



Durham E-Theses

Quantum Field Theory Simulation using Digital Quantum Computers

WADHIA, VIVEK

How to cite:

WADHIA, VIVEK (2021) *Quantum Field Theory Simulation using Digital Quantum Computers*, Durham theses, Durham University. Available at Durham E-Theses Online: <http://etheses.dur.ac.uk/14306/>

Use policy

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a [link](#) is made to the metadata record in Durham E-Theses
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the [full Durham E-Theses policy](#) for further details.

Quantum Field Theory Simulation using Digital Quantum Computers

Vivek Wadhia

A thesis submitted in fulfilment of the degree of
Masters of Science by Research



Department of Physics
Durham University
September 2021

Quantum Field Theory Simulation using Digital Quantum Computers

Vivek Wadhia

Abstract

Quantum field theory (QFT) forms the fundamental basis of many areas in physics and is indispensable for our understanding of the universe. However, there are many different quantum field theories, and the equations associated with each of these are difficult to solve and understand analytically. In recent years, quantum computing has come to the forefront of developments in physics and there is much excitement about potential applications for this new technology. This work seeks to ascertain if it is possible to simulate quantum field theories on this current generation of quantum computers.

Specifically, using quantum walks as an algorithmic implementation, simulated on superconducting quantum computers provided by IBM, using the qiskit software package.

Declaration

I confirm that no part of the material offered has previously been submitted by myself for a degree in this or any other University. Where material has been generated through joint work, the work of others has been indicated.

Vivek Wadhia
September, 2021

The copyright of this thesis rests with the author. No quotation from it should be published without the author's prior written consent and information derived from it should be acknowledged.

Acknowledgments

First and foremost I would like to thank my supervisor Dr. Viv Kendon, for guiding me through the forest that is research, in the most unconventional of years. I would also like to thank Michael Spannowsky and Steve Abel from IPPP, for offering their support in this work. I would be remiss without thanking the guys in the office: Luke, Ben, Anton, Toon, Jack, Joseph, Dan - for all their support and friendship, what good is life without making memories? Finally, I would like to thank the rest of QLM for being so welcoming and friendly.

*Dedicated to my parents,
without whom none of this would be possible.*

Contents

Abstract	i
Declaration	ii
Acknowledgements	iii
Outline	1
1 Introduction to Quantum Computing	2
2 Introduction to QFT Simulation and Motivation	4
2.1 Motivation	5
3 The Dirac Equation	6
4 Introduction to Quantum Walks	12
4.1 Discrete-time Random Walk	12
4.2 Continuous-time Random Walk	13
4.3 Discrete-time Quantum Walk	13
4.4 Continuous-time Quantum Walk	18
4.5 Discrete-time Quantum Walk on a Cycle Graph	18
5 Feynman’s Checkerboard and the Dirac Equation	19
5.1 Checkerboard Path Integral	20
5.2 Connection to the Dirac Equation	22
6 Quantum Walks and Quantum Field Theory	23
7 The Transmon	28
8 Backends and Methods	34
8.1 Methods	35
9 Characterisation of Current Hardware	37
10 Custom Noise Models	46

11 Simulated Quantum Walks vs. Experimental Quantum Walks	53
11.1 Silicon Photonic Quantum Walk	55
11.2 Comparison to Trapped ions	57
12 Conclusions and Further Work	62
12.1 Further Work	65
Appendix	71
A.0 Spinors	71
A.1 Feynman Path Integral and Propagator	72
A.2 Device Topologies	77
A.3 8-node Quantum Walk Circuit	78
A.4 16-node Quantum Walk Circuit	79
A.5 Fock states and Coherent states	80
A.6 Fidelity over 8 days in August 2021	82

List of Figures

4.1	Probability distribution for a classical discrete-time random walk of 100 steps.	13
4.2	Position of the walker on a portion of the number line after 3 steps. Red arrows indicate the state of the coin, up arrow means $ 0\rangle$, down arrow means $ 1\rangle$. Size of the arrow corresponds to the amplitude in the wave function.	16
4.3	Probability distribution for a DTQW of 100 steps. Initial state of the coin is $\frac{1}{\sqrt{2}}(0\rangle + 1\rangle)$	17
4.4	Probability distribution for a DTQW of 100 steps. Coin initialised as $ 1\rangle$ (blue) and coin initialised as $ 0\rangle$ (red).	18
5.1	Illustration of the path of a relativistic particle in (1+1)-d, where x is the space dimension and t is the time dimension.	20
7.1	Basic construction of the transmon.	28
7.2	Diagram of a quantised LC circuit. L is the inductor and C is the capacitor.	29
7.3	Energy diagram of a quantum simple harmonic oscillator. Blue lines indicate energy levels with separation $\hbar\omega$	31
7.4	Energy diagram of the Josephson potential (solid blue) compared to the harmonic oscillator (dashed orange). Wavelength of the potential is given by the quantum flux and the amplitude by twice the Josephson energy.	32
7.5	Energy spectrum of the transmon. Energy levels are no longer equally spaced apart.	33
9.1	8-node QW graph. Binary string (i.e. the state of the quantum register) corresponds to the site number in denary (bold).	37

9.2	Discrete-time quantum walk for an 8-node graph. Panels a) to i) show steps 0 to 8 of the walk respectively (text size on histograms is fixed by the qiskit software). Bars represent the probability of finding the walker at a site denoted by a bit string. The blue bars represent the result of the simulator i.e. the expected result and red bars the real device. Panel j) shows the Hellinger fidelity [60] between the real device and the simulator for each step of the walk. Dotted red-line is to guide the eye. A value of 1.0 indicates perfect agreement between the simulator and the real device.	39
9.3	4-node QW graph. As in Figure 9.1 vertices represent the possible positions of the walker and binary string corresponds to the site number in denary (bold).	40
9.4	Circuit for 4-node quantum walk. Decomposition of STEP gates are enclosed by the dashed lines. Coin operation is given by single Hadamard gates acting on the coin register. These gates are repeated for the desired number of steps.	40
9.5	Discrete-time quantum walk for an 4-node graph. Panels a) to e) show steps 0 to 4 of the walk respectively. Bars represent the probability of finding the walker at a certain site denoted by a bit string. The blue bars represent the result of the simulator and red bars the result of the real device. Panel f) shows the fidelity, comparing the two probability distributions, of the walk for each step between the real device and simulator.	41
9.6	Optimised circuit consisting of basis gates. One step is depicted. First three gates implement the Hadamard coin operator. . . .	43
9.7	Comparison of discrete time quantum walk for a 4-node graph. Panels a) to e) simulator (light blue bars) vs. standard transpiled (red bars) vs. optimised (purple bars) for steps 0 to 4 respectively. Panel f) shows the fidelity, comparing probability distributions, of the standard transpiled (red) and optimised (magenta) for each step of the walk. Error bars represent standard error. . . .	44
10.1	Diagram of the 16-node cycle graph used in the DTQW. . . .	46
10.2	Fidelity at each step of a 16-node, 16 step DTQW, for varying noise strengths of the custom noise model. Error bars represent standard error.	48
10.3	Fidelity at each step of a 16-node, 16 step DTQW, for varying noise strengths of the custom noise model as well as the IBM noise model and Santiago.	49

10.4	Probability distributions for the 16th step of the 16-node DTQW: a) 0% noise strength, b) 2%, c) 6%, d) 10%, e) 20%, f) 50%, g) 100%, h) IBM noise model (magenta), i) Santiago (red).	51
11.1	Probability distribution for 8 steps of the 16-node DTQW run on <code>qasm_simulator</code> a) and using the IBM noise model b). . . .	53
11.2	Probability distributions for 8 steps of the 16-node DTQW. From top left, a) 0%, b) 2%, c) 6%, d) 10%, e) 20%, f) 50%, g) 100% of the full noise strength and h) Santiago (red).	54
11.3	Figure 3 b) of [63]. Red bars represent the experimental probability of finding a photon at that port. Blue bars represent the simulated probability.	56
11.4	Figure 15 of [65] showing the probability distribution (in Fock space) for a DTQW of 3 steps on a line, α_s denote the position on the line e.g. α_0 is position 0 etc. Negative probabilities occur as a result of the authors subtracting overlapping probabilities.	58
11.5	Probability distribution for a 3 step DTQW on a 16-node cycle graph run on <code>qasm_simulator</code> a) and using the IBM noise model b).	58
11.6	Probability distribution for a 3 step DTQW on a 16-node graph. From top left, a) 0%, b) 2%, c) 6%, d) 10%, e) 20% , f) 50%, g) 100% of the full noise strength and h) Santiago (red).	59
A.1	Illustration of a flag in a 2-d plane, used as a loose analogy for a spinor. The flag itself can rotate about the pole (indicated by the dark blue arrow) and the pole itself can change direction.	71
A.2	Double-slit experiment set-up. A source produces identical particles that travel through a screen with two slits, the particles can travel through either of the slits and arrive at some point P on the final screen which is a detector.	73
A.3	Double-slit experiment with additional screens and slits (red).	73
A.4	Topology (connectivity) of qubits in <code>ibmq_santiago</code> . This has the same topology as a straight line.	77
A.5	Topology (connectivity) of qubits in <code>ibmq_quito</code> . This has the same topology as a “T” shape.	77

A.6	Quantum circuit of [58]. Figure depicts one step of the walk. Increment and Decrement gate sets are repeated for the desired number of steps and qubits 1-3 are used to determine the position by evaluating the value of the register.	78
A.7	Quantum circuit for the 16-node DTQW using 5 qubits. Figure depicts one step of the walk. Qubits 1-4 are used to evaluate the position of the walker and qubit 5 acts as the coin.	79
A.8	Fidelity for 8 step, 16-node DTQW (red) run over 8 days in August 2021. Magenta dashed-line indicates the average fidelity over those 8 days. Error bars represent standard error. Dates are formatted dd/mm/yy.	83
A.9	Corresponding probability distributions for the 8 step, 16-node DTQW run over 8 days in August 2021, run on Santiago. Panels a) to h) correspond to dates 3 to 10 of August 2021 respectively.	84

Outline

This thesis is structured as follows, beginning with an introduction to quantum computing and what it means to simulate a quantum field theory, the fundamental concepts for this work are introduced and motivated. This is followed by the introduction of the Dirac equation and Dirac field, as well as classical and quantum random walks. The connection between random walks and the Dirac equation is then established with Feynman's Checkerboard, and the theoretical connection between quantum field theories and quantum walks is concluded in Chapter 6. A brief detour in discourse is then taken in order to introduce the transmon, a type of qubit used in superconducting quantum computers, as well as some technical details related to qiskit and the IBM hardware used. Following this, a series of discrete-time quantum walks are run on multiple IBM devices in order to ascertain the performance of current hardware. This is followed by the exploration of custom-made noise models, and the amount of noise reduction required to perform an effective simulation is determined. Finally, a comparison is made between the noise found in the simulation of quantum walks in IBM quantum computers, and two experimental implementations of quantum walks. The thesis ends with some concluding remarks.

Chapter 1

Introduction to Quantum Computing

The notion of a quantum computer is relatively new to the field of physics and began with Feynman in 1982 [1], who conceptualised the idea of directly using the rules of quantum mechanics to build a computer. However, nobody at the time knew what such a device would look like. In 1985, Deutsch proposed the idea of a quantum Turing machine [2], the first realisation of a quantum computer in a robust theoretical model. Then in 1992, he and Josza gave the first theoretical example of a quantum computer outperforming a classical computer [3]. This was followed by an explosion of work on quantum algorithms in the rest of the 1990s. Bernstein and Vazirani (1993) showed that a quantum computer can be significantly faster than a classical computer, even if the classical computer is allowed a small probability of error [4]. In 1994, Simon showed quantum computers can be exponentially faster [5] and then in 1995 Shor showed that quantum computers can factorise large, prime integers efficiently, i.e. a polynomial scaling with input size [6]. A task that cannot be achieved efficiently by any known classical algorithm. Grover (1996) then developed a quantum algorithm to solve the unstructured search problem, famously known as Grover's algorithm [7]. In the same year, Lloyd proposed a quantum algorithm that is capable of efficiently (polynomial scaling) simulating quantum-mechanical systems [8]. This effectively resolved the suspected final conjecture that Feynman began with, that a quantum system is most effectively simulated by another quantum system.

The late 1990s also saw the start of a period of significant experimental progress in the development of quantum computers. Quantum teleportation was first demonstrated in 1997 by an international collaboration based in Innsbruck [9]. Quantum error-correction was demonstrated at MIT in 1998 [10]. In 2001, IBM used Shor's algorithm to factorise $15 = 3 \times 5$ using NMR [11]. In 2004, 8 qubits were controlled in an ion trap at Innsbruck [12]. In 2012, IBM achieved 100 μs of coherence time for superconducting qubits [13].

Theoretical and experimental development in quantum computation continues at an increasing rate to this day, in part due to the increasing number of research centres, start-ups and large corporations all contributing to the field. This current period of quantum computing is known as the NISQ-era (Noisy Intermediate-Scale Quantum) a term coined by John Preskill [14], and used widely throughout the field to describe the current hardware available as well as its present capabilities. *Noisy* refers to the resilience of each qubit against decohering effects and *Intermediate-Scale* to the size of the quantum register.

Chapter 2

Introduction to QFT Simulation and Motivation

The topic of quantum field theory (QFT) simulation requires us to first define what a quantum field theory is. Surprisingly, this is not such an easy task and depends on what a quantum field theory means to the person in question. Nevertheless, here is one definition: a quantum field theory is a mathematical framework that generalises quantum mechanics to a continuous space and time setting. In very simplistic terms, it is a combination of quantum mechanics and special relativity. However, this is where we encounter the nuances of this topic. For a particle physicist, this definition fits quite well, as quantum field theory is often used as a tool to study the Standard Model as well as other particle physics phenomenology. In other words, a scenario where the particles (usually) have high-energy and (usually) move close to the speed of light. However, quantum field theory also plays an important role in condensed matter physics, for example in the study of both metals and superconductors. Here we have a situation where the objects to study are both low-energy and non-relativistic. Nevertheless, quantum field theory is required to describe the behaviour of, as an example, low energy fluctuations of electrons in metals, i.e. Fermi liquid theory. Quantum field theory can even extend to the classical realm, as the critical point of water (at 647.14K) is described by a 3-d Ising model, which is a quantum field theory [15].

With these nuances in mind, let us define what a quantum field theory means to us, and specifically what it means to simulate one. Wilson was the first to consider quantum field theory simulation [16,17]. His great insight was to consider a quantum field theory existing on discrete points, like a lattice, and taking the long-distance limit of that lattice, so that the spacing of the points effectively goes to zero, in order to recover the full physics of that system, i.e. the field theory. This perspective on quantum field theory lends itself well to simulation, as digital computers are discretised (binary, 0 or 1) and therefore any simulation must also be discretised. Naturally, this also extends to digital quantum computers. So in this sense, a quantum field theory can be thought of as a series of space and time slices that when put together give the full theory. This is the fundamental approach to simulating quantum field theory on digital quantum computers.

2.1 Motivation

From a theoretical perspective, quantum field theory forms the fundamental basis of many areas of study in physics. The aforementioned particle physics and condensed matter physics being two prominent examples but also other areas such as atomic physics, all requiring some kind of quantum field theory to explain fundamental interactions. These underlying interactions give rise to more complicated, larger systems, ultimately tying the microscopic and macroscopic worlds together. Therefore in order to build an understanding of the universe, at any scale, we must first begin on the most fundamental level.

However, due to this very reason, the nature of many quantum field theories leads them to be very complicated, and in general, difficult to solve. It is therefore natural to see if computers can aid in this process. Indeed, using classical computation to solve quantum field theories is a well established method in areas such as lattice QCD, and is very much a field of active research [18–20]. However, it is believed that there are problems that cannot be computed efficiently using classical computers, and quantum computing offers a possible, efficient solution to these. Recently, there has been growing interest in ways of simulating quantum field theories on both analogue and digital quantum computers [21–25]. The standard approach is to use Trotterisation, in order to split the Hamiltonian into time slices and simulate the evolution. However, due to the limitations of NISQ-era digital hardware, the required overheads are currently suspected to be too great to perform an efficient computation. Hence, this work shall explore a novel technique of using quantum walks to simulate quantum field theories, as an alternative to this more general method.

Chapter 3

The Dirac Equation

Perhaps the simplest example of a quantum field theory comes from the Dirac equation, which in turn gives rise to Dirac fields. The Dirac equation is a description of massive spin-1/2 particles, such as electrons and quarks, that is consistent with quantum mechanics and special relativity.

In non-relativistic quantum mechanics, wavefunctions are described by the time-dependent Schrödinger equation

$$-\frac{1}{2m}\nabla^2\psi + V\psi = i\frac{\partial\psi}{\partial t}, \quad (3.1)$$

where the units have been set $\hbar = c = 1$, ψ is the wavefunction, m is the mass of the particle and V is the potential energy. Equation (3.1) can be expressed as kinetic energy plus potential energy equals total energy. From quantum mechanics, as a consequence of the Schrödinger equation, there is also an equivalency between terms and operators

$$p \rightarrow -i\nabla \quad \text{and} \quad E \rightarrow i\frac{\partial}{\partial t}, \quad (3.2)$$

where p is the momentum and E is the total energy.

Energy-momentum conservation, from special relativity, is given by

$$E^2 - p^2 = m^2. \quad (3.3)$$

Substituting the operators from equation (3.2) into (3.3) gives the Klein-Gordon equation

$$-\frac{\partial^2\psi}{\partial t^2} + \nabla^2\psi - m^2\psi = 0. \quad (3.4)$$

In covariant notation, using the (+ - - -) signature, this becomes

$$-\partial^\mu\partial_\mu\psi - m^2\psi = 0. \quad (3.5)$$

Assuming that ψ is a solution of the Klein-Gordon equation and multiplying (3.4) by $-i\psi^*$,

$$i\psi^* \frac{\partial^2 \psi}{\partial t^2} - i\psi^* \nabla^2 \psi + i\psi^* m^2 = 0. \quad (3.6)$$

Taking the complex conjugate of the Klein-Gordon equation and multiplying by $-i\psi^*$

$$i\psi^* \frac{\partial^2 \psi^*}{\partial t^2} - i\psi \nabla^2 \psi^* + i\psi m^2 = 0. \quad (3.7)$$

Subtracting (3.7) from (3.6)

$$\frac{\partial}{\partial t} \left[i \left(\psi^* \frac{\partial \psi}{\partial t} - \psi \frac{\partial \psi^*}{\partial t} \right) \right] + \nabla \cdot [-i(\psi^* \nabla \psi - \psi \nabla \psi^*)] = 0. \quad (3.8)$$

This has the form of a continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \vec{j} = 0, \quad (3.9)$$

with a probability density defined by

$$\rho = i \left(\psi^* \frac{\partial \psi}{\partial t} - \psi \frac{\partial \psi^*}{\partial t} \right), \quad (3.10)$$

and a probability density current of

$$\vec{j} = i(\psi^* \nabla \psi - \psi \nabla \psi^*). \quad (3.11)$$

Suppose a solution to the Klein-Gordon equation is a free particle with energy E and momentum \vec{p}

$$\psi = N e^{-ip_\mu x^\mu}. \quad (3.12)$$

Substituting this solution into (3.10) gives

$$\rho = 2E|N|^2 \quad (3.13)$$

i.e. probability density is proportional to the energy of the particle.

