hold as hermitian operators do not preserve the normalisation of the quantum state after application.

To resolve these complications and determine what shall the solution state $|x\rangle$ be. We start first by mathematically fixing equation 5.2.3 by a normalisation. Suppose, the problem is 2^n-dimensional, express the pure quantum state $|b\rangle = \sum_{j=1}^{2^n} \beta_j |m_j\rangle$ as a linear combination of the normalised eigenvectors of M, $|m_j\rangle$, with the corresponding complex coefficients β_j,

$$M^{-1}|b\rangle = \sum_{j=1}^{2^n} \beta_j M^{-1} |m_j\rangle \tag{5.2.7}$$

$$| \quad \text{Using eigenequation } M^{-1}|m_j\rangle = \frac{1}{m_j}|m_j\rangle.$$

$$= \sum_{j=1}^{2^n} \frac{\beta_j}{m_j} |m_j\rangle \tag{5.2.8}$$

$$| \quad \text{Suppose normalising it to make it a solution state } |x\rangle.$$

$$M^{-1}|b\rangle \xrightarrow{\text{Norm}} \frac{1}{\sqrt{\sum_{j=1}^{2^n} \left|\frac{\beta_j}{m_j}\right|^2}} \sum_{j=1}^{2^n} \frac{\beta_j}{m_j} |m_j\rangle \tag{5.2.9}$$

$$| \quad \text{Let } d = \sqrt{\sum_{j=1}^{2^n} \left|\frac{\beta_j}{m_j}\right|^2}.$$

$$|x\rangle = \frac{1}{d} M^{-1}|b\rangle \tag{5.2.10}$$

This analysis leads to a surprising fact, at least to the author of this thesis, that the solution state $|x\rangle$, in second quantum linear problem, can be interpreted as a normalised pre-collapse quantum solution state of the measurement of $|b\rangle$ by the measurement operator M^{-1} and this revelation forces us reformulate the second quantum linear problem as the following:

> **Problem 2 (Reformulated):** Given a measurement operator M and a pure quantum state $|b\rangle$, find the pre-collapse quantum solution state $|x\rangle$ such that $|x\rangle = \frac{1}{d} M^{-1}|b\rangle$.

Of course, a normalised pre-collapse quantum state does not quantum mechanically exist, however it is possible to create it by a unitary transformation of $|b\rangle$ and we can 'imagined' it as such. This means that the second quantum linear problem is a hypothetical problem unlike the first one which is a physical problem! Fortunately, solving this hypothetical problem is possible, but probable. Aram Harrow, Avinatan Hassidim and Seth Lloyd (HHL) were the first team of researchers to came up with an quantum linear solver (QLS) algorithm that achieves an exponential speedup over the classical algorithm, albeit with a probabilistic success, by supposing that hermitian matrix M is sufficiently sparse [HHL09]. For any other arbitrary hermitian M and pure quantum state $|b\rangle$, the current literature suggests that it solves with at most a quadratic speed up in computation time, using Wossnig et al's adapted QLS algorithm [WZP17].

5.3 Algorithmic Solution Idea Proposition

Solving for $|x\rangle$ using equation 5.2.10, though, mathematically sound, is still not implementable quantum mechanically. As noted in equation 5.2.6, $\frac{1}{d}M^{-1}$ is a hermitian operator, thus, a naive direct application on $|b\rangle$ will obviously not yield the solution state $|x\rangle$. This complication is resolved using the quantum linear solver, as first proposed by Aram Harrow, Avinatan Hassidim and Seth Lloyd (HHL) [HHL09]. It describes an innovative quantum algorithm that makes use of 2 quantum subroutines: Quantum Phase Estimation (QPE) and Ancillary Rotation (AR). The background of QPE can be found in section 4.3 on page 21. The AR subroutine will be described later in this section, its problems will be discussed in the next chapter at section 6.3 and modified for implementation in the following chapters 7 and 8. Before explicitly stating the HHL's quantum linear solver algorithm, it is worthwhile to understand how their quantum algorithm works, which was not apparent in their proposal. I propose three implicit ideas on how one could obtain the solution state $|x\rangle$ using HHL's quantum linear solver algorithm.

First Idea

HHL's first ingenious idea is to leverage the properties of quantum measurements to get the solution state $|x\rangle$: *Use ancillary measurements*. This can be done by appending an ancillary qubit $|0\rangle_a$ to given pure quantum state $|b\rangle_m$ in the main qubit register, where the subscript a and m denotes the state being in the ancillary and main qubit register respectively, and applying an unitary operator A on $|b\rangle_m|0\rangle_a$ such that,

$$
\begin{aligned}
|b\rangle_m|0\rangle_a &= \sum_{j=1}^{2^n}\beta_j\left|m_j\right\rangle_m|0\rangle_a && (5.3.1)\\
&\xrightarrow{A} \sum_{j=1}^{2^n}\left[\frac{\beta_j C}{m_j}\left|m_j\right\rangle_m|1\rangle_a+\beta_j\sqrt{1-\left(\frac{C}{m_j}\right)^2}\left|m_j\right\rangle_m|0\rangle_a\right] && (5.3.2)\\
&\mid \quad \text{for some real parameter } C\in\mathbb{R}.
\end{aligned}
$$

so that the desired un-normalised solution state $|x\rangle_m$ is in a superposition with another orthogonal quantum state. By performing a measurement on the ancillary qubit in the computation basis $\{|0\rangle,|1\rangle\}$ using a Pauli-Z operator and selecting the $|1\rangle_a$ ancillary qubit, whose the measurement outcome is -1, will yield the desired solution state $|x\rangle_m$.

Notice that a free real parameter C is introduced in the amplitudes in equation 5.3.2. It has a limited domain of $C\in[0,min_j(m_j)]$, where $min_j(m_j)$ represents the smallest magnitude real eigenvalue of M. This parameter does not affect desired solution state $|x\rangle_m$, but it affects the probability of obtaining $|x\rangle_m$. We shall see later in the following chapters 7, 8 and 9, that by re-parametrising it (using a new parameter r), we can use C as an independent variable for conducting quantum experiments (algorithms) on a quantum computer to verify the probability of obtaining $|x\rangle_m$. A summary of the measurement results is given in table 5.3.1 below.

	Collapsed State	Measurement Outcomes	Probability
1.	$\underbrace{\frac{1}{\sqrt{\sum_{j=1}^{2^n}\left\|\frac{\beta_j}{m_j}\right\|^2}}\sum_{j=1}^{2^n}\frac{\beta_j}{m_j}\left\|m_j\right\rangle_m\left\|1\right\rangle_a}_{\left\|x\right\rangle_m}$	-1	$\sum_{j=1}^{2^n}\left\|\frac{\beta_j C}{m_j}\right\|^2$
2.	$\frac{1}{\sqrt{\sum_{j=1}^{2^n}\left\|\beta_j\sqrt{1-\left\|\frac{C}{m_j}\right\|^2}\right\|^2}}\sum_{j=1}^{2^n}\beta_j\sqrt{1-\left\|\frac{C}{m_j}\right\|^2}\left\|m_j\right\rangle_m\left\|0\right\rangle_a$	1	$\sum_{j=1}^{2^n}\left\|\beta_j\sqrt{1-\left\|\frac{C}{m_j}\right\|^2}\right\|^2$

Table 5.3.1: Quantum post-measurement results of $A\left|b\right\rangle_m\left|0\right\rangle_a$, where the measurement is done on the ancillary qubit using a Pauli-Z operator.

We check that the probability sum is unity,

$$\sum_{j=1}^{2^n}\left\|\beta_j\sqrt{1-\left\|\frac{C}{m_j}\right\|^2}\right\|^2 = \sum_{j=1}^{2^n}\left|\beta_j\right|^2-\sum_{j=1}^{2^n}\left\|\frac{\beta_j C}{m_j}\right\|^2 \tag{5.3.3}$$

$$\left| \quad \text{Since } \sum_{j=1}^{2^n}\left|\beta_j\right|^2=1.$$

$$= 1-\sum_{j=1}^{2^n}\left\|\frac{\beta_j C}{m_j}\right\|^2 \tag{5.3.4}$$

$$\therefore \quad 1 = \sum_{j=1}^{2^n}\left\|\beta_j\sqrt{1-\left\|\frac{C}{m_j}\right\|^2}\right\|^2+\sum_{j=1}^{2^n}\left\|\frac{\beta_j C}{m_j}\right\|^2 \tag{5.3.5}$$

Note that the first collapsed state in table 5.3.1 contains the desired solution state $\left|x\right\rangle_m$. Therefore, this solution might just work if we can find and construct the unitary operator A. However, it is not easy

to do so. Suppose that we want to construct operator A using operators M^{-1} by working backwards,

$$A|b\rangle_m|0\rangle_a = \sum_{j=1}^{2^n}\left[\frac{\beta_j C}{m_j}|m_j\rangle_m|1\rangle_a + \beta_j\sqrt{1-\left(\frac{C}{m_j}\right)^2}|m_j\rangle_m|0\rangle_a\right] \tag{5.3.6}$$

$$\mid \quad \text{Using eigenequation } M^{-1}|m_j\rangle = \frac{1}{m_j}|m_j\rangle.$$

$$= \sum_{j=1}^{2^n}\left[\beta_j\left(CM^{-1}\otimes\mathbb{I}\right)|m_j\rangle_m|1\rangle_a + \beta_j\sqrt{\mathbb{I}-\left(CM^{-1}\otimes\mathbb{I}\right)^2}|m_j\rangle_m|0\rangle_a\right] \tag{5.3.7}$$

$$= \left(CM^{-1}\otimes\mathbb{I}\right)\sum_{j=1}^{2^n}\beta_j|m_j\rangle_m|1\rangle_a + \sqrt{\mathbb{I}-\left(CM^{-1}\otimes\mathbb{I}\right)^2}\sum_{j=1}^{2^n}\beta_j|m_j\rangle_m|0\rangle_a \tag{5.3.8}$$

$$\mid \quad \text{Subtituting } \sum_{j=1}^{2^n}\beta_j|m_j\rangle = |b\rangle.$$

$$= \left(CM^{-1}\otimes\mathbb{I}\right)|b\rangle_m|1\rangle_a + \sqrt{\mathbb{I}-\left(CM^{-1}\otimes\mathbb{I}\right)^2}|b\rangle_m|0\rangle_a \tag{5.3.9}$$

$$\mid \quad \text{Consider the orthonormal basis set } \{|b\rangle|0\rangle, |b\rangle|1\rangle\},$$

$$\mid \quad \text{that is matrix has basis: } \begin{pmatrix} |b\rangle|0\rangle\langle 0|\langle b| & |b\rangle|0\rangle\langle 1|\langle b| \\ |b\rangle|1\rangle\langle 0|\langle b| & |b\rangle|1\rangle\langle 1|\langle b| \end{pmatrix}.$$

$$\text{Suppose} \quad \Rightarrow \underbrace{\begin{pmatrix} \sqrt{\mathbb{I}-\left(CM^{-1}\otimes\mathbb{I}\right)^2} & -\left(CM^{-1}\otimes\mathbb{I}\right) \\ \left(CM^{-1}\otimes\mathbb{I}\right) & \sqrt{\mathbb{I}-\left(CM^{-1}\otimes\mathbb{I}\right)^2} \end{pmatrix}}_{\text{Let this matrix be } A.}|b\rangle_m|0\rangle_a \tag{5.3.10}$$

We check that operator A is unitary,

$$A^{-1} = \begin{pmatrix} \sqrt{\mathbb{I}-\left(CM^{-1}\otimes\mathbb{I}\right)^2} & -\left(CM^{-1}\otimes\mathbb{I}\right) \\ \left(CM^{-1}\otimes\mathbb{I}\right) & \sqrt{\mathbb{I}-\left(CM^{-1}\otimes\mathbb{I}\right)^2} \end{pmatrix}^{-1} \tag{5.3.11}$$

$$= \frac{1}{\mathbb{I}-\left(CM^{-1}\otimes\mathbb{I}\right)^2+\left(CM^{-1}\otimes\mathbb{I}\right)^2}\begin{pmatrix} \sqrt{\mathbb{I}-\left(CM^{-1}\otimes\mathbb{I}\right)^2} & \left(CM^{-1}\otimes\mathbb{I}\right) \\ -\left(CM^{-1}\otimes\mathbb{I}\right) & \sqrt{\mathbb{I}-\left(CM^{-1}\otimes\mathbb{I}\right)^2} \end{pmatrix} \tag{5.3.12}$$

$$\mid \quad \text{Since } M^{-1} = \left(M^{-1}\right)^\dagger \text{ from equation 5.2.6 and } C = C^*.$$

$$= \begin{pmatrix} \sqrt{\mathbb{I}-\left(C^*\left(M^{-1}\right)^\dagger\otimes\mathbb{I}\right)^2} & \left(C^*\left(M^{-1}\right)^\dagger\otimes\mathbb{I}\right) \\ -\left(C^*\left(M^{-1}\right)^\dagger\otimes\mathbb{I}\right) & \sqrt{\mathbb{I}-\left(C\left(M^{-1}\right)^\dagger\otimes\mathbb{I}\right)^2} \end{pmatrix} \tag{5.3.13}$$

$$= \begin{pmatrix} \sqrt{\mathbb{I}-\left(CM^{-1}\otimes\mathbb{I}\right)^2} & -\left(CM^{-1}\otimes\mathbb{I}\right) \\ \left(CM^{-1}\otimes\mathbb{I}\right) & \sqrt{\mathbb{I}-\left(CM^{-1}\otimes\mathbb{I}\right)^2} \end{pmatrix}^\dagger \tag{5.3.14}$$

$$= A^\dagger \quad \text{(shown)} \tag{5.3.15}$$

This crazy unitary operator A is daunting and it seems that there is no obvious and direct way to construct or even implement it. To understand what operator A actually does on $|b\rangle_m|0\rangle_a$, consider the

eigen-decomposition of M,

$$M = R\Lambda R^{\dagger} \tag{5.3.16}$$

where R is a unitary matrix whose columns are eigenvectors of M and Λ is a diagonal matrix whose elements are eigenvalues of M. The inverse M has a similar decomposition,

$$M^{-1} = R\Lambda^{-1}R^{\dagger} \tag{5.3.17}$$

Applying unitary operator A from equation 5.3.9 on $|b\rangle_m |0\rangle_a$ and using the decomposition of M^{-1} as given in equation 5.3.17 above, yields,

$$\begin{aligned} A|b\rangle_m|0\rangle_a &= \left(CR\Lambda^{-1}R^{\dagger}\otimes\mathbb{I}\right)|b\rangle_m|1\rangle_a \\ &\quad + \sqrt{\mathbb{I}-\left(CR\Lambda^{-1}R^{\dagger}\otimes\mathbb{I}\right)^2}\,|b\rangle_m|0\rangle_a \qquad (5.3.18) \\ &= RC\Lambda^{-1}R^{\dagger}|b\rangle_m\otimes\mathbb{I}|1\rangle_a \\ &\quad + (R\otimes R)\sqrt{\mathbb{I}-\left(C\left(\Lambda^{-1}\right)^2\otimes\mathbb{I}\right)}\left(R^{\dagger}\otimes R^{\dagger}\right)|b\rangle_m|0\rangle_a \qquad (5.3.19) \end{aligned}$$

Focusing on the first term in equation 5.3.19, the following calculation shows how $RC\Lambda^{-1}R^{\dagger}|b\rangle = \sum_{j=1}^{2^n}\frac{\beta_j C}{m_j}|m_j\rangle = |x\rangle$, which is the un-normalised solution state $|x\rangle_m$.

$$\begin{aligned} & RC\Lambda^{-1}R^{\dagger}|b\rangle \\ = \;& RC\Lambda^{-1}R^{\dagger}\sum_{j=1}^{2^n}\beta_j|m_j\rangle \qquad (5.3.20) \\ | \;& \text{Using } R^{\dagger}|m_j\rangle = |j-1\rangle. \\ | \;& \text{where } |j-1\rangle \text{ is a } 2^n\text{D computational basis from the set } \{|j-1\rangle : j=1,2,\ldots,2^n\}. \\ = \;& RC\Lambda^{-1}\sum_{j=1}^{2^n}\beta_j|j-1\rangle \qquad (5.3.21) \\ | \;& \text{Using } \Lambda^{-1}|j-1\rangle = \frac{1}{m_j}|j-1\rangle. \\ = \;& RC\sum_{j=1}^{2^n}\frac{\beta_j}{m_j}|j-1\rangle \qquad (5.3.22) \\ | \;& \text{Using } R|j-1\rangle = |m_j\rangle. \\ = \;& \sum_{j=1}^{2^n}\frac{\beta_j C}{m_j}|m_j\rangle = |x\rangle \qquad (5.3.23) \end{aligned}$$

The above set of equations tell us how unitary operator A does on $|b\rangle_m|0\rangle_a$ to obtain the un-normalised solution state $|x\rangle$:

1. Transform the eigenvectors $|m_j\rangle$ to $|j-1\rangle$ where $|j-1\rangle$ is a 2^nD orthonormal basis from the set$\{|j-1\rangle : j=1,2,\ldots,2^n\}$ using $R^{\dagger}$.

2. Apply an inverse diagonal operator $C\Lambda^{-1}$ so that the inverse eigenvalues amplitudes $\frac{C}{m_j}$ are applied to $|j-1\rangle$ accordingly.

3. Finally, transform $|j-1\rangle$ back to the eigenvectors $\left|m_j\right\rangle$ using R.

Second Idea

Understanding the behaviour of unitary operator A leads to the second HHL's ingenious idea: *Imitate unitary operator A*. By applying a set of controlled Rotate-Y, $R_y(\theta_j)$ quantum gates on the ancillary qubit, conditioned by the control qubit $|b\rangle_m$ being in the eigenstate $\left|m_j\right\rangle_m$ such that $\sin\left(\frac{\theta_j}{2}\right) = \frac{C}{m_j}$, on quantum state $|b\rangle_m|0\rangle_a$. This step is termed as the Ancillary Rotation (AR) subroutine and it does the following,

$$|b\rangle_m|0\rangle_a \xrightarrow{AR(|m_j\rangle)} \sum_{j=1}^{2^n}\left[\frac{\beta_j C}{m_j}\left|m_j\right\rangle_m|1\rangle_a + \beta_j\sqrt{1-\left(\frac{C}{m_j}\right)^2}\left|m_j\right\rangle_m|0\rangle_a\right] \tag{5.3.24}$$

$$| \quad \text{Let } \sin\left(\frac{\theta_j}{2}\right) = \frac{C}{m_j} \text{ and } \cos\left(\frac{\theta_j}{2}\right) = \sqrt{1-\left(\frac{C}{m_j}\right)^2}. \tag{5.3.25}$$

$$= \sum_{j=1}^{2^n}\left[\beta_j \sin\left(\frac{\theta_j}{2}\right)\left|m_j\right\rangle_m|1\rangle_a + \beta_j\cos\left(\frac{\theta_j}{2}\right)\left|m_j\right\rangle_m|0\rangle_a\right] \tag{5.3.26}$$

$$| \quad \text{Using } R_y(\theta_j) = \begin{pmatrix} \cos\left(\frac{\theta_j}{2}\right) & -\sin\left(\frac{\theta_j}{2}\right) \\ \sin\left(\frac{\theta_j}{2}\right) & \cos\left(\frac{\theta_j}{2}\right) \end{pmatrix}.$$

$$\therefore \quad A|b\rangle_m|0\rangle_a = \sum_{j=1}^{2^n}\left[(\mathbb{I}\oplus R_y(\theta_j))\,\beta_j\left|m_j\right\rangle_m|0\rangle_a\right] \tag{5.3.27}$$

The use of controlled-$R_y(\theta_j)$ is unusual as it differs from the convention that controlled-gates are applied on the condition of the control qubit being in either computational basis $\{|0\rangle, |1\rangle\}$. In general, AR step is not implementable if one does not know the eigenvectors $\left|m_j\right\rangle$ and eigenvalues m_j. However, it is actually implementable for special cases and it forms the underlying foundation of the early experiments on HHL algorithm. Often, these basic experiments are trying to solve a simple matrix that has eigenstates that are in the already standard orthonormal basis such as $\{|0\rangle, |1\rangle\}$ or $\{|+\rangle, |-\rangle\}$. Therefore, quantum circuit of the HHL algorithm can be simplified down or 'complied' to a few quantum gates and wires. Despite the initial success of these experiments, we cannot consider these algorithms as the actual HHL algorithm as it cannot be generalised to other cases. For example, consider the case of eigenstates $\left|m_j\right\rangle_m$ being in some arbitrary complex superposition of the standard basis. Finding a decomposition of a controlled-$R_y(\theta_j)$ that is conditioned on $|b\rangle_m$ being in any eigenstate $\left|m_j\right\rangle_m$ might be difficult, let alone implement it.

Third Idea

For the rest of the report, to avoid confusion later in the section, the notation for eigenvalues m_j and eigenstates $\left|m_j\right\rangle$ will be changed to λ_j and $\left|u_j\right\rangle$ respectively. The difficulty in implementing controlled-$R_y(\theta_j)$, that is conditioned conditioned on $|b\rangle_m$ being in any eigenstate $\left|m_j\right\rangle_m$, motivates the HHL's third ingenious idea: *Use the qubit binary representation of the eigenvalues λ_j as condition for controlled-$R_y(\theta_j)$*. That is, to apply AR conditioned by the the eigenvalue qubit register $\left|\lambda_j\right\rangle_e$. This may be easier to implement as it is possible to use controlled-$R_y(\theta)$ on ancillary qubit conditioned by the control qubits

in the eigenvalue qubit register being in the computational basis $\{|0\rangle, |1\rangle\}$. Therefore, by appending an eigenvalue qubit register $|0\rangle_e$ to $|b\rangle_m |0\rangle_a$, the third idea propose the following,

$$|b\rangle_m |0\rangle_e |0\rangle_a \xrightarrow{QPE} |b\rangle_m |\lambda_j\rangle_e |0\rangle_a \tag{5.3.28}$$

$$= \sum_{j=1}^{2^n} \beta_j |u_j\rangle_m |\lambda_j\rangle_e |0\rangle_a \tag{5.3.29}$$

$$\xrightarrow{AR(|\lambda_j\rangle_e)} \sum_{j=1}^{2^n} \left[\frac{\beta_j C}{\lambda_j} |u_j\rangle_m |\lambda_j\rangle_e |1\rangle_a + \beta_j \sqrt{1 - \left(\frac{C}{\lambda_j}\right)^2} |u_j\rangle_m |\lambda_j\rangle_e |0\rangle_a \right] \tag{5.3.30}$$

In summary, the three ideas are:

1. Use ancillary measurements such that the quantum state $A|b\rangle_m |0\rangle_a$ collapse to the desired solution state $|x\rangle_m$.
2. Imitate unitary operator A by applying a set of controlled Rotate-Y, $R_y(\theta_j)$ quantum gates on the ancillary qubit, conditioned by the control qubit $|b\rangle_m$ being in the eigenstate $|u_j\rangle_m$ such that $\sin\left(\frac{\theta_j}{2}\right) = \frac{C}{\lambda_j}$, on quantum state $|b\rangle_m |0\rangle_a$.
3. Use the qubit binary representation $|\lambda_j\rangle_e$ of the eigenvalues λ_j as condition for controlled-$R_y(\theta_j)$ instead of the eigenstate $|u_j\rangle_m$.

With these three ideas in mind, we shall formally describe the quantum linear solver algorithm in the next chapter.

Chapter 6

Original Quantum Linear Solver

Aram Harrow, Avinatan Hassidim and Seth Lloyd (HHL) describe an innovative quantum linear solver (QLS) algorithm that makes use of 2 quantum subroutines: Quantum Phase Estimation (QPE) and Ancillary Rotation (AR) and embodies the three ideas, as proposed in the previous chapter [HHL09]. The QLS algorithm first takes in the quantum state $|b\rangle_m |0\rangle_e |0\rangle_a$ as input and implements QPE to obtain $|\lambda_j\rangle_e$: qubit binary representation of the eigenvalue λ_j. Then, the AR subroutine is employed to imitate unitary operator A, as given in equation 5.3.10. Thereafter, the inverse QPE is called to uncompute $|\lambda_j\rangle_e$ before applying the ancillary measurement. Overall, the QLS algorithm is theoretically simple, as suggested by the quantum circuit given in the figure 6.0.1 below.

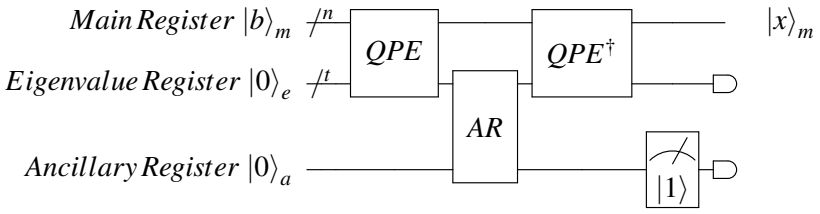

Figure 6.0.1: The quantum circuit for the original QLS algorithm, where QPE and AR refers to Quantum Phase Estimation and Ancillary Rotation subroutine. The set of symbols in the quantum circuit, after the AR gate, represents the ancillary measurement.

In this chapter, we shall explore the QLS algorithm in great detail and analyse any inherent problems of this algorithm, which may necessitate the use of additional modifications to make it implementable on ideal quantum computers (QC). Explicitly, the outline of the original HHL algorithm given below.

Inputs:

- A 2^n dimensional hermitian operator M, a pure quantum state $|b\rangle$.
- Prepare three registers:
  - Main register $|0\rangle_m$ with n qubits to store pure quantum state $|b\rangle_m$.
  - Eigenvalue register $|0\rangle_e$ with t qubits to sufficiently store the all eigenvalues $|\lambda_k\rangle_e$, up to a desired binary precision of t bits.
  - Ancillary register with a single qubit $|0\rangle_a$.

Outputs: A 2^n dimensional quantum state $|x\rangle = \sqrt{\frac{1}{\sum_{k=1}^{2^n} \left|\frac{\beta_k}{\lambda_k}\right|^2}} \sum_{k=1}^{2^n} \frac{\beta_k}{\lambda_k} |u_k\rangle$ on the main register.

Steps:

1. Initialize the registers from $|0\rangle_m |0\rangle_e |0\rangle_a$ to $|b\rangle_m |0\rangle_e |0\rangle_a$.

$$|0\rangle_m |0\rangle_e |0\rangle_a \xrightarrow{\text{Initialize}} |b\rangle_m |0\rangle_e |0\rangle_a = \sum_{j=1}^{2^n} \beta_j \left|u_j\right\rangle_m |0\rangle_e |0\rangle_a \tag{6.0.1}$$

2. (QPE) Compute the eigenvalues and store them in the eigenvalue register using controlled-U quantum gates where $U = exp(2\pi iM)$.

$$\xrightarrow{\text{QPE}} \sum_{j=1}^{2^n} \beta_j \left|u_j\right\rangle_m \left|\lambda_j\right\rangle_e |0\rangle_a \tag{6.0.2}$$

3. (AR) Using the three ideas introduced in section 5.3 on page 28, compute the amplitude coefficients $\frac{C}{\lambda_j}$ and apply it to the eigenvalue qubit $\left|\lambda_j\right\rangle_e$ accordingly, for a fixed real parameter C such that $\frac{C}{\lambda_j} < 1$, for all j.

$$\xrightarrow{\text{AR}} \sum_{j=1}^{2^n} \beta_j \left|u_j\right\rangle_m \left|\lambda_j\right\rangle_e \left(\sqrt{1-\left(\frac{C}{\lambda_j}\right)^2} |0\rangle_a + \frac{C}{\lambda_j} |1\rangle_a\right) \tag{6.0.3}$$

4. Apply the inverse QPE to uncompute the eigenvalues in the eigenvalue register.

$$\xrightarrow{\text{QPE}^\dagger} \sum_{j=1}^{2^n} \beta_j \left|u_j\right\rangle_m |0\rangle_e \left(\sqrt{1-\left(\frac{C}{\lambda_j}\right)^2} |0\rangle_a + \frac{C}{\lambda_j} |1\rangle_a\right) \tag{6.0.4}$$

5. Do a post-selection $|1\rangle_a$ ancilla qubit measurement on the ancillary register with probability $P(|1\rangle_a) = \sum_{j=1}^{2^n} \left|\frac{C\beta_j}{\lambda_j}\right|^2$. Discard all registers, but keep only the main register that will be in the solution state $|x\rangle$.

$$\xrightarrow{\text{Measurement } |1\rangle_a} \sqrt{\frac{1}{\sum_{j=1}^{2^n} \left|\frac{C\beta_j}{\lambda_j}\right|^2}} \sum_{j=1}^{2^n} \beta_j \frac{C}{\lambda_j} \left|u_j\right\rangle_m \tag{6.0.5}$$

$$\Big| \quad \text{Factoring out parameter } C$$

$$|x\rangle_m = \sqrt{\frac{1}{\sum_{j=1}^{2^n} \left|\frac{\beta_j}{\lambda_j}\right|^2}} \sum_{j=1}^{2^n} \frac{\beta_j}{\lambda_j} \left|u_j\right\rangle_m \tag{6.0.6}$$

Upon closer analysis of the original paper [HHL09], it is unfortunate that the AR subroutine was not elaborated sufficiently. This prevented the direct implementation of QLS algorithm for the general case, using common quantum gates. The original paper was focused on the complexity of the algorithm as it was intended to show that it provides an exponential speedup over its best classical counterparts if the following requirements were fulfilled,

1. A sparse hermitian matrix M.
2. The use of black box oracles quantum gates that represents controlled-U^{2^j} for any integer $j > 0$, where $U = \exp(2\pi iM)$ used in QPE subroutine.

3. The condition number of sparse matrix M must have a poly-log$N$ scaling with respect to the size $N \times N$ of the matrix M.

As a result of the paper's open-ended approach to the specifics of the algorithm, additional improvements and modifications are sorely needed for the QLS algorithm to work properly. The Eigenvalue Marking (EM) modification, originally first proposed by [Bar+15] is simple modification that exploits an assumed property of the eigenvalues to truncate the QPE subroutine and gives a simple quantum circuit construction of AR subroutine. EM modification will be thoroughly analysed and used in constructing the simplified QLS algorithm in the next chapter 7 on page 41.

Other notable improvement includes using the amplitude amplification method, adapted from grover's search algorithm, to improve the probability of obtaining the solution state [Amb10]. Another modification is the quantum singular value estimation subroutine developed by Wossnig et al to solve dense matrices in polynomial time with at least a quadratic speedup over its classical counterparts [WZP17]. These two modifications are interesting, but it will not elaborated further in this report.

6.1 Known Applications

Suppose the above requirements for a quantum exponential speedup were fulfilled, there were still application difficulties in other areas that might make its implementation unfeasible. These are,

1. The problem of efficiently preparing an arbitrary state $|b\rangle$.

2. The problem of efficiently extracting the amplitudes of the solution state or measuring the expectation value of an arbitrary observable.

3. The difficulty of finding quantum applications whose its system of linear equation that is hermitian.

One of the early attempts to resolve all the above difficulties was proposed by Dave Clader et al, in which they developed a generic-state preparation subroutine. They also suggested several alternative measurement quantities using simple ancilla measurements and provided a new technique known as preconditioning, that is, to apply a matrix B to a matrix of interest A such that BA is hermitian, so that one will solve $BA|x\rangle = B|b\rangle$ instead [CJS13]. In addition, they have theoretically demonstrated their modified QLS algorithm's ability by computing the electromagnetic scattering cross section of an arbitrary target. Unfortunately, upon further detailed analysis of the electromagnetic scattering problem using the finite element method, it was latter found that their solution could only achieve at most a polynomial speedup due to an additional solution accuracy requirement [MP15].

Nonetheless, QLS algorithm can still be theoretically useful even if the above requirements are not satisfied. For example, QLS algorithm has found itself being modified for quantum data fitting as proposed by Nathan Wiebe et al [WBL12]. Their data fitting QLS algorithm was able to efficiently obtain a least-squares optimised quantum state as an approximation over a given set of known quantum states at least in polynomial time. This can be seen as an alternative method to a full quantum state tomography. The result of their proposal pioneered an emerging field of quantum machine learning (QML) based on QLS algorithm. Recently, many interesting potential QML-QLS applications were proposed such as support vector machine [RML13], quantum gradient descent [KP17] and quantum

amplitude perceptron in quantum neural networks [MKF18], promising at most a polynomial speedup over its classical counterparts.