Substituting the free particle solution into the Klein-Gordon equation gives back

$$E^2 - p^2 = m^2.$$

Therefore

$$E = \pm \sqrt{p^2 + m^2}. \quad (3.14)$$

This gives negative energy solutions. Moreover, the fact probability density is proportional to E implies negative probability densities - which don't exist.

This problem is what motivated Dirac to develop his equation.

Let us begin by factorising the energy-momentum relation. In covariant form this is

$$p^\mu p_\mu - m^2, \quad (3.15)$$

p^μ is the 4-momentum (E, p_x, p_y, p_z) . Factorising gives

$$(\beta^\kappa p_\kappa + m)(\gamma^\lambda p_\lambda - m) \quad (3.16)$$

where $\kappa, \lambda = 0, 1, 2, 3$. Writing this out explicitly, (3.15) is

$$p_0^2 - \vec{p} \cdot \vec{p} = p_0^2 - p_1^2 - p_2^2 - p_3^2 - m^2. \quad (3.17)$$

Equation (3.16) is

$$(\beta^0 p_0 - \beta^1 p_1 - \beta^2 p_2 - \beta^3 p_3 + m)(\gamma^0 p_0 - \gamma^1 p_1 - \gamma^2 p_2 - \gamma^3 p_3 - m). \quad (3.18)$$

Expanding this gives

$$\beta^\kappa \gamma^\lambda p_\kappa p_\lambda - m^2 + m\gamma^\lambda p_\lambda - m\beta^\kappa p_\kappa. \quad (3.19)$$

Choosing $\beta^\kappa = \gamma^\kappa$, removes terms linear in p , giving

$$p^\mu p_\mu - m^2 = \gamma^\kappa \gamma^\lambda p_\kappa p_\lambda - m^2. \quad (3.20)$$

Since $\kappa, \lambda = 0, 1, 2, 3$

$$\begin{aligned} \gamma^\kappa \gamma^\lambda p_\kappa p_\lambda - m^2 &= (\gamma^0)^2 p_0^2 + (\gamma^1)^2 p_1^2 + (\gamma^2)^2 p_2^2 + (\gamma^3)^2 p_3^2 \\ &\quad + (\gamma^0 \gamma^1 + \gamma^1 \gamma^0) p_0 p_1 + (\gamma^0 \gamma^2 + \gamma^2 \gamma^0) p_0 p_2 \\ &\quad + (\gamma^0 \gamma^3 + \gamma^3 \gamma^0) p_0 p_3 + (\gamma^1 \gamma^2 + \gamma^2 \gamma^1) p_1 p_2 \\ &\quad + (\gamma^1 \gamma^3 + \gamma^3 \gamma^1) p_1 p_3 + (\gamma^2 \gamma^3 + \gamma^3 \gamma^2) p_2 p_3 \\ &\quad - m^2 \end{aligned} \quad (3.21)$$

which must be equal to (3.17).

This leads us to the notion of the γ matrices. Dirac's realisation was that in order for the above relation to be true, the γ factors must be 4×4 matrices that also satisfy the anti-commutation relation,

$$\{\gamma^\mu, \gamma^\nu\} = 2g^{\mu\nu} \quad (3.22)$$

where $g^{\mu\nu}$ is the metric,

$$g^{\mu\nu} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}. \quad (3.23)$$

The γ matrices are given by

$$\gamma^0 = \begin{pmatrix} \mathbb{1} & 0 \\ 0 & -\mathbb{1} \end{pmatrix}, \quad \gamma^\mu = \begin{pmatrix} 0 & \sigma^\mu \\ -\sigma^\mu & 0 \end{pmatrix}, \quad (3.24)$$

$\mathbb{1}$ is the 2×2 identity matrix, 0 is a 2×2 null matrix and σ^μ are the 2×2 Pauli matrices, where $\mu = 1, 2, 3$ corresponds to the Pauli x, y, z matrices respectively.

An additional γ matrix is also introduced, $\gamma^5 = i\gamma^0\gamma^1\gamma^2\gamma^3$, and has the property

$$(\gamma^5)^2 = 1, \quad \{\gamma^5, \gamma^\mu\} = 0. \quad (3.25)$$

In other words

$$\gamma^5 = \begin{pmatrix} 0 & \mathbb{1} \\ \mathbb{1} & 0 \end{pmatrix}. \quad (3.26)$$

The Hermitian conjugate of each of these matrices is

$$\gamma^{0\dagger} = \gamma^0, \quad \gamma^{5\dagger} = \gamma^5, \quad \gamma^{\mu\dagger} = \gamma^0\gamma^\mu\gamma^0 = -\gamma \quad \text{for } \mu \neq 0. \quad (3.27)$$

Given that the γ matrices satisfy this algebra, the energy-momentum equation can be factored as

$$p_\mu p^\mu - m^2 = (\gamma^\kappa p_\kappa + m)(\gamma^\lambda p_\lambda - m) = 0. \quad (3.28)$$

Note that the γ matrices need not be exactly the ones specified above, this is just a common physics convention, so long as they satisfy the same conditions the matrices are valid.

The Dirac equation is one of the factors of (3.28) and is conventionally taken to be

$$\gamma^\lambda p_\lambda - m = 0. \quad (3.29)$$

Substituting $p_\mu \rightarrow i\partial_\mu$ gives the covariant form of the Dirac equation.

$$(i\gamma^\mu\partial_\mu - m)\psi = 0 \quad (3.30)$$

where $\partial_\mu = (\frac{\partial}{\partial t}, \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$, m is the particles mass and ψ is a 4-component column matrix known as a *bi-spinor* or *Dirac spinor* (see APPENDIX A.0).

The probability density for the Dirac equation is given by

$$\rho = \psi^\dagger\psi. \quad (3.31)$$

Therefore, probability densities are always positive for the Dirac equation, resolving the issue that motivated its discovery in the first place.

It is the Dirac spinor that leads to the Dirac field. By treating the Dirac equation as a field equation, ψ is promoted to a field. The Dirac spinor is given by

$$\psi = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \end{pmatrix} \quad (3.32)$$

where the ψ s are the components of this field. It is natural to write this four-component field as a pair of two-component fields

$$\psi_L = \begin{pmatrix} \psi_1 \\ \psi_2 \end{pmatrix}, \quad \psi_R = \begin{pmatrix} \psi_3 \\ \psi_4 \end{pmatrix}. \quad (3.33)$$

The subscript L and R stand for left and right. The significance of this notation shall not be expanded upon here but it is related to the notion of helicity.

$$\Rightarrow \psi = \begin{pmatrix} \psi_L \\ \psi_R \end{pmatrix} = \begin{pmatrix} \psi_L \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \psi_R \end{pmatrix} \quad (3.34)$$

here, 0 is a 2-row zero column vector.

The Dirac equation can now be expressed as

$$i \begin{pmatrix} \mathbb{1} & 0 \\ 0 & \mathbb{1} \end{pmatrix} \begin{pmatrix} \partial_0 \psi_L \\ \partial_0 \psi_R \end{pmatrix} + i \begin{pmatrix} -\sigma^\mu & 0 \\ 0 & \sigma^\mu \end{pmatrix} \begin{pmatrix} \partial_\mu \psi_L \\ \partial_\mu \psi_R \end{pmatrix} - m \begin{pmatrix} 0 & \mathbb{1} \\ \mathbb{1} & 0 \end{pmatrix} \begin{pmatrix} \psi_L \\ \psi_R \end{pmatrix} = 0. \quad (3.35)$$

Block multiplying out (3.35) gives two coupled equations

$$\begin{aligned} i\mathbb{1}\partial_0\psi_L - i\sigma^\mu\partial_\mu\psi_L - m\psi_R, \\ i\mathbb{1}\partial_0\psi_R + i\sigma^\mu\partial_\mu\psi_R - m\psi_L. \end{aligned} \quad (3.36)$$

It is convenient to slightly re-define σ^μ :

$$\sigma^\mu = (\mathbb{1}, \sigma^1, \sigma^2, \sigma^3) \quad \text{and} \quad \tilde{\sigma}^\mu = (\mathbb{1}, -\sigma^1, -\sigma^2, -\sigma^3) \quad (3.37)$$

where as before the indices 1,2,3 correspond to Pauli- x, y, z and the index 0 corresponds to the identity matrix.

Equations (3.36) can now be written as

$$\begin{aligned} i\tilde{\sigma}^\mu\partial_\mu\psi_L - m\psi_R &= 0, \\ i\sigma^\mu\partial_\mu\psi_R - m\psi_L &= 0. \end{aligned} \quad (3.38)$$

These are the Dirac field equations.

In order to get the Lagrangian density in terms of ψ_L and ψ_R , multiply out the left-hand side of (3.35) by the row matrix $(\psi_L^\dagger, \psi_R^\dagger)$, where $\psi_L^\dagger = (\psi_1^*, \psi_2^*)$ and $\psi_R^\dagger = (\psi_3^*, \psi_4^*)$.

Block multiplying out gives

$$\mathcal{L} = i\psi_L^\dagger \tilde{\sigma}^\mu \partial_\mu \psi_L + i\psi_R^\dagger \sigma^\mu \partial_\mu \psi_R - m(\psi_L^\dagger \psi_R + \psi_R^\dagger \psi_L). \quad (3.39)$$

Variations $\delta\psi_L^*$ and $\delta\psi_R^*$ in the action give the field equations (3.38).

The Dirac fields may also be further promoted to space and time-dependent operators in a process known as *second quantisation* [26].

Chapter 4

Introduction to Quantum Walks

This chapter concerns the topic of quantum walks on a line and their theoretical background. Quantum walks are analogous to classical random walks but exhibit certain properties that their classical counterparts do not, these shall be discussed later in this section. Recently, quantum walks have experienced a surge interest with the discovery of their potential applications in areas such as quantum algorithms [27–29] and as a model for universal quantum computation [30, 31]. Random walks both quantum and classical, come in two basic kinds, discrete-time and continuous time. For the purposes of this work, only the discrete-time case shall be discussed in detail.

4.1 Discrete-time Random Walk

Firstly, let us consider the case of a classical discrete-time random walk on a line. The setup is as follows: a walker is positioned at the origin on a number line that extends from $-N$ to $+N$, there exists an unbiased, two-sided coin that is flipped, if the coin lands in the state *heads* the walker moves one position to the right ($+1$), if the coin is *tails* the walker moves one position to the left (-1). This process is repeated for a desired number of steps, at any time t , the average position of the walker is the origin. If the walk is run for a sufficient number of iterations, with each time-step being t , and the final position of the walker is plotted - then a binomial distribution is obtained. Figure 4.1 shows the probability distribution for a walker starting at point 0 on a number line that extends from -100 to $+100$, using the paradigm described above.

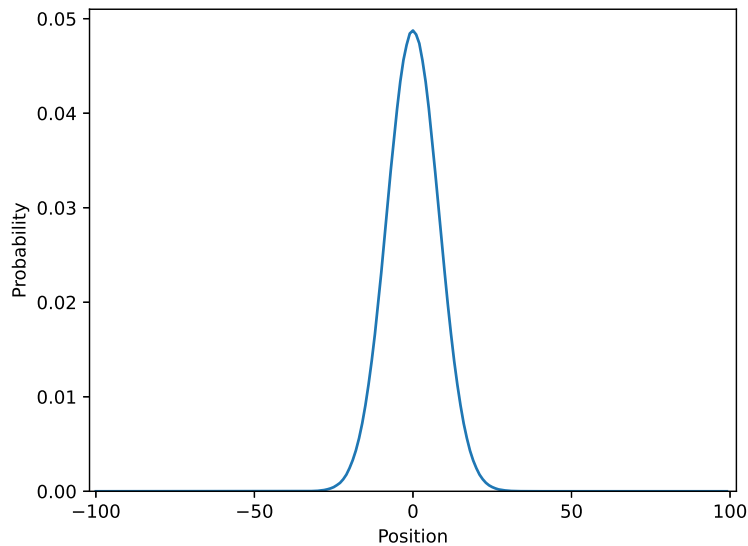


Figure 4.1: Probability distribution for a classical discrete-time random walk of 100 steps.

4.2 Continuous-time Random Walk

A brief mention of classical continuous-time walks, first introduced by Montroll and Weiss in 1965 [32]. The key distinction of a continuous-time walk is that there is a specified probability of the walker moving within a certain time interval. This is known as the hopping rate γ . Consequently, the resulting probability distribution is determined by a probability matrix that contains all the probabilities for each step of the walker. The resulting probability distribution is given by

$$p(t) = \exp[-Mt] p(0) \quad (4.1)$$

where $p(t)$ is the probability distribution (over all sites) at time t , M is the probability matrix and $p(0)$ is the initial probability distribution.

4.3 Discrete-time Quantum Walk

The discrete-time quantum walk (DTQW) is the quantum analogue of the discrete-time random walk and is the version that shall be used to form the basis of this body of work. In order to quantise the discrete-time walk, the coin is promoted to an operator and a step operator is introduced as well. As a corollary, there now also exists a coin state and a position state for the walker. The state of the coin can now be $|0\rangle$ (heads) or $|1\rangle$ (tails), or a superposition of the two. Using these states it is possible to construct the coin operator,

$$\hat{C} = \frac{1}{\sqrt{2}} (|0\rangle \langle 0| + |0\rangle \langle 1| + |1\rangle \langle 0| - |1\rangle \langle 1|). \quad (4.2)$$

This is simply another way of representing a Hadamard gate

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \quad (4.3)$$

Hence this coin is known as a Hadamard coin and is a common choice for the coin operator. However, other coin operators can be chosen and the general case for any coin operator is described by the following matrix,

$$\hat{C}_{general} = \begin{pmatrix} \sqrt{\rho} & e^{i\theta}\sqrt{1-\rho} \\ e^{i\theta}\sqrt{1-\rho} & -e^{i(\theta+\phi)}\sqrt{\rho} \end{pmatrix} \quad (4.4)$$

where ρ is the probability that the walker moves right i.e. $0 \leq \rho \leq 1$, θ and ϕ are arbitrary angles. Taking $\theta = 0$, $\phi = 0$ and $\rho = \frac{1}{2}$ also yields the Hadamard coin.

The step operator acts on the walker (position state) to move it right (+1) if the coin is in the state $|0\rangle$ or left (-1) if it is in the state $|1\rangle$. If the coin is in a superposition then the walker moves both left and right, this operator is also sometimes referred to as the shift operator:

$$\hat{S} = |0\rangle\langle 0| \otimes \sum_m |m+1\rangle\langle m| + |1\rangle\langle 1| \otimes \sum_m |m-1\rangle\langle m|, \quad (4.5)$$

where m is the position of the walker on the line.

The action of the step operator can be summarised as

$$\hat{S} |m, j=0\rangle = |m+1, j=0\rangle, \quad (4.6)$$

$$\hat{S} |m, j=1\rangle = |m-1, j=1\rangle, \quad (4.7)$$

where the second index in the ket is the spin state of the coin (heads or tails) and is given the label j .

It is the action of the coin operator followed by the step operator that creates the DTQW. The combination of these two is sometimes considered as an operator in its own right and is referred to as the walk operator \hat{U} . The walk operator satisfies the condition that it is unitary,

$$\hat{U} = \hat{S} (\hat{C} \otimes I) \quad (4.8)$$

I is the identity matrix corresponding to the number of position states.

The Hilbert space for the discrete-time quantum walk is therefore given by $\mathcal{H} = \mathcal{H}_p \otimes \mathcal{H}_C$. In other words, the tensor product of the position space and the coin space.

Let us consider, as an example, the application of the walk operator 3 times ($\hat{U}^3 = (\hat{S}\hat{C})^3$), on a walker placed at the point 0 on an infinite number line, with the coin initialised in the state $|0\rangle$. The coin operator selected for this particular example is

$$\hat{C}_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix}, \quad (4.9)$$

and its action on the coin state can be summarised as

$$\hat{C}_2 |j = 0\rangle = \frac{1}{\sqrt{2}} (|1\rangle - |0\rangle), \quad (4.10)$$

$$\hat{C}_2 |j = 1\rangle = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle). \quad (4.11)$$

The initial state of the walker is

$$|\psi\rangle_{walker} = |m = 0, j = 0\rangle. \quad (4.12)$$

The first 3 steps are

$$\left(\hat{S}\hat{C}_2\right)^3 |0, 0\rangle = \left(\hat{S}\hat{C}_2\right)^2 \hat{S}\hat{C}_2 |0, 0\rangle \quad (4.13)$$

$$= \left(\hat{S}\hat{C}_2\right)^2 \hat{S} \frac{1}{\sqrt{2}} (-|0, 0\rangle + |0, 1\rangle)$$

$$= \left(\hat{S}\hat{C}_2\right)^2 \frac{1}{\sqrt{2}} (-|-1, 0\rangle + |1, 1\rangle) \quad (\text{1st step}). \quad (4.14)$$

$$= \left(\hat{S}\hat{C}_2\right) \hat{S}\hat{C}_2 \frac{1}{\sqrt{2}} (-|-1, 0\rangle + |1, 1\rangle)$$

$$= \left(\hat{S}\hat{C}_2\right) \hat{S} \frac{1}{2} (-|-1, 0\rangle - |-1, 1\rangle + |1, 0\rangle + |1, 1\rangle)$$

$$= \left(\hat{S}\hat{C}_2\right) \frac{1}{2} (|-2, 0\rangle - |0, 1\rangle + |0, 0\rangle + |2, 1\rangle) \quad (\text{2nd step}). \quad (4.15)$$

$$= \left(\hat{S}\right) \hat{C}_2 \frac{1}{2} (|-2, 0\rangle - |0, 1\rangle + |0, 0\rangle + |2, 1\rangle)$$

$$= \left(\hat{S}\right) \frac{1}{2\sqrt{2}} (-|-2, 0\rangle + |-2, 1\rangle - |0, 0\rangle - |0, 1\rangle - |0, 0\rangle + |0, 1\rangle + |2, 0\rangle + |2, 1\rangle)$$

$$= \frac{1}{\sqrt{8}} (-|-3, 0\rangle + |-1, 1\rangle - |-1, 0\rangle - |1, 1\rangle - |-1, 0\rangle + |1, 1\rangle + |1, 0\rangle + |3, 1\rangle)$$

$$|\psi\rangle_{walker} = \frac{1}{\sqrt{8}} (-|-3, 0\rangle + |-1, 1\rangle - 2|-1, 0\rangle + |1, 0\rangle + |3, 1\rangle) \quad (\text{3rd step}). \quad (4.16)$$

Figure 4.2 shows the position of the walker on a portion of the infinite number line for this quantum walk.

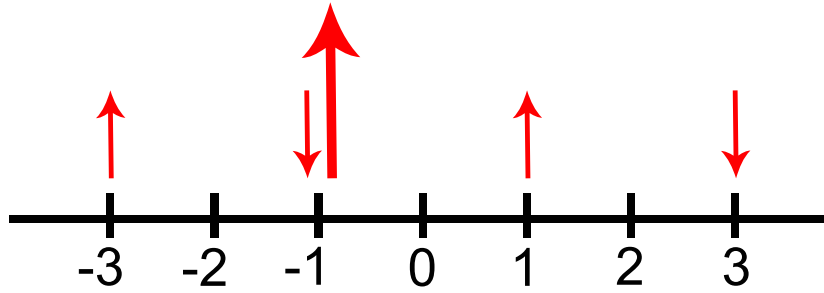


Figure 4.2: Position of the walker on a portion of the number line after 3 steps. Red arrows indicate the state of the coin, up arrow means $|0\rangle$, down arrow means $|1\rangle$. Size of the arrow corresponds to the amplitude in the wave function.

Figure 4.2 displays some key and interesting properties of discrete-time quantum walks. First, and most obvious, is the superposition of the walker at multiple sites on the line. After 3 steps, the walker is in 4 positions. The second interesting property to note is the superposition of the coin at one site. At site -1, the coin is in a superposition of both $|0\rangle$ and $|1\rangle$, something that is not achievable classically. Also note that this superposition is not equal in probability, in fact it is twice as likely that the coin will be in the state $|0\rangle$ than $|1\rangle$. The final state of the walker can be determined by performing a measurement, in line with the rules of quantum mechanics, causing the wavefunction to collapse and giving the site and coin state of the walker. However, it is interesting to note that it is only when this measurement is performed that the walk ends, and in the limit that a measurement is taken after each step of the walk, then a classical discrete-time walk is obtained that follows the same probability statistics as Figure 4.1. It is these properties that make discrete-time quantum walks potential good candidates for use in quantum algorithms and quantum computing in general.

The DTQW analogue of Figure 4.1 is shown in Figure 4.3 , which shows the probability distribution for a DTQW for N in the range -100 to 100. The walker begins at point 0 on the line and the coin is a Hadamard coin that is initiated in a superposition of $|0\rangle$ and $|1\rangle$.

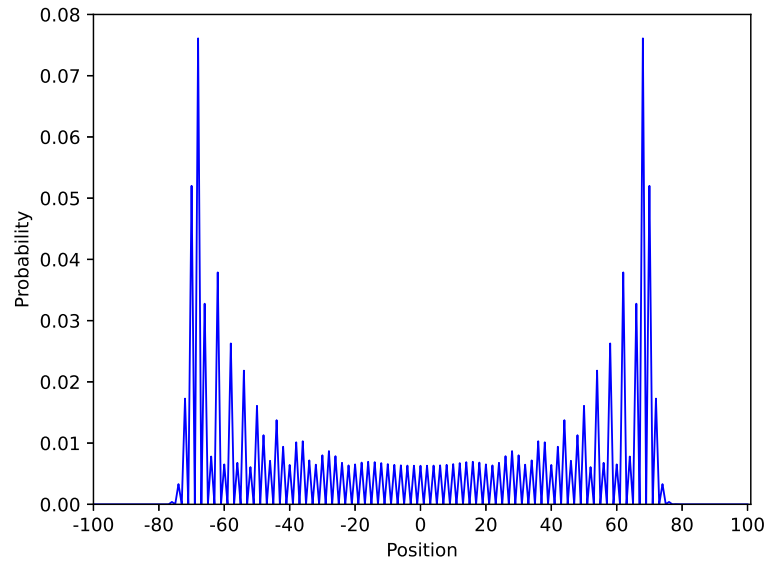


Figure 4.3: Probability distribution for a DTQW of 100 steps. Initial state of the coin is $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.

As can be seen in Figure 4.3, the probability statistics for the DTQW are vastly different to the classical discrete-time case of Figure 4.1. The shape of this distribution is a result of the aforementioned properties of discrete-time quantum walks. The oscillation of the probabilities between sites occurs as a result of the wavefunction of the walker destructively interfering with itself (see Eqns. 4.13-4.16). Furthermore, in the classical walk the highest probability of the walker being found is at the origin whereas in the quantum walk it is almost the opposite, the walker has the greatest probability of being found at the edges of the number line.

The initial conditions of the quantum walk also greatly affect the evolution and resulting probability distributions. Figure 4.4 shows the effect of varying the initial state of the coin. Figure 4.4 also demonstrates the reason for preparing the coin in a superposition of states - so as to avoid biasing the walk.

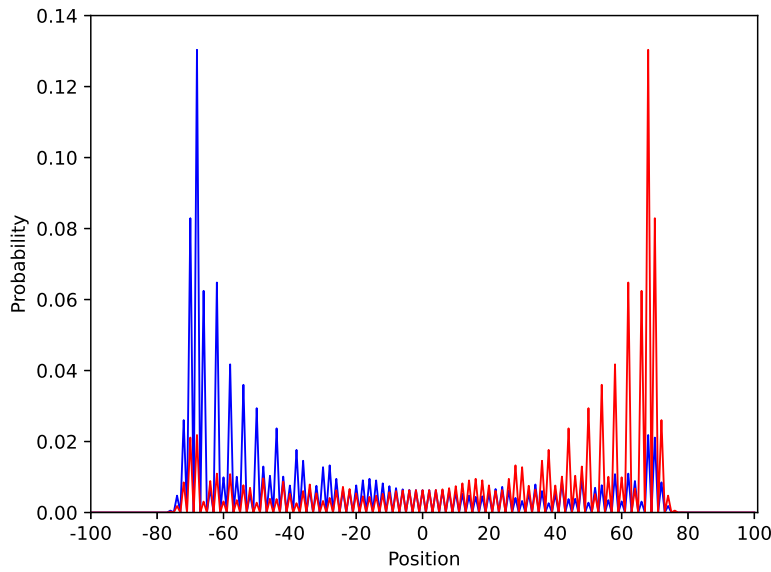


Figure 4.4: Probability distribution for a DTQW of 100 steps. Coin initialised as $|1\rangle$ (blue) and coin initialised as $|0\rangle$ (red).

4.4 Continuous-time Quantum Walk

A brief mention of continuous-time quantum walks (CTQW), which are the quantum analogues of the classical continuous-time walks. Similarly to their classical counterparts, CTQWs evolve as a result of a time-varying unitary matrix that comes from the Hamiltonian of the system. It is believed the earliest consideration of CTQWs was by Farhi and Gutmann in 1998 [33].

4.5 Discrete-time Quantum Walk on a Cycle Graph

Finally, to conclude this chapter, we consider a simple variation of the DTQW on a line, that is a DTQW on a cycle graph. Simply put, this involves taking the ends of the number line and attaching them to each other so the line now forms a cycle. This way once the walker reaches (what was formerly) the end of the line, the walk loops over to what was formerly the other end of the line.

Chapter 5

Feynman's Checkerboard and the Dirac Equation

Feynman was the first to establish a connection between random walks and the Dirac equation, via path integrals, in his 1965 collation with Hibbs titled: “Quantum Mechanics and Path Integrals”. This subtle connection appears as a problem (2-6) on page 34-36 of the text [34]. In this chapter, we explore this connection and present a solution, based-on the work of Jacobson and Schulman [35], which has come to be known as Feynman's Checkerboard or Feynman's Chessboard.

A path integral is a sum over all possible paths of a particle from (x_a, t_a) to (x_b, t_b) . From a path integral it is possible to derive a propagator for the particle (see APPENDIX A.1). For example, for a non-relativistic, free particle in one dimension, the propagator is [34]

$$K^{(NR)}(x_b, t_b; x_a, t_a) = \lim_{N \rightarrow \infty} \left(\frac{-im}{2\pi\hbar\epsilon} \right)^{\frac{1}{2}(N-1)} \int dx_1 \dots dx_N \exp \left[\frac{im \sum (x_{j+1} - x_j)^2}{2\epsilon} \right]. \quad (5.1)$$

Explicitly, this evaluates to

$$K^{(NR)}(x_b, t_b; x_a, t_a) = \sqrt{\frac{m}{2\pi i\hbar(t_b - t_a)}} \exp \left[\frac{im(x_b - x_a)^2}{2\hbar(t_b - t_a)} \right]. \quad (5.2)$$

In more dimensions,

$$K(\vec{x}_b, t_b; \vec{x}_a, t_a) = \langle \vec{x}_b | e^{-\frac{iH(t_b - t_a)}{\hbar}} | \vec{x}_a \rangle \quad (5.3)$$

where H is the Hamiltonian, and

$$\psi(\vec{x}_b, t_b) = \int d^3x_a K(\vec{x}_b, t_b; \vec{x}_a, t_a) \psi(\vec{x}_a, t_a). \quad (5.4)$$

5.1 Checkerboard Path Integral

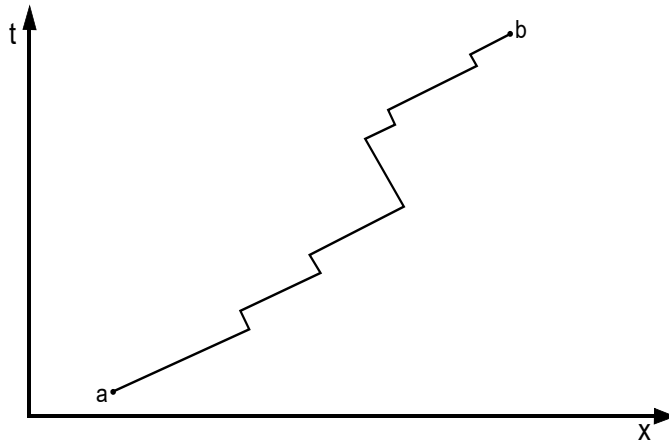


Figure 5.1: Illustration of the path of a relativistic particle in (1+1)-d, where x is the space dimension and t is the time dimension.

Figure 5.1 shows a relativistic random walk for a particle in 1+1 dimensions. The gradient of the segments is constant in magnitude and differs only in sign (direction). Firstly, units are set to $\hbar = c = 1$. The propagator for this particle is

$$K_{\beta\alpha}(x_b, t_b; x_a, t_a) = \lim_{N \rightarrow \infty} \sum_{R \geq 0} \phi_{\beta\alpha}(R) (i\epsilon m)^R. \quad (5.5)$$

N is the number of steps.

$\epsilon = \frac{t_b - t_a}{N}$, i.e. the length of each step.

m is the mass of the particle.

α, β take values of 'left' or 'right'.

$\phi_{\beta\alpha}(R)$ is the number of paths with exactly N steps that start at x_a and in direction α , and end at x_b in the direction β , that switch direction R times.

An important point, the relativistic particle in this problem is both massive and moves at speed c . This is required for the particle to switch directions and for the space and time steps to scale the same with N . While seeming to be in violation of special relativity, the propagator is only evaluated for net subluminal motion i.e.

$$|x_b - x_a| < c(t_b - t_a). \quad (5.6)$$

Consider a path of a particle with R bends, which leaves moving right and arrives moving left and contributes to the sum of the associated propagator, K_{-+} . The particle makes $1 + \frac{(R-1)}{2}$ turns to the left and $\frac{(R-1)}{2}$ turns to the right, where the last turn is to the left and R is odd for K_{-+} . P is defined as the number of steps to the right and Q is defined as the number of steps to the left. It follows that $P + Q = N$.

For K_{-+} , the total number of paths from x_a to x_b is

$$\phi_{-+}(R) = \binom{P-1}{\frac{1}{2}(R-1)} \binom{Q-1}{\frac{1}{2}(R-1)}. \quad (5.7)$$

In the limit $N \rightarrow \infty$,

$$\lim_{N \rightarrow \infty} \phi_{-+}(R) = \frac{(PQ)^{\frac{(R-1)}{2}}}{[(\frac{1}{2}(R-1))!]^2}. \quad (5.8)$$

This means

$$K_{-+}(x_b, t_b; x_a, t_a) = \sum_{R \geq 0, R \text{ odd}} \frac{(PQ)^{\frac{(R-1)}{2}}}{[(\frac{1}{2}(R-1))!]^2} (i\epsilon m)^R. \quad (5.9)$$

Defining $M = (P - Q)$,

$$PQ = \frac{1}{4}(N + M)(N - M) = \left(\frac{N}{2\gamma}\right)^2 \quad (5.10)$$

where $\gamma = (1 - v^2)^{-\frac{1}{2}}$, $v^2 = \frac{M^2}{N^2} = (x_b - x_a)^2 / (t_b - t_a)^2$ and using the definition of ϵ from (5.5), equation (5.9) becomes

$$K_{-+}(x_b, t_b; x_a, t_a) = \frac{2\gamma}{N} \sum_{R \geq 0, R \text{ odd}} \frac{\left(\frac{im(t_b - t_a)}{2\gamma}\right)^R}{[(\frac{1}{2}(R-1))!]^2}. \quad (5.11)$$

Let $z = m\left(\frac{t_b - t_a}{\gamma}\right)$ and replace N with $\frac{(t_b - t_a)}{\epsilon}$ and let $R = 2k + 1$. Equation (5.11) becomes

$$K_{-+}(x_b, t_b; x_a, t_a) = i\epsilon m \sum_{k=0}^{\infty} (-1)^k \frac{(z/2)^{2k}}{(k!)^2} = i\epsilon m J_0(z) \quad (5.12)$$

where J_0 is the zeroth Bessel function of the first kind [36]. Following [35], the next step is to divide by 2ϵ to get the continuum limit as $\epsilon \rightarrow 0$, since K_{-+} vanishes at every other lattice point. Noting that

$$z = m\left(\frac{t_b - t_a}{\gamma}\right) = m(t_b - t_a)\sqrt{1 - v^2} = m\sqrt{(t_b - t_a)^2 - (x_b - x_a)^2} \equiv m\tau. \quad (5.13)$$

Finally, this gives the exact continuum propagator as

$$K_{-+}(x_b, t_b; x_a, t_a) = i\frac{m}{2} J_0(m\tau). \quad (5.14)$$

[35] evaluates the other components, ϕ_{++} and ϕ_{--} as

$$\phi_{++}(R) = \binom{P-2}{\frac{1}{2}R} \binom{Q-1}{\frac{1}{2}(R-1)} \quad (5.15)$$

and

$$\phi_{--}(R) = \binom{P-1}{\frac{1}{2}(R-1)} \binom{Q-2}{\frac{1}{2}R} \quad (5.16)$$

with the two related by

$$\phi_{--} = \phi_{++}(P \leftrightarrow Q). \quad (5.17)$$

Jacobson [37] obtains all four components of the continuum propagator, by letting $x_b = x$, $x_a = 0$, $t_b = t$ and $t_a = 0$

$$K(x, t; 0, 0) = \frac{im}{2} \begin{pmatrix} \frac{i(t+x)}{\tau} J_1(m\tau) & J_0(m\tau) \\ J_0(m\tau) & \frac{i(t-x)}{\tau} J_1(m\tau) \end{pmatrix} \quad (5.18)$$

where τ becomes $\tau = \sqrt{t^2 - x^2}$ with $|x| < t$.

5.2 Connection to the Dirac Equation

The connection between these results and the one-dimensional Dirac equation, which is

$$i\partial_t \Psi = -i\sigma_z \partial_x \Psi - m\sigma_x \Psi, \quad (5.19)$$

is not immediately obvious. Here σ_z and σ_x are the 2×2 Pauli- z and Pauli- x matrices respectively, Ψ is a two-component Dirac spinor. Following the work of Jacobson and Schulman, Kull and Treumann [38] presented a more direct interpretation of (5.18) that shall be discussed first.

Choosing two Dirac spinors given by

$$\Psi_1 = \begin{pmatrix} K_{++} \\ K_{+-} \end{pmatrix}, \quad \Psi_2 = \begin{pmatrix} K_{-+} \\ K_{--} \end{pmatrix} \quad (5.20)$$

where by using (5.17), $K_{+-} = K_{-+}$ and using (5.18), Ψ_1 and Ψ_2 are two independent, exact solutions of (5.19). Thus [38] establishes that Feynman's Checkerboard yields solutions to the one-dimensional Dirac equation and by extension, relates random walks to the Dirac equation.

Jacobson and Schulman [35] provide a more convoluted way of relating the two, connecting the continuum propagator with the partition function of the one-dimensional Ising model. Then establishing a connection between that, and the Hamiltonian of the one-dimensional Dirac equation. The details shall not be discussed here but it is asserted that the continuum propagator for the one-dimensional Ising model is

$$K_{\beta\alpha}(x, t; 0, 0) = \frac{\epsilon}{2\pi} \int dp e^{ipx} \left(e^{it(m\sigma_x - p\sigma_z)} \right)_{\beta\alpha} \quad (5.21)$$

which is the Fourier transform of

$$e^{iHt} \quad (5.22)$$

where H is the Hamiltonian of (5.19). Thus [35] establishes the link between Feynman's Checkerboard, the one-dimensional Ising model and the one-dimensional Dirac equation.

Chapter 6

Quantum Walks and Quantum Field Theory

Following the establishment of the connection between random walks and the Dirac equation, Konno, using a similar method to Jacobson and Schulman, analytically solved the quantum walk on a line [39,40]. This is followed by a series of papers, by various authors, dealing with the continuous limit of quantum walks [41–43].

This chapter focuses on the work of DiMolfetta, who shows that in the continuous limit, a discrete quantum walk yields a Klein-Gordon-like or Dirac-like equation [44–48]. In all cases, the general procedure to retrieve the Dirac-(or Klein-Gordon)-like equation is similar. Starting with a discrete-time quantum walk written as a pair of difference equations, a discrete space and discrete time step is introduced. The continuous limit of these steps is then found by taking them to zero, and imposing certain conditions on the walk, from which the relevant equation can be found. For our purposes, let us consider the simplest case that DiMolfetta examines [49], the relevant parts of which are presented here.

Consider a quantum walk defined over discrete-time and discrete one-dimensional space, using time and space dependent coins, acting on a two-dimensional Hilbert space. The walk is defined by the following finite difference equations, valid for all $(j, m) \in \mathbb{N} \times \mathbb{Z}$.

$$\begin{bmatrix} \psi_{j+1,m}^- \\ \psi_{j+1,m}^+ \end{bmatrix} = B(\theta_{j,m}, \xi_{j,m}, \zeta_{j,m}) \begin{bmatrix} \psi_{j,m+1}^- \\ \psi_{j,m-1}^+ \end{bmatrix} \quad (6.1)$$

where the j index represents the time dimension and the m index represents space.

$$B(\theta_{j,m}, \xi_{j,m}, \zeta_{j,m}) = \begin{bmatrix} e^{i\xi} \cos \theta & e^{i\zeta} \sin \theta \\ -e^{-i\zeta} \sin \theta & e^{-i\xi} \cos \theta \end{bmatrix}, \quad (6.2)$$

is an $SU(2)$ operator and (θ, ξ, ζ) are the three Euler angles.

ψ^-, ψ^+ are two components of the wavefunction Ψ and are the probability amplitudes of the particle moving in the decreasing m or increasing m directions, respectively.

The set of Euler angles define the walk and are, at this stage, arbitrary.