6.2 Physical Implementations

On the experimental side, the limited capability of current quantum computers (QC) have prevented the full implementation of QLS algorithm due to the need of large number of qubits and fault-tolerant quantum operations. One of the more common versions of QLS algorithms that have been successful experimentally demonstrated are complied ones. Complied circuits is a technique that assumes a property or knowledge of the inputs or solution to dramatically simplify the quantum circuit and reduce the number of qubits to a more manageable size that fits existing quantum technologies. However, one severe consequence of applying this technique is that complied circuits are not scalable and generalisable for any other inputs, rendering the algorithm useless outside its restricted finite problem space. Moreover, complied circuits may not be mathematically equivalent to original algorithm due to heavy simplifications, which goes against the definition of the word 'complied'. Thus, complied circuits it is yet another unfortunate and misleading term conjured by the quantum information community.

Nevertheless, several complied versions of QLS algorithms have been demonstrated as a proof of concept on different physical platforms and numerical simulations. These quantum experiments involves optical QC using polarization of photons as qubits [ZYZ19; Cai+13], solid-state QC using superconducting charge circuit qubits of the Xmon type[1] [Zhe+17], IBM QSAM simulator of quantum circuits [Col+18]. Due to its simplicity and ease of implementation, these quantum experiments often report good results largely with errors that are intrinsic to the apparatus used.

The simplified QLS algorithm, which will be introduced in the following chapter 7 on page 41, is not considered as complied, despite its similarities. This simplified algorithm, which uses the Eigenvalue Marking (EM) modification [Bar+15], focuses more on simplifying the quantum circuit based on three ideas rather than on the properties of the inputs. This allows the simplified algorithm to be easily implementable and scalable for other inputs unlike complied ones. Therefore, from a pedagogical standpoint, the simplified algorithm is suitable as the first example and adaptation of QLS algorithm.

6.3 The Ancillary Rotation Problem

The main problem of realising a practical implementation of QLS algorithm was that the quantum circuit decomposition Ancillary Rotation (AR) subroutine was not given in the original paper. Initial workarounds were primarily based on pre-setting AR by determining the rotation angles θ_j for controlled-$R_y(\theta_j)$ gates, calculated from the eigenvalues λ_j of the input matrix M (*See the third idea in section 5.3*). Such methods does not fulfil the input requirements of quantum linear problem as the eigenvalues are not given and must be considered as an unknown quantity. A *proper* AR subroutine must somehow 'read' the unknown eigenvalue qubits in the eigenvalue register $|\lambda_j\rangle_e$, that were created

[1] A superconducting coherent Josephson qubit that is planar and tunable, first developed by a physics experimental group led by Rami Barends in 2013 [Bar+13].

by the QPE subroutine, and apply the corresponding amplitude coefficients $\frac{C}{\lambda_j}$ on the respective eigenstates $\left|u_j\right\rangle_m$.

Among the first solutions to this problem is the Taylor approximation of AR, first developed by Yudong Cao et al [Cao+11]. By having the AR subroutine Taylor expanded up to the first order in the binary representation, it is possible use t number of controlled-$R_y(\theta)$ gates on the ancillary register conditioned by inverse eigenvalue register, where the angular parameter θ is independent of the inputs to the problem. For higher order n^{th} approximation in the Taylor expansion of AR, it requires $O(t^n)$ controlled-$R_y(\theta)$ gates [Ubb19]. An experimental proof of concept of this solution to AR problem was successfully demonstrated on a nuclear magnetic resonance (NMR) QC using the spin of carbon and fluorine atomic nuclei in the iodotrifiuoroethylene C_2F_3I molecule as qubits [Pan+14].

A major disadvantage in implementing such a solution was that it requires an Eigenvalue Inversion quantum subroutine to calculate $\frac{1}{\lambda_j}$ and an extra qubit register $\left|\frac{1}{\lambda_j}\right\rangle_{ie}$ to store it. There are a handful number of quantum algorithms that deals with inverting a number. For arbitrary numbers, represented in binary, one could either use the Newton-Raphson Inverter [Cao+12] or the Thapliyal Inversion algorithm [TVMC16; Tha+18]. The main difference between both of them is the former has an exponential scaling of number of qubits required while the later has a polynomial scaling. Unfortunately, these two algorithms are ridiculously difficult to implement due to the exponential and polynomial scaling of number of qubits, with respect to eigenvalue binary precision t, needed for inverse eigenvalue register. In addition, these algorithms requires the use of quantum arithmetic subroutines which are more complicated than it seems.

Therefore, in order to simplify the construction of the full QLS algorithm, the powers of two inversion quantum subroutine will be used. As the name of the subroutine suggest, the eigenvalues, which will be inverted, must be of integer powers of two. This algorithm is an extremely simple subroutine that uses $\frac{t}{2}$ number of swap gates directly on eigenvalue register, without actually having to append a extra qubit register. Additionally, this subroutine does not affect the overall construction of the QLS algorithm as it can be directly replaced with another suitable inversion quantum subroutine, if desired. Regardless of any Eigenvalue Inversion subroutine, if the real free parameter C is to be set small enough, the 1^{st} order approximation of the Taylor expansion of AR is surprisingly good. The approximated solution state $|\tilde{x}\rangle_m$ is very close, though will *never* be equal, to the exact solution state $|x\rangle_m$, with a high fidelity close to 1. However, the Taylor approximation causes the probability of success to be severely affected. To achieve better fidelity and probability of success, higher order approximation of AR subroutine is needed. A summary of the experiments on QLS algorithm is given in table 6.3.1 on the next page.

Experiment	Year	Type	Matrix	Hamiltonian Simulation	Eigenvalue Inversion	Ancillary Rotation
[Cai+13]	2013	Photonics (Photon polarization)	$\begin{pmatrix} 1.5 & 0.5 \\ 0.5 & 1.5 \end{pmatrix}$	Complied	-	Exact
[Zhe+17]	2017	Solid-State (Superconducting charge circuit of Xmon type)	$\begin{pmatrix} 1.5 & 0.5 \\ 0.5 & 1.5 \end{pmatrix}$	Complied	-	Exact
[Col+18]	2018	Numerical Simulation (IBM QSAM simulator)	$\begin{pmatrix} 1.5 & 0.5 \\ 0.5 & 1.5 \end{pmatrix}$	Complied	-	Exact
[ZYZ19]	2019	Photonics (Photon polarization)	$\begin{pmatrix} 1.5 & 0.5 \\ 0.5 & 1.5 \end{pmatrix}$	Complied	-	Exact
[Bar+15]	2015	Photonics (Photon polarization)	$U_B^\dagger \mathrm{diag}\left(\frac{1}{2}, \frac{3}{4}\right) U_B$	Eigenvalue Qubit Marking	-	Exact
[Cao+11]	2012	Numerical Simulation (Simulator Not Stated)	$\frac{1}{4}\begin{pmatrix} 15 & 9 & 5 & -3 \\ 9 & 15 & 3 & -5 \\ 5 & 3 & 15 & -9 \\ -3 & -5 & -9 & 15 \end{pmatrix}$	Group Leader Optimization Algorithm	Powers of Two	1^{st} order approximation
[Pan+14]	2014	Nuclear Magnetic Resonance (Atomic spin of carbon and fluorine in iodotrifiuoroethylene C_2F_3I molecule)	$\begin{pmatrix} 1.5 & 0.5 \\ 0.5 & 1.5 \end{pmatrix}$	Group Leader Optimization Algorithm	Powers of Two	1^{st} order approximation
[Ubb19]	2019	Numerical Simulation (QX simulator)	$\frac{1}{4}\begin{pmatrix} 15 & 9 & 5 & -3 \\ 9 & 15 & 3 & -5 \\ 5 & 3 & 15 & -9 \\ -3 & -5 & -9 & 15 \end{pmatrix}$	Group Leader Optimization Algorithm	Powers of Two/ Thapliyal Inversion	Up to 5^{th} order approximation

Table 6.3.1: A summary table detailing the experiments involving QLS algorithm. Note that the group leader optimization algorithm is a classical algorithm that finds a quantum circuit decomposition of the Hamiltonian Simulation subroutine using common and rotation quantum gates [DK10], it will not be used in this thesis. Legend: $U_B = R_x(\theta) R_y(\phi)$.

6.4 Algorithm Adaptation Proposal

To mitigate the implementation problems posed by the original QLS algorithm, two adapted versions of QLS algorithms are proposed: A simplified version and a full version. The simplified version will be introduced in the following chapter 7 on the following page. The simplified version uses Eigenvalue Marking (EM) modification to truncate the QPE subroutine. Together with an exact proper AR subroutine, it aims to output an exact solution state $|x\rangle_m$ with the same probability as given by the original. The full version will be introduced in chapter 8 on page 65. The full version aims to output a close approximation of the solution state $|x\rangle_m$ with reduced probability by using a complete version QPE subroutine together with the first order approximation of the AR subroutine. It is called the 'full version' as it can be easily adapted for any hermitian operator M and input quantum state $|b\rangle$ without having to change the structure of the algorithm or knowing the eigenvalues of M. However, to simplify the construction and implementation of the full version, several more assumptions will be needed. For each version of the QLS algorithm, 2D, 4D matrix cases and worked examples will be suggested and implemented by constructing and simulating its quantum circuit using the IBM Qiskit quantum software. The theoretical results will then be generalised to the 2^nD matrix case. Any computational complexity analysis will not done due to author's lack of background knowledge in computer science.

Chapter 7

Simplified Quantum Linear Solver

The simplified quantum linear solver (QLS) algorithm is a QLS that employs Eigenvalue Marking (EM) modification, that was first proposed by [Bar+15]. This chapter will elaborate and explain in detail more than what was given in the original paper. Basically, EM entails a change in the the conditioning of the Rotate-Y gates used in the Ancillary Rotation (AR) subroutine. Instead of using the whole qubit binary sequence of the eigenvalue as a condition, consider using only a *partial* qubit binary sequence of it. This modification is only possible only if the binary sequences of all eigenvalues are the same up to last few qubits. Essentially, EM is a truncated version of QPE. Using this modification, the QLS algorithm can be simplified by considering the following assumptions:

1. Consider 2^n dimensional hermitian operator M and a pure 2^n dimensional quantum state $|b\rangle$.
2. Suppose given the eigenvalue decomposition of hermitian operator $M = R\Lambda R^\dagger$,
   - where R is a unitary operator that has eigenvectors $|u_k\rangle$ in its columns,
   - and Λ is a diagonal operator that has eigenvalues λ_k on its diagonal.
3. Suppose all the eigenvalues λ_k,
   - are real and positive, $\lambda_k \in \mathbb{R}^+$, where index $k = 0, 1, \ldots, 2^n - 1$,
   - has a range of $0 > \lambda_k > 1$,
   - can be finitely represented in binary up to the t^{th} bit, that is,

$$\begin{aligned} \lambda_k &= \left(\frac{x_1}{2} + \frac{x_2}{2^2} + \ldots + \frac{x_t}{2^t}\right)_k && (7.0.1) \\ &= 0.(x_1 x_2 \ldots x_t)_k && (7.0.2) \\ &\mid \quad \text{where } x_j \in \{0,1\}, \\ &\mid \quad \text{for } \forall k \in \{0,1,\ldots,2^n-1\}. \end{aligned}$$

   - has the same binary sequence $\bar{X}$ up to the $(t-n+1)^{th}$ bit and the sequence of the last n bits

is the binary representation of index k, that is all the eigenvalues $\lambda_k = 0.\bar{X}k$,

$$\begin{aligned}
\lambda_k &= 0.x_1x_2\ldots x_{t-n}x_{t-n+1}x_{t-n+2}\ldots x_t && (7.0.3)\\
&\mid \text{Let } \bar{X} = x_1x_2\ldots x_{t-n}.\\
&= 0.\bar{X}\underbrace{x_{t-n+1}x_{t-n+2}\ldots x_t}_{\text{has } n \text{ bits}} && (7.0.4)\\
&\mid x_{t-n+1}x_{t-n+2}\ldots x_t \overset{\text{Relabel}}{=} k_nk_{n-1}\ldots k_2k_1 \overset{\text{In decimal}}{\Rightarrow} k\\
&= 0.\bar{X}k && (7.0.5)
\end{aligned}$$

7.1 2D Case

Consider the two dimensional case, $n = 1$, then, the class of hermitian operator M allowed is such that its two eigenvalues λ_0 and λ_1 in fractional binary representation of t bits are $\lambda_k = 0.\bar{X}k_1$, where index k in binary (using a single bit) $k_1 = \{0,1\}$. The outline of the 2D simplified quantum linear solver algorithm is given as follows,

Inputs:

- A 2D hermitian operator M and a pure 2D quantum state $|b\rangle$.
- Prepare three registers:
  - Main register $|0\rangle_m$ with a single qubit to store the 2D pure quantum state $|b\rangle_m$.
  - Eigenvalue register $|0\rangle_e$ with a single qubit to store the 2D eigenvalue marking qubit $|k_1\rangle_e$.
  - Ancillary register with a single ancilla qubit $|0\rangle_a$.

Outputs: A 2D solution state $|x\rangle = \frac{1}{\sqrt{\sum_{k=0}^{1}\left|\frac{\beta_k}{\lambda_k}\right|^2}}\sum_{k=0}^{1}\frac{\beta_k}{\lambda_k}|u_k\rangle$.

Steps:

1. Initialize the three registers $|b\rangle_m|0\rangle_e|0\rangle_a$.

$$|0\rangle_m|0\rangle_e|0\rangle_a \overset{\text{Initialize}}{\longrightarrow} |b\rangle_m|0\rangle_e|0\rangle_a = \sum_{k=0}^{1}\beta_k|u_k\rangle_m|0\rangle_e|0\rangle_a \tag{7.1.1}$$

2. (Eigenvalue Marking, EM) Extract the last bit k_1 in the binary sequence of the eigenvalue λ_k, corresponding to the eigenstates $|u_k\rangle_m$ and store it in the eigenvalue register in a qubit binary representation as a eigenvalue marking qubit $|k_1\rangle_e$.

$$\overset{\text{EM}}{\longrightarrow} \sum_{k=0}^{1}\beta_k|u_k\rangle_m|k_1\rangle_e|0\rangle_a \tag{7.1.2}$$

3. (Ancillary Rotation, AR) Apply controlled-$R_y(\theta_k)$, with parameter θ_k set such that $\sin\left(\frac{\theta_k}{2}\right) = \frac{C}{\lambda_k}$ for some real parameter C, on the ancilla qubit $|0\rangle_a$, conditioned on the

eigenvalue marking qubit $|k_1\rangle_e$ in the eigenvalue register.

$$\xrightarrow{C[R_y(\theta_k)]} \quad \sum_{k=0}^{1} \beta_k |u_k\rangle_m |k_1\rangle_e \left(\sqrt{1 - \left(\frac{C}{\lambda_k}\right)^2} |0\rangle_a + \frac{C}{\lambda_k} |1\rangle_a \right) \tag{7.1.3}$$

4. Apply the inverse of step 2 to uncompute the eigenvalue marking qubit $|k_1\rangle_e$ in the eigenvalue register.

$$\xrightarrow{\text{EM}^\dagger} \quad \sum_{k=0}^{1} \beta_k |u_k\rangle_m |0\rangle_e \left(\sqrt{1 - \left(\frac{C}{\lambda_k}\right)^2} |0\rangle_a + \frac{C}{\lambda_k} |1\rangle_a \right) \tag{7.1.4}$$

5. Do a post-selection $|1\rangle_a$ ancilla qubit measurement on the ancillary register using Pauli-Z operator, which has a probability $P(|1\rangle_a) = \sum_{k=0}^{1} \left| \frac{C\beta_k}{\lambda_k} \right|^2$. Discard all registers, but keep only the main register which will be in the solution state $|x\rangle$.

$$\xrightarrow{\text{Post-Select } |1\rangle_a \text{ Measurement}} \quad \frac{1}{\sqrt{\sum_{k=0}^{1} \left| \frac{\beta_k}{\lambda_k} \right|^2}} \sum_{k=0}^{1} \frac{\beta_k}{\lambda_k} |u_k\rangle = |x\rangle \tag{7.1.5}$$

Figure 7.1.1 shows quantum circuit structure of the simplified 2D QLS as defined above.

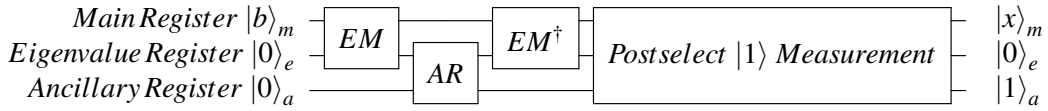

Figure 7.1.1: An outline structure of the simplified 2D QLS quantum circuit. The single qubit registers from top to bottom are main, eigenvalue and ancillary register. Note that compared to the original QLS circuit in figure 6.0.1, the main difference is that EM replaces QPE.

7.1.1 Explicit Circuit Decomposition

This subsection will attempt to describe and explain all steps in detail, as given in section 7.1 above. Following the standard convention in quantum circuit, all qubits starts in state $|0\rangle$. The first step is a state preparation step that correctly prepares the initial state $|b\rangle_m |0\rangle_e |0\rangle_a$ for the algorithm input.

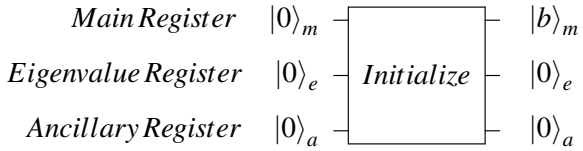

Figure 7.1.2: State preparation of qubit registers starting from standard initial state $|0\rangle_m |0\rangle_e |0\rangle_a$ to $|b\rangle_m |0\rangle_e |0\rangle_a$.

The second step is the Eigenvalue Marking (EM) step. It is an important step as it is an dramatically simplified QPE algorithm. The purpose of this step is to extract the last bit k_1 of the binary sequence of the eigenvalue λ_k, use it to mark the corresponding eigenvectors $|u_k\rangle_m$ as an eigenvalue marking qubit

$|k_1\rangle_e$ and store it in the eigenvalue register. That is,

$$|u_0\rangle_m |0\rangle_e \longrightarrow |u_0\rangle_m |0\rangle_e \tag{7.1.6}$$

$$|u_1\rangle_m |0\rangle_e \longrightarrow |u_1\rangle_m |1\rangle_e \tag{7.1.7}$$

$$\sum_{k=0}^{1} \beta_k |u_k\rangle_m |k_1\rangle_e \xrightarrow{\text{Step 2}} \beta_0 |u_0\rangle_m |0\rangle_e + \beta_1 |u_1\rangle_m |1\rangle_e \tag{7.1.8}$$

To realise this EM step , we need to first consider the application of the unitary operator $U = \exp\left(2\pi i 2^{t-1} M\right)$ on $|b\rangle$, then construct a controlled-U gate so as to apply a dramatically simplified QPE algorithm that involves one qubit as shown in figure 7.1.3.

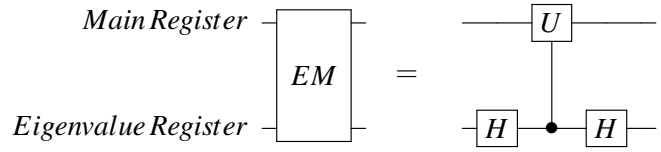

Figure 7.1.3: A quantum circuit for the Eigenvalue Marking (EM) subroutine, which is a simplified QPE algorithm that involves one qubit, where $U = \exp\left(2\pi i 2^{t-1} M\right)$.

The calculation below explicitly shows the application of the unitary operator $U = \exp\left(2\pi i 2^{t-1} M\right)$ on $|b\rangle$,

$$U|b\rangle = \exp\left(2\pi i 2^{t-1} M\right)|b\rangle \tag{7.1.9}$$

$$| \quad \text{Let } |b\rangle = \sum_{k=0}^{1} \beta_k |u_k\rangle,$$

$$| \quad \text{where } |u_k\rangle \text{ are normalized eigenvectors of } M \text{ for some real coefficients of } \beta_k.$$

$$= \sum_{k=0}^{1} \exp\left(2\pi i 2^{t-1} M\right) \beta_k |u_k\rangle \tag{7.1.10}$$

$$| \quad \text{Note the eigenequation: } M|u_k\rangle = \lambda_k |u_k\rangle = 0.\bar{X}k_1 |u_k\rangle.$$

$$= \sum_{k=0}^{1} \exp\left(2\pi i 2^{t-1} 0.\bar{X}k_1\right) \beta_k |u_k\rangle \tag{7.1.11}$$

$$= \sum_{k=0}^{1} \exp\left(2\pi i 2^{t-1} 0.x_1 x_2 \ldots x_{t-1} k_1\right) \beta_k |u_k\rangle \tag{7.1.12}$$

$$= \sum_{k=0}^{1} \exp\left[2\pi i 2^{t-1} \left(\frac{x_1}{2} + \frac{x_2}{2^2} + \ldots + \frac{x_{t-1}}{2^{t-1}} + \frac{k_1}{2^t}\right)\right] \beta_k |u_k\rangle \tag{7.1.13}$$

$$= \sum_{k=0}^{1} \exp\left[2\pi i \left(2^{t-2} x_1 + 2^{t-3} x_2 + \ldots + 2^0 x_{t-1} + 2^{-1} k_1\right)\right] \beta_k |u_k\rangle \tag{7.1.14}$$

$$| \quad \text{Since } 2^{t-1-j} x_j \text{ are integers for } j = 1, \ldots, t-1. \text{Then, } \exp\left[2\pi i 2^{t-1-j} x_j\right] = 1.$$

$$= \sum_{k=0}^{1} \exp\left[\pi i k_1\right] \beta_k |u_k\rangle \tag{7.1.15}$$

$$= \sum_{k=0}^{1} (-1)^{k_1} \beta_k |u_k\rangle \tag{7.1.16}$$

$$\therefore U|b\rangle = \beta_0 |u_0\rangle - \beta_1 |u_1\rangle \tag{7.1.17}$$

Essentially, the unitary operator U applies a phase flip from 1 to -1 on $|u_1\rangle$ and an identity $\mathbb{I}$ on $|u_0\rangle$.

The controlled-U gate can be deconstructed as $(R\otimes\mathbb{I})(C(Z))\left(R^{\dagger}\otimes\mathbb{I}\right)$, where R represents an arbitrary 2D unitary operator, such that $M = R\Lambda R^{\dagger}$, and Z is the Pauli-Z operator. The calculation below explains the gate deconstruction of controlled-U, start by considering the substitution of $M = R\Lambda R^{\dagger}$ during the analysis of the application of the unitary operation U to $|b\rangle$:

$$
\begin{aligned}
U|b\rangle &= \sum_{k=0}^{1}\exp\left(2\pi i 2^{t-1}M\right)\beta_k|u_k\rangle && (7.1.18)\\
&\mid \text{Subtitute } M = R\Lambda R^{\dagger}.\\
&= \sum_{k=0}^{1}\exp\left(2\pi i 2^{t-1}R\Lambda R^{\dagger}\right)\beta_k|u_k\rangle && (7.1.19)\\
&\mid \text{Using } f\left(UBU^{\dagger}\right) = Uf(B)U^{\dagger},\\
&\mid \text{for any analytical function } f \text{ acting on any operator } B \text{ with any unitary } U.\\
&= R\sum_{k=0}^{1}\exp\left(2\pi i 2^{t-1}\Lambda\right)R^{\dagger}\beta_k|u_k\rangle && (7.1.20)\\
&\mid \text{Note eigenequation: } R^{\dagger}\beta_k|u_k\rangle = r_k|k\rangle.\\
&= R\sum_{k=0}^{1}\exp\left(2\pi i 2^{t-1}\Lambda\right)r_k|k\rangle && (7.1.21)\\
&\mid \text{Note eigenequation: } \Lambda|k\rangle = \lambda_k|k\rangle = 0.\bar{X}k_1|k\rangle.\\
&= R\sum_{k=0}^{1}\exp\left[2\pi i 2^{t-1}0.\bar{X}k_1\right]r_k|k\rangle && (7.1.22)\\
&\mid \text{Following similar steps from equation 7.1.11 to 7.1.16.}\\
&= R\sum_{k=0}^{1}(-1)^{k_1}r_k|k\rangle && (7.1.23)\\
&\mid \underline{\text{This suggest that } \exp\left(2\pi i 2^{t-1}\Lambda\right) = Z.}\\
&= \sum_{k=0}^{1}(-1)^{k_1}\beta_k|u_k\rangle && (7.1.24)\\
\therefore U|b\rangle &= \beta_0|u_0\rangle - \beta_1|u_1\rangle && (7.1.25)
\end{aligned}
$$

The steps below details the truncated QPE algorithm that involves one qubit using controlled-U gate.

1. Apply a Hadamard transformation on the eigenvalue register.

$$
|b\rangle_m|0\rangle_e \xrightarrow{\mathbb{I}\otimes H} |b\rangle_m\left(\frac{|0\rangle_e+|1\rangle_e}{\sqrt{2}}\right) = \frac{1}{\sqrt{2}}\left(|b\rangle_m|0\rangle_e + |b\rangle_m|1\rangle_e\right) \qquad (7.1.26)
$$

2. Apply controlled-U on the main register, controlled by eigenvalue register, so that eigenvectors $|u_k\rangle_m$ are marked with qubits in the eigenvalue register .

$$
\begin{aligned}
&\xrightarrow{C(U)} \frac{1}{\sqrt{2}}\left(|b\rangle_m|0\rangle_e + U|b\rangle_m|1\rangle_e\right) && (7.1.27)\\
&\mid \text{Apply operator } U \text{ using equation 7.1.25} && (7.1.28)\\
&= \frac{1}{\sqrt{2}}\left[\left(\beta_0|u_0\rangle_m + \beta_1|u_1\rangle_m\right)|0\rangle_e + \left(\beta_0|u_0\rangle_m - \beta_1|u_1\rangle_m\right)|1\rangle_e\right]\\
&= \beta_0|u_0\rangle_m|+\rangle_e + \beta_1|u_1\rangle_m|-\rangle_e && (7.1.29)
\end{aligned}
$$

3. Apply an inverse Quantum Fourier Transformation on the eigenvalue register, which is just a Hadamard transformation for a single qubit, so that the eigenvalue marking qubits $|k_1\rangle_e$ are in the computational basis.

$$\xrightarrow{\mathbb{I}\otimes H} \quad \beta_0 |u_0\rangle_m |0\rangle_e + \beta_1 |u_1\rangle_m |1\rangle_e \tag{7.1.30}$$

In summary, step 2 is simplified down to being equivalent in applying $(R\otimes H)(C(Z))\left(R^\dagger \otimes H\right)$ on initial state $|b\rangle_m |0\rangle_e$. The quantum circuit for the EM step is shown in figure 7.1.4,

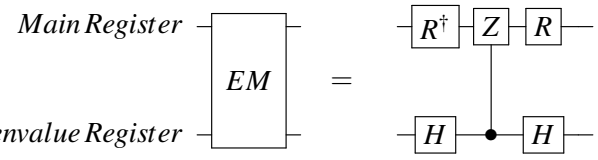

Figure 7.1.4: A quantum circuit for Eigenvalue Marking step decomposed using common quantum gates and gate R which represents an arbitrary 2D unitary operator, such that $M = R\Lambda R^\dagger$.

The third step is the Ancillary Rotation (AR) step, where an ancillary register, initialized in the state $|0\rangle_a$, is introduced. The goal is to apply amplitude coefficients $\frac{C}{\lambda_k}$ to the respective eigenstates $|u_k\rangle_m$ in the main register, where C is a real free parameter that can be set depending on circumstances to be described.

$$\beta_0 |u_0\rangle_m |0\rangle_e + \beta_1 |u_1\rangle_m |1\rangle_e \quad \xrightarrow{\text{Step 3}} \quad \sum_{k=0}^{1} \beta_k |u_k\rangle_m |k_1\rangle_e \left[\sqrt{1-\left(\frac{C}{\lambda_k}\right)^2}\,|0\rangle_a + \frac{C}{\lambda_k}|1\rangle_a\right] \tag{7.1.31}$$

Note the desired amplitude coefficient $\frac{C}{\lambda_k}$ is with the ancilla qubit $|1\rangle_a$. Consider applying a Rotate-Y $R_y(\theta_k)$ gate on the ancillary qubit $|0\rangle_a$ with an angle $\theta_k = 2\arcsin\left(\frac{C}{\lambda_k}\right)$,

$$|0\rangle_a \quad \xrightarrow{R_y(\theta)} \quad \cos\left(\frac{\theta_k}{2}\right)|0\rangle_a + \sin\left(\frac{\theta_k}{2}\right)|1\rangle_a \tag{7.1.32}$$

$$\Big| \quad \text{Let } \theta_k = 2\arcsin\left(\frac{C}{\lambda_k}\right).$$

$$= \quad \sqrt{1-\left(\frac{C}{\lambda_k}\right)^2}\,|0\rangle_a + \frac{C}{\lambda_k}|1\rangle_a \tag{7.1.33}$$

From equation 7.1.33, it is can be determined that the domain of the arcsin function limits parameter C to $0 < |C| \leq \min_k(\lambda_k)$. The domain of parameter C does not include zero as it would not fulfil the goal of introducing an amplitude coefficient $\frac{C}{\lambda_k}$ and it is upper bounded by $\min_k(\lambda_k)$. This limitation will have an effect on the probability of success of obtaining the solution state. The probability function will be given in the next subsection 7.1.2 on page 48. Particularly, it has a maximum probability less than 1 for any C. With two eigenstates $|u_0\rangle_m$, $|u_1\rangle_m$ in the main register being 'marked' by the qubits $|0\rangle_e$, $|1\rangle_e$ in the eigenvalue register by the previous step 2, we can use two controlled-$R_y(\theta_k)$ on the ancillary register, conditioned on the marking qubit in the eigenvalue register being $|0\rangle_e$, $|1\rangle_e$ respectively as shown figure 7.1.5.

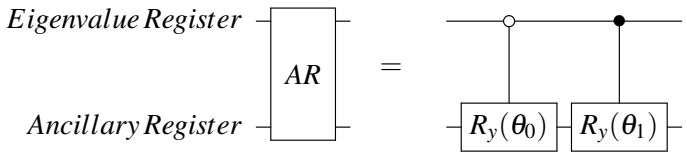

Figure 7.1.5: Ancillary Rotation (AR) quantum circuit using controlled-$R_y(\theta_k)$ with the purpose to introduce the amplitude coefficients $\frac{C}{\lambda_k}$ to the respective eigenstates $|u_k\rangle_m$ in the main register by using the eigenvalue marking qubits $|k_1\rangle_e$ in the eigenvalue register.

The explicit calculation below elaborates the processes that happened in step 3.

1. Append ancillary register $|0\rangle_a$

$$\beta_0 |u_0\rangle_m |0\rangle_e + \beta_1 |u_1\rangle_m |1\rangle_e \xrightarrow{\text{Append } |0\rangle_a} \beta_0 |u_0\rangle_m |0\rangle_e |0\rangle_a + \beta_1 |u_1\rangle_m |1\rangle_e |0\rangle_a \quad (7.1.34)$$

2. Apply controlled-$R_y(\theta_0)$ on the ancilla qubit, conditioned on the marking qubit in the eigenvalue register being $|0\rangle_e$.

$$\xrightarrow{\mathbb{I}\otimes C'[R_y(\theta_0)]} \beta_0 |u_0\rangle_m |0\rangle_e \left[\sqrt{1-\left(\frac{C}{\lambda_0}\right)^2}|0\rangle_a + \frac{C}{\lambda_0}|1\rangle_a\right] + \beta_1 |u_1\rangle_m |1\rangle_e |1\rangle_a \quad (7.1.35)$$

3. Apply controlled-$R_y(\theta_1)$ on the ancilla qubit, conditioned on the marking qubit in the eigenvalue register being $|1\rangle_e$.

$$\begin{aligned} \xrightarrow{I\otimes C[R_y(\theta_1)]} \quad & \beta_0 |u_0\rangle_m |0\rangle_e \left[\sqrt{1-\left(\frac{C}{\lambda_0}\right)^2}|0\rangle_a + \frac{C}{\lambda_0}|1\rangle_a\right] \\ & + \beta_1 |u_1\rangle_m |1\rangle_e \left[\sqrt{1-\left(\frac{C}{\lambda_1}\right)^2}|0\rangle_a + \frac{C}{\lambda_1}|1\rangle_a\right] \quad (7.1.36) \\ = \quad & \sum_{k=0}^{1} \beta_k |u_k\rangle_m |k_1\rangle_e \left[\sqrt{1-\left(\frac{C}{\lambda_k}\right)^2}|0\rangle_a + \frac{C}{\lambda_k}|1\rangle_a\right] \quad (7.1.37) \end{aligned}$$

The fourth step is the inverse application of second step, that is to unmark or uncompute the eigenvalue marking qubits $|k_1\rangle_e$ in the eigenvalue register as shown in figure 47 below.

$$\xrightarrow{\text{Step 4}} \sum_{k=0}^{1} \beta_k |u_k\rangle_m |0\rangle_e \left[\sqrt{1-\left(\frac{C}{\lambda_k}\right)^2}|0\rangle_a + \frac{C}{\lambda_k}|1\rangle_a\right] \quad (7.1.38)$$

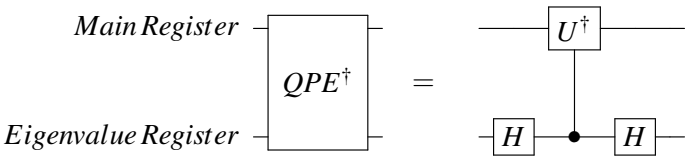

Figure 7.1.6: A quantum circuit for inverse Eigenvalue Marking step where $U = \exp\left(2\pi i 2^{t-1} M\right)$. Note that $H = H^\dagger$.

In the final fifth step, we have to post-select $|1\rangle_a$ ancilla qubit in the ancillary register. Then, keep only the main register and discard the rest to get the solution state $|x\rangle_m$, as shown in figure 7.1.7.

$$\xrightarrow{\text{Step 5}} \quad \sqrt{\frac{1}{\sum_{k=0}^{1}\left|\frac{\beta_k}{\lambda_k}\right|^2}}\sum_{k=0}^{1}\frac{\beta_k}{\lambda_k}|u_k\rangle = |x\rangle \tag{7.1.39}$$

$$\mid \quad \text{where the real free parameter } C \text{ was factored out.} \tag{7.1.40}$$

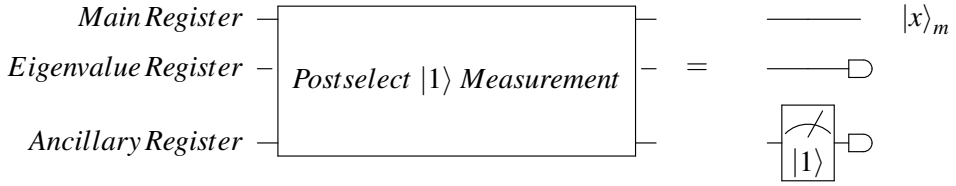

Figure 7.1.7: The post selection of $|1\rangle_a$ ancilla qubit measurement in the ancillary register, with the later discard of both eigenvalue and ancillary register while keeping only the main register.

It is important to note that the parameter C has no role in the normalisation of $|x\rangle$. However, the parameter C is important as it determines the probability of measuring $|1\rangle_a$ ancilla qubit in the ancillary register. Composing all the quantum circuits components together, yields a quantum circuit that solves the quantum linear problem with assumptions laid on page 41.

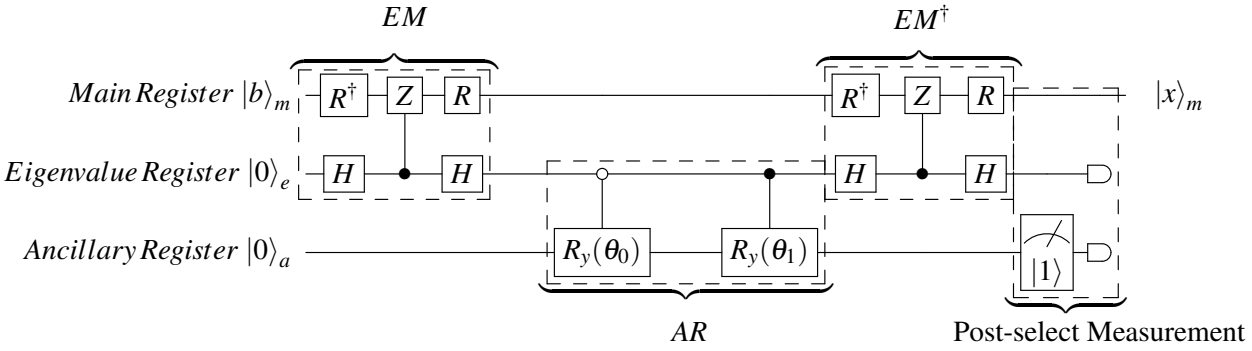

Figure 7.1.8: The quantum circuit of the 2D simplified quantum linear solver.

7.1.2 Theoretical Results

Consider the spectral decomposition of a 2D hermitian operator M,

$$M = R\Lambda R^\dagger \tag{7.1.41}$$

$$= R\begin{pmatrix} 0.\bar{X}0 & 0 \\ 0 & 0.\bar{X}1 \end{pmatrix}R^\dagger \tag{7.1.42}$$

$$\mid \quad \text{where the eigenvalues in the diagonal matrix are in fractional binary form.} \tag{7.1.43}$$

Suppose the given input pure 2D quantum state $|b\rangle$ is decomposed in terms of linear combinations of eigenvectors of U with complex coefficients β_j, that is $|b\rangle = \sum_{k=0}^{1}\beta_k|u_k\rangle$. The 2D simplified QLS algorithm is tasked to obtain an exact 2D solution state $|x\rangle$ such that, $|x\rangle = \frac{1}{d}M^{-1}|b\rangle$, where d is a

normalisation factor. The solution state $|x\rangle$ is found to be,

$$|x\rangle = \sqrt{\frac{1}{\left|\frac{\beta_0}{0.\bar{X}0}\right|^2+\left|\frac{\beta_1}{0.\bar{X}1}\right|^2}}\left[\frac{\beta_0}{0.\bar{X}0}|u_0\rangle+\frac{\beta_1}{0.\bar{X}1}|u_1\rangle\right] \tag{7.1.44}$$

The probability of successfully obtaining the solution state $P(|1\rangle_a)$ is,

$$P(|1\rangle_a) = \left|\frac{\beta_0 C}{\lambda_0}\right|^2+\left|\frac{\beta_1 C}{\lambda_1}\right|^2 \tag{7.1.45}$$

$$| \quad \text{Subsitute } C=\frac{1}{2^r} \text{ and } \lambda_j = 0.\bar{X}j.$$

$$= \frac{1}{2^{2r}}\left[\left|\frac{\beta_0}{0.\bar{X}0}\right|^2+\left|\frac{\beta_1}{0.\bar{X}1}\right|^2\right] \tag{7.1.46}$$

Note that we have **re-parametrised** C using r such that,

$$C = \frac{1}{2^r} \tag{7.1.47}$$

$$r = -\log_2(C) \tag{7.1.48}$$

This is done so that expressions and calculations involving binary numbers can be simplified. The maximum possible probability of success $P_{max}(|1\rangle_a)$ occurs at the maximum possible C_{max} such that, $C_{max} = \min_k(\lambda_k)$,

$$C_{max} = 0.\bar{X}0 \tag{7.1.49}$$

$$r_{min} = -\log_2(0.\bar{X}0) \tag{7.1.50}$$

$$\therefore P_{max}(|1\rangle_a) = \frac{1}{2^{2\log_2(0.\bar{X}0)}}\left[\left|\frac{\beta_0}{0.\bar{X}0}\right|^2+\left|\frac{\beta_1}{0.\bar{X}1}\right|^2\right] \tag{7.1.51}$$

$$= (0.\bar{X}0)^2\left[\left|\frac{\beta_0}{0.\bar{X}0}\right|^2+\left|\frac{\beta_1}{0.\bar{X}1}\right|^2\right] \tag{7.1.52}$$

7.1.3 2D Worked Example

Consider $R = H$ and $\bar{X} = 11$, in binary. Then, 2D hermitian operator M of interest has the following matrix representation:

$$M = \frac{1}{16}\begin{pmatrix} 13 & -1 \\ -1 & 13 \end{pmatrix} \tag{7.1.53}$$

where the eigenvalues are $\lambda_0 = 0.110 = \frac{3}{4}$ and $\lambda_1 = 0.111 = \frac{7}{8}$. Consider the input pure 2D quantum state $|b\rangle$ to be $|\uparrow_y\rangle$ as given in equation 3.5.2,

$$\begin{aligned} |b\rangle &= |\uparrow_y\rangle && (7.1.54)\\ &\quad | \quad \text{Linear decomposing in terms eigenvectors of } M. \\ &= \frac{1+i}{2}\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} + \frac{1-i}{2}\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{pmatrix} && (7.1.55)\\ &\quad | \quad \therefore \text{The cofficients are: } \beta_0 = \frac{1+i}{2},\ \beta_1 = \frac{1-i}{2}. \end{aligned}$$

The vector solution $\vec{x}$ for the classical linear problem $\vec{x} = M^{-1}\vec{b}$ is found to be,

$$\vec{x} = \frac{1}{21\sqrt{2}}\begin{pmatrix} 13+i \\ 1+13i \end{pmatrix} \quad (7.1.56)$$

The solution state $|x\rangle$ is found to be,

$$|x\rangle = \frac{1}{\sqrt{340}}\begin{pmatrix} 13+i \\ 1+13i \end{pmatrix} \quad (7.1.57)$$

The probability of success obtaining solution state $P(|1\rangle_a)$ is,

$$P(|1\rangle_a) = \frac{1}{4^r}\frac{680}{441} \quad (7.1.58)$$

The maximum possible probability of success $P_{max}(|1\rangle_a)$ occurs at the maximum possible C_{max} such that,

$$\begin{aligned} C_{max} &= \min_k(\lambda_k) = \frac{3}{4} && (7.1.59)\\ \Rightarrow \quad r_{min} &= -\log_2\frac{3}{4} \approx 0.415 \ \text{(3s.f)} && (7.1.60)\\ \therefore P_{max}(|1\rangle_a) &= \frac{85}{98} \approx 0.867 \ \text{(3s.f)} && (7.1.61) \end{aligned}$$

The quantum circuit for the 2D worked example is shown in figure 7.1.9 below.

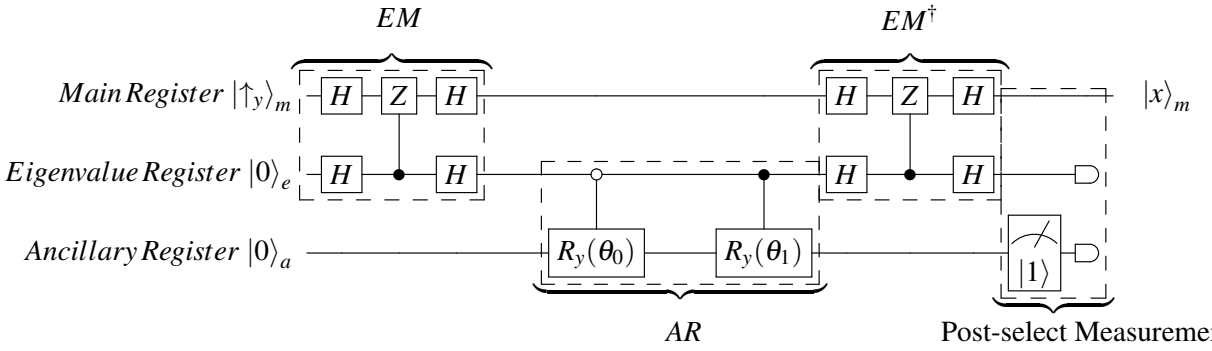

Figure 7.1.9: The quantum circuit of the simplified QLS for the 2D worked example where $\theta_0 = 2\arcsin\left(\frac{4}{3\cdot 2^r}\right)$ and $\theta_1 = 2\arcsin\left(\frac{8}{7\cdot 2^r}\right)$.

7.1.4 Experimental Results

Using IBM Qiskit quantum simulator, the quantum circuit given in figure 7.1.9 is implemented with the inputs prepared according to the 2D worked example given in subsection 7.1.3 on page 49. The table 7.1.1 below shows a breakdown of the quantum resources used this quantum circuit.

No.	Resource Type	Number of
1	Qubits	3
2	Circuit Depth	15
3	Single Qubit Arbitrary Unitary Gate $u3$	5
4	Hadamard Gate H	4
5	Measure Gate Pauli-Z	3
6	Controlled-Phase Gate $C[u1]$	2
7	Pauli-X Gate X	2
8	$C[u3]$	2
9	Total Quantum Operations (No. 3 to 8)	18

Table 7.1.1: A summary of the resource breakdown of the quantum circuit of the simplified QLS algorithm for the 2D worked example.

The independent variable is the real parameter $r = -\log_2(C)$, which is parametrized over a range between $0 \leq r < 4$ with increments of 0.01. The dependent variable is the probability of successes $P(|1\rangle_a)$ which is measured by calculating the fraction of the number of $|1\rangle_a$ ancilla qubit obtained over 10^4 independent runs. The theoretical probability of success $P(|1\rangle_a) = \frac{1}{4^r}\frac{680}{441}$ is numerically calculated and plotted against the experimental $P(|1\rangle_a)$ values as shown in figure 7.1.10. The average deviation of N number of experimental probability data points x_{ej} about in the actual value x_a is measured using the root mean squared percentage error (RMSPE) defined by the formula $RMSPE = \sqrt{\frac{\sum_{j=1}^{N}\left(\frac{x_a - x_{ej}}{x_a}\right)^2}{N}}$.

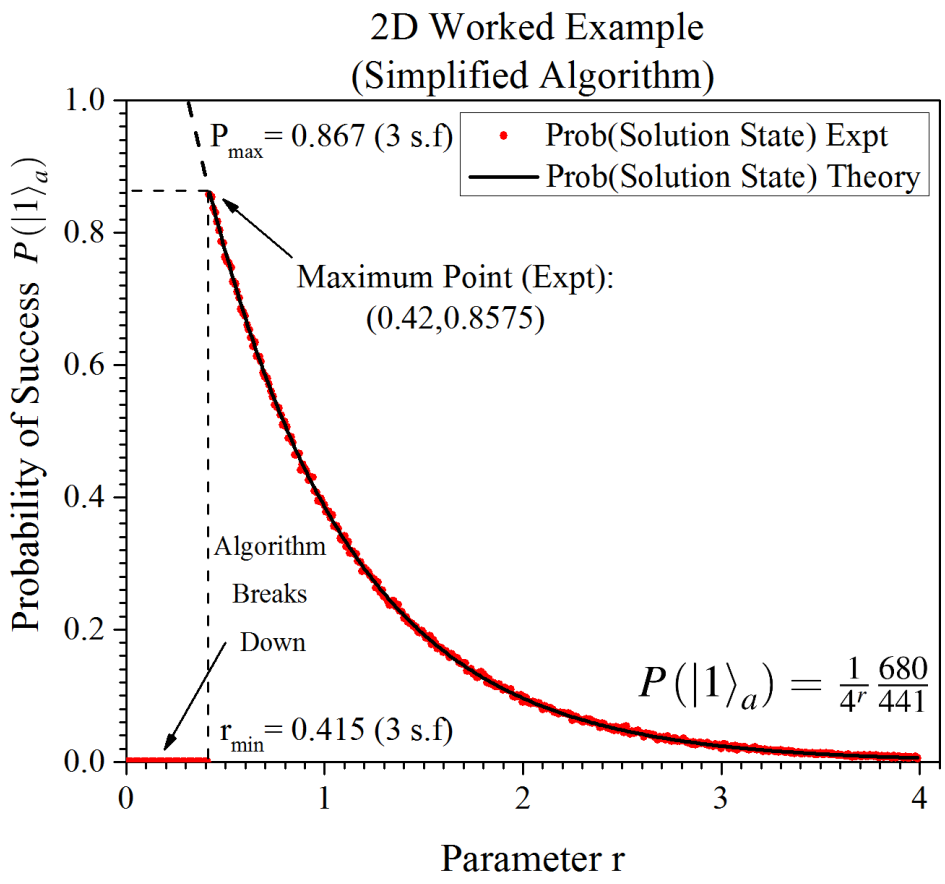

Figure 7.1.10: A graph of the probability of success $P(|1\rangle_a)$ against the parameter $r = -\log_2(C)$ for both experiment and theory using the simplified QLS for the 2D worked example.

The experimental $P(|1\rangle_a)$ plot given in figure 7.1.10 shows a decreasing trend from $r = 0.42$ to 3.99 with the corresponding probability $P(|1\rangle_a) = 0.8575 \pm 0.0001$ and 0.0054 ± 0.0001 respectively. The decreasing trend fits the theoretical $P(|1\rangle_a)$ prediction closely. It is found that these experimental values deviate about $RMSPE = \pm 6\%$ of the theoretical value on average. The maximum probability was experimentally found to be $P_{max}(|1\rangle_a) = 0.8575 \pm 0.0001$ corresponding to $r_{min} = 0.420 \pm 0.005$. Both of these experimental values matches the theoretical prediction as given in equation 7.1.61 and 7.1.60 respectively, within the uncertainty range. Note that the experimental probability data that ranges from $r = 0.00$ to $r = 0.41$ was manually adjusted by setting probability to zero to reflect the break down of the algorithm as the corresponding rotation angles θ do not exist for complex amplitudes.

The order of magnitude of the average simulation runtime of the quantum circuit is found to be about $10^{-3}s$, which is of two orders more than the order of magnitude of the average runtime of the classical solver (using the built-in python numpy linear solver) which is about $10^{-5}s$. The discussion of results is deferred to section 10.1 on page 90.

7.2 4D Case

The 4D case is largely similar in structure to the 2D case with the main difference in the quantum circuit construction of the EM and AR subroutines in step 2 and 3 respectively. The assumptions are

also similar to the 2D case with the exception that because the matrix of interest M is 4D, the four assumed eigenvalues are λ_0, λ_1, λ_2 and λ_3 in fractional binary representation of t bits, are: $\lambda_0 = 0.\bar{X}00$, $\lambda_1 = 0.\bar{X}01$, $\lambda_2 = 0.\bar{X}10$, $\lambda_3 = 0.\bar{X}11$. In general, $\lambda_k = 0.\bar{X}k_2k_1$ such that $k_2k_1 = \{00, 01, 10, 11\}$. The outline of the 4D simplified algorithm is almost the same as the 2D case, but with more qubits needed to store more information of the input quantum state $|b\rangle_m$ and eigenvalues marking qubits $|k_2k_1\rangle_e$.

Inputs:

- A 4D hermitian operator M and a pure 4D quantum state $|b\rangle$.
- Prepare three registers:
  - Main register $|0\rangle_m$ with 2 qubits to store the pure 4D quantum state $|b\rangle_m$.
  - Eigenvalue register $|0\rangle_e$ with 2 qubits to store the eigenvalue marking qubits $|k_2k_1\rangle_e$.
  - Ancillary register with a single ancilla qubit $|0\rangle_a$.

Outputs: A 4D solution state $|x\rangle = \frac{1}{\sqrt{\sum_{k=0}^{3}\left|\frac{\beta_k}{\lambda_k}\right|^2}} \sum_{k=0}^{3} \frac{\beta_k}{\lambda_k} |u_k\rangle$.

Steps:

1. Initialize the three registers $|b\rangle_m |0\rangle_e |0\rangle_a$.

$$|0\rangle_m |0\rangle_e |0\rangle_a \xrightarrow{\text{Initialize}} |b\rangle_m |0\rangle_e |0\rangle_a = \sum_{k=0}^{3} \beta_k |u_k\rangle_m |0\rangle_e |0\rangle_a \tag{7.2.1}$$

2. (Eigenvalue Marking, EM) Extract the two last bits k_2k_1 in the binary sequence of the eigenvalue λ_k, corresponding to the eigenstates $|u_k\rangle_m$ and store it in the eigenvalue register in a qubit binary representation as a eigenvalue marking qubit $|k_2k_1\rangle_e$.

$$\xrightarrow{\text{EM}} \sum_{k=0}^{3} \beta_k |u_k\rangle_m |k_2k_1\rangle_e |0\rangle_a \tag{7.2.2}$$

3. (Ancillary Rotation, AR) Apply controlled-$R_y(\theta_k)$, with parameter θ_k set such that $\sin\left(\frac{\theta_k}{2}\right) = \frac{C}{\lambda_k}$ for some real parameter C, on the ancilla qubit $|0\rangle_a$, conditioned on the eigenvalue marking qubit $|k_2k_1\rangle_e$ in the eigenvalue register.

$$\xrightarrow{C[R_y(\theta_k)]} \sum_{k=0}^{3} \beta_k |u_k\rangle_m |k_2k_1\rangle_e \left(\sqrt{1-\left(\frac{C}{\lambda_k}\right)^2} |0\rangle_a + \frac{C}{\lambda_k} |1\rangle_a\right) \tag{7.2.3}$$

4. Apply the inverse of step 2 to uncompute the eigenvalue marking qubit $|k_2k_1\rangle_e$ in the eigenvalue register.

$$\xrightarrow{\text{EM}^\dagger} \sum_{k=0}^{3} \beta_k |u_k\rangle_m |0\rangle_e \left(\sqrt{1-\left(\frac{C}{\lambda_k}\right)^2} |0\rangle_a + \frac{C}{\lambda_k} |1\rangle_a\right) \tag{7.2.4}$$

5. Do a post-selection $|1\rangle_a$ ancilla qubit measurement on the ancillary register using Pauli-Z operator, which has a probability $P(|1\rangle_a) = \sum_{k=0}^{3}\left|\frac{C\beta_k}{\lambda_k}\right|^2$. Discard all registers, but keep only

the main register which will be in the solution state $|x\rangle$.

$$\xrightarrow{\text{Post-Select } |1\rangle_a \text{ Measurement}} \quad \frac{1}{\sqrt{\sum_{k=0}^{3}\left|\frac{\beta_k}{\lambda_k}\right|^2}} \sum_{k=0}^{3} \frac{\beta_k}{\lambda_k} |u_k\rangle = |x\rangle \tag{7.2.5}$$

Figure 7.2.1 shows quantum circuit structure of the simplified 4D QLS as defined above.

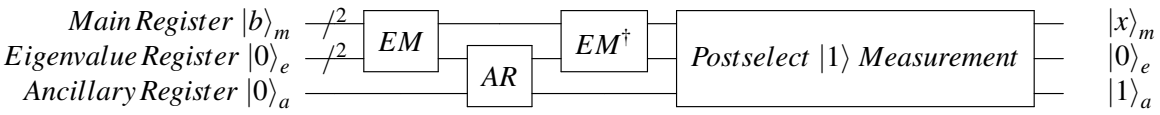

Figure 7.2.1: An outline structure of the simplified 4D QLS quantum circuit. The qubit registers from top to bottom are main, eigenvalue and ancillary register.

7.2.1 Explicit Circuit Decomposition

The first, fourth and fifth step is similar to that of the 2D case, and thus will not be repeated. We shall focus on the second and third step as it is slightly different in implementation compared to the 2D case. Recall the main purpose of the second step is to extract the two last bits k_2k_1 of the binary sequence of the eigenvalue λ_k, use it to mark the corresponding eigenvectors $|u_k\rangle_m$ as an eigenvalue marking qubit $|k_2k_1\rangle_e$ and store it in the eigenvalue register.

$$|u_0\rangle_m|00\rangle_e \longrightarrow |u_0\rangle_m|00\rangle_e \tag{7.2.6}$$

$$|u_1\rangle_m|00\rangle_e \longrightarrow |u_1\rangle_m|01\rangle_e \tag{7.2.7}$$

$$|u_2\rangle_m|00\rangle_e \longrightarrow |u_0\rangle_m|10\rangle_e \tag{7.2.8}$$

$$|u_3\rangle_m|00\rangle_e \longrightarrow |u_1\rangle_m|11\rangle_e \tag{7.2.9}$$

$$\sum_{k=0}^{3} \beta_k |u_k\rangle_m|00\rangle_e \xrightarrow{\text{Step 2}} \sum_{k=0}^{3} \beta_k |u_k\rangle_m|k_2k_1\rangle_e \tag{7.2.10}$$

Similar to the 2D case, 4D eigenvalue marking step is a truncated QPE algorithm that uses 2 qubits, as given figure 7.2.2 below.

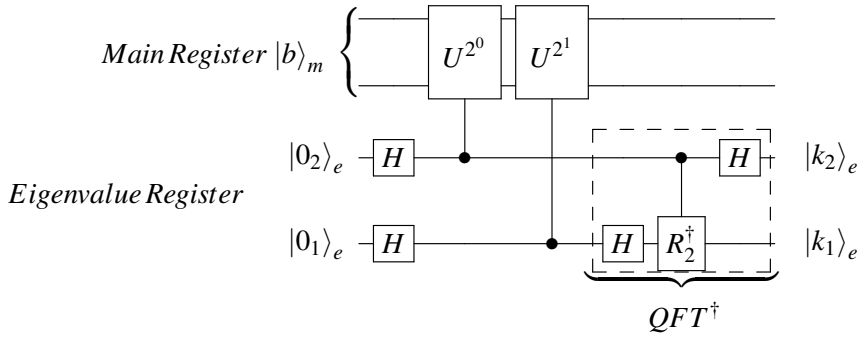

Figure 7.2.2: The quantum circuit of the eigenvalue marking step of the 4D case, where $U = \exp\left(2\pi i 2^{t-2} M\right)$ and $R_j = \begin{bmatrix} 1 & 0 \\ 0 & \exp\left(2\pi i 2^{-j}\right) \end{bmatrix}$.

To construct such a quantum circuit for this step, we need to first consider the application of the unitary operation $U^{2^1} = \exp\left(2\pi i 2^{t-1} M\right)$ and $U^{2^0} = \exp\left(2\pi i 2^{t-2} M\right)$ on $|b\rangle$. Then, we can construct

controlled versions of U^{2^1} and U^{2^0}. The calculation below explicitly shows the application of operator U^{2^1} and U^{2^0} on $|b\rangle$. The steps are largely the same in the 2D case, starting from equation 7.1.9.

$$
\begin{aligned}
U^{2^1}|b\rangle &= \exp\left(2\pi i 2^{t-1} M\right)|b\rangle && (7.2.11)\\
&\mid \text{Let } |b\rangle = \sum_{k=0}^{3} \beta_k |u_k\rangle, \\
&\mid \text{where } |u_k\rangle \text{ are normalized eigenvectors of } M \text{ for some real coefficients of } \beta_k. \\
&= \sum_{k=0}^{3} \exp\left(2\pi i 2^{t-1} M\right) \beta_k |u_k\rangle && (7.2.12)\\
&\mid \text{Note eigenequation: } M|u_k\rangle = \lambda_k |u_k\rangle. \\
&= \sum_{k=0}^{3} \exp\left(2\pi i 2^{t-1} 0.\bar{X} k_2 k_1\right) \beta_k |u_k\rangle && (7.2.13)\\
&= \sum_{k=0}^{3} \exp\left(2\pi i 2^{t-1} 0.x_1 x_2 \ldots x_{t-2} k_2 k_1\right) \beta_k |u_k\rangle && (7.2.14)\\
&= \sum_{k=0}^{3} \exp\left[2\pi i 2^{t-1}\left(\frac{x_1}{2} + \frac{x_2}{2^2} + \ldots + \frac{x_{t-2}}{2^{t-2}} + \frac{k_2}{2^{t-1}} + \frac{k_1}{2^t}\right)\right] \beta_k |u_k\rangle && (7.2.15)\\
&= \sum_{k=0}^{3} \exp\left[2\pi i \left(2^{t-2} x_1 + 2^{t-3} x_2 + \ldots + 2^1 x_{t-2} + 2^0 k_2 + 2^{-1} k_1\right)\right] \beta_k |u_k\rangle && (7.2.16)\\
&\mid \text{Since } 2^0 k_2 \text{ and } 2^{t-1-i} x_i \text{ are integers for } i = 1, \ldots, t-2. \\
&\mid \text{Then, } \exp\left[2\pi i 2^0 k_2\right] = \exp\left[2\pi i 2^{t-1-i} x_i\right] = 1. \\
&= \sum_{k=0}^{3} \exp\left[2\pi i 0.k_1\right] \beta_k |u_k\rangle && (7.2.17)\\
&= \sum_{k=0}^{t} (-1)^{k_1} \beta_k |u_k\rangle && (7.2.18)\\
\therefore U^{2^1}|b\rangle &= \beta_0 |u_0\rangle - \beta_1 |u_1\rangle + \beta_2 |u_2\rangle - \beta_3 |u_3\rangle && (7.2.19)
\end{aligned}
$$

$$
\begin{aligned}
U^{2^0}|b\rangle &= \exp\left(2\pi i 2^{t-2} M\right)|b\rangle && (7.2.20)\\
&= \sum_{k=0}^{3} \exp\left[2\pi i 2^{t-2}\left(\frac{x_1}{2} + \frac{x_2}{2^2} + \ldots + \frac{x_{t-2}}{2^{t-2}} + \frac{k_2}{2^{t-1}} + \frac{k_1}{2^t}\right)\right] \beta_k |u_k\rangle && (7.2.21)\\
&= \sum_{k=0}^{3} \exp\left[2\pi i \left(2^{t-3} x_1 + 2^{t-4} x_2 + \ldots + 2^0 x_{t-2} + 2^{-1} k_2 + 2^{-2} k_1\right)\right] \beta_k |u_k\rangle && (7.2.22)\\
&\mid \text{Since } 2^{t-2-i} x_i \text{ are integers for } i = 1, \ldots, t-2. \text{ Then, } \exp\left[2\pi i 2^{t-2-i} x_i\right] = 1. \\
&= \sum_{k=0}^{3} \exp\left[2\pi i\left(\frac{k_2}{2} + \frac{k_1}{4}\right)\right] \beta_k |u_k\rangle && (7.2.23)\\
&= \sum_{k=0}^{3} \exp\left[2\pi i 0.k_2 k_1\right] \beta_k |u_k\rangle && (7.2.24)\\
&= \sum_{k=0}^{3} (-1)^{k_2} (i)^{k_1} \beta_k |u_k\rangle \\
\therefore U^{2^0}|b\rangle &= \beta_0 |u_0\rangle + i\beta_1 |u_1\rangle - \beta_2 |u_2\rangle - i\beta_3 |u_3\rangle && (7.2.25)
\end{aligned}
$$

Note that U^{2^1} and U^{2^0}applies a set of phase shifts amplitudes to each eigenstate $|u_3\rangle$. To implement controlled-U^{2^1} and controlled-U^{2^0} using common quantum gates, recall the assumption that the eigen-decomposition of $M = R\Lambda R^\dagger$ is given in page 1. The purpose of $R^\dagger$ is to transform the eigenstates $|u_k\rangle$ of M into $|k-1\rangle$ where $|k-1\rangle$ is a 2^nD orthonormal basis from the set $\{|k-1\rangle : k = 1,2,\ldots,2^n\}$. Then, diagonal operators $\exp\left(2\pi i 2^{t-2}\Lambda\right)$ and $\exp\left(2\pi i 2^{t-1}\Lambda\right)$ must be responsible for applying the correct amplitudes to the basis, corresponding to the eigenstates accordingly. By applying the eigen-decomposition, using the operator $R^\dagger$and comparing the result given in equation 7.2.19 and 7.2.25, we easily can find the decomposition of diagonal operators,

$$\begin{aligned} U^{2^0} &= R\exp\left(2\pi i 2^{t-2}\Lambda\right)R^\dagger && (7.2.26)\\ &\mid \text{ Using the amplitudes of } |u_k\rangle \text{ in equation 7.2.25.} \\ \Rightarrow \quad \exp\left(2\pi i 2^{t-2}\Lambda\right) &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -i \end{pmatrix} = Z\otimes S && (7.2.27) \end{aligned}$$

$$\begin{aligned} U^{2^1} &= R\exp\left(2\pi i 2^{t-1}\Lambda\right)R^\dagger && (7.2.28)\\ &\mid \text{ Using the amplitudes of } |u_k\rangle \text{ in equation 7.2.19.} \\ \Rightarrow \quad \exp\left(2\pi i 2^{t-1}\Lambda\right) &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} = \mathbb{I}\otimes Z = (Z\otimes S)^{2^1} && (7.2.29) \end{aligned}$$

Then, the controlled versions of U^{2^1} and U^{2^0} gates can be deconstructed as shown in figure 7.2.3 below.