The total probability is defined by

$$\pi_j = \sum_m \left(|\psi_{j,m}^-|^2 + |\psi_{j,m}^+|^2 \right). \quad (6.3)$$

To take the continuous limit of this walk, let us introduce time Δt and space Δx steps.

Let us also define, for any quantity a appearing in (6.1), a function \tilde{a} , such that the number $a_{j,m}$ is the value taken by \tilde{a} at the space-time point $(t_j = j\Delta t, x_m = m\Delta x)$.

Equation (6.1) then becomes

$$\begin{bmatrix} \psi^-(t_j + \Delta t, x_m) \\ \psi^+(t_j + \Delta t, x_m) \end{bmatrix} = B(\theta(t_j, x_m), \xi(t_j, x_m), \zeta(t_j, x_m)) \begin{bmatrix} \psi^-(t_j, x_m + \Delta x) \\ \psi^+(t_j, x_m - \Delta x) \end{bmatrix} \quad (6.4)$$

where the tildes have been dropped on all functions to simplify the notation.

Suppose that all functions can be chosen at least C^2 in both space and time variables for all sufficiently small values of Δt and Δx . The formal continuous limit is defined as the couple of differential equations obtained from (6.4), by letting both Δt and Δx tend to zero.

Introducing a time-scale τ , a length-scale λ and an infinitesimal ϵ ,

$$\begin{aligned} \Delta t &= \tau\epsilon, \\ \Delta x &= \lambda\epsilon^\delta \end{aligned} \quad (6.5)$$

where $\delta > 0$ allows for Δt and Δx to tend to zero differently.

For the continuous limit to exist, the operator $B(\theta, \xi, \zeta)$ must also tend to unity as ϵ tends to zero. This is because the two column vectors on the left-hand side and right-hand side of (6.4) both tend to $\Psi(t_j, x_m)$ when Δt and Δx tend to zero. This implies that the two functions θ and ξ must depend on ϵ and tend to zero or π as ϵ tends to zero.

Therefore, it generally does not make sense to consider the continuous limit of a given walk, defined by ϵ -independent angles, but rather the limit of a family of walks indexed by ϵ , whose defining angles depend on ϵ and tend to zero with this infinitesimal. This family of quantum walks is known as a jet, and in this particular case a 1-jet. We therefore write,

$$\begin{aligned} \theta(t, x) &= p\pi + \bar{\theta}(t, x)\epsilon^\alpha \\ \xi(t, x) &= p\pi + \bar{\xi}(t, x)\epsilon^\beta, \end{aligned} \quad (6.6)$$

where $p = 0$ or 1 and $\alpha, \beta > 0$.

The continuous limit can be investigated by Taylor expanding $\psi^\pm(t, x \mp \Delta x)$, $\psi^\pm(t \pm \Delta t, x)$, $\cos\theta$, $\sin\theta$ and $e^{i\xi}$ in powers of ϵ . The possible scaling obeyed by the continuous limit are found by examining the lowest order contributions.

$$\psi^\pm(t \pm \Delta t, x) = \psi^\pm(t, x) + \mathcal{O}(\epsilon), \quad (6.7)$$

$$\psi^\pm(t, x \mp \Delta x) = \psi^\pm(t, x) + \mathcal{O}(\epsilon^\delta), \quad (6.8)$$

$$e^{i\xi} \cos \theta = 1 + \mathcal{O}(\epsilon^\beta) + \mathcal{O}(\epsilon^{2\alpha}), \quad (6.9)$$

and

$$e^{i\zeta} \sin \theta = \mathcal{O}(\epsilon^\alpha). \quad (6.10)$$

Equation (6.4) then becomes

$$\begin{bmatrix} \psi^-(t_j, x_m) \\ \psi^+(t_j, x_m) \end{bmatrix} + \mathcal{O}(\epsilon) = \begin{bmatrix} \psi^-(t_j, x_m) \\ \psi^+(t_j, x_m) \end{bmatrix} + \mathcal{O}(\epsilon^\alpha) + \mathcal{O}(\epsilon^\beta) + \mathcal{O}(\epsilon^\delta). \quad (6.11)$$

The most interesting scaling corresponds to when $\alpha = \beta = \delta = 1$, because all contributions to (6.11) are of equal importance. This is the scaling that shall be explored next.

We shall now find the equations of motion for ψ^- and ψ^+ . Defining the null coordinates, u^- and u^+ in terms of t and x :

$$\begin{aligned} u^- &= \frac{1}{2} \left(\frac{t}{\tau} - \frac{x}{\lambda} \right) \\ u^+ &= \frac{1}{2} \left(\frac{t}{\tau} + \frac{x}{\lambda} \right). \end{aligned} \quad (6.12)$$

Partial derivatives with respect to these null coordinates are

$$\begin{aligned} \partial_- &= \partial_{u^-} = \tau \partial_t - \lambda \partial_x \\ \partial_+ &= \partial_{u^+} = \tau \partial_t + \lambda \partial_x, \end{aligned} \quad (6.13)$$

and

$$\partial_- \partial_+ = \tau^2 \partial_{tt} - \lambda^2 \partial_{xx} = \lambda^2 \square, \quad (6.14)$$

where \square is the d'Alembertian,

$$\square = \frac{1}{c^2} \partial_{tt} - \partial_{xx}, \quad (6.15)$$

with $c = \lambda/\tau$. The expansion of the discrete equations around $\epsilon = 0$ then leads to the following equations of motion for ψ^- and ψ^+ ,

$$\partial_- \psi^- = \left(i\bar{\xi} \psi^- - (-1)^{p+1} \bar{\theta} e^{+i\zeta} \psi^+ \right), \quad (6.16)$$

$$\partial_+ \psi^+ = \left(i\bar{\xi} \psi^+ - (-1)^{p+1} \bar{\theta} e^{-i\zeta} \psi^- \right). \quad (6.17)$$

These two equations are sometimes known as the Dirac form of the continuous quantum walk dynamics.

These coupled first-order equations are best transcribed in operator form, i.e. $\hat{D}\Psi = 0$, with the operator \hat{D} acting on the two-component wave function Ψ given by

$$\hat{D} = \Gamma^- \partial_- + \Gamma^+ \partial_+ - i\sigma^3 \bar{\xi} + i\sigma^2 (-1)^{p+1} \bar{\theta} \cos \zeta + i\sigma^1 (-1)^{p+1} \bar{\theta} \sin \zeta, \quad (6.18)$$

where $\sigma^{1,2,3}$ are the Pauli- x, y, z matrices respectively, and

$$\Gamma^\mp = \frac{1}{2}(\mathbb{1} \pm \sigma^3). \quad (6.19)$$

The operator \hat{D} defines the formal continuous limit of the quantum walk. These first-order equations imply that each component of Ψ obeys, at points where θ does not vanish, an uncoupled Klein-Gordon equation. For example, the equation obeyed by ψ^- can be obtained using (6.16) to express ψ^+ in terms of ψ^- and $\partial_- \psi^-$, and by then replacing ψ^+ by this expression in (6.17). This leads to

$$\lambda^2 \square \psi^- = [\partial_+ (\ln \bar{\theta} + i\zeta) - i\xi] \partial_- \psi^- + i\bar{\xi} \partial_+ \psi^- - [\bar{\theta}^2 + \bar{\xi}^2 + i\bar{\xi} \partial_+ (\ln \bar{\theta} + i\zeta) - i\partial_+ \xi] \psi^-, \quad (6.20)$$

$$\lambda^2 \square \psi^+ = -i\bar{\xi} \partial_- \psi^+ + [\partial_- (\ln \bar{\theta} - i\zeta) + i\xi] \partial_+ \psi^+ - [\bar{\theta}^2 + \bar{\xi}^2 + i\bar{\xi} \partial_- (-\ln \bar{\theta} + i\zeta) + i\partial_- \xi] \psi^+. \quad (6.21)$$

These equations resemble Klein-Gordon equations but contain terms that violate time-reversibility.

Let us now examine the connection with the Dirac field. The Lagrangian density is defined by

$$\mathcal{L}[\Psi, \Psi^\dagger] = \Psi^\dagger \hat{D} \Psi, \quad (6.22)$$

and leads to the first-order equations of motion (6.16) and (6.17).

Another form of the Lagrangian density is

$$\mathcal{L}[\Psi, \Psi^\dagger] = \Psi^* \Gamma^\mu \nabla_\mu \Psi \quad (6.23)$$

where $\mu \in \{+, -\}$ and

$$\nabla_\mp = \partial_\mp - i\sigma^3 \bar{\xi} + i\sigma^2 (-1)^{p+1} \bar{\theta} \cos \zeta + i\sigma^1 (-1)^{p+1} \bar{\theta} \sin \zeta. \quad (6.24)$$

This can be derived from (6.22), noting that $\Gamma^- + \Gamma^+ = 1$.

Defining

$$\gamma_\mu = -\sigma^3 \Gamma_\mu, \quad (6.25)$$

and

$$D_\mu = \partial_\mu + i\sigma^1 (-1)^{p+1} \bar{\theta} \sin \zeta + i\sigma^2 (-1)^{p+1} \bar{\theta} \cos \zeta. \quad (6.26)$$

The Lagrangian density can then be written as

$$\mathcal{L}[\Psi, \Psi^\dagger] = \Psi^\dagger (i\gamma^\mu D_\mu - \bar{\xi}) \Psi. \quad (6.27)$$

This looks like the Lagrangian density for a Dirac field. However, in this case, Ψ belongs to a two-dimensional Hilbert space and is not a Dirac spinor.

Thus, the connection between discrete-time quantum walks and the Dirac equation/Dirac field, has been established. So, as a first step to simulating a quantum field theory on a quantum computer, this single particle Dirac equation shall be considered, and a discrete-time quantum walk shall be simulated.

Chapter 7

The Transmon

Before we consider the execution of quantum walks on a digital quantum computer, it is worth taking a brief detour to discuss the basic construction of its most fundamental building block - the qubit. Over the years a variety of different methods for implementing qubits have been developed [50]. For the case of a superconducting quantum computer, the most well-known is the transmon. This section shall explore the basic construction of a transmon as well as its associated mode of operation.

The transmon, *Transmission-line Shunted Plasma Oscillation qubit*, is a type of superconducting qubit used in IBM quantum computers. At the simplest level, a transmon can be deconstructed into four components - two islands connected to each other via a Josephson junction and some form of capacitance. A Josephson junction is a non-linear circuit element made of two superconducting layers separated by a non/weakly superconducting layer. The two islands represent the two levels ($|0\rangle$ and $|1\rangle$) of the qubit, as well as other physical components [51]. Figure 7.1 depicts this basic construction.

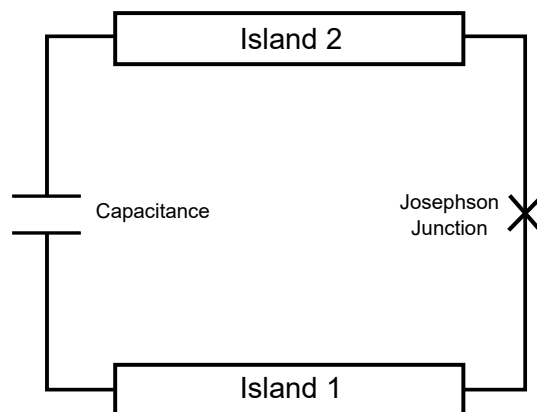


Figure 7.1: Basic construction of the transmon.

The original design of the transmon was similar to the Cooper pair box. In essence, a superconducting qubit may be categorised into three basic kinds: charge, phase and flux [52]. The contemporary transmon is any combination of these. At a fundamental level, the transmon functions as follows: Cooper pairs tunnel between islands (states) by excitation from microwave pulses - typically of the order 4-8 GHz and roughly 20ns long. Therefore, it is possible to prepare states and execute single qubit gates using only these pulses. In order to execute two or multiple-qubit gates the transmons must be connected via a transmission-line resonator. Specifically, there are two kinds of transmission-line resonators, a bus resonator is used to connect multiple transmons to each other in order to implement multiple-qubit gates via a current flowing between them. A read-out resonator is used to perform measurements. This is achieved by inputting a very weak light pulse into the read-out resonator, of the order of femto watts. Depending on what state the transmon is in, $|0\rangle$ or $|1\rangle$, there is a shift in the resonant frequency of the read-out resonator. This shift can then be detected as the light pulse is reflected back out of the resonator and thus measurement of the state is possible. The subject of circuit Quantum Electrodynamics (circuit QED) deals with the construction of quantum processors and ways in which superconducting qubits may be connected. The interested reader may wish to consult [53], as this shall not be discussed further.

Notice the basic design of a transmon is similar to a quantised LC oscillator, shown in Figure 7.2.

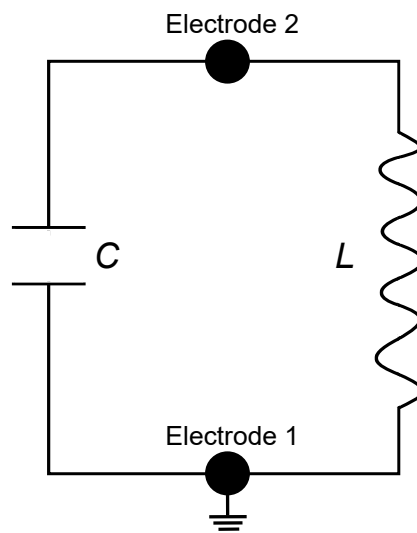


Figure 7.2: Diagram of a quantised LC circuit. L is the inductor and C is the capacitor.

The Hamiltonian of this system is given by

$$\hat{H}_{LC} = \frac{\hat{Q}^2}{2C} + \frac{\hat{\phi}^2}{2L}, \quad (7.1)$$

where \hat{Q} is the charge on the capacitor and $\hat{\phi}$ is the flux through the inductor. Note that these quantities do not commute

$$[\hat{\phi}, \hat{Q}] = i\hbar. \quad (7.2)$$

The quantised LC oscillator itself corresponds directly to the quantum simple harmonic oscillator (SHO):

$$\hat{H}_{SHO} = \frac{\hat{P}^2}{2m} + \frac{k\hat{X}^2}{2} \quad (7.3)$$

with k being the spring constant, m being the mass and \hat{P} , \hat{X} being the momentum and position operators respectively. Note that

$$[\hat{X}, \hat{P}] = i\hbar. \quad (7.4)$$

The exact correspondence is

$$\hat{\phi} \longleftrightarrow \hat{X}$$

$$\hat{Q} \longleftrightarrow \hat{P}$$

$$L \longleftrightarrow \frac{1}{k}$$

$$C \longleftrightarrow m.$$

This system may be solved using ladder operators:

$$\hat{a} = \left(\frac{\hat{Q}}{Q_{zpf}} - i \frac{\hat{\phi}}{\phi_{zpf}} \right) \quad (7.5)$$

$$\hat{a}^\dagger = \left(\frac{\hat{Q}}{Q_{zpf}} + i \frac{\hat{\phi}}{\phi_{zpf}} \right) \quad (7.6)$$

with $\phi_{zpf} = \sqrt{2\hbar Z}$, $Q_{zpf} = \sqrt{\frac{2\hbar}{Z}}$ and $Z = \omega L = \frac{1}{\omega C} = \sqrt{\frac{L}{C}}$.

This gives the Hamiltonian as

$$\hat{H}_{LC} = \hbar\omega \left(\hat{a}^\dagger \hat{a} + \frac{1}{2} \right) \quad (7.7)$$

with

$$[\hat{a}, \hat{a}^\dagger] = 1. \quad (7.8)$$

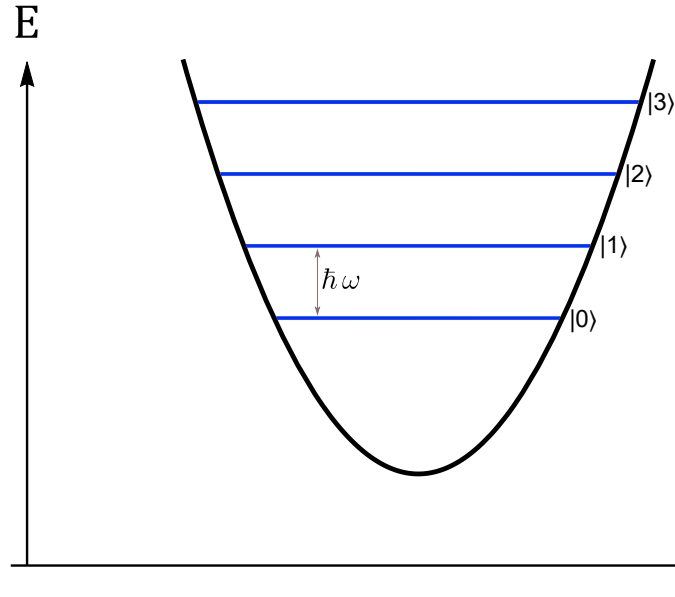


Figure 7.3: Energy diagram of a quantum simple harmonic oscillator. Blue lines indicate energy levels with separation $\hbar\omega$.

However, this does not make a very good qubit, since the energy levels are equally spaced this allows for leakage out of the qubit-subspace. Figure 7.3 shows the energy diagram for this system. This brings us to the role of the Josephson junction, to introduce anharmonicity into the system and therefore create unequal spacing between energy levels. Thus reducing the chances accessing of other energy levels (states). Since we are designing a quantum *bit* we only require two energy levels, typically chosen to be the ground and first excited state.

The current through a Josephson junction is given by

$$I = I_c \sin\left(2\pi \frac{\phi}{\phi_0}\right), \quad (7.9)$$

where

$$I_c = \frac{\pi \Delta}{2e R}, \quad (7.10)$$

and ϕ_0 is the quantum flux $\frac{h}{2e}$.

The energy stored in the junction is

$$E_{stored} = E_J \left(1 - \cos\left(2\pi \frac{\phi}{\phi_0}\right)\right), \quad (7.11)$$

E_J is the Josephson energy,

$$E_J = \frac{I_c \phi_0}{2\pi}. \quad (7.12)$$

Figure 7.4 shows the energy level diagram for the Josephson potential as well as the quantum simple harmonic oscillator.

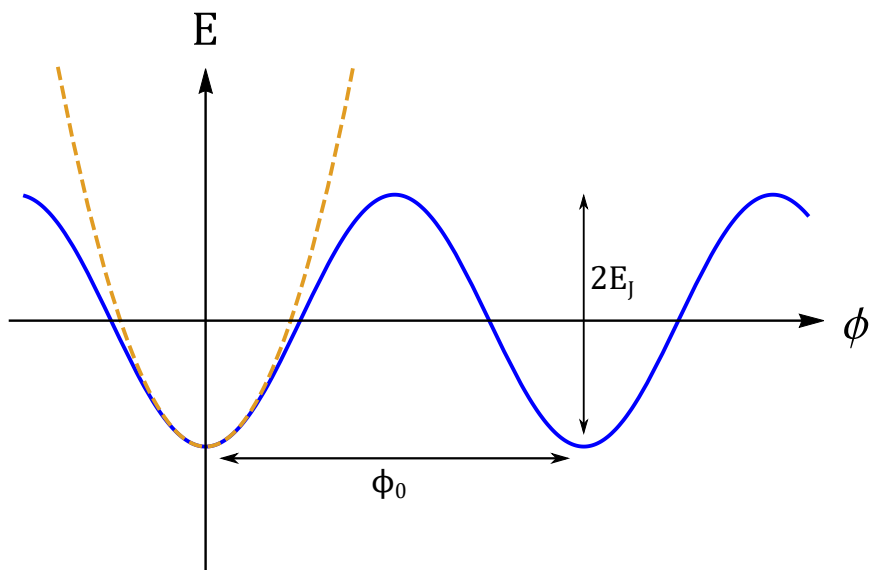


Figure 7.4: Energy diagram of the Josephson potential (solid blue) compared to the harmonic oscillator (dashed orange). Wavelength of the potential is given by the quantum flux and the amplitude by twice the Josephson energy.