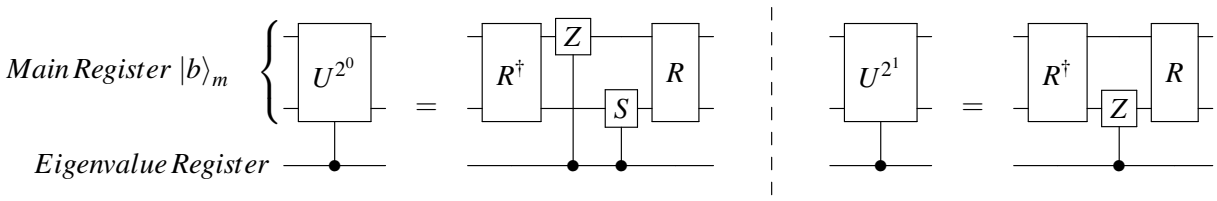

Figure 7.2.3: A quantum circuit of the decomposition of controlled-U^{2^0} and controlled-U^{2^1} gates using common quantum gates, Pauli-Z and Phase S, where R is a 2D arbitrary unitary quantum gate.

The steps below details the truncated QPE algorithm that uses 2 qubits using controlled versions of U^{2^1} and U^{2^0} gates.

1. Apply Walsh-Hadamard transformation on the eigenvalue register.

$$\begin{aligned} |b\rangle_m|0_2\rangle_e|0_1\rangle_e &\xrightarrow{\mathbb{I}\otimes H\otimes H} |b\rangle\left(\frac{|0_2\rangle_e+|1_2\rangle_e}{\sqrt{2}}\right)\left(\frac{|0_1\rangle_e+|1_1\rangle_e}{\sqrt{2}}\right) && (7.2.30)\\ &\mid \text{ Dropping the indexes.} \\ &= \frac{1}{2}|b\rangle\left(|00\rangle_e+|01\rangle_e+|10\rangle_e+|11\rangle_e\right) && (7.2.31) \end{aligned}$$

2. Apply controlled-U^{2^0} on the main register, controlled by marking qubit $|k_2\rangle_e$ in the eigenvalue register, using equation 7.2.24.

$$\xrightarrow{C\left(U^{2^0}\right)} \frac{1}{2}\left(\sum_{k=0}^{3}\beta_k|u_k\rangle_m|00\rangle_e+\sum_{k=0}^{3}\beta_k|u_k\rangle_e|01\rangle_e + \sum_{k=0}^{3}\exp[2\pi i 0.k_2k_1]\beta_k|u_k\rangle_m|10\rangle_e+\sum_{k=0}^{3}\exp[2\pi i 0.k_2k_1]\beta_k|u_k\rangle_e|11\rangle_e\right) \quad (7.2.32)$$

3. Apply controlled-U^{2^1} on the main register, controlled by marking qubit $|k_1\rangle_e$ in the eigenvalue register, using equation 7.2.17.

$$\xrightarrow{C\left(U^{2^1}\right)} \frac{1}{2}(\sum_{k=0}^{3}\beta_k|u_k\rangle_m|00\rangle_e+\sum_{k=0}^{3}\exp[2\pi i 0.k_1]\beta_k|u_k\rangle_m|01\rangle_e$$

$$= +\sum_{k=0}^{3}\beta_k\exp[2\pi i 0.k_2k_1]|u_k\rangle_m|10\rangle_e+\sum_{k=0}^{3}\exp[2\pi i 0.k_1]\exp[2\pi i 0.k_2k_1]\beta_k|u_k\rangle_m|11\rangle_e \quad (7.2.33)$$

$$= \sum_{k=0}^{3}\beta_k|u_k\rangle_m\left(\frac{|0\rangle_e+\exp[2\pi i 0.k_2k_1]|1\rangle_e}{\sqrt{2}}\right)\left(\frac{|0\rangle_e+\exp[2\pi i 0.k_1]|1\rangle_e}{\sqrt{2}}\right) \quad (7.2.34)$$

4. Apply an inverse Fourier transformation on the eigenvalue register using equation 4.2.8, so that the eigenvalue marking qubits $|k_2k_1\rangle_e$ are in the computational basis.

$$\xrightarrow{QFT^\dagger} \sum_{k=0}^{3}\beta_k|u_k\rangle|k_2k_1\rangle \quad (7.2.35)$$

$$= \beta_0|u_0\rangle|00\rangle+\beta_1|u_1\rangle|01\rangle+\beta_2|u_2\rangle|10\rangle+\beta_3|u_3\rangle|11\rangle \quad (7.2.36)$$

The third step is the AR step for the 4D case. The purpose of this step is apply the amplitude coefficients $\frac{C}{\lambda_k}$ to the eigenvalue qubit marker $|k_2k_1\rangle_e$, corresponding to the eigenstate $|u_k\rangle_m$. We use four controlled-$R_y(\theta_k)$ quantum gates, with parameter θ_k set such that $\sin\left(\frac{\theta_k}{2}\right)=\frac{C}{\lambda_k}$ for some real parameter C, on the ancilla qubit $|0\rangle_a$, conditioned by the eigenvalue marking qubit $|k_2k_1\rangle_e$.

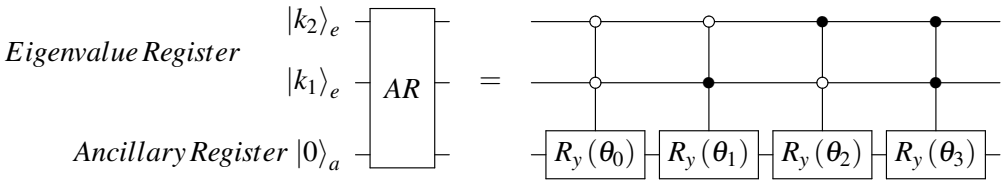

Figure 7.2.4: A quantum circuit of the decomposition of the 4D Ancillary Rotation step using controlled-$R_y(\theta_k)$ gates, with parameter θ_k set such that $\sin\left(\frac{\theta_k}{2}\right)=\frac{C}{\lambda_k}$ for some real parameter C.

Composing all the quantum circuits components together, yields a quantum circuit that solve the quantum linear problem with assumptions laid on page 41.

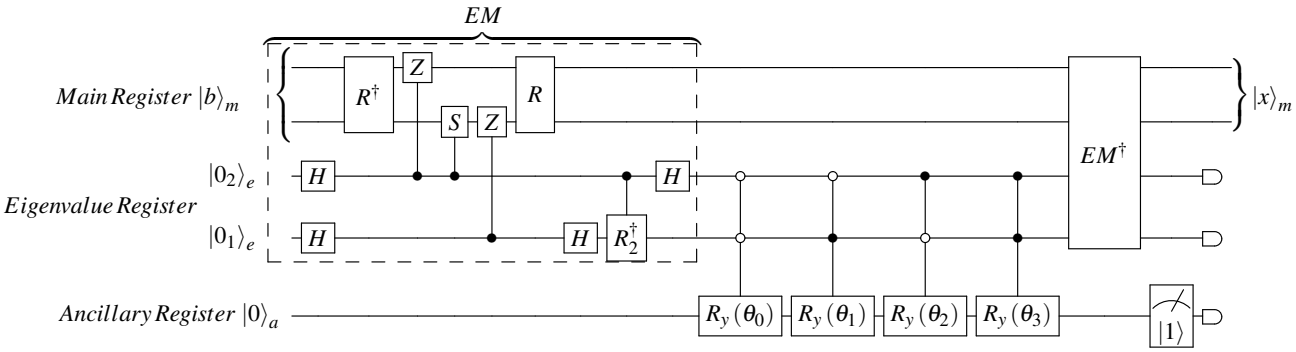

Figure 7.2.5: The quantum circuit of the 4D simplified QLS. Note that R represents an arbitrary 2D unitary operator acting on main register, such that $M = R\Lambda R^\dagger$, $R_j = \begin{bmatrix} 1 & 0 \\ 0 & \exp\left(2\pi i 2^{-j}\right) \end{bmatrix}$ are phase rotation gates acting on eigenvalue register and $R_y(\theta_k)$ are controlled Rotate-Y quantum gates acting on the ancillary qubit.

7.2.2 Theoretical Results

Consider the spectral decomposition of a 4D hermitian operator M,

$$M = R\Lambda R^\dagger \tag{7.2.37}$$

$$= R \begin{pmatrix} 0.\bar{X}00 & 0 & 0 & 0 \\ 0 & 0.\bar{X}01 & 0 & 0 \\ 0 & 0 & 0.\bar{X}10 & 0 \\ 0 & 0 & 0 & 0.\bar{X}11 \end{pmatrix} R^\dagger \tag{7.2.38}$$

| where the eigenvalues in the diagonal matrix are in fractional binary form.

Suppose the given input pure 4D quantum state $|b\rangle$ is decomposed in terms of linear combinations of eigenvectors of U with complex coefficients β_k, that is $|b\rangle = \sum_{k=0}^{3} \beta_k |u_k\rangle$. The 4D simplified QLS algorithm is tasked to obtain an exact 4D solution state $|x\rangle$ such that, $|x\rangle = \frac{1}{d} M^{-1} |b\rangle$, where d is a normalisation factor. The exact solution state $|x\rangle$ is found to be,

$$|x\rangle = \sqrt{\frac{1}{\sum_{k=0}^{3} \left| \frac{\beta_k}{0.\bar{X}k} \right|^2}} \sum_{j=0}^{3} \frac{\beta_k}{0.\bar{X}k} |u_k\rangle \tag{7.2.39}$$

The probability of successfully obtaining the exact solution state $P(|1\rangle_a)$ is,

$$P(|1\rangle_a) = \sum_{k=0}^{3} \left| \frac{\beta_k C}{0.\bar{X}k} \right|^2 \tag{7.2.40}$$

| Subsitute $C = \dfrac{1}{2^r}$ and $\lambda_j = 0.\bar{X}k$.

$$= \frac{1}{2^{2r}} \left[\sum_{k=0}^{3} \left| \frac{\beta_k}{0.\bar{X}k} \right|^2 \right] \tag{7.2.41}$$

Similar to 2D case on page 7.1.2, note that C has been re-parametrised using r such that $C = \frac{1}{2^r}$. The maximum possible probability of success $P_{max}(|1\rangle_a)$ occurs at the maximum possible C_{max} such that,

$C_{max} = \min_k(\lambda_k)$,

$$
\begin{aligned}
C_{max} &= 0.\bar{X}00 && (7.2.42)\\
r_{min} &= -\log_2(0.\bar{X}00) && (7.2.43)\\
\therefore P_{max}(|1\rangle_a) &= (0.\bar{X}00)^2 \left[\sum_{j=0}^{3} \left| \frac{\beta_k}{0.\bar{X}k} \right|^2 \right] && (7.2.44)
\end{aligned}
$$

7.2.3 4D Worked Example

Consider $R = H \otimes X_p$ and $\bar{X} = 1$ in binary, where X_p is the Pauli-X gate. Then, 4D hermitian operator M of interest has the following matrix representation:

$$
M = \frac{1}{16} \begin{pmatrix} 12 & 0 & -2 & 0 \\ 0 & 10 & 0 & -2 \\ -2 & 0 & 12 & 0 \\ 0 & -2 & 0 & 10 \end{pmatrix} \tag{7.2.45}
$$

where the eigenvalues are $\lambda_0 = 0.100 = \frac{4}{8}$, $\lambda_1 = 0.101 = \frac{5}{8}$, $\lambda_2 = 0.110 = \frac{7}{8}$, $\lambda_3 = 0.111$. Consider the input pure 4D quantum state $|b\rangle$ to be $|+\rangle \otimes |\uparrow_y\rangle$ where $|+\rangle$ and $|\uparrow_y\rangle$ are given in equation 3.5.2,

$$
\begin{aligned}
|b\rangle &= |+\rangle \otimes |\uparrow_y\rangle && (7.2.46)\\
&\quad | \quad \text{Linear decomposing in terms eigenvectors of } M.\\
&= \frac{i}{\sqrt{2}} \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} + \frac{1}{\sqrt{2}} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix} + 0 \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \frac{-1}{\sqrt{2}} \end{pmatrix} + 0 \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{-1}{\sqrt{2}} \\ 0 \end{pmatrix} && (7.2.47)\\
&\quad | \quad \therefore \text{The cofficients are: } \beta_0 = \frac{i}{\sqrt{2}},\ \beta_1 = \frac{1}{\sqrt{2}},\ \beta_2 = 0,\ \beta_3 = 0.
\end{aligned}
$$

The vector solution $\vec{x}$ for the classical linear problem $\vec{x} = M^{-1}\vec{b}$ is found to be,

$$
\vec{x} = \frac{1}{5} \begin{pmatrix} 4 \\ 5i \\ 4 \\ 5i \end{pmatrix} \tag{7.2.48}
$$

The solution state $|x\rangle$ is found to be

$$
|x\rangle = \sqrt{\frac{25}{82}} \begin{pmatrix} \frac{8}{10} \\ i \\ \frac{8}{10} \\ i \end{pmatrix} \tag{7.2.49}
$$

The probability of success obtaining solution state $P(|1\rangle_a)$ is,

$$P(|1\rangle_a) = \frac{1}{4^r}\frac{82}{25} \tag{7.2.50}$$

The maximum possible probability of success $P_{max}(|1\rangle_a)$ occurs at the maximum possible C_{max} such that,

$$C_{max} = \min(\lambda_j) = \frac{4}{8} \tag{7.2.51}$$

$$\Rightarrow \quad r_{min} = -\log_2 \frac{4}{8} = 1 \tag{7.2.52}$$

$$\therefore P_{max}(|1\rangle_a) = \frac{41}{50} = 0.82 \tag{7.2.53}$$

The quantum circuit for the 4D worked example is shown in figure 7.2.6 below.

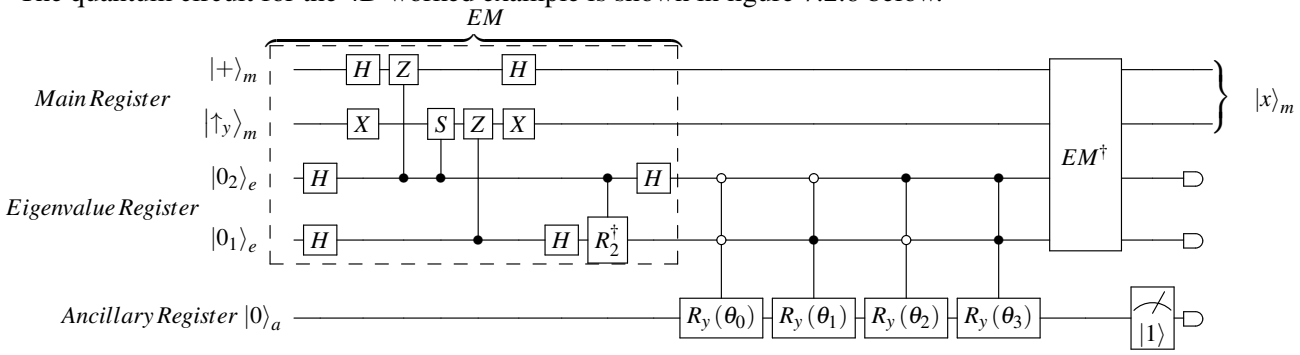

Figure 7.2.6: The quantum circuit of the simplified QLS for the 4D worked example where $\theta_0 = 2\arcsin\left(\frac{8}{4\cdot 2^r}\right)$, $\theta_1 = 2\arcsin\left(\frac{8}{5\cdot 2^r}\right)$, $\theta_2 = 2\arcsin\left(\frac{8}{6\cdot 2^r}\right)$ and $\theta_3 = 2\arcsin\left(\frac{8}{7\cdot 2^r}\right)$.

7.2.4 Experimental Results

Using IBM Qiskit quantum simulator, the quantum circuit given in figure 7.2.6 is implemented with the inputs prepared according to the 4D worked example given in subsection 7.2.3 on the previous page. The table 7.2.1 below shows a breakdown of the quantum resources used this quantum circuit.

No.	Resource Type	Number of
1	Qubits	6
2	Circuit Depth	37
3	Single Qubit Arbitrary Unitary Gate $u3$	10
4	Hadamard Gate H	8
5	Controlled-Phase Gate $C[u1]$	8
6	Pauli-X Gate X	8
7	Toffoli Gate CC_x	8
8	Measure Gate Pauli-Z	5
9	$C[u3]$	4
10	Total Quantum Operations (No. 3 to 12)	51

Table 7.2.1: A summary of the resource breakdown of the quantum circuit of the simplified QLS algorithm for the 4D worked example.

The independent variable is the real parameter $r = -\log_2(C)$, which is parametrized over a range

between $0 \leq r < 4$ with increments of 0.01. The dependent variable is the probability of successes $P(|1\rangle_a)$, which is measured by calculating the fraction of the number of $|1\rangle_a$ ancilla qubit obtained over 10^4 independent runs. The theoretical probability of success $P(|1\rangle_a) = \frac{1}{4^r}\frac{82}{25}$ is numerically calculated and plotted against the experimental $P(|1\rangle_a)$ values as shown in figure 7.2.7. The average deviation of N number of experimental probability data points x_{ej} about in the actual value x_a is measured using the root mean squared percentage error (RMSPE) defined by the formula $RMSPE = \sqrt{\frac{\sum_{j=1}^{N}\left(\frac{x_a - x_{ej}}{x_a}\right)^2}{N}}$.

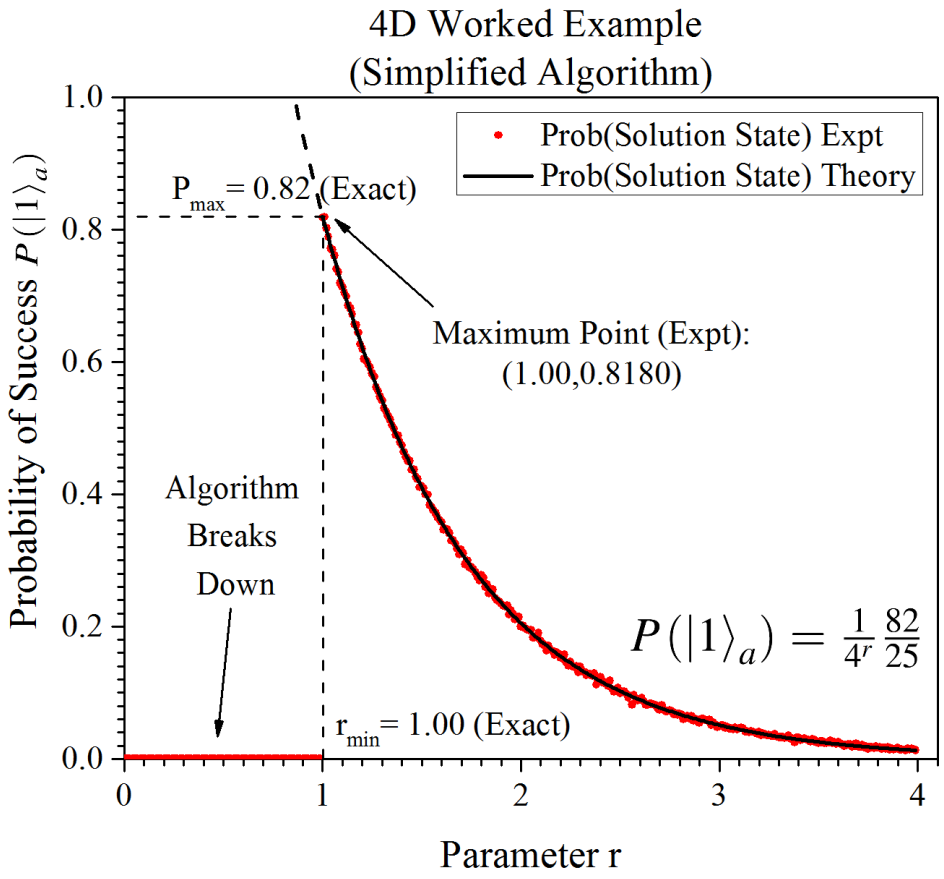

Figure 7.2.7: A graph of the probability of success $P(|1\rangle_a)$ against the parameter $r = -\log_2(C)$ for both experiment and theory using the simplified QLS for the 4D worked example.

The experimental $P(|1\rangle_a)$ plot given in figure 7.2.7 shows a decreasing trend from $r = 1.00$ to 3.99 with the corresponding probability $P(|1\rangle_a) = 0.8180 \pm 0.0001$ and 0.0132 ± 0.0001 respectively. The decreasing trend fits the theoretical $P(|1\rangle_a)$ prediction closely. It is found that these experimental values deviate about $RMSPE = \pm 5\%$ of the theoretical value on average. The maximum probability was experimentally found to be $P_{max}(|1\rangle_a) = 0.8180 \pm 0.0001$ corresponding to $r_{min} = 1.000 \pm 0.005$. Both of these experimental values matches the theoretical prediction as given in equation 7.2.53 and 7.2.52 respectively, within the uncertainty range. Note that the experimental data that ranges from parameter $r = 0.00$ to $r = 0.99$ was manually adjusted by setting probability to zero to reflect the break down of the algorithm as the corresponding rotation angles θ do not exist for complex amplitudes.

The order of magnitude of the average simulation runtime of the quantum circuit is found to be about $10^{-3}s$, which is of two orders more than the order of magnitude of the average runtime of the classical

solver (using the built-in python numpy linear solver) which is about $10^{-5}s$. The discussion of results is deferred to section 10.1 on page 90.

7.3 2^nD Case

By working through the 2D and 4D cases, it is possible to generalise the quantum algorithm, circuit and theoretical results for the 2^nD case, under same assumptions and approximations. One could easily deduce them simply by observation. Thus, the algorithm outline for the 2^nD case will not be given. However, we shall focus on the circuit decomposition and the theoretical results. Consider the spectral decomposition of a 2^nD hermitian matrix $M = R\Lambda R^\dagger$, where R is a 2^nD arbitrary unitary operator and Λ is a 2^nD diagonal matrix whose elements are distinct integer powers of two. The explicit 2^nD hermitian matrix M is,

$$M = R \begin{pmatrix} 0.\bar{X}0_n \dots 0_1 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & & \ddots & \vdots \\ 0 & & 0.\bar{X}k_n \dots k_1 & & 0 \\ \vdots & \ddots & & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0.\bar{X}1_n \dots 1_1 \end{pmatrix} R^\dagger \tag{7.3.1}$$

Suppose the given input pure 2^nD quantum state $|b\rangle$ is decomposed in terms of linear combinations of eigenvectors of R, $|u_k\rangle$ with complex coefficients β_k, that is $|b\rangle = \sum_{k=0}^{2^n-1} \beta_k |u_k\rangle$. The 2^nD simplified QLS algorithm is tasked to obtain an exact 2^nD solution state $|x\rangle$ such that $|x\rangle = \frac{1}{d} M^{-1} |b\rangle$, where d is a normalisation factor.

7.3.1 Explicit Circuit Decomposition

The first, fourth and fifth algorithm step is similar to that of 2D and 4D case. We only have to focus on generalising the quantum circuit decomposition of the second and third step. By generalising the second step in the 2D and 4D case, notice that truncated QPE algorithm in the EM step always require 2^nD quantum gates $U^{2^0}, U^{2^1}, \dots, U^{2^{n-1}}$ such that the last gate $U^{2^{n-1}}$ must be of the form

$$U^{2^{n-1}} = R \left(\underbrace{\mathbb{I} \otimes \dots \otimes \mathbb{I}}_{n-1 \text{ times}} \otimes Z \right) R^\dagger \tag{7.3.2}$$

where n is the number of qubits needed for the main register and for the eigenvalue register. In addition, notice that the Pauli-Z gate and Phase gate S, that were used in the construction of controlled-U^{2^j} gates in the 2D and 4D case can be simply stated in terms of $u1(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$, that is

$$Z = u1(\pi) \tag{7.3.3}$$

$$S = u1\left(\frac{\pi}{2}\right) \tag{7.3.4}$$

Therefore, so generalising to the 2^nD case, it is conjectured that the quantum gate $U^{2^{n-1-j}}$ can be constructed as,

$$U^{2^{n-1-j}} = R\left(\underbrace{\mathbb{I}\otimes\ldots\otimes\mathbb{I}}_{n-1-j \text{ times}}\otimes u1\left(\frac{\pi}{2^0}\right)\otimes\ldots\otimes u1\left(\frac{\pi}{2^{j-1}}\right)\otimes u1\left(\frac{\pi}{2^j}\right)\right)R^\dagger \tag{7.3.5}$$

Explicitly for $n = 1,2,3,4$, the required quantum gates $U^{2^{n-1-j}}$ for the truncated QPE algorithm gate decomposition is shown in table 7.3.1.

n	1	2	3	4
U^{2^0}	$u1(\pi)$	$u1(\pi)\otimes u1\left(\frac{\pi}{2}\right)$	$u1(\pi)\otimes u1\left(\frac{\pi}{2}\right)\otimes u1\left(\frac{\pi}{4}\right)$	$u1(\pi)\otimes u1\left(\frac{\pi}{2}\right)\otimes u1\left(\frac{\pi}{4}\right)\otimes u1\left(\frac{\pi}{8}\right)$
U^{2^1}	-	$\mathbb{I}\otimes u1(\pi)$	$\mathbb{I}\otimes u1(\pi)\otimes u1\left(\frac{\pi}{2}\right)$	$\mathbb{I}\otimes u1(\pi)\otimes u1\left(\frac{\pi}{2}\right)\otimes u1\left(\frac{\pi}{4}\right)$
U^{2^2}	-	-	$\mathbb{I}\otimes\mathbb{I}\otimes u1(\pi)$	$\mathbb{I}\otimes\mathbb{I}\otimes u1(\pi)\otimes u1\left(\frac{\pi}{2}\right)$
U^{2^3}	-	-	-	$\mathbb{I}\otimes\mathbb{I}\otimes\mathbb{I}\otimes u1(\pi)$

Table 7.3.1: A table of the quantum gates $U^{2^{n-1-j}}$ for 2^n dimension truncated Quantum Phase Estimation subroutine for the simplified quantum linear solver for $n = 1,2,3,4$. Note that the 2^n unitary quantum gates R and $R^\dagger$ are dropped to avoid cumbersome writing.

In general, the quantum circuit for the truncated QPE is given as,

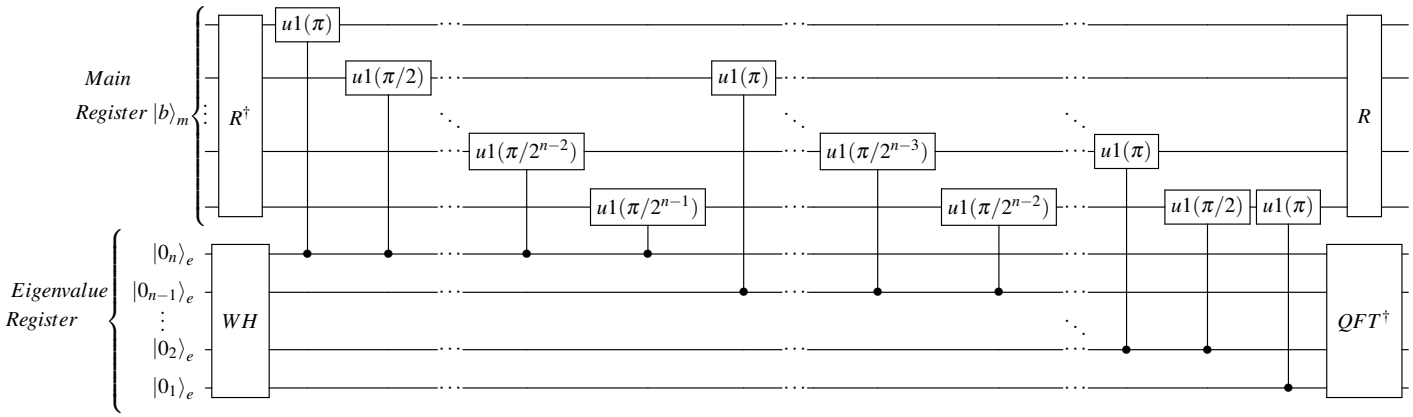

Figure 7.3.1: A quantum circuit of the decomposition of the truncated Quantum Phase Estimation in the Eigenvalue Marking step for 2^n dimensions. WH and QFT represents Walsh-Hadamard and Quantum Fourier Transform subroutines respectively (*See section 4.3 for more details*).

Generalising the third step for 2^nD is relatively simple as it is just a matter of applying the correct amplitude coefficients based on the condition of the eigenvalue marking qubits, which are in the computational basis of 2^nD. Without explicit derivation, the quantum circuit for the AR subroutine is given as,

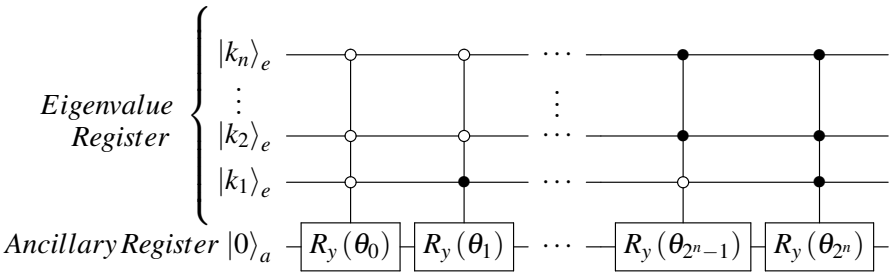

Figure 7.3.2: A quantum circuit of the decomposition of the Ancillary Rotation step using controlled-$R_y(\theta_k)$ gates, with parameter θ_k set such that $\sin\left(\frac{\theta_k}{2}\right) = \frac{C}{\lambda_k}$ for some real parameter C, for 2^n dimensions.

7.3.2 Theoretical Results

Consider the spectral decomposition of a 2^nD hermitian operator M, such that $M = R\Lambda R^\dagger$. Suppose the given input pure 2^nD quantum state $|b\rangle$ is decomposed in terms of linear combinations of eigenvectors of U with complex coefficients β_k, that is $|b\rangle = \sum_{k=0}^{2^n-1} \beta_k |u_k\rangle$. The 2^nD simplified QLS algorithm is tasked to obtain an exact 2^nD solution state $|x\rangle$ such that, $|x\rangle = \frac{1}{d} M^{-1} |b\rangle$ where d is a normalisation factor. The solution state $|x\rangle$ is found to be,

$$|x\rangle = \sqrt{\frac{1}{\sum_{k=0}^{2^n-1} \left|\frac{\beta_k}{0.\bar{X}k}\right|^2}} \sum_{j=0}^{2^n-1} \frac{\beta_k}{0.\bar{X}k} |u_k\rangle \tag{7.3.6}$$

The probability of successfully obtaining the exact solution state $P(|1\rangle_a)$ is,

$$P(|1\rangle_a) = \sum_{k=0}^{2^n-1} \left|\frac{\beta_k C}{\lambda_j}\right|^2 \tag{7.3.7}$$

$$| \quad \text{Subsitute } C = \frac{1}{2^r} \text{ and } \lambda_k = 0.\bar{X}k.$$

$$= \frac{1}{2^{2r}} \left[\sum_{k=0}^{2^n-1} \left|\frac{\beta_k}{2^{r+1}}\right|^2\right] \tag{7.3.8}$$

Similar to 2D case on page 7.1.2, note that C has been re-parametrised using r such that $C = \frac{1}{2^r}$. The maximum possible probability of success $P_{max}(|1\rangle_a)$ occurs at the maximum possible C_{max} such that, $C_{max} = \min_k(\lambda_k)$,

$$C_{max} = 0.\bar{X}0_n \ldots 0_1 \tag{7.3.9}$$

$$r_{min} = \log_2(0.\bar{X}0_n \ldots 0_1) \tag{7.3.10}$$

$$\therefore P_{max}(|1\rangle_a) = (0.\bar{X}0_n \ldots 0_1)^2 \left[\sum_{k=0}^{2^n} \left|\frac{\beta_k}{0.\bar{X}k}\right|^2\right] \tag{7.3.11}$$

Hence, in general for arbitrary dimension, the free parameter r has domain $[-\log_2(0.\bar{X}\mathbf{0}), \infty)$, where $\mathbf{0} = 0_n \ldots 0_1$ represents the value zero, using n number of bits.

Chapter 8

Full Quantum Linear Solver

The full version of the Quantum Linear Solver (QLS) algorithm is an enhanced QLS algorithm that includes the 1^{st} order Taylor approximation of the Ancillary Rotation (AR) subroutine, first introduced in section 6.3. The main purpose of this enhancement to avoid the use of parametrized angles θ_j for controlled-$R_y(\theta_k)$ gates which requires the knowledge of the unknown eigenvalues λ_k of the hermitian operator M as $\sin\left(\frac{\theta_k}{2}\right) = \frac{C}{\lambda_k}$ from equation 5.3.25 under section 5.3. It is called the 'full version' as it can used to solve for solution state $|x\rangle = \frac{1}{d}M^{-1}|b\rangle$, where d is a normalization factor, for any input hermitian operator M and quantum state $|b\rangle$ without having to change the structure of the algorithm. However, in this project, the implementation of the full QLS algorithm will be simplified to reduce to the number of qubits and this requires several more assumptions.

First, assume that quantum gates R and $R^\dagger$ and the controlled-exp$(2\pi i 2^{-t}\Lambda)$ are given, where $M = R\Lambda R^\dagger$. Second, assume that the eigenvalue of M are integer powers of two. That is, $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$, where $\lambda_k = 2^{p_k}$ and $p_k \in \mathbb{Z}$. A key thing to note is that these assumptions will not be used together to design the whole quantum circuit of the algorithm as doing so would ruin the generality or scalability of the circuit. In other words, the assumptions will be implemented within the relevant quantum subroutines. A summary is given in table 8.0.1 showing the assumptions and enhancements used in the full QLS algorithm in its respective subroutines.

	Name	Type	Purpose
1.	Given R, $R^\dagger$ and controlled-exp$(2\pi i 2^{-t}\Lambda)$ quantum gates	Assumption	To simplify HS in QPE.
2.	Integer powers of two eigenvalues	Assumption	To simplify EI and avoid an extra qubit register.
3.	1^{st} order Taylor approximation of AR	Enhancement	To avoid the use of unknown eigenvalues.

Table 8.0.1: A summary table showing the enhancement and assumptions used for simplifying implementation of the full QLS algorithm. Legend: Hamiltonian Simulation (HS), Quantum Phase Estimation (QPE), Eigenvalue Inversion (EI), Ancillary Rotation (AR).

The outline of the full QLS algorithm is given explicitly below.

Inputs:

- A 2^n dimensional hermitian operator M and a pure 2^n dimensional quantum state $|b\rangle$.
- Prepare three registers:
  - Main register $|0\rangle_m$ with n qubits to store the pure 2^n dimensional quantum state $|b\rangle_m$.
  - Eigenvalue register $|0\rangle_e$ with t qubits to sufficiently store the all eigenvalues λ_k in qubit binary representation $|\lambda_k\rangle_e$ up to a desired precision of t bits.
  - Ancillary register with a single qubit $|0\rangle_a$.

Outputs: A 2^n dimensional 1^{st} order Taylor approximated solution state $|\tilde{x}\rangle$.

$$|\tilde{x}\rangle \quad = \quad \sqrt{\frac{1}{\sum_{k=1}^{2^n}\left|\beta_k \sin\left(\frac{C}{\lambda_k}\right)\right|^2}} \sum_{k=1}^{2^n} \beta_k \sin\left(\frac{C}{\lambda_k}\right)|u_k\rangle_m \tag{8.0.1}$$

Steps:

1. Initialize the registers from $|0\rangle_m|0\rangle_e|0\rangle_a$ to $|b\rangle_m|0\rangle_e|0\rangle_a$.

$$|0\rangle_m|0\rangle_e|0\rangle_a \quad \xrightarrow{\text{Initialize}} \quad |b\rangle_m|0\rangle_e|0\rangle_a = \sum_{k=1}^{2^n} \beta_k|u_k\rangle_m|0\rangle_e|0\rangle_a \tag{8.0.2}$$

2. (Quantum Phase Estimation, QPE) Compute the eigenvalues and store them in the eigenvalue register by applying a Quantum Phase Estimation subroutine using the controlled-$\exp(2\pi i 2^{-t}M)$ quantum gate.

$$\xrightarrow{\text{QPE}} \quad \sum_{k=1}^{2^n} \beta_k|u_k\rangle_m|\lambda_k\rangle_e|0\rangle_a \tag{8.0.3}$$

3. (Eigenvalue Inversion, EI) Compute the inverted eigenvalues and store them in the inverted register.

$$\xrightarrow{\text{EI}} \quad \sum_{k=1}^{2^n} \beta_k|u_k\rangle_m\left|\frac{1}{\lambda_k}\right\rangle_e|0\rangle_a \tag{8.0.4}$$

4. (1^{st} order Ancillary Rotation, $\tilde{\text{AR}}$) Compute the amplitude coefficients $\frac{C}{\lambda_k}$ and apply it to the respective inverse eigenvalue qubit $\left|\frac{1}{\lambda_k}\right\rangle_e$, for a real parameter C such that $\frac{C}{\lambda_k} < 1$, for all k.

$$\xrightarrow{\tilde{\text{AR}}} \quad \sum_{k=1}^{2^n} \beta_k|u_k\rangle_m\left|\frac{1}{\lambda_k}\right\rangle_e\left(\cos\left(\frac{C}{\lambda_k}\right)|0\rangle_a + \sin\left(\frac{C}{\lambda_k}\right)|1\rangle_a\right) \tag{8.0.5}$$

5. Apply the inverse QPE and EI to uncompute the both the eigenvalues and inverted ones in the eigenvalue register respectively.

$$\xrightarrow{(\text{EI}^\dagger)(\text{QPE}^\dagger)} \quad \sum_{k=1}^{2^n} \beta_k|u_k\rangle_m|0\rangle_e\left(\cos\left(\frac{C}{\lambda_k}\right)|0\rangle_a + \sin\left(\frac{C}{\lambda_k}\right)|1\rangle_a\right) \tag{8.0.6}$$

6. Do a post-selection $|1\rangle_a$ qubit measurement on the ancillary register using Pauli-Z operator, which has a probability $P(|1\rangle_a) = \sum_{k=1}^{2^n}\left|\beta_k \sin\left(\frac{C}{\lambda_k}\right)\right|^2$. Discard all registers, but keep only

the main register that will be in the 1^{st} order Taylor approximated solution state $|\tilde{x}\rangle$.

$$\xrightarrow{\text{Post-Select } |1\rangle_a \text{ Measurement}} \sqrt{\frac{1}{\sum_{k=1}^{2^n} \left|\beta_k \sin\left(\frac{C}{\lambda_k}\right)\right|^2}} \sum_{k=1}^{2^n} \beta_k \sin\left(\frac{C}{\lambda_k}\right) |u_k\rangle_m = |\tilde{x}\rangle \quad (8.0.7)$$

Figure 8.0.1 shows the general quantum circuit structure of the full QLS algorithm as defined above.

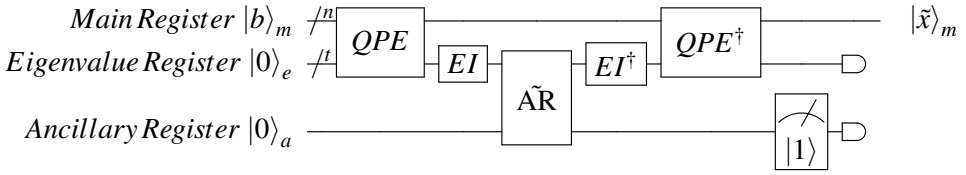

Figure 8.0.1: The quantum circuit structure of the full QLS algorithm. The qubit registers from top to bottom are main, eigenvalue and ancillary register. Legend: Quantum Phase Estimation (QPE), Eigenvalue Inversion (EI), Ancillary Rotation (AR).

8.1 Quantum Subroutines Simplifications

In this section, the following subroutines will be simplified and explained in detail: Quantum Phase Estimation, Hamiltonian Simulation and Quantum Eigenvalue Inversion. Note that any assumptions used in these subroutines are not used elsewhere in the algorithm.

8.1.1 Quantum Phase Estimation

Quantum Phase Estimation (QPE), in essence, is to obtain a quantum state $|\phi\rangle_e$ that is a qubit binary representation of ϕ, where ϕ is a phase factor of eigenvalue $\delta = \exp(2\pi i 2^{-t}\phi)$ of the unitary operator U. If we let $U = \exp(2\pi i 2^{-t} M)$, where M is a hermitian operator, then it will have eigenvalues $\delta_k = \exp(2\pi i 2^{-t}\lambda_k)$ with the corresponding eigenvalues of M, λ_k. Using QPE, with $U = \exp(2\pi i 2^{-t} M)$, will allow the qubit binary representation of the eigenvalue λ_k to be extracted and stored as a quantum state $|\lambda_k\rangle_e$ in the eigenvalue register. A basic introduction to QPE can be found on this thesis in section 4.3 on page 21. The following outline the QPE algorithm.

Inputs:

- A 2^n dimensional unitary operator $U = \exp(2\pi i 2^{-t} M)$ and a pure 2^n dimensional quantum state $|b\rangle_m = \sum_{k=1}^{2^n} \beta_k |u_k\rangle_m$, such that, $U|b\rangle_m = \sum_{j=1}^{2^n} \beta_k \exp(2\pi i 2^{-t}\lambda_k)|u_k\rangle_m$ where $M|u_k\rangle = \lambda_k|u_k\rangle$ and $\sum_{k=1}^{2^n} \beta_k = 1$.
- Prepare the main register $|0\rangle_m$ with n qubits to store the pure 2^n dimensional quantum state $|b\rangle_m$.
- Eigenvalue register $|0\rangle_e$ with t qubits to sufficiently store the all eigenvalues λ_k in qubit binary representation $|\lambda_k\rangle_e$ up to a desired precision of t bits.

Outputs:

- A quantum state $|\lambda_k\rangle_e$ that is a qubit binary representation of the eigenvalue λ_k.

Implementation steps:

1. Initialize quantum state $|b\rangle_m |0\rangle_e$ from $|0\rangle_m |0\rangle_e$.

$$|0\rangle_m |0\rangle_e \xrightarrow{\text{Initialize}} |b\rangle_m |0\rangle_e \tag{8.1.1}$$

2. (Walsh–Hadamard Transformation, WS) Apply Hadamard gates to all qubits in the eigenvalue register to create superposition of all possible states.

$$\xrightarrow{\text{WS}} \frac{1}{\sqrt{2^t}} \bigotimes_{j=1}^{t} \left(|b\rangle_m |0_j\rangle_e + |b\rangle_m |1_j\rangle_e \right) \tag{8.1.2}$$

3. (Hamiltonian Simulation, HS) Apply a sequence of controlled-$U^{2^{t-j}}$ gates on the main register conditioned on the j^{th}qubit of the eigenvalue register .

$$\xrightarrow{\text{HS}} \frac{1}{\sqrt{2^t}} \bigotimes_{j=1}^{t} \left(\frac{|b\rangle_m |0_j\rangle_e + \left(U^{2^{t-j}} \oplus \mathbb{I}\right) \sum_{k=1}^{2^n} \beta_k |u_k\rangle_m |1_j\rangle_e}{\sqrt{2}} \right) \tag{8.1.3}$$

$$= \frac{1}{\sqrt{2^t}} \left\{ \sum_{k=1}^{2^n} \beta_k |u_k\rangle_m \left(|0_t\rangle_e + \exp\left(2\pi i 0.\lambda_{tk} \ldots \lambda_{1k}\right) |1_t\rangle_e\right) \otimes \ldots \otimes \left(|0_1\rangle_e + \exp\left(2\pi i 0.\lambda_{1k}\right) |1_1\rangle_e\right) \right\} \tag{8.1.4}$$

4. (Inverse QFT) Apply inverse Quantum Fourier Transform on the eigenvalue register.

$$\xrightarrow{\text{QFT}^\dagger} \sum_{k=1}^{2^n} \beta_k |u_k\rangle_m |\lambda_{tk} \ldots \lambda_{1k}\rangle_e = \sum_{k=1}^{2^n} \beta_k |u_k\rangle_m |\lambda_k\rangle_e \tag{8.1.5}$$

8.1.2 Hamiltonian Simulation

If $U = \exp\left(2\pi i 2^{-t} M\right)$, where M is Hermitian, then the third step of QPE at equation 8.1.3 is called the Hamiltonian Simulation (HS) subroutine. Any hermitian matrix M can be decompose spectrally as $M = R\Lambda R^\dagger$, where R is unitary whose columns are the eigenstates and Λ is a diagonal matrix whose elements are the eigenvalues of M. Using this spectral decomposition and the property, $f\left(U'BU'^\dagger\right) = U' f(B) U'^\dagger$, for any analytical function f of any operator B with any unitary U', $U = \exp\left(2\pi i 2^{-t} M\right)$ is re-expressed as $U = R\exp\left(2\pi i 2^{-t}\Lambda\right) R^\dagger$. To implement HS on a quantum circuit, controlled-$U^{2^{t-j}}$ quantum gates are required. We need to first consider the decomposition of $U^{2^{t-j}}$ gate so that controlled-$U^{2^{t-j}}$ gates can be decomposed. $U^{2^{t-j}}$ gate can be decomposed as follows,

$$U^{2^{t-j}} = \underbrace{\left[R\exp\left(2\pi i 2^{-t}\Lambda\right) R^\dagger\right] \cdot \ldots \cdot \left[R\exp\left(2\pi i 2^{-t}\Lambda\right) R^\dagger\right]}_{2^{t-j} \text{ multiplicative product terms}} \tag{8.1.6}$$

$$\mid \quad \text{multiply } \left[R\exp\left(2\pi i 2^{-t}\Lambda\right) R^\dagger\right] \; 2^{t-j} \text{ times and using } RR^\dagger = \mathbb{I}$$

$$= R\left[\exp\left(2\pi i 2^{-t}\Lambda\right)\right]^{2^{t-j}} R^\dagger \tag{8.1.7}$$

Using the decomposition of $U^{2^{t-j}}$ gate above, we can construct controlled-$U^{2^{t-j}}$ using quantum gates of R, $R^\dagger$ and controlled-$\exp\left(2\pi i 2^{-t}\Lambda\right)$ as shown in figure 8.1.1 by noticing that if we apply controlled-

$U^{2^{t-j}}$ on $|b\rangle_m$ on the condition of $|1\rangle_e$, it results in following properties,

$$|b\rangle_m|0\rangle_e \xrightarrow{C\left(U^{2^{t-j}}\right)} |b\rangle_m|0\rangle_e \quad | \quad \text{Equivalent to applying } RR^\dagger = \mathbb{I} \tag{8.1.8}$$

$$= RR^\dagger|b\rangle_m|0\rangle_e \tag{8.1.9}$$

$$|b\rangle_m|1\rangle_e \xrightarrow{C\left(U^{2^{t-j}}\right)} U^{2^{t-j}}|b\rangle_m|1\rangle_e \tag{8.1.10}$$

$$= R\left[\exp\left(2\pi i 2^{-t}\Lambda\right)\right]^{2^{t-j}} R^\dagger|b\rangle_m|1\rangle_e \tag{8.1.11}$$

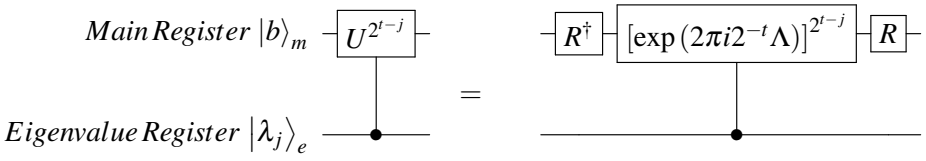

Figure 8.1.1: Decomposition of controlled-$U^{2^{t-j}}$ gate.

where controlled-$[\exp(2\pi i 2^{-t}\Lambda)]^{2^{t-j}}$ can be implemented using controlled-$[\exp(2\pi i 2^{-t}\Lambda)]$ gate 2^{t-j} successive times as shown in figure 8.1.2.

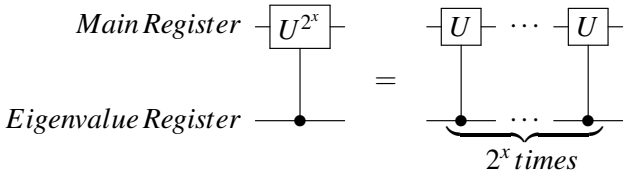

Figure 8.1.2: Decomposition of controlled-U^{2^x} gate into 2^x number of controlled-U gates for any unitary U. For the example given above, $U = \exp(2\pi i 2^{-t}\Lambda)$.

Unfortunately, this decomposition introduces a practical resource problem as an exponential, 2^{t-j} number of controlled-$\exp(2\pi i 2^{-t}\Lambda)$ gates is needed to construct a single controlled-$[\exp(2\pi i 2^{-t}\Lambda)]^{2^{t-j}}$ gate. There is no possible further simplifications to reduce the exponential scaling. The final quantum circuit form of HS subroutine, with precision of up to t bits, is shown in figure 8.1.3 below. Note that the explicit gate decomposition of controlled-$[\exp(2\pi i 2^{-t}\Lambda)]$ will be differed till the next chapter 9 as it is dimensionally dependent.

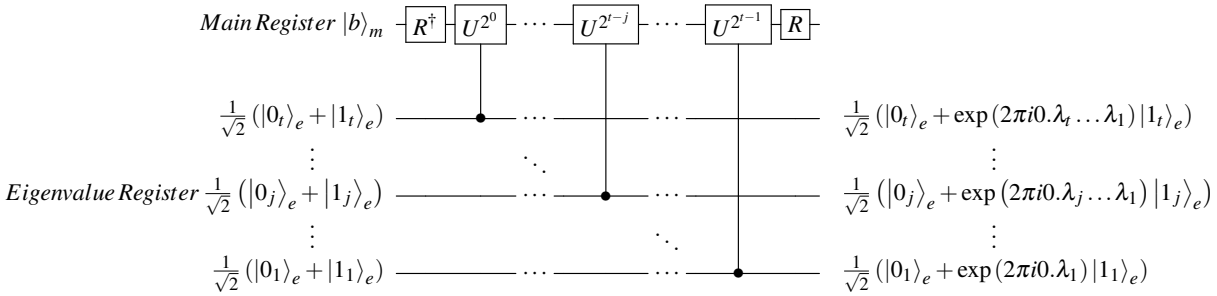

Figure 8.1.3: Simplified quantum circuit of the Hamiltonian Simulation subroutine, where $U = \exp(2\pi i 2^{-t}\Lambda)$. Note the notation change: $\lambda_k \overset{\text{in binary}}{\Longrightarrow} \lambda_{tk}\ldots\lambda_{1k} \overset{\text{drop k}}{\Longrightarrow} \lambda_t\ldots\lambda_1$, so as to avoid cumbersome writing.

8.1.3 Quantum Eigenvalue Inversion

The Quantum Eigenvalue Inversion (EI) is the most challenging step as it involves division of an arbitrary number in qubit binary representation. The main problem lies with classical notion of division that it requires an division iteration and a stop condition. This poses two major challenges when dealing with unknown eigenvalues using quantum division. First, how does quantum division perform division iterations when loops are forbidden in quantum circuit formalism? Second, how will quantum division determine whether if the division has successfully completed or not, due to indivisibility of certain pairs of numbers. Fortunately, there are several quantum division algorithms in the current literature. However, none of them are practical to implement, due to resource constraints. Therefore, for simplification sake, we shall apply the assumption that the eigenvalues are integers powers of two, so that the quantum circuit for inverting them is short. Inverting powers of two eigenvalues quantum mechanically is rather simple due to its neat property that shall be explained as follows:

Recalling and using equation 3.2.1 from section 3.2 on page 12, if the largest eigenvalue is $\lambda_t = 2^{t-1}$ for some $t \in \mathbb{N}$, then it can be written as,

$$\begin{aligned} 2^{t-1} &= 1_t \cdot 2^{t-1} + 0_{t-1} \cdot 2^{t-2} + \ldots + 0_2 \cdot 2^1 + 0_1 \cdot 2^0 && (8.1.12) \\ &\Rightarrow 1_t 0_{t-1} \ldots 0_2 0_1 && (8.1.13) \end{aligned}$$

Then, its inverse is $\frac{1}{\lambda_t} = 2^{1-t}$, which can be written as,

$$\begin{aligned} 2^{1-t} &= \frac{0_1}{2^0} + \frac{0_2}{2^1} + \ldots + \frac{0_{t-1}}{2^{t-2}} + \frac{1_t}{2^{t-1}} && (8.1.14) \\ &\Rightarrow 0_1.0_2 \ldots 0_{t-1} 1_t && (8.1.15) \end{aligned}$$

Consider multiplying $\frac{1}{\lambda_t}$ by 2^{t-1}, that is $\frac{2^{t-1}}{\lambda_t} = 2^{t-1}2^{1-t} = 2^0$, then this yields,

$$2^0 \Rightarrow 0_1 0_2 \ldots 0_{t-1} 1_t \quad (8.1.16)$$

For smaller eigenvalues $\lambda_k = 2^{k-1}$, where $k < t$, multiplying its inverse $\frac{1}{\lambda_k} = 2^{1-k}$ by 2^{t-1} , that is $\frac{2^{t-1}}{\lambda_k} = 2^{t-1}2^{1-k} = 2^{t-k}$, this yields,

$$\begin{aligned} 2^{t-k} &= 0_1 \cdot 2^{t-1} + \ldots + 0_{k-1} \cdot 2^{t-k+1} + 1_k \cdot 2^{t-k} + 1_{k+1} \cdot 2^{t-k-1} + \ldots + 0_t \cdot 2^0 && (8.1.17) \\ &\Rightarrow 0_1 \ldots 0_{k-1} 1_k 0_{k+1} \ldots 0_t && (8.1.18) \\ &\mid \text{ Compare the binary representation of } \lambda_k = 2^{k-1}. \\ 2^{k-1} &\Rightarrow 0_t \ldots 0_{k+1} 1_k 0_{k-1} \ldots 0_1 && (8.1.19) \end{aligned}$$

Comparing equation 8.1.18 and 8.1.19, we can define a inversion of power of two mapping function as follows,

$$\begin{aligned} \lambda_k &\overset{\text{Maps to}}{\longrightarrow} \frac{2^{t-1}}{\lambda_k} \quad \mid \text{ for } 1 \leq k \leq t && (8.1.20) \\ 2^{k-1} &\longrightarrow 2^{t-k} \quad \text{(inversion of power of two)} && (8.1.21) \end{aligned}$$

This inversion mapping function 8.1.20 operationally results in the swapping of the position of the binary

bits about the middle position. Therefore, we can compute $\frac{1}{\lambda_k}$ easily by reversing the binary sequence of λ_k. The quantum circuit that does this with an inverted register is simply a cascading operation of CNOT gates as shown in figure 8.1.4.

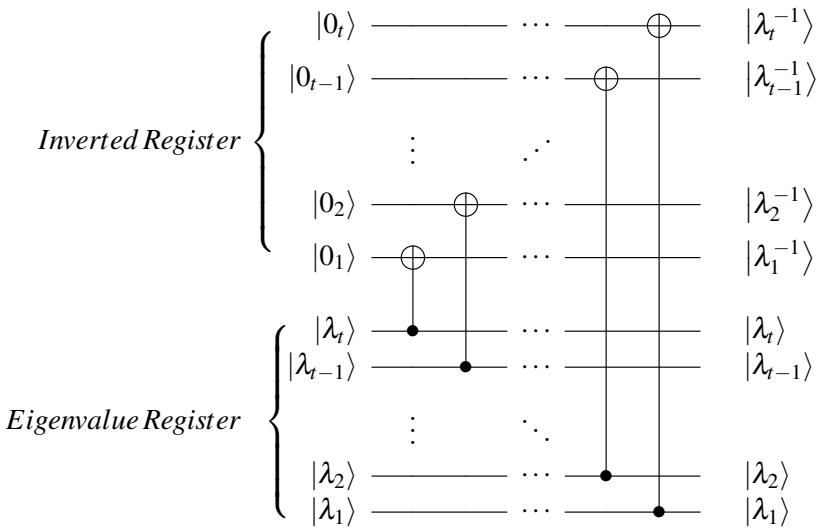

Figure 8.1.4: The quantum circuit of the Eigenvalue Inversion subroutine of powers of two with the use of an inverted register. Note the notation change: $\lambda_k \overset{\text{in binary}}{\Longrightarrow} \lambda_{tk}\dots\lambda_{1k} \overset{\text{drop k}}{\Longrightarrow} \lambda_t\dots\lambda_1$, so as to avoid cumbersome writing.

To simplify this operation even further, we can avoid the use of an inverted register by using swap gates on the eigenvalue register itself, so that the inverted eigenvalues are computed and held in the eigenvalue register as shown figure 8.1.5 below.

$$|\lambda_k\rangle_e \overset{\text{EI(Simplified)}}{\longrightarrow} \left|\frac{1}{\lambda_k}\right\rangle_e \quad | \quad \text{(without an inverted register)} \tag{8.1.22}$$

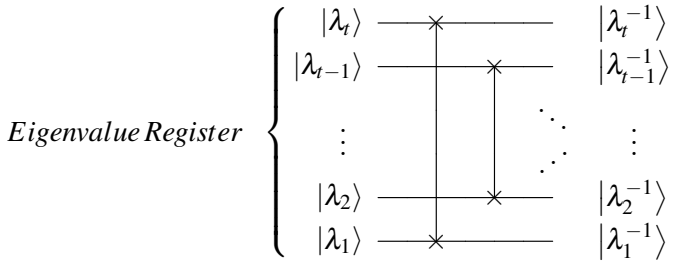

Figure 8.1.5: The simplified quantum circuit of the Eigenvalue Inversion subroutine without the use of an inverted register, assuming eigenvalues are powers of two. Note the notation change: $\lambda_k \overset{\text{in binary}}{\Longrightarrow} \lambda_{tk}\dots\lambda_{1k} \overset{\text{drop k}}{\Longrightarrow} \lambda_t\dots\lambda_1$, so as to avoid cumbersome writing.

8.2 1^{st} Order Ancillary Rotation Approximation

Note that in this subsection only, the indexes k shall be dropped that is, $\lambda_k \to \lambda, \theta_k \to \theta$ to avoid cumbersome writing. Ancillary Rotation (AR) is the most critical subroutine as it applies the corresponding amplitudes coefficients $\frac{C}{\lambda}$ to the inverted eigenvalues states $\left|\frac{1}{\lambda}\right\rangle_e$ for all eigenvalues λ

such that,

$$\left|\frac{1}{\lambda}\right\rangle_e |0\rangle_a \xrightarrow{\text{AR}} \left|\frac{1}{\lambda}\right\rangle_e \left(\sqrt{1-\left(\frac{C}{\lambda}\right)^2}|0\rangle_a + \frac{C}{\lambda}|1\rangle_a\right) \tag{8.2.1}$$

It makes use of the controlled-$R_y(\theta)$ quantum gate applied on the ancilla register conditioned on $\left|\frac{1}{\lambda}\right\rangle_e$ in the computational basis. By definition, Rotate-Y$R_y(\theta)$ has matrix representation $R_y(\theta) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}$ in the computational basis. When $R_y(\theta)$ is applied on $|0\rangle_a$ for $\theta \in [0,\pi]$, the resulting state is $R_y(\theta)|0\rangle_a = \sqrt{1-\sin^2\left(\frac{\theta}{2}\right)}|0\rangle_a + \sin\left(\frac{\theta}{2}\right)|1\rangle_a$. By substituting $\theta = 2\arcsin\left(\frac{C}{\lambda}\right)$, yields

$$R_y\left(2\arcsin\left(\frac{C}{\lambda}\right)\right)|0\rangle_a = \sqrt{1-\left(\frac{C}{\lambda}\right)^2}|0\rangle_a + \frac{C}{\lambda}|1\rangle_a \tag{8.2.2}$$

This is an exact rotation that will eventually gives an exact solution state $|x\rangle$. Note that the required amplitude coefficients $\frac{C}{\lambda}$ is in the quantum state that consists of the ancilla qubit $|1\rangle_a$. A post-selection of $|1\rangle_a$ ancilla qubit is therefore required at the end of the computation to extract out the exact solution state $|x\rangle$ using the Pauli-Z measurement operator. However, there are two challenges to implementing this exact AR subroutine. First, the AR should not have input-dependent gates, that is the quantum gates used in AR must be independent of the inputs M or $|b\rangle$ of the quantum linear problem. Second, AR should automatically determine the inverted eigenvalues $\frac{1}{\lambda}$ by 'reading' the states $\left|\frac{1}{\lambda}\right\rangle_e$ in eigenvalue register and implement the amplitude coefficients $\frac{C}{\lambda}$ accordingly. We can overcome these two challenges if AR were to be approximated. As suggested by Cao et al [Cao+11], first consider the Taylor expansion of $\arcsin(x)$ about $x = 0$,

$$\arcsin(x) = \sum_{n=0}^{\infty} \frac{(2n)!}{4^n (n!)^2 (2n+1)} x^{2n+1} \tag{8.2.3}$$

$$= x + \frac{1}{6}x^3 + \frac{3}{40}x^5 + \ldots \tag{8.2.4}$$

Therefore, the rotation gate $R_y\left(2\arcsin\left(\frac{C}{\lambda}\right)\right)$ can be Taylor expanded about $\frac{C}{\lambda} = 0$ to yield an approximated rotation gate $R_y\left(2\frac{C}{\lambda}\right)$ as follows,

$$R_y\left(2\arcsin\left(\frac{C}{\lambda}\right)\right) = R_y\left(2\frac{C}{\lambda} + \frac{1}{3}\left(\frac{C}{\lambda}\right)^3 + \frac{3}{20}\left(\frac{C}{\lambda}\right)^5 + \ldots\right) \tag{8.2.5}$$

$$\begin{aligned} &| \quad \text{Choose real paramter } C \text{ such that } 0 < \frac{C}{\lambda} \ll 1. \\ &| \quad \text{Drop all third and higher order terms.} \\ &\approx R_y\left(2\frac{C}{\lambda}\right) \end{aligned} \tag{8.2.6}$$

Note that the approximated rotation gate $R_y\left(2\frac{C}{\lambda}\right)$ rotates $|0\rangle_a$ to,

$$R_y\left(2\frac{C}{\lambda}\right)|0\rangle_a = \cos\left(\frac{C}{\lambda}\right)|0\rangle_a + \sin\left(\frac{C}{\lambda}\right)|1\rangle_a \tag{8.2.7}$$

where the domain $\frac{C}{\lambda} \in \left[0, \frac{\pi}{2}\right]$. Comparing equation 8.2.2 and 8.2.7, it can be seen that the solution state will not be exact. As a result of the approximation, there is a trade off between the accuracy and probability of success of the solution state. Both of which are dependent on the choice of real parameter C. A larger $\frac{C}{\lambda}$ close to $\frac{\pi}{2}$ yields a better chance of success, that is getting the approximated solution state, but the accuracy of the solution state suffers as the approximation no longer holds. Conversely, a smaller $\frac{C}{\lambda}$ close to 0 yields a more accurate solution state but the chance of obtaining that accurate solution state becomes lower. Unlike in the simplified version of QLS algorithm, the notion of finding the maximum probability or a maximum parameter C is now meaningless as the desired exact solution state cannot be obtained at maximum probability. Instead, it is better to find out what is the maximum probability of an approximated solution state given desired fidelity.