For small flux:

$$E_J \approx \frac{\phi^2}{2L_J} + \text{const.}$$

$$L_J = \frac{\phi_0}{2\pi I_c}.$$

Thus the transmon energy spectrum becomes anharmonic as desired, shown in Figure 7.5.

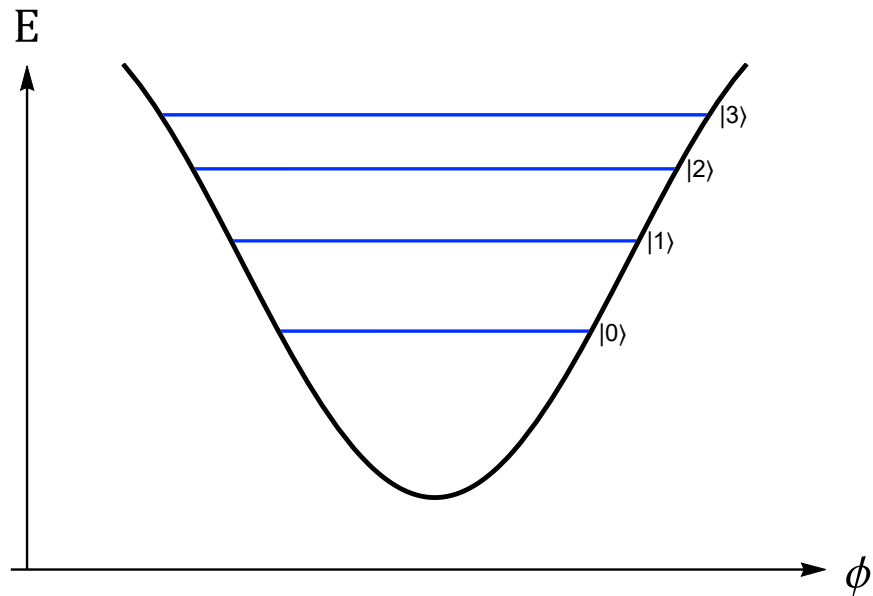


Figure 7.5: Energy spectrum of the transmon. Energy levels are no longer equally spaced apart.

Due to this unequal spacing, microwave pulses can be tuned to exact transitions. Therefore, in principle, confining the transmon to two energy levels.

To conclude, let us consider the characteristic parameters associated with each individual transmon. There are many such terms, however the two that shall be discussed here are the T_1 and T_2 coherence times [54]. Terms that are borrowed from NMR physics. Due to the fabrication process of each superconducting qubit, no two are identical, so it can be said that each transmon exhibits a sort-of personality. The coherence times therefore form a fundamental part of this personality and can vary depending on environmental factors.

Firstly, the T_1 time also known as the spin-lattice time, longitudinal coherence time, amplitude damping time (among others) is a measure of the energy lost from the system. Specifically, it is time taken (more accurately the decay constant) for a state to decay from $|1\rangle$ to $|0\rangle$, i.e. excited state to ground state. T_1 times are measured using inversion recovery [55], which is a magnetisation preparation technique followed by an imaging sequence.

The T_2 time, known as the spin-spin relaxation time, transverse coherence time or phase coherence time, is the time (decay constant) for a state to decay from say, $|+\rangle$ to an equal probabilistic outcome of $|+\rangle$ and $|-\rangle$. Note that the difference between the two states being the phase, hence this is known as phase decoherence. T_2 times are measured using Hahn echoes [56]. A Hahn echo is a refocusing of spins via a pulse of electromagnetic radiation.

Chapter 8

Backends and Methods

This chapter shall briefly describe the IBM hardware used to execute programmes as well as the methods used in doing so. A quick caveat, IBM hardware and the qiskit software package are constantly being updated and so the information provided here is correct as of the time of publication. The following are some key definitions and components.

Qiskit - Simply put, qiskit is an open-source software package developed by IBM, using Python, to design and execute quantum circuits on backends.

Backend - A backend is a term used by IBM to describe the various devices qiskit programmes are run on. They include the real quantum computers as well as various simulators. For this body of work, only three backends shall ever be used.

`qasm_simulator` - This is a simulator that runs locally on the user's machine and emulates a quantum computer. The maximum size of the quantum register is 32 qubits¹, and all the qubits have full connectivity, i.e. every qubit can connect with every other qubit. The standard output for the `qasm_simulator` is counts - how many times a certain output state occurs, from counts qiskit is able to produce a probability distribution for that output. This is the standard mode of operation for real quantum devices as well. However, the `qasm_simulator` is also capable of outputting statevectors and density matrices etc., in various representations such as the Bloch sphere, something that is not possible in real devices. The output from the `qasm_simulator` can be thought of as the “correct” or “ideal” answer for a computation, as it contains no errors due to noise or any other effects and simulates what an ideal quantum computer would output. In this document, `qasm_simulator` will sometimes be referred to as simply, the simulator.

¹There are ways of increasing this limit but this involves different methods of simulation that are beyond the scope of this explanation.

`ibmq_santiago` - This is a quantum computer (device) consisting of a total quantum register of 5 qubits. Figure A.4 in APPENDIX A.2 shows the topology, i.e. the connectivity of qubits in the device. Here, the definition of connectivity is two qubits that may be connected together in order to implement two qubit gates, for example a CNOT-gate. This devices only outputs counts, meaning as it is a physical device, it can only output a finite number of samples as opposed to exact probabilities. The architecture of this device is `Falcon_r4L`, “Falcon” is the generation, “r4” means revision-4 and “L” is the topology, in this case the letter “L” stands for the word *line*. This will be the most commonly used real device and is referred to in the text as Santiago.

`ibmq_quito` - This a quantum computer (device) with a quantum register of 5 qubits, and the architecture is `Falcon_r4T`, similar to Santiago but the qubits are connected in a “T” shape. Figure A.5 in APPENDIX A.2 shows this topology. The output of this device is counts. This is sometimes referred to as Quito in the text.

For a more in-depth explanation of the various backends and features of qiskit, the interested reader should consult the online qiskit documentation [57].

8.1 Methods

Design and execution of programmes is done using `qiskit_v0.24.0` running on `Python 3.8.5`. `Spyder 4.1.5` is used as the IDE to write code along with `IPython 7.19.0`. Table 8.1 shows the version for each qiskit module.

<u>Module</u>	<u>Version</u>
<code>qiskit-terra</code>	0.16.4
<code>qiskit-aer</code>	0.7.6
<code>qiskit-ignis</code>	0.5.2
<code>qiskit-ibmq-provider</code>	0.12.1
<code>qiskit-aqua</code>	0.8.2

Table 8.1: Qiskit module versions.

The method for execution is as follows, the quantum circuit is coded in Spyder and then run on the `qasm_simulator`. This is to check that the programme performs as expected and for the initial phase of debugging, as the simulator gives the ideal results. Once the programme performs as expected on the simulator, the programme is then executed on one of the real quantum devices. The results of this execution are compared to results from the simulator to identify any anomalies that may have occurred and this is followed by any more potential debugging and re-running on backends. Once the programme runs to a satisfactory level on both the simulator and real device(s) then the whole process of execution is complete.

The choice of which device to run on, Santiago or Quito, is a result of which backends

are available at the time of execution. This is because backends are often taken down for maintenance or decommissioning. Frequently for long periods of time (days, weeks, months).

Chapter 9

Characterisation of Current Hardware

This chapter explores the limits of current hardware that is available from IBM, namely it examines the fidelity of the device(s) by running a discrete-time quantum walk programme. The base component for simulating a quantum field theory in this work.

Using the work of Kemp, Nishio, Satoh, Vogt-Lee and Bassan [58], who in turn build on the work of Douglas and Wang [59], it is possible to build a quantum circuit that is able to perform a quantum walk on a cycle graph. For our purposes, we shall consider the example presented in [58], namely, that of a 4-qubit circuit that corresponds to an 8-node cyclic graph, this is depicted in Figure 9.1. The vertices of the graph denote the possible positions of the walker.

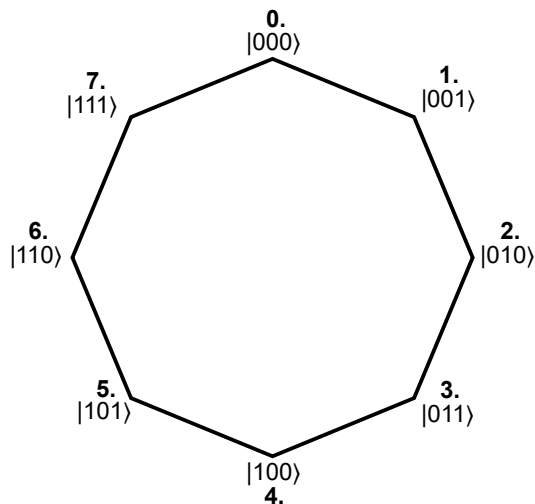


Figure 9.1: 8-node QW graph. Binary string (i.e. the state of the quantum register) corresponds to the site number in denary (bold).

The quantum register consists of 4 qubits, 3 of which are used to determine the position of the walker, the final qubit is used as the coin operator and is not measured. Essentially the circuit is built of two large sets of gates, an increment step gate and a decrement step gate used to move the walker, which themselves consist of CNOT and X gates. These increment and decrement gates are repeated depending on the number of steps required. One step involves one increment and one decrement, two steps involve increment, decrement, increment, decrement and so forth. The coin to be used is the standard Hadamard coin (implemented with a H gate). The circuit for this walk is shown in APPENDIX A.3. The results of the execution are given in Figure 9.2. The programme is incremented up to 8 steps and executed on `ibmq_quito`.

Figure 9.2 j) shows the Hellinger fidelity [60] between the real device and the simulator. For the purpose of this work, the Hellinger fidelity is a means of comparing how similar two classical probability distributions are, i.e. the results of the execution.

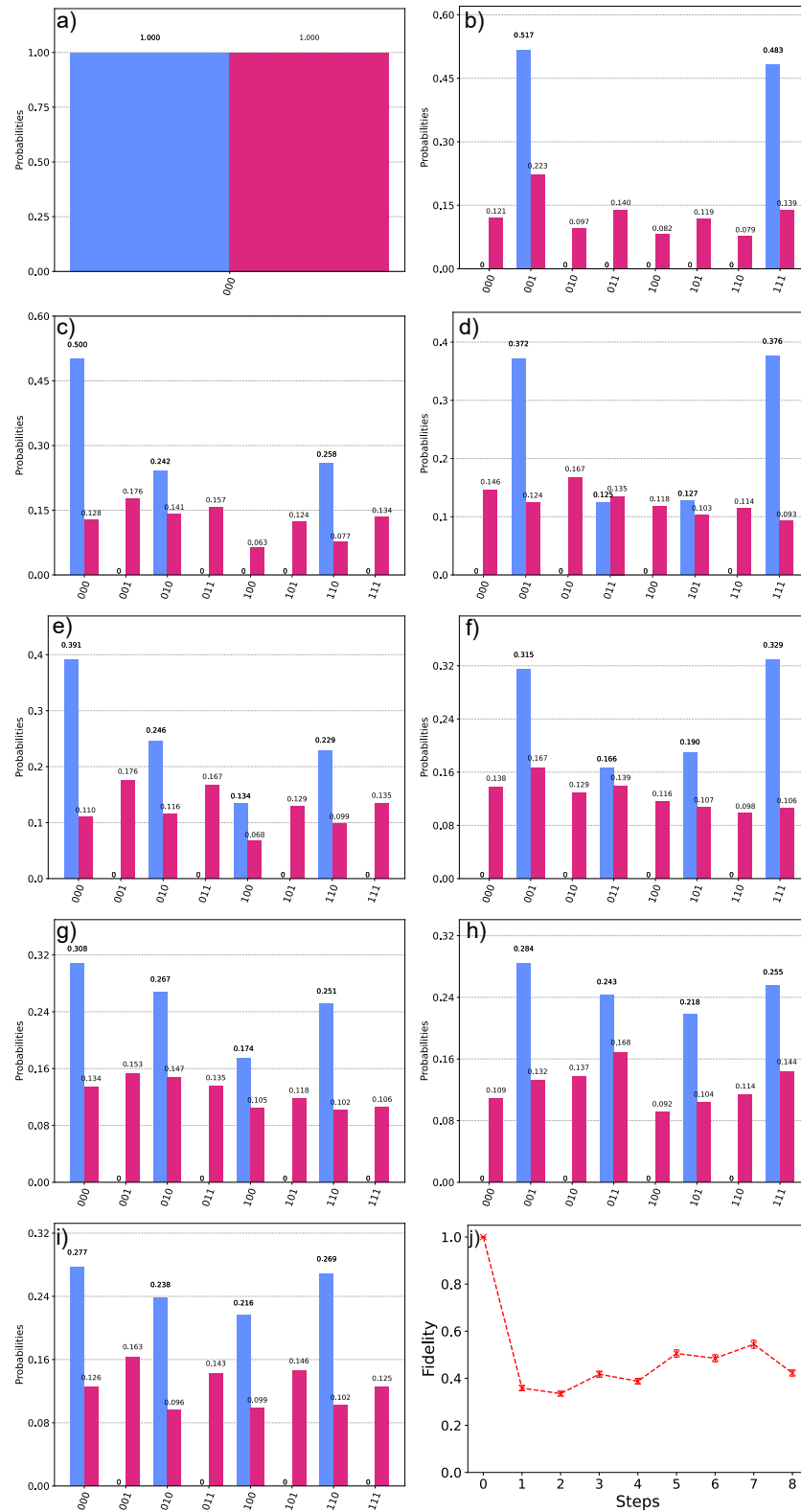


Figure 9.2: Discrete-time quantum walk for an 8-node graph. Panels a) to i) show steps 0 to 8 of the walk respectively (text size on histograms is fixed by the qiskit software). Bars represent the probability of finding the walker at a site denoted by a bit string. The blue bars represent the result of the simulator i.e. the expected result and red bars the real device. Panel j) shows the Hellinger fidelity [60] between the real device and the simulator for each step of the walk. Dotted red-line is to guide the eye. A value of 1.0 indicates perfect agreement between the simulator and the real device.

As can be seen the IBM device performs poorly. The minimum expectation would be that the real device follows the first few steps of the walk with a reasonable level of fidelity. However, panel j) of Figure 9.2 shows that the performance of the hardware is very poor, with an average fidelity of 0.45. In other words, the results just look like noise from the real device.

To examine this result let us consider a simpler example, that of a 4-node cyclic graph, as depicted in Figure 9.3.

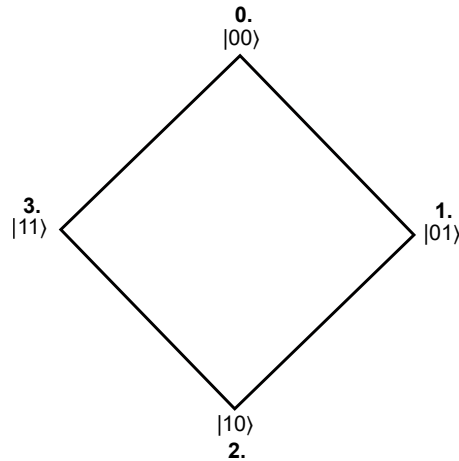


Figure 9.3: 4-node QW graph. As in Figure 9.1 vertices represent the possible positions of the walker and binary string corresponds to the site number in denary (bold).

The circuit now reduces to 3 qubits in total, 2 used to describe position and 1 as the coin. This is shown in Figure 9.4. Specifically the pattern the circuit follows is STEP, STEP, H acting on coin, STEP, H on coin, STEP etc. However, notice now that the circuit only consists of one large gate, a STEP gate that incorporates the functionality of the increment and decrement gates of [58].

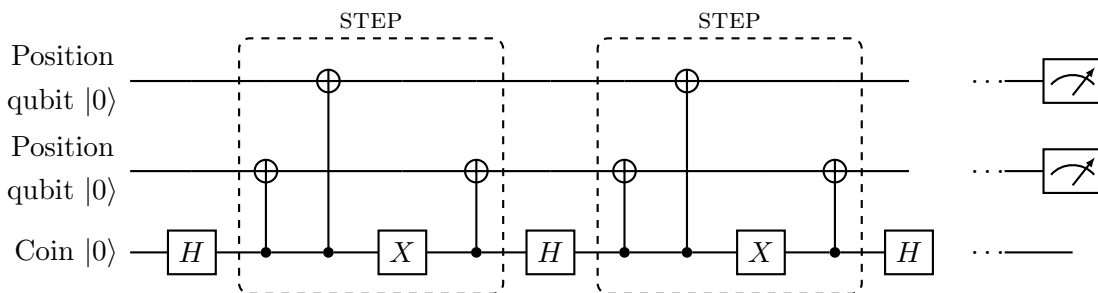


Figure 9.4: Circuit for 4-node quantum walk. Decomposition of STEP gates are enclosed by the dashed lines. Coin operation is given by single Hadamard gates acting on the coin register. These gates are repeated for the desired number of steps.

This programme was executed on `ibmq_quito` for 4 iterations, with the position of the walker being measured at the end of the desired number of steps. The results are presented below.