With the approximation, it is now possible to construct an input-independent quantum circuit of AR. Consider decomposing approximated rotation $R_y(2Cy)$ gate, where $\frac{1}{\lambda} = y$, in binary representation,

$$\begin{aligned} R_y(2Cy) &= R_y\left[2C\left(y_t \cdot 2^0 + y_{t-1} \cdot 2^{-1} + \ldots + y_j \cdot 2^{j-t} + \ldots + y_1 \cdot 2^{1-t}\right)\right] && (8.2.8) \\ &\mid \text{ Since } R_y(\theta+\phi) = R_y(\theta) R_y(\phi). \\ &= R_y\left(2Cy_t \cdot 2^0\right) \cdots R_y\left(2Cy_j \cdot 2^{j-t}\right) \ldots R_y\left(2Cy_1 \cdot 2^{1-t}\right) && (8.2.9) \end{aligned}$$

Note that since binary bit y_j is can be either 1 or 0, it makes the approximated gate $R_y\left(2Cy_j \cdot 2^{j-t}\right)$ be either $\mathbb{I}$ or $R_y\left(2C2^{j-t}\right)$ respectively. Effectively, this is equivalent to applying controlled-rotate-Y $C\left[R_y\left(2C2^{j-t}\right)\right]$ gate on ancillary register conditioned by the qubits $|y\rangle = \left|\frac{1}{\lambda}\right\rangle$ in inverse register being $|1\rangle_e$, that is,

$$\begin{aligned} C\left[R_y\left(2C2^{j-t}\right)\right] &= C\left[R_y\left(2Cy_t \cdot 2^0\right)\right] \cdots C\left[R_y\left(2Cy_j \cdot 2^{j-t}\right)\right] \cdots C\left[R_y\left(2Cy_1 \cdot 2^{1-t}\right)\right] && (8.2.10) \\ &= C^{|y_t\rangle}\left[R_y\left(2C2^0\right)\right] \cdots C^{|y_j\rangle}\left[R_y\left(2C2^{j-t}\right)\right] \cdots C^{|y_1\rangle}\left[R_y\left(2C2^{1-t}\right)\right] && (8.2.11) \\ &\mid \text{ Re-parameterise } C = \frac{\pi}{2^r}, \text{ where } r \geq 0. \\ &= C^{|y_t\rangle}\left[R_y\left(\frac{2\pi}{2^r}\right)\right] \cdots C^{|y_j\rangle}\left[R_y\left(\frac{2\pi}{2^{r+t-j}}\right)\right] \cdots C^{|y_1\rangle}\left[R_y\left(\frac{2\pi}{2^{r+t-1}}\right)\right] && (8.2.12) \end{aligned}$$

Therefore, the quantum circuit of the 1^{st} order Taylor approximation Ancillary Rotation subroutine can be constructed as shown figure 8.2.1 below. Note that we have **re-parametrised** C using r such that,

$$C = \frac{\pi}{2^r} \qquad (8.2.13)$$

This is done so that expressions and calculations involving binary numbers and trigonometry can be simplified. This re-parameterisation differs from the simplified case, on page 7.1.2, by a factor of π.

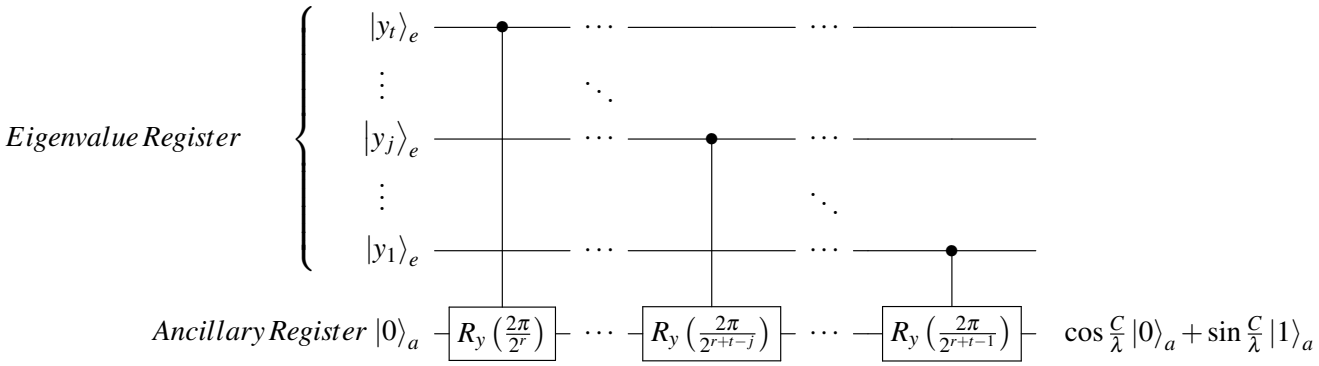

Figure 8.2.1: The quantum circuit of the 1^{st} order Taylor approximation Ancillary Rotation subroutine, where real parameter $C = \frac{\pi}{2^r}$ has been substituted, for some real parameter $r > 0$.

8.3 Post-Selection Measurement

After running the 1^{st} order Taylor approximation AR subroutine, we apply the inverse of EI and QPE to uncompute the eigenvalues. Finally, we measure the ancillary qubit in the computational basis using the Pauli-Z operator. If the ancillary qubit measured is $|0\rangle_a$, then repeat the algorithm as it has failed. Otherwise, if the ancillary qubit measured is $|1\rangle_a$, discard all qubits registers except the main register, which will be in the approximated solution state $|\tilde{x}\rangle_m$.

$$|\tilde{x}\rangle_m = \sqrt{\frac{1}{\sum_{k=1}^{2^n}\left|\beta_k \sin\left(\frac{C}{\lambda_k}\right)\right|^2}} \sum_{k=1}^{2^n} \beta_j \sin\left(\frac{C}{\lambda_k}\right)|u_k\rangle_m \tag{8.3.1}$$

where probability of success is $P(|1\rangle_a) = \sum_{k=1}^{2^n}\left|\beta_k \sin\left(\frac{C}{\lambda_k}\right)\right|^2$.

8.4 Approximated Solution State Fidelity

The fidelity measures how close the approximated solution state $|\tilde{x}\rangle_m$ is to the exact solution state $|x\rangle_m$. The fidelity $F(|a\rangle, |b\rangle)$ between two pure quantum states $|a\rangle = \sum_{j=1}^{2^n} \gamma_j |i\rangle$ and $|b\rangle = \sum_{j=1}^{n} \delta_j |i\rangle$, for some orthonormal basis set $\{|i\rangle\}$, is defined as,

$$F(|a\rangle, |b\rangle) = |\langle a|b\rangle|^2 = \left|\sum_{j=1}^{2^n} \gamma_j^* \delta_j\right|^2 \tag{8.4.1}$$

By substituting the exact solution state $|x\rangle_m$ and its approximated version $|\tilde{x}\rangle_m$ from equation 6.0.6 and 8.3.1 respectively into equation 8.4.1, the fidelity of the approximate solution state $F(|x\rangle, |\tilde{x}\rangle)$ is,

$$F(|x\rangle, |\tilde{x}\rangle) = \left|\sum_{j=1}^{2^n} \frac{\left(\beta_j \frac{1}{\lambda_j}\right)^*}{\sqrt{\sum_{k=1}^{2^n}\left|\frac{\beta_k}{\lambda_k}\right|^2}} \frac{\beta_j \sin\left(\frac{C}{\lambda_j}\right)}{\sqrt{\sum_{i=1}^{2^n}\left|\beta_i \sin\left(\frac{C}{\lambda_i}\right)\right|^2}}\right|^2 \tag{8.4.2}$$

Chapter 9

Full Quantum Linear Solver Implementation

This chapter focuses on the implementation of the full Quantum Linear Solver (QLS) algorithm elaborated in the previous chapter 8. As shown table 8.0.1, the implementation will requires some enhancements and assumptions that will simplify the quantum subroutines for practical implementation reasons. The 2D, 4D and 2^nD cases will investigated and several worked examples will be suggested for theoretical and experimental analysis. The quantum circuit construction of the full QLS algorithm is built using the IBM Qiskit Python library package and implemented on the IBM Qiskit quantum simulator. Note that, we have **re-parametrised** C using r, such that, $C = \frac{\pi}{2^r}$ in equation 8.2.13 and it will be used throughout this chapter. It is done so that expressions and calculations involving binary numbers and trigonometry can be simplified.

9.1 2D Case

Consider the spectral decomposition of a general 2D hermitian operator M, where the eigenvalues 2^{p_k} are integer powers of two where $p_2 > p_1 \in \mathbb{Z}$.

$$M = U\Lambda U^\dagger \tag{9.1.1}$$

$$\begin{array}{l|l} & \text{Subtitute the arbitrary 2D unitary operator } U \text{ from equation 2.2.3.} \\ = & \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -e^{i\delta}\sin\left(\frac{\theta}{2}\right) \\ e^{i\phi}\sin\left(\frac{\theta}{2}\right) & e^{i(\phi+\delta)}\cos\left(\frac{\theta}{2}\right) \end{pmatrix} \begin{pmatrix} 2^{p_1} & 0 \\ 0 & 2^{p_2} \end{pmatrix} \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -e^{-i\delta}\sin\left(\frac{\theta}{2}\right) \\ e^{-i\phi}\sin\left(\frac{\theta}{2}\right) & e^{-i(\phi+\delta)}\cos\left(\frac{\theta}{2}\right) \end{pmatrix} \end{array} \tag{9.1.2}$$

Suppose the given input pure 2D quantum state $|b\rangle$ is decomposed in terms of linear combinations of eigenvectors of U with complex coefficients β_k, that is $|b\rangle = \beta_1 \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) \\ e^{i\phi}\sin\left(\frac{\theta}{2}\right) \end{pmatrix} + \beta_2 \begin{pmatrix} -e^{i\delta}\sin\left(\frac{\theta}{2}\right) \\ e^{i(\phi+\delta)}\cos\left(\frac{\theta}{2}\right) \end{pmatrix}$. The 2D full QLS algorithm is tasked to obtain an approximated 2D solution state $|\tilde{x}\rangle$ such that it is close to $|x\rangle = \frac{1}{d}M^{-1}|b\rangle$, where d is a normalisation factor. The quantum circuit for 2D full QLS algorithm is shown in figure 9.1.1 below, it follows from the general circuit given in figure 8.0.1 on page 67.

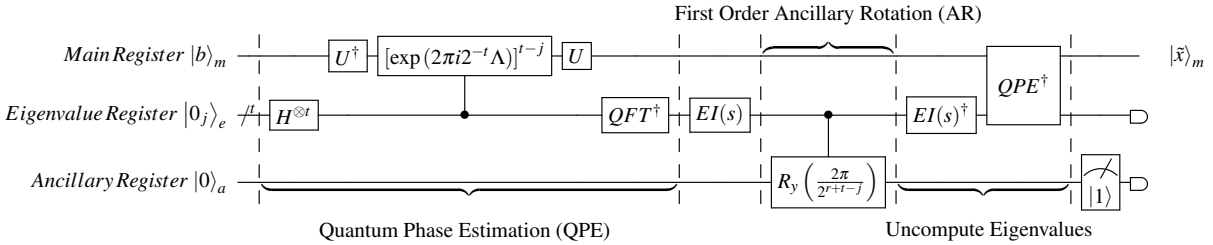

Figure 9.1.1: The quantum circuit of the 2D full QLS algorithm where EI(s) refers to the Eigenvalue Inversion subroutine (Powers of Two).

9.1.1 $C\left[\exp\left(2\pi i 2^{-t}\Lambda\right)\right]$ Quantum Gate Decomposition

This section continues from section 8.1.2 in the previous chapter. The explicit construction of the 2D controlled-$\exp\left(2\pi i 2^{-t}\Lambda\right)$ quantum gate was intentionally deferred until this section as it is dimensionally dependent. Consider the matrix representation of the 2D $\exp\left(2\pi i 2^{-t}\Lambda\right)$ quantum gate in the computational basis,

$$\begin{aligned} \exp\left(2\pi i 2^{-t}\Lambda\right) &= \exp\left(2\pi i 2^{-t}\begin{pmatrix} 2^{p_1} & 0 \\ 0 & 2^{p_2} \end{pmatrix}\right) && (9.1.3) \\ &\mid \text{ Subtitute the number of qubits } t = p_2 + 1. \\ &= \begin{pmatrix} \exp\left(\pi i 2^{p_1-p_2}\right) & 0 \\ 0 & -1 \end{pmatrix} && (9.1.4) \end{aligned}$$

where we will have to use $t = p_2 + 1$ number of qubits in the eigenvalue register for this to work. Note that $\exp\left(2\pi i 2^{-t}\Lambda\right)$ can be decomposed into $u1\left(\phi\right)R_z\left(\theta\right)$ using two common parametrised diagonal quantum gates, phase gate $u1\left(\phi\right) = \text{diag}\left(1, e^{i\phi}\right)$ and rotate-Z gate $R_z\left(\theta\right) = \text{diag}\left(e^{-\frac{i\theta}{2}}, e^{\frac{i\theta}{2}}\right)$.

$$u1\left(\phi\right)R_z\left(\theta\right) = \begin{pmatrix} e^{-i\left(\frac{\theta}{2}\right)} & 0 \\ 0 & e^{i\left(\frac{\theta}{2}+\phi\right)} \end{pmatrix} \quad (9.1.5)$$

By comparing the equations 9.1.4 and 9.1.5, we have the following,

$$\begin{aligned} \theta &= -2^{p_1-p_2+1}\pi && (9.1.6) \\ \phi &= \left(2^{p_1-p_2}+1\right)\pi && (9.1.7) \\ \therefore \exp\left(2\pi i 2^{-t}\Lambda\right) &= u1\left(\left(2^{p_1-p_2}+1\right)\pi\right)R_z\left(-2^{p_1-p_2+1}\pi\right) && (9.1.8) \end{aligned}$$

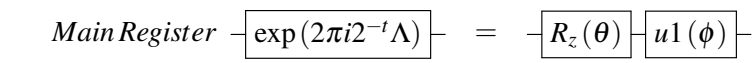

Figure 9.1.2: Decomposition of 2D $\exp\left(2\pi i 2^{-t}\Lambda\right)$ into phase and rotate-Z gates.

It is important to note that the above decomposition is valid even for non-integer powers of 2, that is by considering $p_k \in \mathbb{R}$. Using the above gate decomposition of $\exp\left(2\pi i 2^{-t}\Lambda\right)$, it is simple to construct the controlled version of it,

$$C\left[\exp\left(2\pi i 2^{-t}\Lambda\right)\right] = C\left[u1\left(\phi\right)\right]C\left[R_z\left(\theta\right)\right] \quad (9.1.9)$$

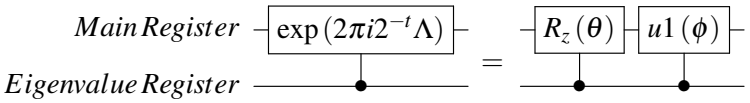

Figure 9.1.3: Decomposition of 2D controlled-exp $(2\pi i 2^{-t}\Lambda)$ into controlled phase and controlled rotate-Z gates.

9.1.2 Theoretical Results

Solving the 2D quantum linear problem using the original QLS algorithm, the exact solution state $|x\rangle$ is found to be,

$$|x\rangle \quad = \quad \sqrt{\frac{1}{\left|\frac{\beta_1}{2^{p_1}}\right|^2+\left|\frac{\beta_2}{2^{p_2}}\right|^2}}\left[\frac{\beta_1}{2^{p_1}}|u_1\rangle+\frac{\beta_2}{2^{p_2}}|u_2\rangle\right] \tag{9.1.10}$$

with probability of success $P(|1\rangle_a)$,

$$P(|1\rangle_a) \quad = \quad \left|\frac{\pi\beta_1}{2^{r+p_1}}\right|^2+\left|\frac{\pi\beta_2}{2^{r+p_2}}\right|^2 \tag{9.1.11}$$

Using the full QLS algorithm will yield the approximated solution state $|\tilde{x}\rangle$,

$$|\tilde{x}\rangle \quad = \quad \sqrt{\frac{1}{\left|\beta_1\sin\left(\frac{\pi}{2^{r+p_1}}\right)\right|^2+\left|\beta_2\sin\left(\frac{\pi}{2^{r+p_2}}\right)\right|^2}}\left[\beta_1\sin\left(\frac{\pi}{2^{r+p_1}}\right)|u_1\rangle+\beta_2\sin\left(\frac{\pi}{2^{r+p_2}}\right)|u_2\rangle\right] \tag{9.1.12}$$

with probability of success $P\left(\left|\tilde{1}\right\rangle_a\right)$,

$$P\left(\left|\tilde{1}\right\rangle_a\right) \quad = \quad \left|\beta_1\sin\left(\frac{\pi}{2^{r+p_1}}\right)\right|^2+\left|\beta_2\sin\left(\frac{\pi}{2^{r+p_2}}\right)\right|^2 \tag{9.1.13}$$

The fidelity $F(|x\rangle,|\tilde{x}\rangle)$ of between the exact solution state $|x\rangle$ and the approximated state $|\tilde{x}\rangle$ is,

$$F(|x\rangle,|\tilde{x}\rangle) \quad = \quad \frac{\left|\left(\frac{|\beta_1|^2}{2^{p_1}}\sin\left(\frac{\pi}{2^{r+p_1}}\right)+\frac{|\beta_2|^2}{2^{p_2}}\sin\left(\frac{\pi}{2^{r+p_2}}\right)\right)\right|^2}{\left(\left|\beta_1\sin\left(\frac{\pi}{2^{r+p_1}}\right)\right|^2+\left|\beta_2\sin\left(\frac{\pi}{2^{r+p_2}}\right)\right|^2\right)\left(\left|\frac{\beta_1}{2^{p_1}}\right|^2+\left|\frac{\beta_2}{2^{p_2}}\right|^2\right)} \tag{9.1.14}$$

9.1.3 2D Worked Example

Consider the simplest set of non-trivial eigenvalues and the following parameters for the 2D hermitian operator M in equation 9.1.2: $p_1=1$, $p_2=2$, $\delta=\frac{\pi}{2}$, $\phi=-\frac{\pi}{2}$, $\theta=\frac{\pi}{3}$. Then, the 2D hermitian operator M of interest has the following complex matrix representation:

$$M \quad = \quad \frac{1}{2}\begin{pmatrix} 5 & -i\sqrt{3} \\ i\sqrt{3} & 7 \end{pmatrix} \tag{9.1.15}$$

Consider the input pure 2D quantum state $|b\rangle$ to be $|+\rangle$ as given in equation 3.5.2,

$$
\begin{aligned}
|b\rangle &= |+\rangle && (9.1.16)\\
&| \quad \text{Linear decomposing in terms eigenvectors of } M. && (9.1.17)\\
&= \frac{\sqrt{3}+i}{2\sqrt{2}}\begin{pmatrix} \frac{\sqrt{3}}{2} \\ \frac{-i}{2} \end{pmatrix} + \frac{\sqrt{3}+i}{2\sqrt{2}}\begin{pmatrix} \frac{-i}{2} \\ \frac{\sqrt{3}}{2} \end{pmatrix} && (9.1.18)\\
&| \quad \therefore \text{The cofficients are } \beta_1 = \beta_2 = \frac{\sqrt{3}+i}{2\sqrt{2}}.
\end{aligned}
$$

Using equation 9.1.8, the quantum gate $\exp\left(2\pi i 2^{-t}\Lambda\right)$ can be decomposed as,

$$\exp\left(2\pi i 2^{-t}\Lambda\right) = u1\left(\frac{3\pi}{2}\right) R_z(-\pi) = \begin{pmatrix} i & 0 \\ 0 & -1 \end{pmatrix} \tag{9.1.19}$$

The vector solution $\vec{x}$ for the classical linear problem $\vec{x} = M^{-1}\vec{b}$ is found to be,

$$\vec{x} = \frac{1}{16\sqrt{2}}\begin{pmatrix} 7+i\sqrt{3} \\ 5-i\sqrt{3} \end{pmatrix} \tag{9.1.20}$$

Using the original QLS algorithm, the exact solution state $|x\rangle$ is found to be,

$$|x\rangle = \sqrt{\frac{1}{80}}\begin{pmatrix} 7+i\sqrt{3} \\ 5-i\sqrt{3} \end{pmatrix} \tag{9.1.21}$$

with probability of success $P(|1\rangle_a)$,

$$P(|1\rangle_a) = \pi^2\left[2^{-2r-3} + 2^{-2r-5}\right] \tag{9.1.22}$$

The approximated solution state $|\tilde{x}\rangle$ obtained using the full QLS algorithm is expected to be,

$$|\tilde{x}\rangle = \sqrt{\frac{2}{\sin^2\left(\frac{\pi}{2^{r+1}}\right) + \sin^2\left(\frac{\pi}{2^{r+2}}\right)}}\left(\frac{\sqrt{3}+i}{2\sqrt{2}}\right)\begin{pmatrix} \frac{\sqrt{3}}{2}\sin\left(\frac{\pi}{2^{r+1}}\right) - \frac{i}{2}\sin\left(\frac{\pi}{2^{r+2}}\right) \\ \frac{\sqrt{3}}{2}\sin\left(\frac{\pi}{2^{r+2}}\right) - \frac{i}{2}\sin\left(\frac{\pi}{2^{r+1}}\right) \end{pmatrix} \tag{9.1.23}$$

with probability of success $P\left(\left|\tilde{1}\right\rangle_a\right)$,

$$P\left(\left|\tilde{1}\right\rangle_a\right) = \frac{1}{2}\left(\sin^2\left(\frac{\pi}{2^{r+1}}\right) + \sin^2\left(\frac{\pi}{2^{r+2}}\right)\right) \tag{9.1.24}$$

The fidelity $F(|x\rangle, |\tilde{x}\rangle)$ of between these two states is,

$$F(|x\rangle, |\tilde{x}\rangle) = \frac{16\left|\frac{1}{2}\sin\left(\frac{\pi}{2^{r+1}}\right) + \frac{1}{4}\sin\left(\frac{\pi}{2^{r+2}}\right)\right|^2}{5\left[\sin^2\left(\frac{\pi}{2^{r+1}}\right) + \sin^2\left(\frac{\pi}{2^{r+2}}\right)\right]} \tag{9.1.25}$$

9.1.4 Experimental Results

Using the IBM Qiskit quantum simulator, the quantum circuit given in figure 9.1.1 on page 76 is implemented with the input prepared according to the 2D worked example given in section 9.1.3. The

table 9.1.1 below shows a breakdown of the quantum resources used this quantum circuit.

No.	Resource Type	Number of
1	Qubits	5
2	Circuit Depth	51
3	Controlled-Phase Gate $C[u1]$	20
4	Controlled-Rotate-Z Gate $C[R_z]$	14
5	Hadamard Gate H	13
6	Measure Gate Pauli-Z	5
7	Single Qubit Arbitrary Unitary Gate $u3$	4
8	$C[u3]$	3
9	Swap Gate S_{wap}	2
10	Total Quantum Operations (No. 3 to 9)	61

Table 9.1.1: A summary of the resource breakdown of the quantum circuit of the full QLS algorithm for the 2D worked example.

The independent variable is the real parameter $r = -\log_2\left(\frac{C}{\pi}\right)$, which is parametrized over a range between $0 \leq r < 4$ with increments of 0.01. There are three dependent variables: the probability of obtaining the approximated solution state $P\left(\left|\tilde{1}\right\rangle_a\right)$ using the full QLS, the probability of obtaining the exact solution state $P\left(\left|1\right\rangle_a\right)$ using the original QLS and the fidelity $F\left(\left|x\right\rangle, \left|\tilde{x}\right\rangle\right)$.

$P\left(\left|\tilde{1}\right\rangle_a\right)$ is measured by calculating the fraction the number of $\left|\tilde{1}\right\rangle_a$ ancilla qubit obtained over 10^4 independent runs. The theoretical $P\left(\left|1\right\rangle_a\right)$ and $P\left(\left|\tilde{1}\right\rangle_a\right)$ which are given in equations 9.1.22 and 9.1.24 respectively, are numerically calculated and plotted against the experimental $P\left(\left|\tilde{1}\right\rangle_a\right)$ values. The fidelity of state $F\left(\left|x\right\rangle, \left|\tilde{x}\right\rangle\right)$ is measured by calculating the dot product squared of the experimentally obtained $\left|\tilde{x}\right\rangle$ with the exact solution state $\left|x\right\rangle$ for each independent run. A combined graph of these three dependent variables against the r is plotted in figure 9.1.4 below. The average deviation of N number of experimental probability data points x_{ej} about in the actual value x_a is measured using the root mean squared percentage error (RMSPE) defined by the formula $RMSPE = \sqrt{\frac{\sum_{j=1}^{N}\left(\frac{x_a - x_{ej}}{x_a}\right)^2}{N}}$.

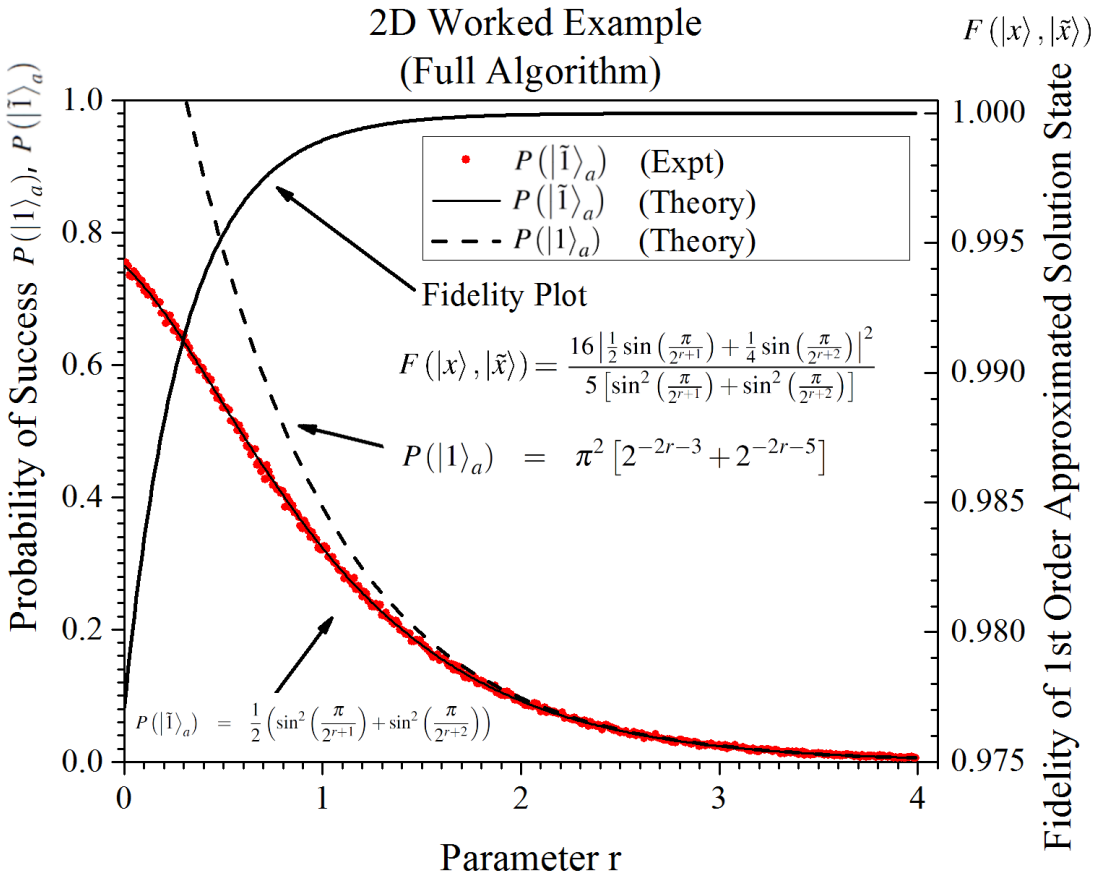

Figure 9.1.4: A combined graph of probability of successes $P\left(|\tilde{1}\rangle_a\right)$, $P(|1\rangle_a)$ and fidelity of 1st order approximated solution state $F(|x\rangle, |\tilde{x}\rangle)$ against parameter r, using the full QLS for the 2D worked example.

The experimental $P\left(|\tilde{1}\rangle_a\right)$ plot given in figure 9.1.4 shows a decreasing trend from $r = 0.00$ to 3.99 with the corresponding probability $P\left(|\tilde{1}\rangle_a\right) = 0.7526 \pm 0.0001$ and 0.0059 ± 0.0001 respectively. The decreasing trend fits the theoretical $P\left(|\tilde{1}\rangle_a\right)$ prediction closely. It is found that these experimental values deviate about $RMSPE = \pm 5\%$ of the theoretical value on average. The experimental $P\left(|\tilde{1}\rangle_a\right)$ plot also fits the theoretical probability $P(|1\rangle_a)$ for about $r > 2$. The experimental results for the fidelity of state is found to be exactly equal to the theoretical results. The experimental fidelity $F(|x\rangle, |\tilde{x}\rangle) > 0.999$ is achieved for $r > 1.010 \pm 0.005$.

The order of magnitude of the average simulation runtime of the quantum circuit is found to be about $10^{-1}s$, which is of four orders of magnitude more than that of the average runtime of the classical solver (using the built-in python numpy linear solver) which is about $10^{-5}s$. The discussion of results is deferred to section 10.2 on page 91.