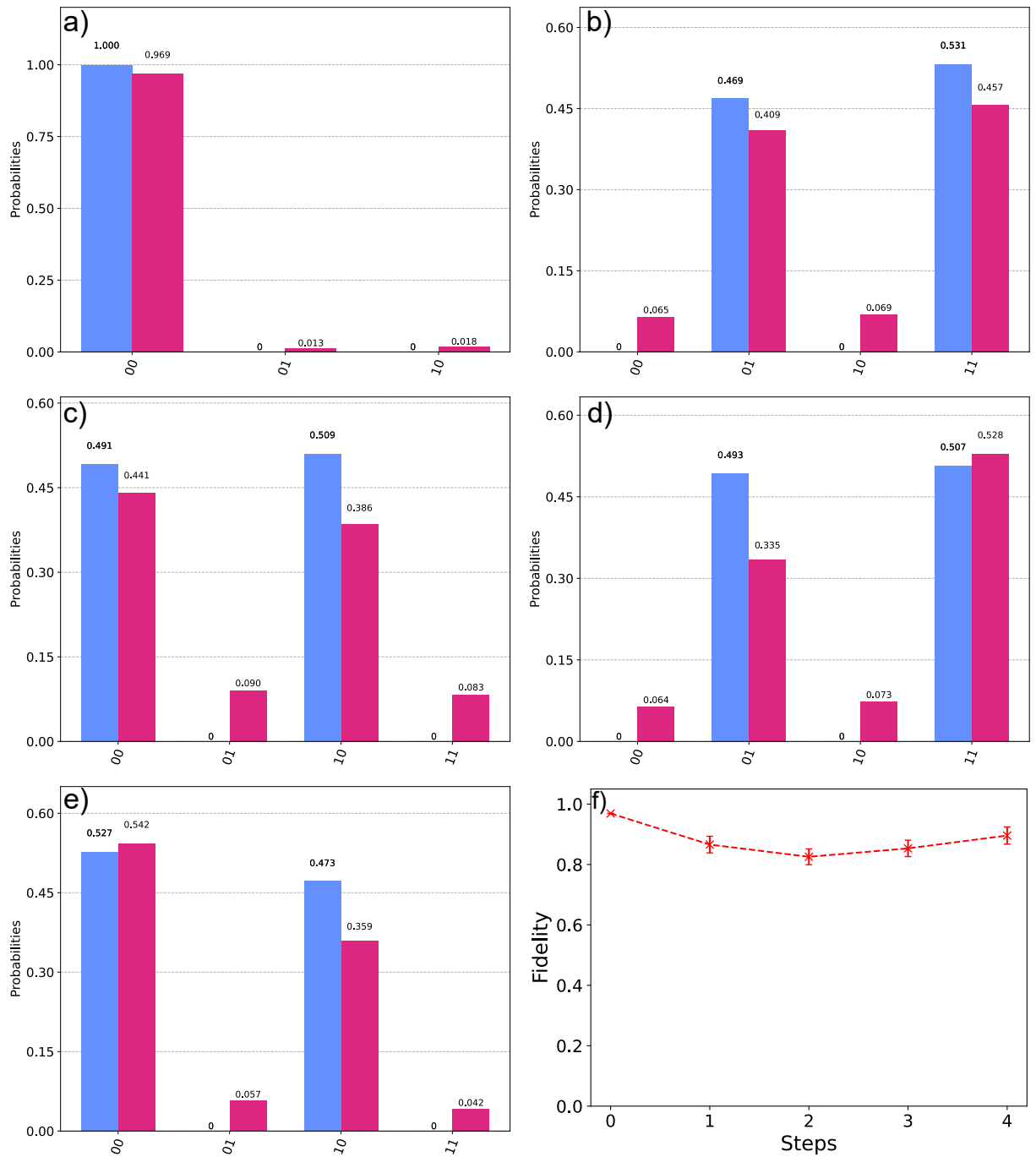


Figure 9.5: Discrete-time quantum walk for an 4-node graph. Panels a) to e) show steps 0 to 4 of the walk respectively. Bars represent the probability of finding the walker at a certain site denoted by a bit string. The blue bars represent the result of the simulator and red bars the result of the real device. Panel f) shows the fidelity, comparing the two probability distributions, of the walk for each step between the real device and simulator.

As can be seen in Figure 9.5, this particular IBM device performs the 4-node quantum walk far better than the 8-node. Although, in general, the fidelity of the results do degrade slightly as the walk progresses from the initial steps. However not as much as in Figure 9.2, it is still clear to see what the accurate value would be, so this variation can be attributed most likely to noise and not major errors occurring in the execution of the circuit. In other words, errors are considered to be improbable in a circuit with so few gates.

Let us now examine why the 4-node quantum walk performs similarly to the simulator whilst the 8-node counterpart does not. On inspection of their respective quantum circuits (shown in Figure A.6 APPENDIX A.3 and Figure 9.4 respectively), both appear to be relatively simple. However, at this point it is worth discussing the way in which IBM processes circuits and how they are implemented. This process is known as transpiling. Currently, all IBM backends utilise the same basis gates: {CNOT, ID, RZ, SX, X} which form a universal set of gates. In other words, all other gates must be implemented using only these. Focusing on the increment and decrement gates of the 8-node circuit, they contain Toffoli and C4-NOT (C4-X) gates. Implementation of these gates using the basis gates proves to be a non-trivial task. The qiskit software has a method of implementing Toffoli gates by reducing them into a set of basis gates. In total, this decomposition consists of 18 gates - 6 CNOT gates, 10 RZ gates and 2 SX gates for each Toffoli. This is already a large increase in the number of gates required to perform a single Toffoli operation. However, this is not the case for the C4-NOT or generally any C_n -NOT gate (where $n > 4$). There currently exists no optimised way of implementing these gates in qiskit. For the case of the C4-NOT, qiskit transpiles this into a 34 gate circuit: 14 C-NOT gates, 18 RZ and 2 SX. A single C4-NOT gate thus needs a large number of gates that in and of itself forms a fairly large circuit and [58] contains two Toffolis and two C4-NOT gates for a single step of the quantum walk. This coupled with the other gates (i.e. X and H) being transpiled explains why the quantum walk appears as just noise. The number of gates being implemented, even for a single step, has enough noise associated with each gate to degrade the results to the point where it is unrecognisable. In comparison, the 4-node circuit does not contain any Toffoli or C4-NOT gates, so the only gates to transpile are H and X, which demonstrates why these results are more accurate. This circuit, when transpiled is much closer to the original design and contains a modest number of gates so noise degradation does not ruin the result. However, some degradation due to noise is present and this can explain why, in general, the results appear to decrease in fidelity as the number of steps increase - there is more noise from more gates.

It is worth mentioning that the transpilation process is automatic and occurs when the circuit is sent to the device (transpilation is obviously not necessary when run on the simulator). Furthermore, it does not solely consist of transforming the circuit into basis gates but also deciding which qubits to use in the device (determined by the hardware's topology) among other tasks [61]. However, as discussed above, this process is not always a hundred-percent optimal and can lead to the end result being inefficient. It is possible to manually bypass at least some part of this transpilation process, namely the part of converting the original circuit into one consisting of solely basis gates. This is done simply by designing the initial circuit in terms of the IBM basis gates ($\{\text{CNOT, ID, RZ, SX, X}\}$) in the first place.

We will now consider what happens when the 4-node quantum walk is designed using only the basis gates. This is a very straightforward task as the original circuit itself is simple. The advantage of this is the user can decide exactly how the gates will be implemented and pick the best circuit decompositions for their needs. The figure below shows the optimised version of the 4-node quantum walk to be used.

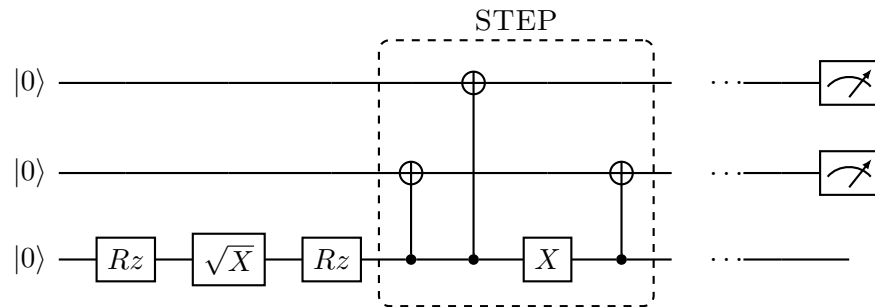


Figure 9.6: Optimised circuit consisting of basis gates. One step is depicted. First three gates implement the Hadamard coin operator.

This optimised version (shown in Figure 9.6) can now be compared to the original one that relies on the automatic transpilation process.

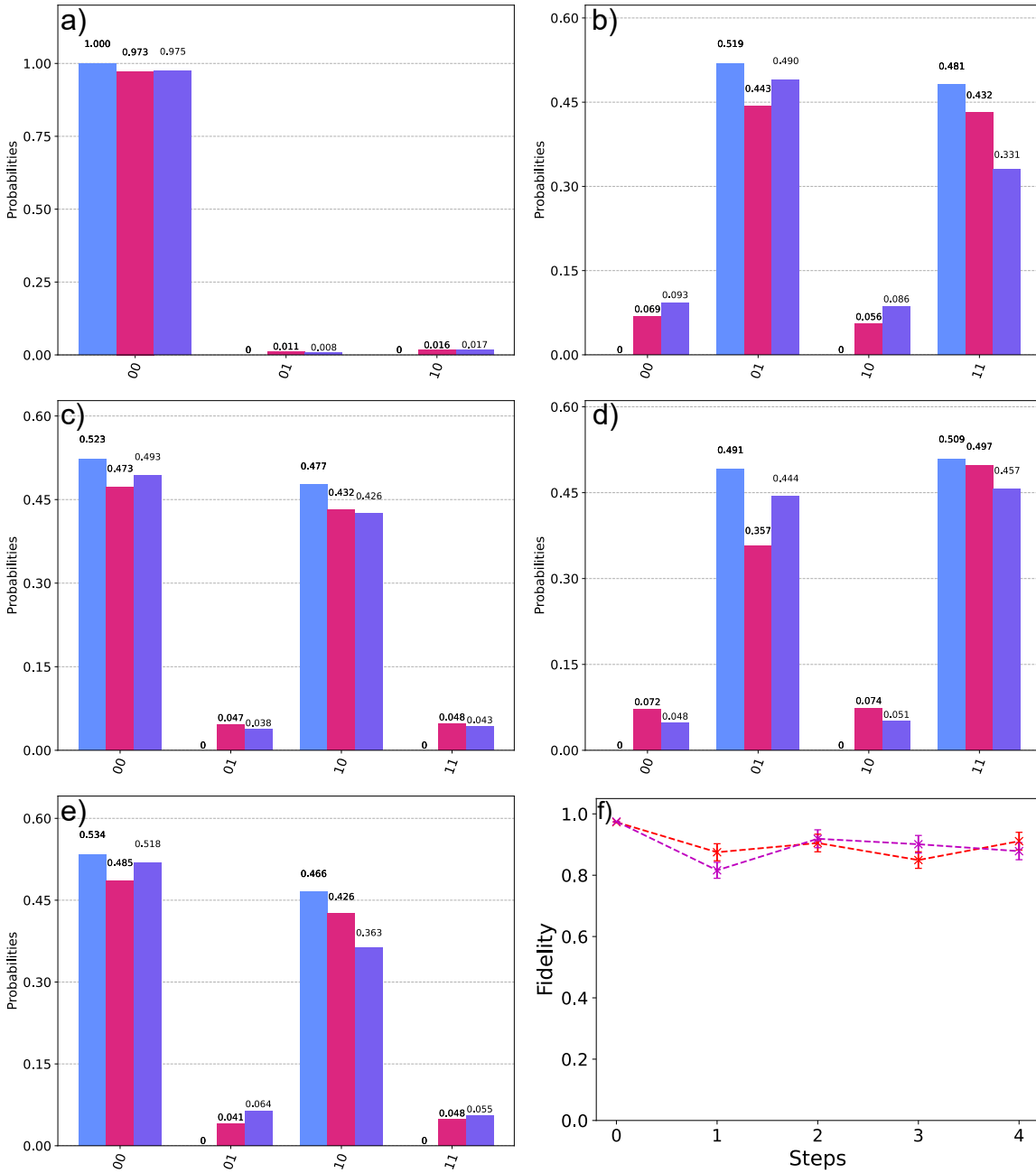


Figure 9.7: Comparison of discrete time quantum walk for a 4-node graph. Panels a) to e) simulator (light blue bars) vs. standard transpiled (red bars) vs. optimised (purple bars) for steps 0 to 4 respectively. Panel f) shows the fidelity, comparing probability distributions, of the standard transpiled (red) and optimised (magenta) for each step of the walk. Error bars represent standard error.

Figure 9.7 shows the similarity in performance of the manually optimised and automatically transpiled circuits. The optimised walk performs slightly better for some steps, with an improvement of approximately 1% overall, over the automatically transpiled version (comparing average fidelities). It is interesting to note that this implies that the qiskit transpiler does not choose the most optimum configuration for this case, which is trivial to do when computed manually. In this specific case the performance increase may not appear to change the overall result drastically, but by extrapolation, it is speculated this optimisation will have a greater effect, e.g. more steps in a quantum walk with higher fidelity. For the case of a more complicated circuit, it is speculated that it would be worth the effort, if not practically mandatory, to perform this optimisation in order to achieve the best results.

To conclude this section it is worth remarking on the current state of IBM hardware as well as software. It has been shown that by using careful programming it is possible to achieve effective quantum simulations on NISQ-era hardware to the level that one would intuitively expect. The qiskit software package has been designed to, and indeed does, operate on a high-level, capable of performing complicated tasks such as transpiling as well as offering a range of tools for the programmer. However, it has been shown that this can sometimes be detrimental to execution and that current progress regarding the actual quantum hardware is not on the same level as the software. In other words, although qiskit is able to do tasks such as conversion of circuits into basis gates, it is not necessarily the best course of action. The end user should ideally perform some of these tasks by operating on a low-level programming regime. This is to enable a more direct level of communication between the hardware and the user to better optimise their programme for their needs.

Chapter 10

Custom Noise Models

A feature of the qiskit software package is the ability to create and use noise models. IBM offers a noise model of each available backend that, in principle, models the noise levels of that device, on that day, as accurately as the software will allow. However, it is also possible for the user to create their own custom noise models by varying, and including/not including, parameters associated with noise in superconducting quantum computers. Noise models are created using the `qasm_simulator` as a base template, from this the noise parameters are introduced on top of this template. The goal of this chapter is to ascertain how much current noise levels in an IBM backend would need to be reduced in order to achieve a consistent, high-fidelity result for a discrete-time quantum walk.

To investigate this, consider the DTQW on a cycle graph, using the same paradigm as the walks in the previous chapter. However, now the quantum register is extended to a total of 5 qubits, this being the maximum number available in IBM's current hardware. 4 qubits give the binary encoding of the position of the walker and the 5th qubit acts as the coin operator. Accordingly, the number of sites on the graph for this quantum walk now increases from the walks in the previous section to a total of 16, as shown in Figure 10.1.

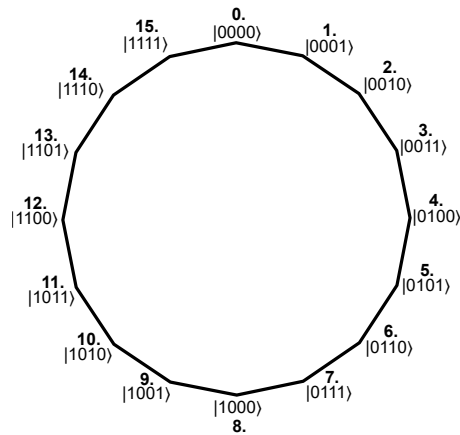


Figure 10.1: Diagram of the 16-node cycle graph used in the DTQW.

The corresponding quantum circuit for this walk can be seen in APPENDIX A.4.

Before we consider the execution of the quantum walk let us discuss what a custom noise model consists of. The parameters that make-up a noise model are: single-qubit gate errors, two-qubit gate errors, three-qubit gate errors and multiple (i.e. four and five) qubit gate errors. The single qubit gate errors correspond to the single-qubit unitary operators in the circuit, in the case of the 16-node DTQW that shall be used, this equates to Hadamard gates and X gates. The two, three and multiple qubit errors correspond to C-NOT, Toffoli and C4-X/C5-X gates respectively. In practice these errors represent depolarising error rates associated with each of the gates and are numbers arbitrarily chosen by the user. The method for choosing the errors involves trial and error by gradually tweaking the numbers until the results of simulating a DTQW on a noise model match the results on a real device, as closely as possible. The chosen values for each of the errors, which represent the probability per gate, are shown below.

<u>Type of gate</u>	<u>Error</u>
single-qubit	0.005
two-qubit	0.02
three-qubit	0.04
multiple-qubit	0.6

Table 10.1: Errors used in custom noise model to emulate `ibmq_santiago`.

The values shown in Table 10.1 are referred to in this text as **full noise strength**, as they refer to the maximum amount of noise modelled, i.e. the amount of noise that most closely resembles the real device.

As mentioned previously, IBM also provide a noise model for each of their backends. This noise model has the error rates pre-determined by automated analysis of each of the devices every so often, e.g. every 24 hours. Due to the susceptibility of transmons to environmental noise, these error rates vary over time and hence this method of modelling noise is a means of keeping track of that variation and providing the most accurate error values. The other advantage of this, of course, is the user does not need to define each individual error rate, this is done automatically. In addition to depolarising errors the IBM noise model also contains: readout errors, errors associated with the measurement of qubits, and a thermal relaxation error that essentially consists of the T_1 and T_2 relaxation time constants [62]. It is important to note however, that the depolarising and thermal relaxation errors only apply to single and two-qubit gates, so these errors do not apply to three-qubit and higher gates. In this text, the noise model provided by IBM will be referred to as the **IBM noise model**.

In order to investigate noise on IBM quantum computers, the DTQW for 16 nodes is executed, first on the base `qasm_simulator`, and then the custom noise model for 16 steps of the walk, in single step increments. The noise level is then reduced in gradual decrements until there are eventually no depolarising errors in the custom noise model and the walk is run for each of these decrements. The same DTQW is then executed on the IBM noise model of

`ibmq_santiago` and the real `ibmq_santiago`. The fidelity for each step of the quantum walk is recorded up to a maximum of 16 steps. The fidelity for each backend: custom noise model, IBM noise model and Santiago, is found by comparing the probability distributions of each step of the walk with the corresponding step from the `qasm_simulator`, which can be thought of as the “perfect” or expected result, as it contains no noise.

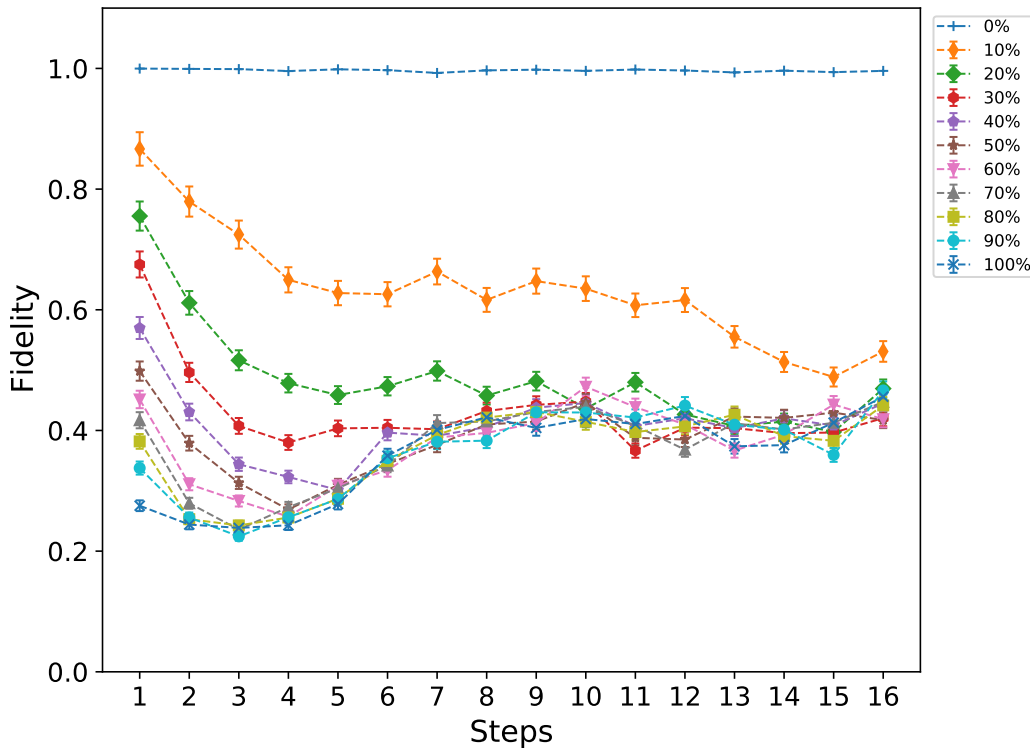


Figure 10.2: Fidelity at each step of a 16-node, 16 step DTQW, for varying noise strengths of the custom noise model. Error bars represent standard error.