9.2 4D Case

In the 4D case, consider the spectral decomposition of a 4D hermitian operator $M = U\Lambda U^{\dagger}$. Suppose $U = U_{2_1} \otimes U_{2_2}$ where U_{2_j} is a 2D arbitrary unitary operator from equation 2.2.3. Λ is a 4D diagonal operator whose elements are distinct integer powers of two. For reasons which will be explained in

the following subsection 9.2.1, make another assumption, just for this 4D case, that the magnitude of the diagonal elements are descending from the top, that is $p_1 > p_2 > p_3 > p_4 \in \mathbb{Z}$. Note that quantum operation $U_{2_1} \otimes U_{2_2}$ is a subset of the infinite possible configurations for arbitrary 4D quantum gate U in the quantum circuit formalism. The explicit matrix representation of the 4D hermitian operator M is,

$$M = \left(U_{2_1} \otimes U_{2_2}\right) \underbrace{\begin{pmatrix} 2^{p_1} & 0 & 0 & 0 \\ 0 & 2^{p_2} & 0 & 0 \\ 0 & 0 & 2^{p_3} & 0 \\ 0 & 0 & 0 & 2^{p_4} \end{pmatrix}}_{\Lambda} \left(U_{2_1}^{\dagger} \otimes U_{2_2}^{\dagger}\right) \tag{9.2.1}$$

| Note: $p_1 > p_2 > p_3 > p_4 \in \mathbb{Z}$ for this 4D case only.

$$| \quad U_{2_1} \otimes U_{2_2} = \begin{pmatrix} \cos\left(\frac{\theta_1}{2}\right)\cos\left(\frac{\theta_2}{2}\right) & -e^{i\delta_2}\cos\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right) & -e^{i\delta_1}\sin\left(\frac{\theta_1}{2}\right)\cos\left(\frac{\theta_2}{2}\right) & e^{i\delta_1}e^{i\delta_2}\sin\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right) \\ e^{i\phi_2}\cos\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right) & e^{i(\phi_2+\delta_2)}\cos\left(\frac{\theta_1}{2}\right)\cos\left(\frac{\theta_2}{2}\right) & -e^{i\delta_1}e^{i\phi_2}\sin\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right) & -e^{i\delta_1}e^{i(\phi_2+\delta_2)}\sin\left(\frac{\theta_1}{2}\right)\cos\left(\frac{\theta_2}{2}\right) \\ e^{i\phi_1}\sin\left(\frac{\theta_1}{2}\right)\cos\left(\frac{\theta_2}{2}\right) & -e^{i\phi_1}e^{i\delta_2}\sin\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right) & e^{i(\phi_1+\delta_1)}\cos\left(\frac{\theta_1}{2}\right)\cos\left(\frac{\theta_2}{2}\right) & -e^{i(\phi_1+\delta_1)}e^{i\delta_2}\cos\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right) \\ e^{i\phi_1}e^{i\phi_2}\sin\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right) & e^{i\phi_1}e^{i(\phi_2+\delta_2)}\sin\left(\frac{\theta_1}{2}\right)\cos\left(\frac{\theta_2}{2}\right) & e^{i(\phi_1+\delta_1)}e^{i\phi_2}\cos\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right) & e^{i(\phi_1+\delta_1)}e^{i(\phi_2+\delta_2)}\cos\left(\frac{\theta_1}{2}\right)\cos\left(\frac{\theta_2}{2}\right) \end{pmatrix}$$

Suppose the given input pure 4D quantum state $|b\rangle$ is decomposed in terms of linear combinations of eigenvectors of U with complex coefficients β_k, that is

$$\begin{aligned} |b\rangle = \beta_1 &\begin{pmatrix} \cos\left(\frac{\theta_1}{2}\right)\cos\left(\frac{\theta_2}{2}\right) \\ e^{i\phi_2}\cos\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right) \\ e^{i\phi_1}\sin\left(\frac{\theta_1}{2}\right)\cos\left(\frac{\theta_2}{2}\right) \\ e^{i\phi_1}e^{i\phi_2}\sin\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right) \end{pmatrix} + \beta_2 \begin{pmatrix} -e^{i\delta_2}\cos\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right) \\ e^{i(\phi_2+\delta_2)}\cos\left(\frac{\theta_1}{2}\right)\cos\left(\frac{\theta_2}{2}\right) \\ -e^{i\phi_1}e^{i\delta_2}\sin\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right) \\ e^{i\phi_1}e^{i(\phi_2+\delta_2)}\sin\left(\frac{\theta_1}{2}\right)\cos\left(\frac{\theta_2}{2}\right) \end{pmatrix} + \\ &+\beta_3 \begin{pmatrix} -e^{i\delta_1}\sin\left(\frac{\theta_1}{2}\right)\cos\left(\frac{\theta_2}{2}\right) \\ -e^{i\delta_1}e^{i\phi_2}\sin\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right) \\ e^{i(\phi_1+\delta_1)}\cos\left(\frac{\theta_1}{2}\right)\cos\left(\frac{\theta_2}{2}\right) \\ e^{i(\phi_1+\delta_1)}e^{i\phi_2}\cos\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right) \end{pmatrix} + \beta_4 \begin{pmatrix} e^{i\delta_1}e^{i\delta_2}\sin\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right) \\ -e^{i\delta_1}e^{i(\phi_2+\delta_2)}\sin\left(\frac{\theta_1}{2}\right)\cos\left(\frac{\theta_2}{2}\right) \\ -e^{i(\phi_1+\delta_1)}e^{i\delta_2}\cos\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right) \\ e^{i(\phi_1+\delta_1)}e^{i(\phi_2+\delta_2)}\cos\left(\frac{\theta_1}{2}\right)\cos\left(\frac{\theta_2}{2}\right) \end{pmatrix} \end{aligned} \tag{9.2.2}$$

The 4D full QLS algorithm is tasked to obtain an approximated 4D solution state $|\tilde{x}\rangle$ such that it is close to $|x\rangle = \frac{1}{d}M^{-1}|b\rangle$, where d is a normalisation factor. The quantum circuit for 4D full QLS algorithm is shown in figure 9.2.1 below, it is similar to the 2D case, as it follows from the general circuit given in figure 8.0.1 on page 67.

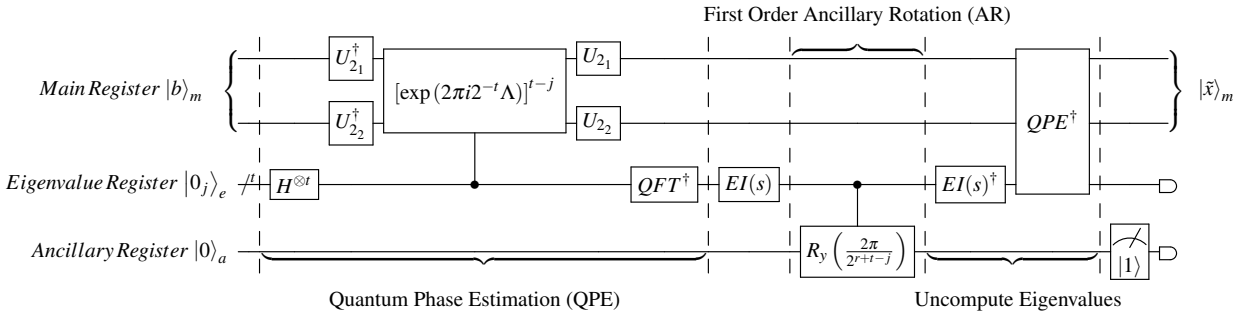

Figure 9.2.1: The quantum circuit of the 4D full QLS algorithm, where EI(s) refers to the Eigenvalue Inversion Subroutine (Powers of Two).

9.2.1 $C[\exp(2\pi i 2^{-t}\Lambda)]$ Quantum Gate Decomposition

This section continues from section 8.1.2 in the previous chapter. The explicit construction of the 4D controlled-$\exp(2\pi i 2^{-t}\Lambda)$ quantum gate was intentionally deferred until this section as it is dimensionally dependent. Consider the matrix representation of the 4D $\exp(2\pi i 2^{-t}\Lambda)$ diagonal

quantum gate, in the computational basis.

$$\begin{aligned} \exp\left(2\pi i 2^{-t}\Lambda\right) &= \operatorname{diag}\left[-1, \exp\left(\pi i 2^{p_2-p_1}\right), \exp\left(\pi i 2^{p_3-p_1}\right), \exp\left(\pi i 2^{p_4-p_1}\right)\right] \qquad (9.2.3) \\ &\mid \text{ Using number of qubits } t = p_1 + 1. \end{aligned}$$

where we will have to use $t = p_1 + 1$ number of qubits in the eigenvalue register for this to work. Note that the $\exp(2\pi i 2^{-t}\Lambda)$ quantum gate can be decomposed into the following gate operation,

$$C^2(R_z(\theta_2))\,C^2(u1(\phi_2))\,C^1(R_z(\theta_1))\,X_1 Z_1 X_1 \qquad (9.2.4)$$

with its quantum circuit representation as given on the right hand side of figure 9.2.2 below. Its matrix representation is found to be,

$$\begin{aligned} & C^2(R_z(\theta_2))\,C^2(u1(\phi_2))\,C^1(R_z(\theta_1))\,X_1 Z_1 X_1 \\ \mid\ & \text{Since } XZX = -Z. \\ = \ & \underbrace{\operatorname{diag}\left[1, \exp\left(-\frac{i\theta_2}{2}\right), 1, \exp\left(\frac{i\theta_2}{2}\right)\right]}_{C^2(R_z(\theta_2))} \underbrace{\operatorname{diag}\left[1,1,1, \exp\left(\frac{i\phi_2}{2}\right)\right]}_{C^2(u1(\phi_2))} \underbrace{\operatorname{diag}\left[1,1, \exp\left(-\frac{i\theta_1}{2}\right), \exp\left(\frac{i\theta_1}{2}\right)\right]}_{C^1(R_z(\theta_1))} \underbrace{\operatorname{diag}[-1,1,-1,1]}_{\mathbb{I}\otimes(-Z)} \qquad (9.2.5) \\ = \ & \operatorname{diag}\left[-1, \exp\left(-\frac{i\theta_2}{2}\right), -\exp\left(\frac{i\theta_1}{2}\right), \exp\left(\frac{i\theta_2}{2} + \frac{i\theta_1}{2} + i\phi_2\right)\right] \qquad (9.2.6) \end{aligned}$$

Comparing equation 9.2.3 and 9.2.6, we have the following,

$$\begin{aligned} \theta_1 &= 2\pi\left(1 - 2^{p_3-p_1}\right) && (9.2.7) \\ \theta_2 &= -\pi 2^{p_2-p_1+1} && (9.2.8) \\ \phi_2 &= \pi\left(2^{p_4-p_1} + 2^{p_2-p_1} + 2^{p_3-p_1} - 1\right) && (9.2.9) \\ \therefore \exp\left(2\pi i 2^{-t}\Lambda\right) &= C^2(R_z(\theta_2))\,C^2(u1(\phi_2))\,C^1(R_z(\theta_1))\,X_1 Z_1 X_1 && (9.2.10) \end{aligned}$$

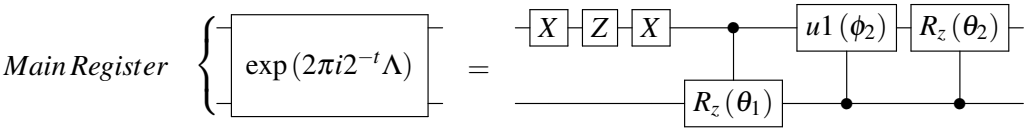

Figure 9.2.2: Decomposition of 4D $\exp(2\pi i 2^{-t}\Lambda)$ into parametrised quantum gates.

From the decomposition of 4D $\exp(2\pi i 2^{-t}\Lambda)$ quantum gate in equation 9.2.5, it is simple to construct the controlled-$\exp(2\pi i 2^{-t}\Lambda)$, as shown in figure 9.2.3 below. The reason for assuming that the magnitude of the diagonal elements, the eigenvalues, are descending from the top is that a general gate decomposition for a 4D diagonal gate using IBM Qiskit quantum gates is yet unknown. Thus, the gate decomposition of 9.2.4 was found manually by trial and error. However, it was unfortunately restricted to the case where the top diagonal element is -1.

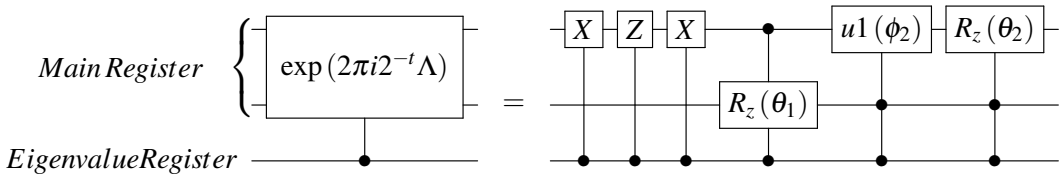

Figure 9.2.3: Decomposition of 4D $C\left[\exp\left(2\pi i 2^{-t}\Lambda\right)\right]$ into parametrised quantum gates.

9.2.2 Theoretical Results

Solving the 4D quantum linear problem using the original QLS algorithm, the exact solution state $|x\rangle$ is found to be,

$$|x\rangle \quad = \quad \sqrt{\frac{1}{\sum_{k=1}^{4}\left|\frac{\beta_k}{2^{p_k}}\right|^2}}\sum_{k=1}^{4}\frac{\beta_k}{2^{p_k}}|u_k\rangle \tag{9.2.11}$$

with probability of success $P\left(|1\rangle_a\right)$,

$$P\left(|1\rangle_a\right) \quad = \quad \sum_{k=1}^{4}\left|\frac{\pi\beta_k}{2^{r+p_k}}\right|^2 \tag{9.2.12}$$

Using the full QLS algorithm will yield the approximated solution state $|\tilde{x}\rangle$,

$$|\tilde{x}\rangle \quad = \quad \sqrt{\frac{1}{\sum_{k=1}^{4}\left|\beta_k \sin\left(\frac{\pi}{2^{r+p_k}}\right)\right|^2}}\sum_{k=1}^{4}\beta_k \sin\left(\frac{\pi}{2^{r+p_k}}\right)|u_k\rangle \tag{9.2.13}$$

with probability of success $P\left(\left|\tilde{1}\right\rangle_a\right)$,

$$P\left(\left|\tilde{1}\right\rangle_a\right) \quad = \quad \sum_{k=1}^{4}\left|\beta_k \sin\left(\frac{\pi}{2^{r+p_k}}\right)\right|^2 \tag{9.2.14}$$

The fidelity $F\left(|x\rangle, |\tilde{x}\rangle\right)$ of between the exact solution state $|x\rangle$ and the approximated state $|\tilde{x}\rangle$ is,

$$F\left(|x\rangle, |\tilde{x}\rangle\right) \quad = \quad \frac{\left|\sum_{k=1}^{4}\frac{|\beta_k|^2}{2^{p_k}}\sin\left(\frac{\pi}{2^{r+p_k}}\right)\right|^2}{\left(\sum_{k=1}^{4}\left|\beta_k \sin\left(\frac{\pi}{2^{r+p_k}}\right)\right|^2\right)\left(\sum_{k=1}^{4}\left|\frac{\beta_k}{2^{p_k}}\right|^2\right)} \tag{9.2.15}$$

9.2.3 4D Worked Example

Consider the simplest set of non-trivial eigenvalues and the following parameters for the 4D hermitian operator M in equation 9.2.1: $p_k = 5-k$, where $k = \{1,2,3,4\}$, $\delta_1 = \delta_2 = -\frac{\pi}{2}$, $\phi_1 = \phi_2 = \theta_1 = \theta_2 = \frac{\pi}{2}$. Then, the 4D hermitian operator M of interest has the following complex matrix representation:

$$M \quad = \quad \frac{1}{2}\begin{pmatrix} 15 & -5i & -9i & -3 \\ 5i & 15 & 3 & -9i \\ 9i & 3 & 15 & -5i \\ -3 & 9i & 5i & 15 \end{pmatrix} \tag{9.2.16}$$

Consider the input pure 4D quantum state $|b\rangle$ to be the tensor product of two equal $|+\rangle$, where $|+\rangle$

is a quantum state as given in equation 3.5.2,

$$
\begin{aligned}
|b\rangle &= |+\rangle \otimes |+\rangle && (9.2.17)\\
&| \quad \text{Linear decomposing in terms eigenvectors of } M. && (9.2.18)\\
&= -\frac{i}{2}\begin{pmatrix} 1/2 \\ i/2 \\ i/2 \\ -1/2 \end{pmatrix} - \frac{i}{2}\begin{pmatrix} i/2 \\ 1/2 \\ -1/2 \\ i/2 \end{pmatrix} - \frac{i}{2}\begin{pmatrix} i/2 \\ -1/2 \\ 1/2 \\ i/2 \end{pmatrix} - \frac{i}{2}\begin{pmatrix} -1/2 \\ i/2 \\ i/2 \\ 1/2 \end{pmatrix} && (9.2.19)\\
&| \quad \therefore \text{The cofficients are } \beta_{1,2,3,4} = -\frac{i}{2}.
\end{aligned}
$$

Using equation 9.2.10, the quantum gate $\exp\left(2\pi i 2^{-t}\Lambda\right)$ can be decomposed as,

$$
\exp\left(2\pi i 2^{-t}\Lambda\right) = C^2\left(R_z\left(\frac{3\pi}{2}\right)\right) C^2\left(u1\left(-\frac{\pi}{8}\right)\right) C^1\left(R_z\left(-\pi\right)\right) X_1 Z_1 X_1 \qquad (9.2.20)
$$

The vector solution $\vec{x}$ for the classical linear problem $\vec{x} = M^{-1}\vec{b}$ is found to be,

$$
\vec{x} = \frac{1}{64}\begin{pmatrix} 6+7i \\ 9+2i \\ 9-2i \\ 6-7i \end{pmatrix} \qquad (9.2.21)
$$

Using the original QLS algorithm, the exact solution state $|x\rangle$ is found to be,

$$
|x\rangle = \frac{1}{2\sqrt{85}}\begin{pmatrix} 6+7i \\ 9+2i \\ 9-2i \\ 6-7i \end{pmatrix} \qquad (9.2.22)
$$

with probability of success $P\left(|1\rangle_a\right)$,

$$
P\left(|1\rangle_a\right) = \pi^2\left[2^{-2r-4} + 2^{-2r-6} + 2^{-2r-8} + 2^{-2r-10}\right] \qquad (9.2.23)
$$

The approximated solution state $|\tilde{x}\rangle$ obtained using the full QLS algorithm is expected to be,

$$
|\tilde{x}\rangle = \sqrt{\frac{1}{\sum_{k=1}^{4}\sin^2\left(\frac{\pi}{2^{r+k}}\right)}}\begin{pmatrix} -\frac{i}{2}\sin\left(\frac{\pi}{2^{r+4}}\right) + \frac{1}{2}\sin\left(\frac{\pi}{2^{r+3}}\right) + \frac{1}{2}\sin\left(\frac{\pi}{2^{r+2}}\right) + \frac{i}{2}\sin\left(\frac{\pi}{2^{r+1}}\right) \\ \frac{1}{2}\sin\left(\frac{\pi}{2^{r+4}}\right) - \frac{i}{2}\sin\left(\frac{\pi}{2^{r+3}}\right) + \frac{i}{2}\sin\left(\frac{\pi}{2^{r+2}}\right) + \frac{1}{2}\sin\left(\frac{\pi}{2^{r+1}}\right) \\ \frac{1}{2}\sin\left(\frac{\pi}{2^{r+4}}\right) + \frac{i}{2}\sin\left(\frac{\pi}{2^{r+3}}\right) - \frac{i}{2}\sin\left(\frac{\pi}{2^{r+2}}\right) + \frac{1}{2}\sin\left(\frac{\pi}{2^{r+1}}\right) \\ \frac{i}{2}\sin\left(\frac{\pi}{2^{r+4}}\right) + \frac{1}{2}\sin\left(\frac{\pi}{2^{r+3}}\right) + \frac{1}{2}\sin\left(\frac{\pi}{2^{r+2}}\right) - \frac{i}{2}\sin\left(\frac{\pi}{2^{r+1}}\right) \end{pmatrix} \qquad (9.2.24)
$$

with probability of success $P\left(\left|\tilde{1}\right\rangle_a\right)$,

$$
P\left(\left|\tilde{1}\right\rangle_a\right) = \frac{1}{4}\sum_{k=1}^{4}\sin^2\left(\frac{\pi}{2^{r+k}}\right) \qquad (9.2.25)
$$

The fidelity of between $F\left(|x\rangle, |\tilde{x}\rangle\right)$ these two states are,

$$F\left(|x\rangle, |\tilde{x}\rangle\right) = \frac{256\left|\sum_{k=1}^{4} \frac{1}{2^k} \sin\left(\frac{\pi}{2^{r+k}}\right)\right|^2}{85\left[\sum_{k=1}^{4} \sin^2\left(\frac{\pi}{2^{r+k}}\right)\right]} \tag{9.2.26}$$

9.2.4 Experimental Results

Similar to the 2D example, using IBM Qiskit quantum simulator, the quantum circuit given in figure 9.2.1 is implemented with the inputs prepared according to the 4D worked example given in section 9.2.3 on page 83. The table 9.2.1 below shows a breakdown of the quantum resources used this quantum circuit.

No.	Resource Type	Number of
1	Qubits	9
2	Circuit Depth	779
3	Toffoli Gate CC_x	372
4	CNOT Gate C_x	124
5	Controlled-Rotate-Z Gate $C\left[R_z\right]$	124
6	Controlled-Phase Gate $C\left[u1\right]$	82
7	Controlled-Pauli-Z Gate $C\left[Z\right]$	62
8	Hadamard Gate H	22
9	Measure Gate Pauli-Z	9
10	Single Qubit Arbitrary Unitary Gate $u3$	8
11	$C\left[u3\right]$	5
12	Swap Gate S_{wap}	4
13	Total Quantum Operations (No. 3 to 12)	812

Table 9.2.1: A summary of the resource breakdown of the quantum circuit of the full QLS algorithm for the 4D worked example.

The independent variable is the real parameter $r = -\log_2\left(\frac{C}{\pi}\right)$ which is parametrized over a range between $0 \leq r < 4$ with increments of 0.01. There are three dependent variables: the probability of obtaining the approximated solution state $P\left(\left|\tilde{1}\right\rangle_a\right)$ using the full QLS, the probability of obtaining the exact solution state $P\left(|1\rangle_a\right)$ using the original QLS and the fidelity $F\left(|x\rangle, |\tilde{x}\rangle\right)$.

$P\left(\left|\tilde{1}\right\rangle_a\right)$ is measured by calculating the fraction the number of $\left|\tilde{1}\right\rangle_a$ ancilla qubit obtained over 10^4 independent runs. The theoretical $P\left(|1\rangle_a\right)$ and $P\left(\left|\tilde{1}\right\rangle_a\right)$ which are given in equations 9.1.22 and 9.1.24 respectively, are numerically calculated and plotted against the experimental $P\left(\left|\tilde{1}\right\rangle_a\right)$ values. The fidelity of state $F\left(|x\rangle, |\tilde{x}\rangle\right)$ is measured by calculating the dot product squared of the experimentally obtained $|\tilde{x}\rangle$ with the exact solution state $|x\rangle$ for each independent run. A combined graph of these three dependent variables against the r is plotted in figure 9.1.4 below. The average deviation of N number of experimental probability data points x_{ej} about in the actual value x_a is measured using the root mean squared percentage error (RMSPE) defined by the formula $RMSPE = \sqrt{\frac{\sum_{j=1}^{N}\left(\frac{x_a - x_{ej}}{x_a}\right)^2}{N}}$.

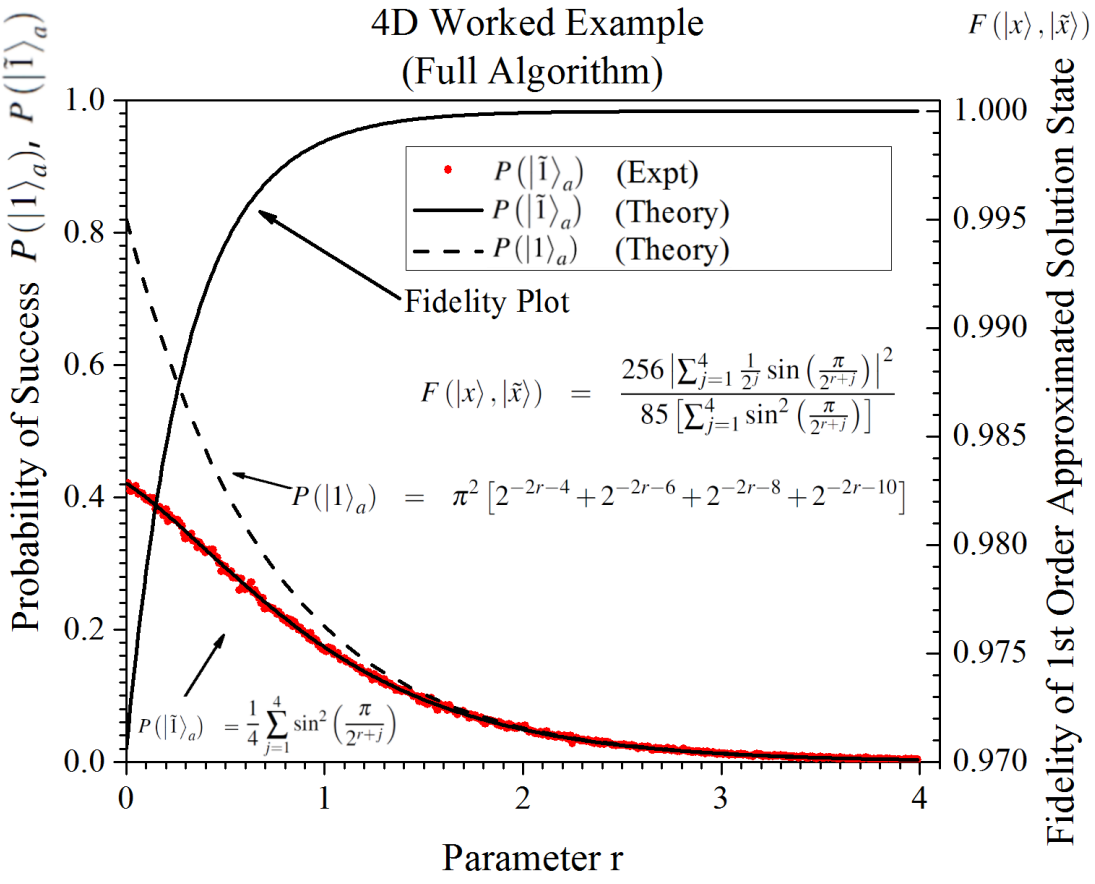

Figure 9.2.4: A combined graph of probability of successes $P\left(\left|\tilde{1}\right\rangle_a\right)$, $P(|1\rangle_a)$ and fidelity of 1st order approximated solution state $F(|x\rangle, |\tilde{x}\rangle)$ against parameter r, using the full QLS for the 4D worked example.

The experimental $P\left(\left|\tilde{1}\right\rangle_a\right)$ plot given in figure 9.2.4 shows a decreasing trend from parameter $r = 0.00$ to 3.99 with the corresponding probability $P\left(\left|\tilde{1}\right\rangle_a\right) = 0.4188 \pm 0.0001$ and 0.0032 ± 0.0001 respectively. The decreasing trend fits the theoretical $P\left(\left|\tilde{1}\right\rangle_a\right)$ prediction closely. It is found that these experimental values deviate about $RMSPE = \pm 8\%$ of the theoretical value on average. The experimental $P\left(\left|\tilde{1}\right\rangle_a\right)$ plot also fits the theoretical probability $P(|1\rangle_a)$ for about $r > 1.5$. The experimental results for the fidelity of state is found to be exactly equal to the theoretical results. The experimental fidelity $F(|x\rangle, |\tilde{x}\rangle) > 0.999$ is achieved for $r > 1.110 \pm 0.005$.

The order of magnitude of the average simulation runtime of the quantum circuit is found to be about $10^0 s$, which is of five orders of magnitude more than that of the average runtime of the classical solver (using the built-in python numpy linear solver) which is about $10^{-5} s$. The discussion of results is deferred to section 10.2 on page 91.

9.3 2^nD Case

By working through the 2D and 4D case, it is possible to generalise the theoretical results for the 2^nD case under same assumptions and approximations. One could easily deduce some results simply by observation. In the 2^nD case, consider the spectral decomposition of a 2^nD hermitian matrix $M = U\Lambda U^\dagger$,

where U is a 2^nD arbitrary unitary operator and Λ is a 2^nD diagonal matrix whose elements are distinct integer powers of two. The explicit 2^nD hermitian matrix M is,

$$M = U\text{diag}\left(2^{p_1},\ldots,2^{p_k},\ldots,2^{p_{2^n}}\right)U^\dagger \tag{9.3.1}$$

Suppose the given input pure 2^nD quantum state $|b\rangle$ is decomposed in terms of linear combinations of eigenvectors of U, $|u_k\rangle$ with complex coefficients β_k, that is $|b\rangle = \sum_{k=0}^{2^n}\beta_k|u_k\rangle$. The full QLS algorithm is tasked to obtain an approximated 2^nD solution state $|\tilde{x}\rangle$ such that it is close to $|x\rangle = \frac{1}{d}M^{-1}|b\rangle$, where d is a normalisation factor. The quantum circuit for 2^nD full QLS algorithm is shown in figure 9.3.1 below, it is similar to the 2D and 4D case, as it follows from the general circuit given in figure 8.0.1 on page 67. It may look seemingly simple but deceptively, it is a monster quantum circuit that consists of exponential number of common quantum gates.

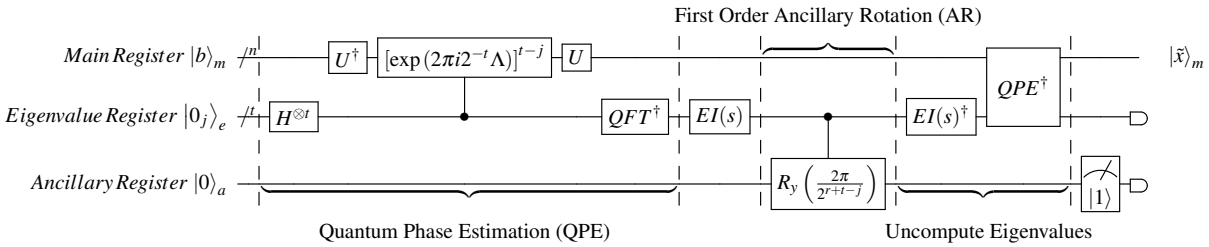

Figure 9.3.1: The quantum circuit of the 2^nD full quantum linear solver, where EI(s) refers to the Eigenvalue Inversion Subroutine (Simplified Version).

The construction of the $\exp(2\pi i 2^{-t}\Lambda)$ for arbitrary 2^n dimension is not a trivial problem, despite being one of the relatively restrictive and simplest class of unitary matrices. Unfortunately, this is one of the quantum gates that cannot be generalised by observation, and shall be postponed for future research.

9.3.1 Theoretical Results

Fortunately, the theoretical results can be easily generalised and derived simply by observation. A rigorous derivation or proof of the generalised results shall be left as an exercise for the reader. Solving the 2^nD quantum linear problem using the original QLS algorithm, the exact solution state $|x\rangle$ is found to be,

$$|x\rangle = \sqrt{\frac{1}{\sum_{k=1}^{2^n}\left|\frac{\beta_k}{2^{p_k}}\right|^2}}\sum_{k=1}^{2^n}\frac{\beta_k}{2^{p_k}}|u_k\rangle \tag{9.3.2}$$

with probability of success $P(|1\rangle_a)$

$$P(|1\rangle_a) = \sum_{j=1}^{2^n}\left|\frac{\pi\beta_k}{2^{r+p_k}}\right|^2 \tag{9.3.3}$$

Using the full QLS algorithm will yield the the approximated solution state $|\tilde{x}\rangle$,

$$|\tilde{x}\rangle = \sqrt{\frac{1}{\sum_{k=1}^{2^n}\left|\beta_k\sin\left(\frac{\pi}{2^{r+p_k}}\right)\right|^2}}\sum_{k=1}^{2^n}\beta_k\sin\left(\frac{\pi}{2^{r+p_k}}\right)|u_k\rangle \tag{9.3.4}$$

with probability of success $P\left(\left|\tilde{1}\right\rangle_a\right)$ is,

$$P\left(\left|\tilde{1}\right\rangle_a\right) = \sum_{k=1}^{2^n}\left|\beta_k \sin\left(\frac{\pi}{2^{r+p_k}}\right)\right|^2 \tag{9.3.5}$$

The fidelity $F\left(|x\rangle, |\tilde{x}\rangle\right)$ of between the exact solution state $|x\rangle$ and the approximated state $|\tilde{x}\rangle$ is,

$$F\left(|x\rangle, |\tilde{x}\rangle\right) = \frac{\left|\sum_{k=1}^{2^n} \frac{|\beta_k|^2}{2^{p_k}} \sin\left(\frac{\pi}{2^{r+p_k}}\right)\right|^2}{\left(\sum_{k=1}^{2^n}\left|\beta_k \sin\left(\frac{\pi}{2^{r+p_k}}\right)\right|^2\right)\left(\sum_{k=1}^{2^n}\left|\frac{\beta_k}{2^{p_k}}\right|^2\right)} \tag{9.3.6}$$

Part III

Project Conclusion

Chapter 10

Discussion of Results

This chapter shall discuss theoretical and experimental results obtained from all four worked examples in the previous chapters.

10.1 On the Theoretical Results

The theoretical analysis of the simplified QLS algorithm shows that it can solve the quantum linear problem probabilistically by generating an exact solution quantum state $|x\rangle$, which is a normalised solution vector $\vec{x}$. This is unlike classical linear solver algorithms which always guarantee a solution for any hermitian matrix M. The probability of success $P(|1\rangle_a)$ is found to be dependent on the input pure quantum state $|b\rangle$, the eigenvalues $0.\bar{X}k$ of input hermitian matrix M, and the free parameter $C = \frac{1}{2^r}$ as shown in equation 7.3.7. This tells us that we cannot theoretically predict probability of success if eigenvalues $0.\bar{X}k$ are unknown, that is, when $\bar{X}$ is not given.