Figure 10.2 shows the fidelity for each step of the quantum walk and for decreasing noise levels. 100% refers to 100% of the full noise strength, i.e. the values in Table 10.1. 90% refers to 90% of each error in Table 10.1 etc. At 0% of full noise strength the errors all take a value of 0. However, it is important to note that this is not exactly the same (although extremely close) as the base `qasm_simulator`. The distinction being that at 0% noise strength the noise model still takes into account the small errors introduced by the transpilation process.

The Hellinger fidelity is a measure of how similar two probability distributions are, it takes values between 0 and 1. A value of 1 indicates perfect agreement between the distributions, a value of 0 indicates a perfect anti-correlation between distributions. A value of 0.5 indicates no correlation between distributions. As a general rule, a fidelity of 0.8 or higher is considered a very good result that shows a clear correlation between the two distributions.

As can be seen from Figure 10.2, For all noise strengths except 0%, the performance is poor. Noise strengths 60-100% all begin with a fidelity below 0.5, in other words they cannot even produce an accurate result for 1 step of the quantum walk. Similarly, for noise strengths 30-50%, which show some improvement but still not enough to be considered accurate. Serious improvements in the fidelity only begin to be seen when at 10% and 20% of full noise strength. However, the fidelity soon declines in the case of 20% noise strength and after just 2 steps the results of the walk are indiscernible from noise. In the case of 10% of full noise strength the performance is markedly better, with the fidelity being consistently higher than all other noise strengths for every step of the walk. Here, approximately 13 steps of the walk are achieved at a reasonable standard with a final fidelity around 0.6. This is roughly the limit before the results of the walk start to become indistinguishable from the noise. 0% noise strength achieves a constant fidelity of almost exactly 1.0, with minor fluctuations occurring due to the transpilation errors (see Chapter 9 page 42). This confirms that indeed, 0% noise strength and the `qasm_simulator` are very similar. In all other cases for the custom noise model it can be seen that the fidelity tends to an equilibrium value of approximately 0.5 as the number of steps tends to 16. Thus showing the effect of noise overcoming the results of the walk, as noise corresponds to no correlation, in other words it is random.

Figure 10.3 shows the fidelity for the same DTQW for selected noise strengths and includes the noise model of `ibmq_santiago` as well as `ibmq_santiago` itself.

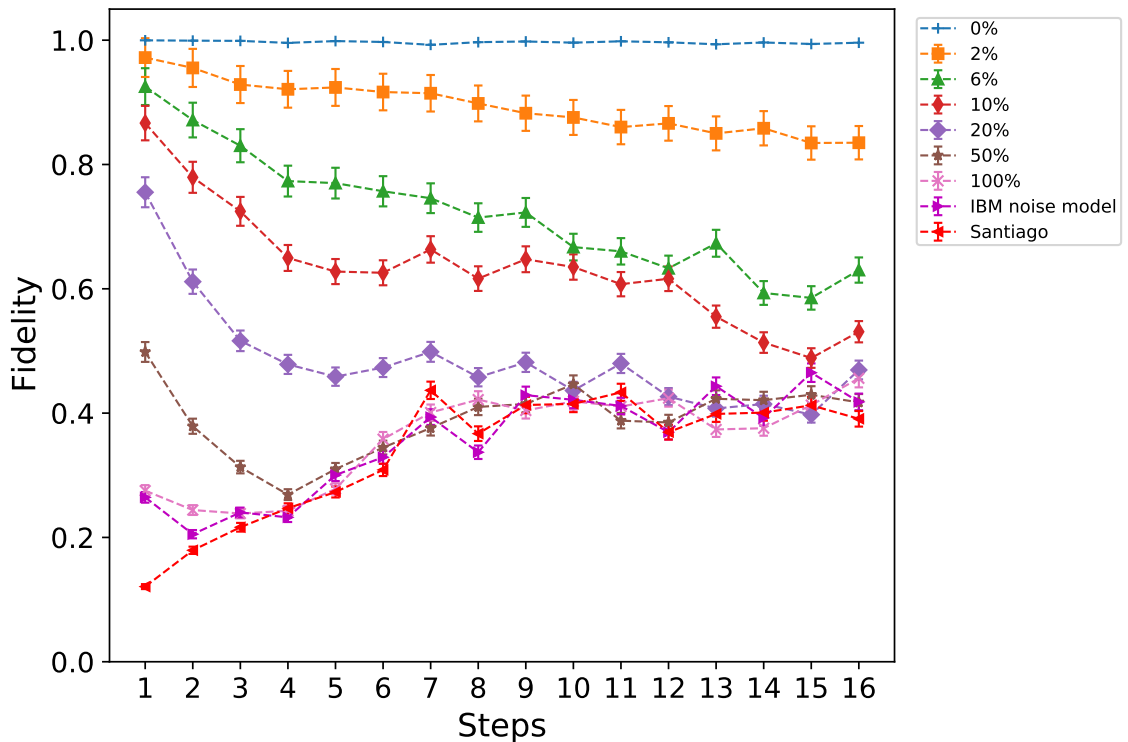


Figure 10.3: Fidelity at each step of a 16-node, 16 step DTQW, for varying noise strengths of the custom noise model as well as the IBM noise model and Santiago.

From Figure 10.3 a direct comparison between the noise models and `ibmq_santiago` can be made. Overall, the worst performing is the real device, starting with a low fidelity that gradually increases but never exceeds 0.5. This indicates a slight anti-correlation with the expected results, particularly for the first few steps, but beyond roughly 6 steps, this is just noise. Interesting to note is the similar performance of the IBM noise model, 100% strength and Santiago. For the IBM noise model and Santiago this would be expected but the custom noise model also follows the real device very closely, at points overlapping, e.g. 4 steps, 9 steps and 10 steps. Although overall, the performance of the IBM noise model is slightly closer to the real device, it is interesting to see this overlap or lack there-of at certain points. This also demonstrates that depolarising errors are the dominant contribution to noise, since it is possible to so closely model (as shown in Figures 10.2 and 10.3) the real device without the addition of other errors, for instance Pauli errors, Kraus errors etc. The addition of 2% and 6% noise strength also shows the non-linear improvement in performance once the level of noise is below 10% of the initial parameters. For 6% noise strength the entire walk of 16 steps is achievable with a reasonable fidelity, only dropping to slightly sub-optimal levels towards 16 steps, even at this point the results of the walk are distinguishable from any noise. 2% noise strength performs excellently and by far has the highest overall fidelity (aside from 0% of course), never dropping below 0.8 . All steps of the walk are clearly distinguishable from noise and extrapolation of Figure 10.3 would seem to suggest that many more additional (probably around 8) steps could be achieved with high or good fidelity results for 2% noise strength. We therefore conclude that in order for a DTQW on a 16-node cycle graph to be executed with consistent, high fidelity results, for each step, the current noise levels within the Santiago device would need to be reduced by approximately, at least, 94%.

Table 10.2 & Figure 10.4 show the corresponding fidelities and histograms for the 16th step of the DTQW, for each of the noise strengths/devices in Figure 10.3. This shows how fidelity relates to human-readable results.

Noise strength/device	Fidelity	Error
0%	0.99	0.000
2%	0.84	0.027
6%	0.63	0.020
10%	0.53	0.017
20%	0.47	0.015
50%	0.41	0.013
100%	0.45	0.015
IBM noise model	0.42	0.013
Santiago	0.39	0.013

Table 10.2: Fidelities of varying noise strengths/devices for 16th step of 16-node DTQW.

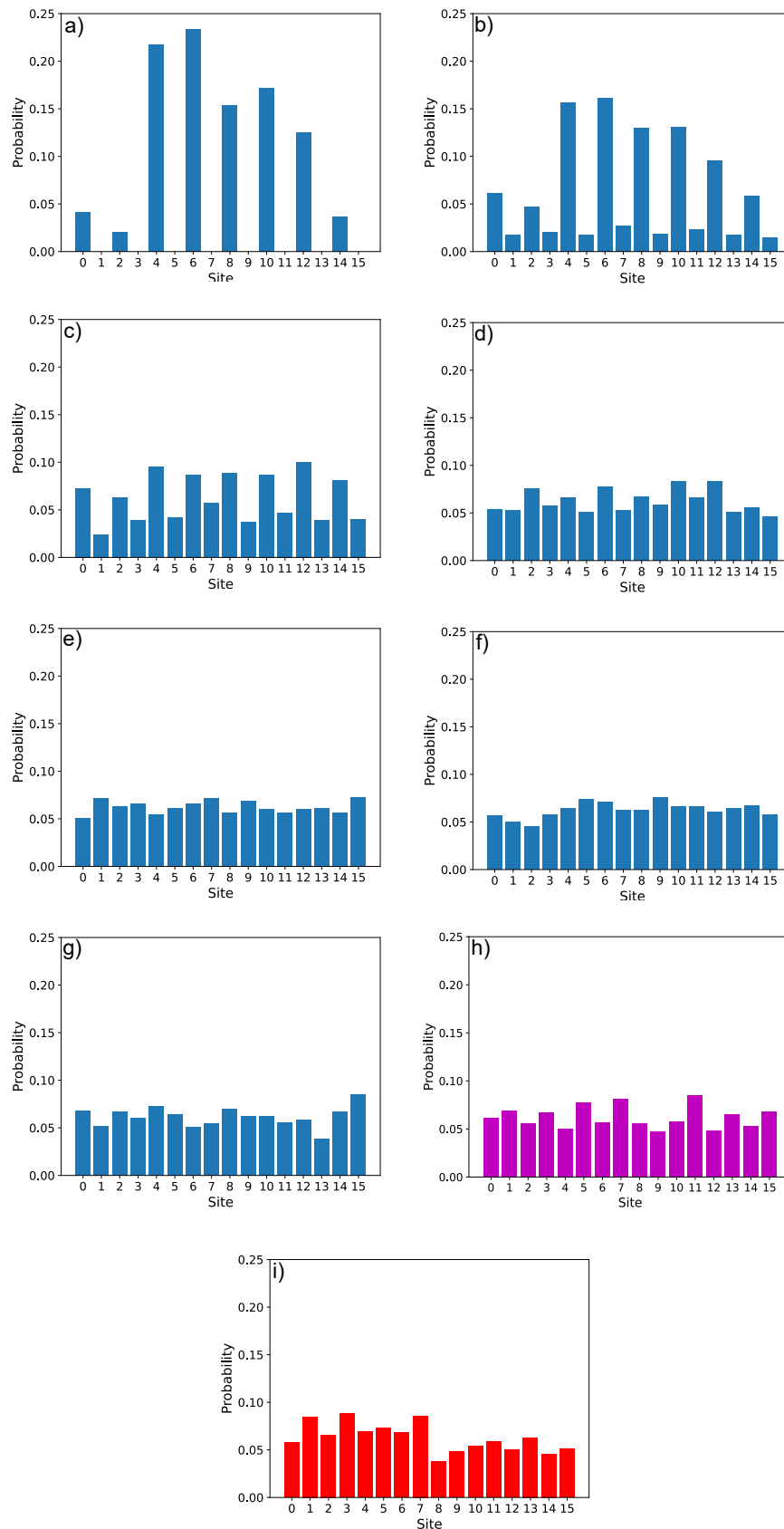


Figure 10.4: Probability distributions for the 16th step of the 16-node DTQW: a) 0% noise strength, b) 2%, c) 6%, d) 10%, e) 20%, f) 50%, g) 100%, h) IBM noise model (magenta), i) Santiago (red).

Considering how much current noise levels in the backend have to be reduced before clearly-readable results are produced, the natural question to pose is why are the results so poor for both the real device and (most) of the noise models? The answer is related to the discussion in the previous chapter regarding the transpilation process. It has been established that qiskit does not currently have a way to efficiently, meaning economical in terms of gates used, implement anything more than a Toffoli gate, that is to say a C4-X or C5-X gate. The very gates that are required for this encoding of a DTQW. The transpilation process resulting in a large increase in the number of gates used to execute the programme and therefore a large increase in the noise. The noise ultimately coming from each gate in the circuit, and it has now been established particularly from the depolarising errors associated with each gate. Therefore the more gates that are added due to transpilation, the more noisy the circuit becomes. This effect is exacerbated for larger quantum registers, requiring larger (more qubit) Cn -X gates. Again, as similarly discussed in the preceding chapter. Therefore, the only way to mitigate this effect is either, develop an efficient way of implementing Cn -X gates¹, which also may not be possible within the restrictions of IBM's transpiling process or reduce the noise levels associated with the transmons through better engineering, development of circuit QED etc. In practice, improvement of superconducting quantum computers will probably consist of a combination of these two.

¹Assuming of course a more efficient way exists.

Chapter 11

Simulated Quantum Walks vs. Experimental Quantum Walks

This chapter explores the differences between quantum walks executed on IBM hardware, as demonstrated in this work, and quantum walks executed experimentally by others prior to this work. Specifically, exploring if noise from experimental implementations of quantum walks can be used to simulate noise found in this current generation of IBM quantum computers. The chapter is structured as follows, we begin with the examination of an 8 step DTQW, run using IBM hardware, and re-iterate the conclusions of the previous chapter but for this specific walk. This is then compared to the first experimental implementation. Following this, the second experimental implementation is introduced and a comparison to a 3 step, IBM, DTQW is made.

Firstly, let us consider the DTQW using 5 qubits and consisting of 16 nodes. Figure 11.1 shows 8 steps (as that is the minimum number required to cover the whole graph) of the walk executed on the `qasm_simulator` as well as using IBM's noise model of the Santiago backend.

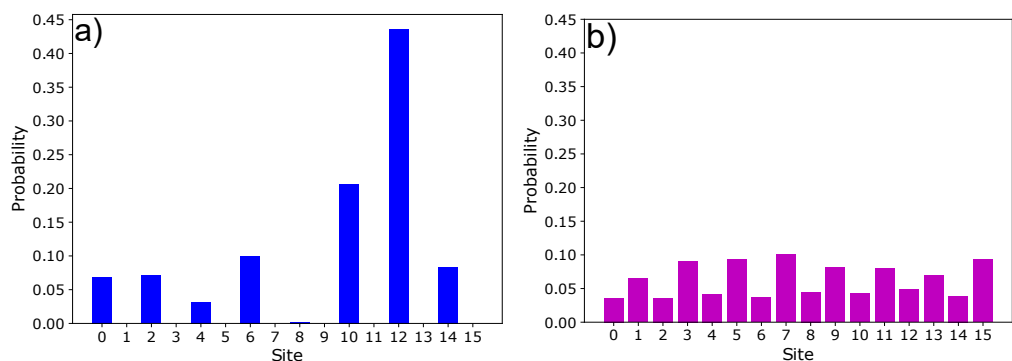


Figure 11.1: Probability distribution for 8 steps of the 16-node DTQW run on `qasm_simulator` a) and using the IBM noise model b).

Figure 11.2 shows the DTQW for varying noise strengths, using the custom noise model, as well as results from execution on the real device `ibmq_santiago`.

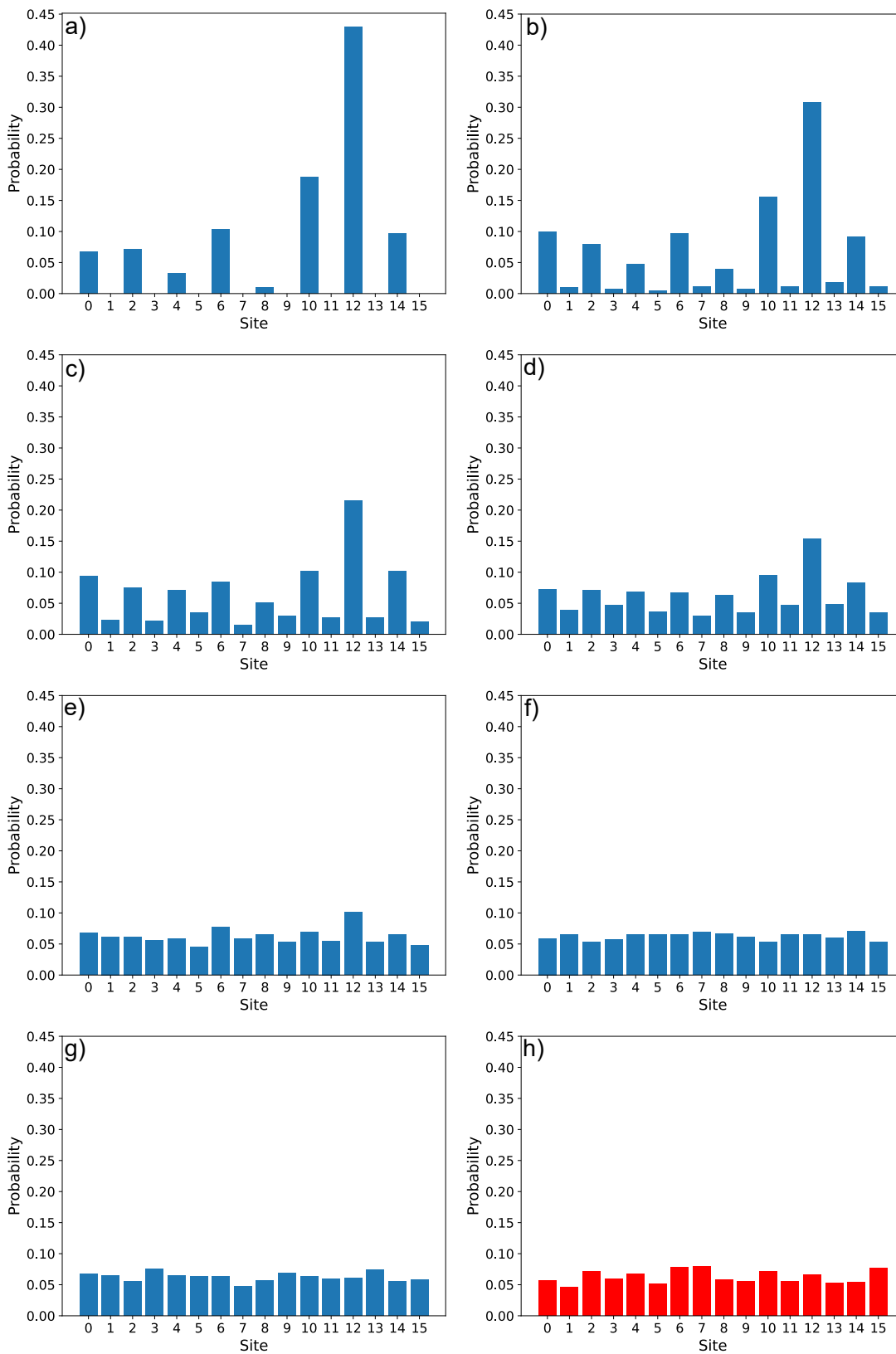


Figure 11.2: Probability distributions for 8 steps of the 16-node DTQW. From top left, a) 0%, b) 2%, c) 6%, d) 10%, e) 20%, f) 50%, g) 100% of the full noise strength and h) Santiago (red).