Suppose, eigenvalues $0.\bar{X}k$ are known, the maximum parameter C that it can take must be equal to the smallest eigenvalue, that is $0.\bar{X}\mathbf{0}$, which corresponds to the maximum probability of success. The lower limit of C is zero, which corresponds to a complete failure of the algorithm. The maximum probability of success $P_{max}(|1\rangle_a)$ was theoretically found to be about 0.86 and 0.82 for the 2D and 4D case in each of its worked examples respectively. This is an encouraging result that shows quantum algorithms could solve problems with acceptable probability with more than half. For a simplified QLS implementation, the user would not have to worry about setting the parameter $C = \frac{1}{2^r}$ above the upper bound, as it is physically unattainable. This is because, it is impossible to find a controlled-$R_y(\theta_k)$ quantum gate such that $\theta_k = 2\arcsin\left(\frac{1}{\lambda_k \cdot 2^r}\right)$ where $\frac{1}{\lambda_k \cdot 2^r} > 1$. Furthermore, by setting the parameter $C = \frac{1}{2^r}$ within its range, the simplified QLS algorithm guarantees a non-zero probability of obtaining the solution quantum state $|x\rangle$.

The theoretical analysis of the full QLS algorithm shows that it can approximately solve the quantum linear problem probabilistically by generating an approximated solution quantum state $|\tilde{x}\rangle$, Taylor expanded up to the first order. The full QLS algorithm avoids the need of knowing of any eigenvalues to construct the quantum circuit. However, similar to the simplified case, the probability function $P\left(\left|\tilde{1}\right\rangle_a\right)$ is dependent on the same three variables, particularly the unknown eigenvalues. Therefore, in order to predict the probability of success one has to carry out some heavy quantum pre-processing on the given quantum inputs to obtain unknown eigenvalues. At worst, it would require the use some QLS algorithm to obtain the unknown eigenvalues as it may be reformulated as another

quantum linear problem, resulting in a 'chicken or the egg' dilemma.

The probability function $P\left(\left|\tilde{1}\right\rangle_a\right)$ is a continuous function that will never exceed unity one for all values of parameter $C = \frac{\pi}{2^r}$, which is unlike upper bounded probability function for the simplified QLS case. The fidelity function $F\left(|x\rangle, |\tilde{x}\rangle\right)$ as shown in equation 9.3.6, was found to be a strictly increasing function that is upper bounded by unity one for increasing values of parameter r above zero. These theoretical results definitively shows the trade-off between the fidelity and probability. An approximated solution state with a higher probability of success will have a lower fidelity, the converse is true. For example, in the 2D case, an approximate solution state with fidelity at least 0.999, has a probability of success of up to 0.38. In the 4D case, it has a probability of success of up to 0.18.

Additionally, it can be deduced that the full QLS algorithm will have absolutely zero chance of outputting an exact solution state $|x\rangle$ for all values of parameter r above zero. Therefore, it cannot be used as algorithm to solve the quantum linear problem exactly, instead, it is suitable for problems which accept approximated quantum states as a solution, such as quantum optimisation problems.

10.2 On the Experimental Results

The experimental $P(|1\rangle_a)$ and $P\left(\left|\tilde{1}\right\rangle_a\right)$ values in worked examples of both simplified and full version deviates significantly, between $\pm 5\%$ to $\pm 8\%$ even under ideal conditions. This an unexpected result, at least to the author, as the experiments were conducted using a quantum simulator. Perhaps, this might be a consequence of the quantum measurements employed by the IBM quantum simulator, which uses quantum noise models to emulate real quantum devices [IBM19]. The quantum noise are inherent and systematic noise in real quantum devices that cannot eliminated by any classical error correcting methods. This shows any quantum state or operation cannot be perfectly represented by a real quantum computer in general even under ideal conditions. Unfortunately, IBM has not made any explicit mention on what are the ideal noise conditions or the simulated real quantum devices. It is hypothesize and suspected that results reflects the ideal thermal noise of real quantum devices, particularly on superconducting qubits which are extensively used by IBM as a quantum computer. Another reason for the deviation might be due to the use of insufficient runs. 10^4 independent runs may not be statistically significant. Higher number of runs can be conducted, but it leads to a longer experimental completion time.

The simplified and full QLS algorithm are found, unsurprisingly, slower than a typical classical solver by at least 2 and 4 orders of magnitude respectively. In general, the runtime of the 4D case is found to be slower than that 2D case. This may be due be fact that the quantum circuit for the 4D case uses twice as many qubits than the 2D case and has significantly more quantum operation than the 2D case. Therefore, it contributes significantly to the classical computation time needed to classically process highly complex data structure of a quantum circuit, unlike that of the 2D and 4D classical solver.

However, the experimental run-times of the quantum circuits obtained should not be interpreted as actual runtime of the quantum simulation of the quantum circuit, as it is highly dependent on the efficiency of the quantum circuit building functions codes, which may be unoptimised. An anomalous result could be seen be comparing the 4D simplified QLS with the 2D full QLS worked example. The 4D simplified QLS has of 2 order of magnitude faster runtime than that of the 2D full QLS despite both having relatively similar number of quantum gates and qubits, this may be due to inefficient building

functions code used in the full QLS quantum circuit. The author believes that all experimental run-times could be further improved, if the code for the quantum circuit building function codes were to be further optimised.

10.3 Complications

The Dimensionality Caveat

One caveat of the full QLS algorithm is that it only applies to linear systems of 2^n dimension with positive integers n. Any hermitian matrix M of other dimensions cannot be simulated using this algorithm. There are several workaround to this caveat. One 'simple' solution that leverage the above quantum circuit construction is to consider using other j-level quantum states and quantum operations. An example of a QLS algorithm using a three, $(j = 3)$, level quantum state, called qutrits, was first proposed by Marek Sawerwain et al [SL13] who showed that it can be used to solve a 3^n-dimension quantum linear system. However, this 'simple' solution may not always be a practical one, as there are limited physical examples of j-level quantum states and quantum operations. Another possible solution would be to consider appending the matrix of interest with extra trivial elements so that the dimension becomes 2^n. Then, one could solve the quantum linear problem using the full QLS algorithm easily. However, author does not know how to implement the appending process, that is to expand the dimensionality of an quantum operator and embed an unknown quantum operator to it. The author suspects that heavy amounts of pre and post quantum processing may be needed to recover the intended solution state as any appending process might drastically change the intended outcome.

The Quantum Gate Problem

The full QLS algorithm demands a special set of quantum gates that must be readily available for quantum circuit construction. In this thesis, under the 2D and 4D worked example, we have use quantum gates from a set provided by IBM Qiskit quantum simulator. It is possible to use other sets of quantum gates, such as the universal quantum gate set proposed in Nielsen's textbook [NC10]. However, it is important to understand and know the restrictions imposed on the configurability of that set, as it affects the implementability of the full QLS algorithm to solve the quantum linear problem. In this section, we will note the differences in the between the quantum gate set used for constructing the part of the algorithm that is input-independent and another that is input-dependent.

Simplifying the analysis, suppose Alice wants to solve her quantum linear problem for her hermitian operator M and pure quantum state $|b\rangle$ by obtaining the solution state $|x\rangle = \frac{1}{d}M^{-1}|b\rangle$, where d is a normalisation factor. She employs Bob who has a quantum computer preloaded with his input-independent part of the full QLS algorithm and supplies him with her quantum circuit part that is input-dependent. The only quantum circuit part that is input-dependent is the Hamiltonian Simulation (HS) subroutine in the QPE subroutine as it is dependent on the hermitian matrix M. The rest of the quantum circuit is input-independent.

From Bob's perspective, to construct his input-independent part of the full QLS algorithm, he can use quantum gates provided by either the IBM Qiskit quantum gate set or the Nielsen's textbook universal set. He does not need to know anything about Alice's hermitian operator M and pure quantum state

$|b\rangle$, other than the dimensionally of the problem. From Alice's perspective, she need to construct her own HS quantum circuit. What other quantum gates does Alice need, given that she has only quantum measurement gate M? This is the Alice's quantum gate problem that concerns the minimal gate resources or knowledge needed to construct a quantum circuit of the HS subroutine.

The answer is surprisingly simple and yet profound, she is only required to possess multiple copies of a single quantum gate, controlled-$\exp(2\pi i 2^{-t} M)$ quantum gate or at least know its full gate decomposition using Bob's gate set, otherwise, she cannot solve her quantum linear problem. This is the reason for introducing the necessary, yet insufficient, assumption that the eigen-decomposition of M is given: $M = U\Lambda U^{\dagger}$, that is, U, $U^{\dagger}$ and Λ quantum gates are given, in chapter 8. The eigen-decompostion assumption only helps to Alice to construct $\exp(2\pi i 2^{-t} M)$ quantum gate, however, she still might have difficulty in creating a controlled-$\exp(2\pi i 2^{-t} M)$ gate. This is one of the consequences of the no-go theorem of adding control to an arbitrary quantum gate, which is proven by Mateus Araújo et al [Ara+14], where Alice cannot make a controlled-$\exp(2\pi i 2^{-t} M)$ gate from $\exp(2\pi i 2^{-t} M)$ quantum gates and any other other arbitrary quantum gates, without having the ability to fully decompose M or $\exp(2\pi i 2^{-t} M)$ into Pauli operators and quantum gates from Bob's gate set. Hence, this is another reason for deferment of the subsections 9.1.1 and 9.2.1 on the 2D and 4D gate decomposition of controlled-$\exp(2\pi i 2^{-t} M)$ from chapter 8 to 9 so as to allow the author to fully and manually decompose $\exp(2\pi i 2^{-t} M)$ gate, for different dimension cases, into quantum gates from IBM Qiskit gate set for implementation.

The implications of such are counter-intuitive as one usually expects the user to input just an operator M and state $|b\rangle$ to the quantum linear solver similar to the classical case. Some possible workarounds to this input problem are the complete avoidance of using controlled versions of input dependent quantum gates by assuming special case, such as consider using modulus of M, or introducing the assumption that the knowledge and the ability of decomposing unknown quantum gate and constructing controlled version of it are given [Tho+18].

Chapter 11

Future Research

11.1 Algorithm Extensions

The two proposed versions of the QLS algorithm used in this thesis may lead to many other possible extensions as its subroutines can be easily modified or substituted. For example, there are alternative HS subroutines such as Group Leadership Optimization algorithm to find alternative decomposition of the controlled-exp$(2\pi i 2^{-t}M)$ using classical methods by Anmer Daskin et al [DK10], Quantum Walks Algorithm that decomposes controlled-exp$(2\pi i 2^{-t}M)$ into a sum of sparse matrices for efficient implementation by Andrew M. Childs et al [Chi+02; Chi08; CW12]. For the Eigenvalue Inversion subroutine, to invert a number that is finitely represented in binary, one could either use the Thapliyal inversion algorithm by Himanshu Thapliyal et al [TVMC16; Tha+18] or Newton-Raphson Method by Yudong Cao et al [Cao+12]. Readers are also encouraged to consult an excellent review on the comparison analysis of several other possible alternative subroutines by Otmar ubbens [Ubb19]. Other possible extension of QLS is the repurpose of the amplitude amplification (AA) quantum subroutine [Bra+00] to increase the probability of the success of obtaining the solution state. Andris Ambainis took this even further by applying his modified AA, called variable time amplitude amplification, on the QLS algorithm to improve the computation time of the original by a linear factor [Amb10].

Subroutine	Current	Alternative
Hamiltonian Simulation	Eigenvalue Decomposition	• Group Leadership Optimization Algorithm • Quantum Walks Algorithm
Eigenvalue Inversion	Powers of Two	• Thapliyal Inversion Algorithm • Newton-Raphson Method
Ancillary Rotation	1^{st} Order Approximation	3^{rd} and Higher Order Approximation
Ancillary Measurement	-	Amplitude Amplification (Enhancement)

Table 11.1.1: A summary table showing the potential extensions to QLS algorithm.

11.2 Alternate Quantum Linear Solvers

There are alternate versions of quantum linear solver algorithms that solves the quantum linear problem in a different form. Quantum Recommendation System (QRS) is another QLS algorithm that use Quantum Singular Value Estimation algorithm (QSVE) which is first proposed by Iordanis Kerenidis and Anupam Prakash [KP16]. In the simplest sense, QRS is trying to solve a product recommendation problem, a standard problem in the online commerce industry, in a quantum way. The product recommendation problem is described as follows:

> Given a database of the past online user's preferences and ratings for a set of products which have are labelled with certain properties, what kinds of products should the computer system recommends to other future potential online users so that they may purchase it with a high probability.

Soon after QRS was proposed, Leonard Wossnig et al integrated QSVE into QLS algorithm to solve the quantum linear problem for dense hermitian operator M, achieving at most a polynomial speedup in computational time over its classical counterparts [WZP17].

Alternatively, Hsin-Yuan Huang et al investigated the use of variational algorithms for solving linear systems, that is to prepare potential a candidate quantum solution state using a set of parametrized quantum gates and perform measurements on the candidate to determine its quality based a given loss function. Then, based on the loss function, they use a classical parameter optimization loop algorithm to maximise the quality of the candidate quantum solution state [HBR19]. The ideas behind alternate QLS algorithms can be used to further improve the performance of the original QLS.

Chapter 12

Conclusion and Summary

In conclusion, we have analysed the quantum linear problem of obtaining the solution state $|x\rangle = \frac{1}{d}M^{-1}|b\rangle$, given a hermitian operator M and input quantum state $|b\rangle$, where d is a normalisation factor. We have found that $|x\rangle$ can be quantum mechanically interpreted as the normalised pre-collapse quantum state of the measurement of input quantum state $|b\rangle$ by the hermitian operator M^{-1}. By leveraging the power of quantum measurements, together with two other ideas implicitly developed by Aram Harrow, Avinatan Hassidim and Seth Lloyd (HHL), it is possible to construct a quantum linear solver (QLS) algorithm that takes quantum state $|b\rangle$ as input and obtain the solution state $|x\rangle$. Because of the lack of implementation details in the original paper, the author sought to seek improvements and modifications suggested in current literature. Both simplified and full versions of the QLS algorithm were proposed and thoroughly analysed for the 2 and 4, with a extension to the 2^n dimensions. The simplified version aims solve for a particular form of a hermitian operator M which reduces the number of quantum operations by using a truncated Quantum Phase Estimation subroutine, while the full version aims to output a close approximation of the solution state $|x\rangle$ without the need of knowing the eigenvalues. The theoretical results suggest that predicting the probability of obtaining the solution state $|x\rangle$ or its approximated one $|\tilde{x}\rangle$ requires the knowledge of the eigenvalues of M. In addition, constructing any quantum circuit version of the HHL QLS algorithm requires multiple copies or its full gate decomposition of controlled-$\exp(2\pi i 2^{-t}M)$ quantum gate where t is the number of qubits needed to express the eigenvalues of M in qubit binary representation up to a desired precision. Several worked examples were suggested and implemented by constructing and simulating its quantum circuit using the IBM Qiskit quantum simulator. The experimental probability distribution of success was found to deviate by about $\pm 7\%$ of the theoretical value on average. The deviation source is currently unknown but is suspected to be either quantum noise implemented by the IBM Qiskit quantum simulator or due to insufficient number of quantum circuit experiment runs. A summary of the features, assumptions, advantages and disadvantages of the various QLS algorithm analysed in this thesis are given the tables 12.0.1 and 12.0.2 below.

QLS Algorithm	Original	Simplified	Full
Outputs	• Outputs an exact solution state $\|x\rangle$ with probability given in row 1 of table 5.3.1.	• Outputs an exact solution state $\|x\rangle$ with probability given in equation 7.3.7.	• Outputs an approximated solution state $\|\tilde{x}\rangle$ with probability given in equation 9.3.5.
Assumptions	• The QPE and AR subroutines can be efficiently implemented for sparse hermitian operator M. • The input quantum state $\|b\rangle$ can be efficiently prepared.	• The quantum gates R and $R^\dagger$ are given, such that $M = R\Lambda R^\dagger$. • Eigenvalues λ_j of M are of the form $0.\bar{X}\,\bar{j}$, where $\bar{X}$ is a fixed arbitrary finite binary sequence and $\bar{j}$ is the binary sequence of index j.	• The R and $R^\dagger$ and controlled-$\exp(2\pi i 2^{-t}\Lambda)$ quantum gates are given, such that $M = R\Lambda R^\dagger$. • Eigenvalues λ_j of M are integer powers of two. (For a practical implementation only, otherwise assumption not needed in general)
Advantages	• The algorithm guarantees a non-zero probability of obtaining an exact solution state, if parameter C is within allowable limits. • Theoretically, it computes and solves with at most an exponential speedup.	• The algorithm guarantees a non-zero probability of obtaining an exact solution state, if the parameter C is within allowable limits. • The algorithm breaks down when parameter C is outside the allowable limits. However, this is physically unattainable when using AR.	• Due to the approximation, the parameter C can be arbitrary set without affecting the overall quantum circuit structure. • In general, this algorithm can be used for any hermitian operator M with real positive eigenvalues with a finite binary representation. • Its quantum circuit can be constructed without the knowing the the eigenvalues of M. • It can be easily adapted and modified to suit other potential applications.

Table 12.0.1: A summary of the features, assumptions and advantages of the original, simplified and full QLS algorithms. AR refers to the Ancillary Rotation quantum subroutine.

QLS Algorithm	Original	Simplified	Full
Disadvantages	• The AR implementation details were not given in the original paper. Further improvements and modifications original algorithm to the are needed to construct a valid quantum circuit.	• All eigenvalues of hermitian operator M must be known so that it can be used to compute the rotation angles for the Ancillary Rotation subroutine to work. • Extremely limited in its potential applications due to the restrictive assumptions imposed on the type of hermitian operator M.	• It requires the use of an Eigenvalue Inversion subroutine, which may be highly complicated and impractical to implement. • For higher n^{th} order of approximation, a exponential power scaling $O(t^n)$ number of gates are needed for AR, where t is the number of qubits needed to store the eigenvalues using the qubit binary representation. • There is a trade-off between the probability of obtaining the approximated state and its fidelity. • It impossible construct quantum circuit without having multiple copies or the gate decomposition of controlled-$\exp(2\pi i 2^{-t} M)$ quantum gate for the Hamiltonian Simulation subroutine.

Table 12.0.2: A summary of the disadvantages of the original, simplified and full QLS algorithms. AR refers to the Ancillary Rotation quantum subroutine.

Chapter 13

Appendix

13.1 The Experiment codes

All quantum circuits are constructed and simulated on the IBM Qiskit quantum simulator that uses python version 3.6 programming language under the Jupyter Notebook Environment. The link to the codes are given below.

- Simplified QLS Algorithm (2D and 4D Worked Examples)
  - `https://gist.github.com/cheechonghian/ea7ed914a264444b540aa57c428df497`
- Full QLS Algorithm (2D and 4D Worked Examples)
  - `https://gist.github.com/cheechonghian/fc1c1a321f1e36b216eb7615d7edba65`

The author recommends Anaconda to download the following environment: `https://anaconda.org/chee0122/myQuantumCircuitSimulator2/files` and Jupyter Notebook Version 6.0.1 to access, view and run the python codes which have all the worked examples elaborated in this thesis.

Bibliography

[Amb10] Andris Ambainis. "Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations". In: (2010), pp. 1–17. arXiv: 1010.4458. URL: http://arxiv.org/abs/1010.4458.

[Ara+14] Mateus Araújo et al. "Quantum circuits cannot control unknown operations". In: *New Journal of Physics* 16.9 (2014), p. 093026. ISSN: 1367-2630. DOI: 10.1088/1367-2630/16/9/093026. URL: http://stacks.iop.org/1367-2630/16/i=9/a=093026?key=crossref.665d2000dcb8d8581cfae47ce9005e26.

[Bar+13] R. Barends et al. "Coherent Josephson Qubit Suitable for Scalable Quantum Integrated Circuits". In: *Physical Review Letters* 111.8 (2013), p. 080502. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.111.080502. arXiv: 1304.2322v1. URL: https://link.aps.org/doi/10.1103/PhysRevLett.111.080502.

[Bar+15] Stefanie Barz et al. "A two-qubit photonic quantum processor and its application to solving systems of linear equations". In: *Scientific Reports* 4.1 (2015), p. 6115. ISSN: 2045-2322. DOI: 10.1038/srep06115. arXiv: 1302.1210. URL: https://arxiv.org/abs/1302.1210.

[Bar+95] Adriano Barenco et al. "Elementary gates for quantum computation". In: *Physical Review A* 52.5 (1995), pp. 3457–3467. ISSN: 1050-2947. DOI: 10.1103/PhysRevA.52.3457. arXiv: 9503016v1 [quant-ph]. URL: https://link.aps.org/doi/10.1103/PhysRevA.52.3457.

[Ben82] Paul Benioff. "Quantum Mechanical Models of Turing Machines That Dissipate No Energy". In: *Physical Review Letters* 48.23 (1982), pp. 1581–1585. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.48.1581. URL: https://link.aps.org/doi/10.1103/PhysRevLett.48.1581.

[Bra+00] Gilles Brassard et al. "Quantum Amplitude Amplification and Estimation". In: (2000), pp. 53–74. DOI: 10.1090/conm/305/05215. arXiv: 0005055 [quant-ph]. URL: http://arxiv.org/abs/quant-ph/0005055.

[Cai+13] X.-D. Cai et al. "Experimental Quantum Computing to Solve Systems of Linear Equations". In: *Physical Review Letters* 110.23 (2013), p. 230501. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.110.230501. arXiv: 1302.4310. URL: https://link.aps.org/doi/10.1103/PhysRevLett.110.230501.

[Cao+11] Yudong Cao et al. “Quantum Circuit Design for Solving Linear Systems of Equations”. In: *Molecular Physics* 110.15-16 (2011), pp. 1675–1680. ISSN: 0026-8976. DOI: 10.1080/00268976.2012.668289. arXiv: 1110.2232. URL: http://arxiv.org/abs/1110.2232http://dx.doi.org/10.1080/00268976.2012.668289.

[Cao+12] Yudong Cao et al. “Quantum algorithm and circuit design solving the Poisson equation”. In: *New Journal of Physics* 15 (2012). ISSN: 13672630. DOI: 10.1088/1367-2630/15/1/013021. arXiv: 1207.2485. URL: http://arxiv.org/abs/1207.2485.

[Chi+02] Andrew M. Childs et al. “Exponential algorithmic speedup by quantum walk”. In: (2002), pp. 1–24. DOI: 10.1145/780542.780552. arXiv: 0209131 [quant-ph]. URL: http://arxiv.org/abs/quant-ph/0209131.

[Chi08] Andrew M Childs. “On the relationship between continuous- and discrete-time quantum walk”. In: (2008), pp. 1–22. DOI: 10.1007/s00220-009-0930-1. arXiv: 0810.0312. URL: http://arxiv.org/abs/0810.0312.

[CJS13] B. D. Clader, B. C. Jacobs and C. R. Sprouse. “Preconditioned quantum linear system algorithm”. In: *Physical Review Letters* 110.25 (2013), pp. 1–5. ISSN: 00319007. DOI: 10.1103/PhysRevLett.110.250504. arXiv: 1301.2340. URL: http://arxiv.org/abs/1301.2340.

[Cle+98] R. Cleve et al. “Quantum algorithms revisited”. In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 454.1969 (1998), pp. 339–354. ISSN: 1364-5021. DOI: 10.1098/rspa.1998.0164. arXiv: 9708016v1 [quant-ph]. URL: https://royalsocietypublishing.org/doi/10.1098/rspa.1998.0164.

[Col+18] Patrick J. Coles et al. “Quantum Algorithm Implementations for Beginners”. In: (2018). arXiv: 1804.03719. URL: http://arxiv.org/abs/1804.03719.

[CW12] Andrew M. Childs and Nathan Wiebe. “Hamiltonian Simulation Using Linear Combinations of Unitary Operations”. In: *Quantum Information and Computation* 12.11-12 (2012), pp. 901–924. ISSN: 15337146. DOI: 10.26421/QIC12.11-12. arXiv: 1202.5822. URL: http://arxiv.org/abs/1202.5822.

[DK10] Anmer Daskin and Sabre Kais. “Group Leaders Optimization Algorithm”. In: *Making Things Happen: A Model of Proactive Motivation* 36.4 (2010), pp. 827–856. ISSN: 0026-8976. DOI: 10.1080/00268976.2011.552444. arXiv: 1004.2242. URL: http://arxiv.org/abs/1004.2242.

[Gro96] Lov K. Grover. “A fast quantum mechanical algorithm for database search”. In: *Proceedings 28th Annual ACM Symposium on the Theory of Computing*. 1996, pp. 212–219. arXiv: 9605043 [quant-ph]. URL: http://arxiv.org/abs/quant-ph/9605043.

[HBR19] Hsin-Yuan Huang, Kishor Bharti and Patrick Rebentrost. “Near-term quantum algorithms for linear systems of equations”. In: (2019), pp. 1–17. arXiv: 1909.07344. URL: http://arxiv.org/abs/1909.07344.

[HHL09] Aram W. Harrow, Avinatan Hassidim and Seth Lloyd. “Quantum Algorithm for Linear Systems of Equations”. In: *Physical Review Letters* 103.15 (2009), p. 150502. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.103.150502. arXiv: 0811.3171v3. URL: https://link.aps.org/doi/10.1103/PhysRevLett.103.150502.

[IBM19] IBM. *IBM Qiskit Aer Backend Quantum Simulator*. 2019. URL: https://qiskit.org/documentation/the{_}elements.html{\#}aer (visited on 23/10/2019).

[Kit95] A. Yu. Kitaev. “Quantum measurements and the Abelian Stabilizer Problem”. In: (1995), pp. 1–22. arXiv: 9511026v1 [quant-ph]. URL: https://arxiv.org/abs/quant-ph/9511026v1.

[KP16] Iordanis Kerenidis and Anupam Prakash. “Quantum Recommendation Systems”. In: *Leibniz International Proceedings in Informatics, LIPIcs* 67 (2016), pp. 1–22. ISSN: 18688969. DOI: 10.4230/LIPIcs.ITCS.2017.49. arXiv: 1603.08675. URL: http://arxiv.org/abs/1603.08675.

[KP17] Iordanis Kerenidis and Anupam Prakash. “Quantum gradient descent for linear systems and least squares”. In: (2017). arXiv: 1704.04992. URL: http://arxiv.org/abs/1704.04992.

[MKF18] Kosuke Mitarai, Masahiro Kitagawa and Keisuke Fujii. “Quantum Analog-Digital Conversion”. In: *Physical Review A* 99.1 (2018), p. 012301. ISSN: 2469-9926. DOI: 10.1103/PhysRevA.99.012301. arXiv: 1805.11250. URL: http://arxiv.org/abs/1805.11250.

[MP15] Ashley Montanaro and Sam Pallister. “Quantum algorithms and the finite element method”. In: (2015), pp. 1–16. DOI: 10.1103/PhysRevA.93.032324. arXiv: 1512.05903. URL: http://arxiv.org/abs/1512.05903.

[MV05] M. Mottonen and J. J. Vartiainen. “Decompositions of general quantum gates”. In: (2005). arXiv: 0504100 [quant-ph]. URL: http://arxiv.org/abs/quant-ph/0504100.

[NC10] Michael a. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge: Cambridge University Press, 2010, p. 702. ISBN: 9780511976667. DOI: 10.1017/CBO9780511976667. arXiv: arXiv:1011.1669v3. URL: http://ebooks.cambridge.org/ref/id/CBO9780511976667.

[Pan+14] Jian Pan et al. “Experimental realization of quantum algorithm for solving linear systems of equations”. In: *Physical Review A* 89.2 (2014), p. 022313. ISSN: 1050-2947. DOI: 10.1103/PhysRevA.89.022313. arXiv: 1302.1946v1. URL: https://link.aps.org/doi/10.1103/PhysRevA.89.022313.

[RML13] Patrick Rebentrost, Masoud Mohseni and Seth Lloyd. “Quantum support vector machine for big data classification”. In: *Physical Review Letters* 113.13 (2013), p. 130503. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.113.130503. arXiv: 1307.0471. URL: http://arxiv.org/abs/1307.0471.

[SB+16] Horst Schmidt-Böcking et al. “The Stern-Gerlach experiment revisited”. In: *The European Physical Journal H* 41.4-5 (2016), pp. 327–364. ISSN: 2102-6459. DOI: 10.1140/epjh/e2016-70053-2. arXiv: 1609.09311v1. URL: http://link.springer.com/10.1140/epjh/e2016-70053-2.

[SBM04] Vivek V. Shende, Stephen S. Bullock and Igor L. Markov. “Synthesis of Quantum Logic Circuits”. In: *Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC* 1 (2004), pp. 272–275. ISSN: 02780070. DOI: 10.1109/TCAD.2005.855930. arXiv: 0406176 [quant-ph]. URL: http://arxiv.org/abs/quant-ph/0406176.

[Sho94] P.W. Shor. “Algorithms for quantum computation: discrete logarithms and factoring”. In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. IEEE Comput. Soc. Press, 1994, pp. 124–134. ISBN: 0-8186-6580-7. DOI: 10.1109/SFCS.1994.365700. URL: http://ieeexplore.ieee.org/document/365700/.

[SL13] Marek Sawerwain and Wiesław Leoński. “Quantum circuits based on qutrits as a tool for solving systems of linear equations”. In: (2013). arXiv: 1309.0800. URL: http://arxiv.org/abs/1309.0800.

[SN17] Jun John Sakurai and Jim Napolitano. *Modern Quantum Mechanics*. Cambridge University Press, 2017. ISBN: 9781108499996. DOI: 10.1017/9781108499996. URL: https://doi.org/10.1017/9781108499996.

[Tha+18] Himanshu Thapliyal et al. “Quantum Circuit Designs of Integer Division Optimizing T-count and T-depth”. In: (2018), pp. 1–11. arXiv: 1809.09732. URL: http://arxiv.org/abs/1809.09732.

[Tho+18] Jayne Thompson et al. “Quantum computing with black-box subroutines”. In: *New Journal of Physics* 20.1 (2018), p. 013004. ISSN: 1367-2630. arXiv: 1310.2927v5. URL: https://arxiv.org/abs/1310.2927v5.

[TVMC16] Himanshu Thapliyal, T. S. S. Varun and Edgard Munoz-Coreas. “Quantum Circuit Design of Integer Division Optimizing Ancillary Qubits and T-Count”. In: (2016), pp. 2–5. arXiv: 1609.01241. URL: http://arxiv.org/abs/1609.01241.

[Ubb19] Otmar Ubbens. “Practical Implementation of a Quantum Algorithm for the Solution of Systems of Linear Systems of Equations Otmar Ubbens”. Bachelor Thesis. Delft University of Technology, 2019. URL: https://repository.tudelft.nl/islandora/object/uuid{\%}3A444580f0-a661-4adc-a937-51c5660916d9.

[WBL12] Nathan Wiebe, Daniel Braun and Seth Lloyd. “Quantum Algorithm for Data Fitting”. In: *Physical Review Letters* 109.5 (2012), p. 050505. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.109.050505. arXiv: 1204.5242. URL: http://arxiv.org/abs/1204.5242.

[WZP17] Leonard Wossnig, Zhikuan Zhao and Anupam Prakash. “A quantum linear system algorithm for dense matrices”. In: *Physical Review Letters* 120.5 (2017), p. 050502. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.120.050502. arXiv: 1704.06174. URL: http://arxiv.org/abs/1704.06174.

[Zhe+17] Yarui Zheng et al. “Solving Systems of Linear Equations with a Superconducting Quantum Processor”. In: *Physical Review Letters* 118.21 (2017), p. 210504. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.118.210504. arXiv: 1703.06613v1. URL: http://link.aps.org/doi/10.1103/PhysRevLett.118.210504.

[ZYZ19] Xiong Zhang, Zhenwei Yang and Xiangdong Zhang. “Simplified experimental scheme of quantum algorithm for solving linear equations with single photons”. In: *Optics Express* 27.3 (2019), p. 3369. ISSN: 1094-4087. DOI: 10.1364/OE.27.003369. URL: https://www.osapublishing.org/abstract.cfm?URI=oe-27-3-3369.