As can be seen in Figure 11.1 & 11.2 the effect of noise becomes drastic even at low levels. At approximately 10% of the full noise strength (Figure 11.2 d)), the walk becomes indiscernible from random noise. Beyond this point the probability distributions fail to resemble anything close to what would be expected for an asymmetric DTQW. In other words, the results are just noise. Conversely, for the case of the `qasm_simulator` and up to 6% noise strength (Figure 11.2 c)) a DTQW is visible. In summary, there would have to be a 94% reduction in current noise levels, in the Santiago device, to accurately execute a DTQW, for 8 steps, using 5 qubits, on a 16-node graph.

11.1 Silicon Photonic Quantum Walk

The work of Xiao-Chun et al. [63], that in turn builds-on the work of [64], showed that it is possible to implement quantum walks on a silicon photonic chip.

The experimental procedure involves a 775 nm Ti:Sapphire laser collected into a single-mode fibre. The light from this laser is then vertically polarised. The horizontally polarised component is not required for this experiment. The vertically polarised light is incident on a periodically poled potassium titanyl phosphate (PPKTP) crystal, which in turn produces 1550 nm photon pairs. Half of these pairs are then vertically polarised and are used in the chip. The silicon chip consists of a series of ports for the photons to enter, and then a honeycomb-like structure that is essentially a waveguide (where the quantum walk occurs), on the other side of this structure are a further set of ports for the photons to exit. So, the vertically polarised photons are coupled into the chip by a grating, enter via the ports, move through the structure that implements the quantum walk, and finally exit from the ports on the other side where they are detected by a superconducting nanowire single photon detector. Figure 2 of [63] shows the structure of the silicon chip as well as the experimental setup.

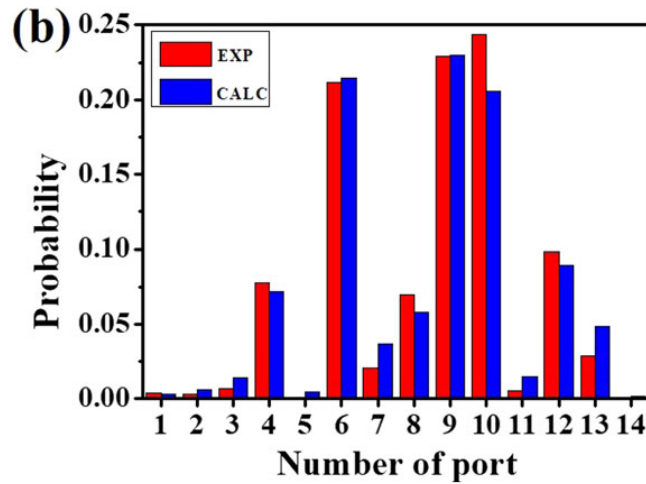


Figure 11.3: Figure 3 b) of [63]. Red bars represent the experimental probability of finding a photon at that port. Blue bars represent the simulated probability.

Figure 11.3 shows the results for a quantum walk on a 14-port silicon chip, with the photons entering into port 7 of 14, i.e. the middle.

Comparing Figure 11.3 and Figures 11.1 & 11.2, it is interesting to note some discrepancies. The silicon chip quantum walk exhibits some clustering of probabilities that are not seen in an ideal DTQW, for example the high probabilities for finding a photon at adjacent sites 9 and 10. This is not the case for the `qasm_simulator` walk or generally for the noise models, up to 10% noise strength Figure 11.2 d). Clustering of some states does appear at 20 - 100% noise strength and in `ibmq_santiago`. However this has been attributed to noise and is not present to the same extent as Figure 11.3. Furthermore, it is interesting to note that the authors of [63], at least qualitatively, are able to predict this behaviour of their quantum walk to a high degree of accuracy, as indicated by the blue bars in Figure 11.3. However, in any case, the implementation of this quantum walk on a silicon photonic chip does not appear to behave in the same way as a simulated quantum walk (with a binary encoding) on the `qasm_simulator`, let alone `ibmq_santiago` and hence is an unsuitable candidate for modelling noise. In other words, the implementation of this experimental quantum walk does not model the noise levels currently found in the Santiago device well.

11.2 Comparison to Trapped ions

In 2012 Matjeschk et al. [65] showed that it is possible to experimentally implement a DTQW using trapped ions. The experiment involves using the hyperfine energy-levels of a single $^{25}\text{Mg}^+$ ion in order to implement a DTQW on a line.

The $^{25}\text{Mg}^+$ ion is confined in a linear Paul trap. The two coin states labelled $|H\rangle$ and $|T\rangle$, for heads and tails respectively, are chosen to be the

$$\begin{aligned} |H\rangle &\equiv |^2S_{1/2}, F = 2, m_F = 2\rangle, \\ |T\rangle &\equiv |^2S_{1/2}, F = 3, m_F = 3\rangle \end{aligned}$$

electronic states. In addition,

$$|A\rangle \equiv |^2S_{1/2}, F = 2, m_F = -2\rangle$$

where $|A\rangle$ is the motional state.

Coin and shift operators are implemented via means of laser and radiofrequency pulses between electronic states of the ion. Figure 4 of [65] shows in detail how these states are related and their role within the walk. The coin is initialised in the $|T\rangle$ state by optical pumping. Read-out of the final state is done by photon scattering and hence position states of the walker correspond to coherent states of the emitted photons.

$$|\alpha_k\rangle = e^{-|\alpha_k|^2/2} \sum_{n=0}^{\infty} \frac{\alpha_k^n}{\sqrt{n!}} |n\rangle \quad (11.1)$$

where $|\alpha_k\rangle$ denotes the position of the walker, α_k are motional states and $|n\rangle$ are the Fock states (see APPENDIX A.5).

Figure 11.4 shows the experimental results for a DTQW of 3 steps on a line.

Figure 11.5 shows the DTQW on a 16-node cycle graph for 3 steps, run on `qasm_simulator` and using IBM's noise model of Santiago. In the cyclic walk the walker moves to positions 15, 14, 13 whereas in the experimental walk, the walker moves to position -1, -2, -3 due to being on a line. Considering the walk is only for 3 steps, any interference effects from being on a cyclic graph are not an issue and the walk may be treated as if it was on a line. In addition the coin qubit (in our circuit) is now initialised in the $|1\rangle$ state, whereas in previous runs it has been in the state $|0\rangle$. This is in order to match the initialisation of the coin in the trapped ion setting.

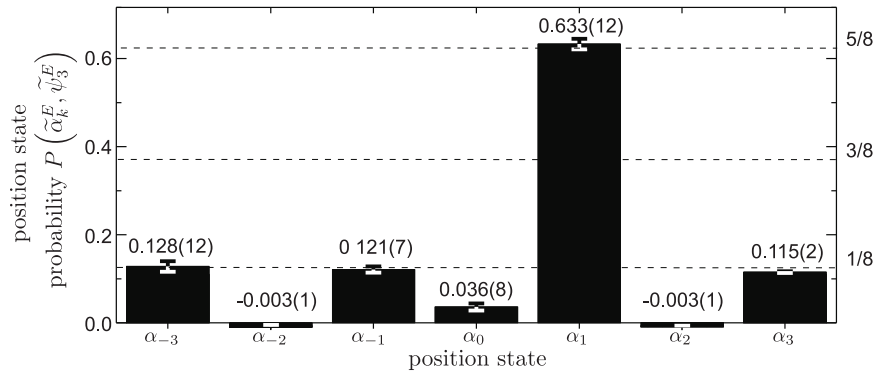


Figure 11.4: Figure 15 of [65] showing the probability distribution (in Fock space) for a DTQW of 3 steps on a line, α_s denote the position on the line e.g. α_0 is position 0 etc. Negative probabilities occur as a result of the authors subtracting overlapping probabilities.

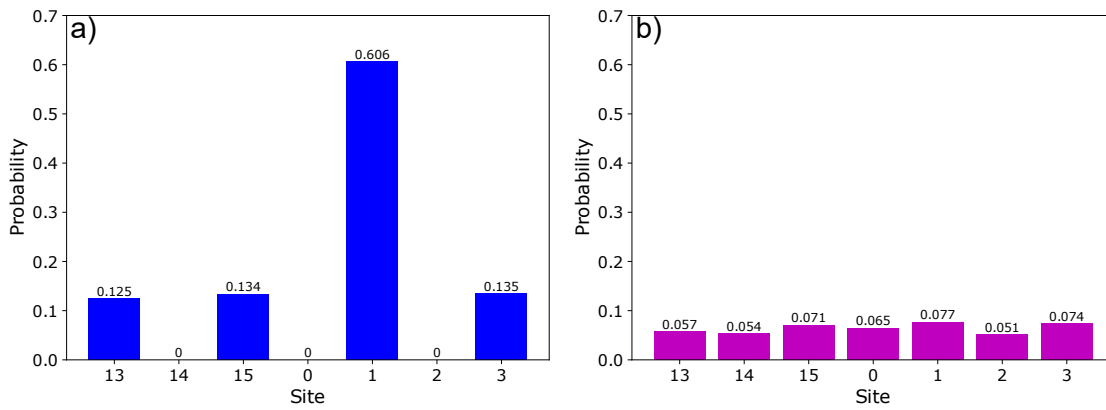


Figure 11.5: Probability distribution for a 3 step DTQW on a 16-node cycle graph run on `qasm_simulator` a) and using the IBM noise model b).

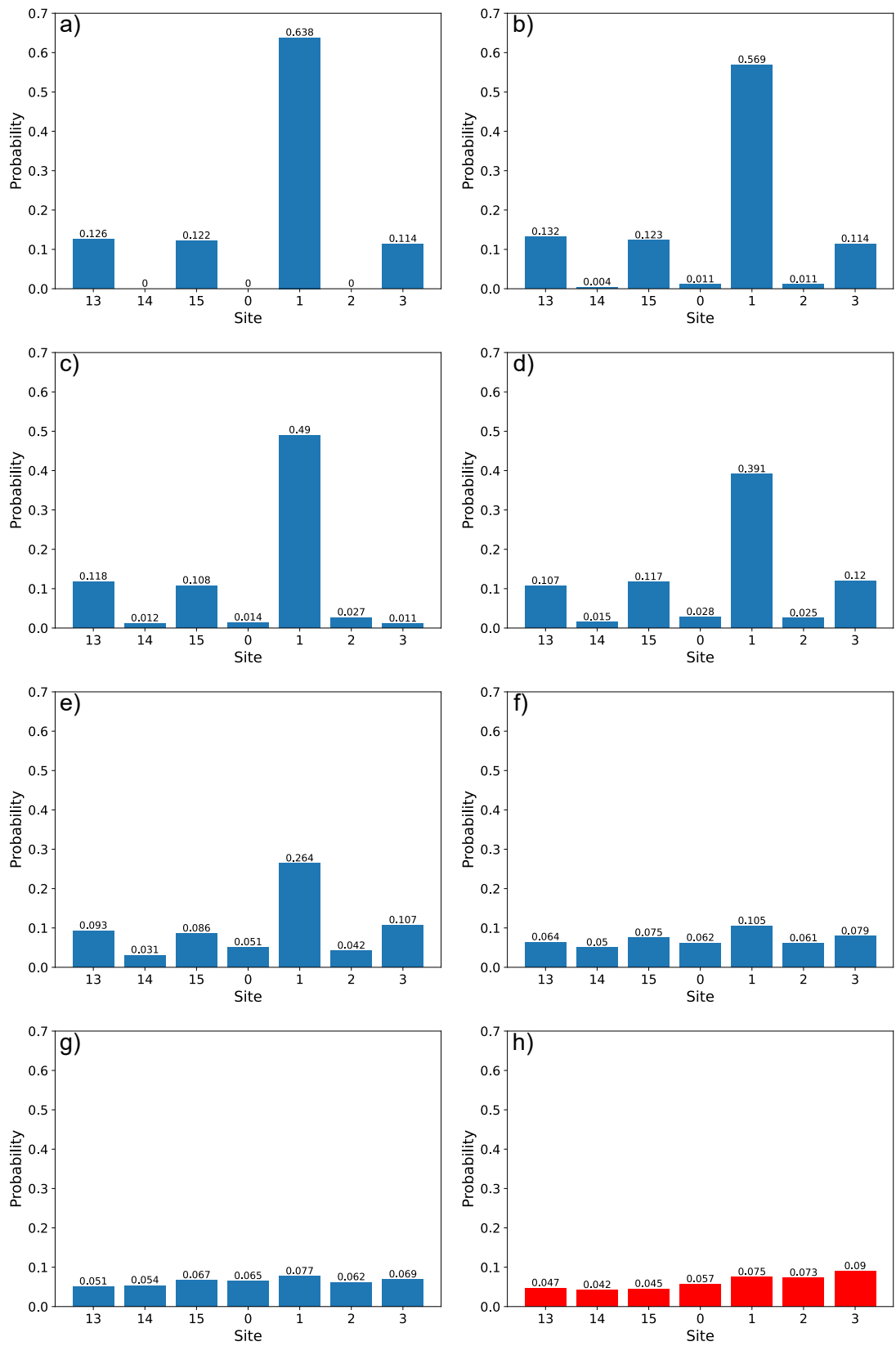


Figure 11.6: Probability distribution for a 3 step DTQW on a 16-node graph. From top left, a) 0%, b) 2%, c) 6%, d) 10%, e) 20% , f) 50%, g) 100% of the full noise strength and h) Santiago (red).

Table 11.1 shows the fidelities of the different noise strengths/devices against the experimental data Figure 11.4.

Noise strength/Device	Fidelity
Simulator	0.96
0%	0.96
2%	0.99
6%	0.92
10%	0.97
20%	0.91
50%	0.73
100%	0.67
IBM Noise Model	0.68
Santiago	0.62

Table 11.1: Fidelities for varying noise strengths including the `qasm.simulator` and `ibmq.santiago`.

As a reminder, the (Hellinger) fidelity is bounded from 0 to 1. A value of 1 indicates perfect agreement between the two distributions. A value of 0 indicates perfect anti-correlation between the two distributions. Therefore, a value of 0.5 indicates no correlation at all.

From Table 11.1 it is clear to see that the 2% custom noise model most closely resembles the experimental quantum walk, to a very high degree. This is followed by 0% noise strength and the simulator, then 10% noise strength. The fact that 10% noise strength is closer to the experimental data than 6% implies a possible anomalous result for 6%. This can be seen in Figure 11.6 c) the probability for site 3 is lower than would be expected. The general trend would suggest, the more you decrease the noise the more the distribution resembles experimental data, up to 2% noise strength, clearly the experiment contains noise and the simulator and 0% do not. In a similar fashion, 100% noise strength has a fidelity closer to the real device than the IBM Noise Model, which suggests the IBM noise model is a less accurate model of the noise on Santiago than the custom noise model. The IBM noise model has a fidelity of 0.92 compared with Santiago, and the custom noise model at 100% strength has a fidelity of 0.94, compared with Santiago. This result is discussed further in the conclusion. By far the worse performing is the real device, Santiago, with a fidelity of 0.62. Below a fidelity of approximately 0.7, the results of the execution become indistinguishable from noise. Again, this can be verified by examining Figure 11.6. This further reinforces the conclusion that the result of this execution, on `ibmq_santiago`, is just noise.

Naturally the question to pose, the one that motivated this section is: can noise from experimental quantum walks be used to model noise in simulated quantum walks (on a real quantum computer)? Analysis of [65] suggests, that in the cases examined, noise simulation of real devices does not work well. Somewhat ironically, noise levels on current devices are much greater than were observed in trapped ions in [65] so this does not give an accurate representation of the noise experienced on this current generation of IBM transmons, and possibly will not for quite some time. However if there is a reduction in current noise levels, e.g. a reduction of 98%, then this experimental paradigm would model superconducting noise very accurately. So, an encoding of an experimental quantum walk in Fock space, that at first glance appears to be far different from that of the encoding of a circuit model, simulated quantum walk, could potentially be used as a noise model between the two systems further in time.

Chapter 12

Conclusions and Further Work

In summary, this work has presented the theoretical connection between discrete-time quantum walks and quantum field theory in the form of the Dirac equation/field. From there, a series of discrete-time quantum walks have been executed on quantum computers as well as simulations of quantum computers, to ascertain the feasibility of simulating quantum field theories, at this moment in time. In addition, the noise levels in these devices has also been examined, and an estimate for the required amount of reduction in these noise levels, for adequately simulating a discrete-time quantum walk, has been made. Finally, a comparison between the noise in quantum computers and experimental implementations of quantum walks has been made to see if the experiments exhibit a similar level/type of noise as the superconducting quantum computers provided by IBM.

In conclusion, it is not currently possible to simulate a quantum field theory, on an IBM digital quantum computer, using quantum walks. Even for a very simplistic model, this is not possible let alone for a full quantum field theory such as an $SU(2) \times U(1)$. More specifically, it is not possible to simulate large, as in containing many sites and steps, quantum walks, that form the basis for this work. Currently, the only simulations that can be run effectively are for simple quantum walks that do not contain many sites and/or steps. Therefore, it is concluded that this current generation of superconducting NISQ computers, provided by IBM, are unsuitable for simulation of quantum field theories using discrete-time quantum walks. Due to current levels of noise associated with each device. However, in general, this method of using quantum walks still has merit. As it avoids known computationally intensive methods such as Trotterisation, that similarly all suffer from noise in quantum devices. So if current noise levels are reduced in the IBM quantum computers, admittedly by a significant amount, then this method would be effective in simulating quantum field theories.

In this work, a particular encoding of a quantum walk has been used. Namely, a binary encoding used to create a DTQW on a cycle graph. It is worth mentioning that this method has a potential major drawback that occurs in the presence of bit-flip errors, and one that can easily ruin a set of results. Consider the 8-node DTQW and the position $|010\rangle$ i.e. site number 2. Imagine a bit-flip error occurs on qubit 1, resulting in the state $|110\rangle$, i.e. site

number 6. This is clearly many sites away from the expected position, practically on the opposite site of the graph. Whilst this error may be easy to spot for a small number of steps, for a large number of steps this task may prove difficult, especially when trying to ascertain an accurate probability amplitude. This effect would be further compounded by multiple bit-flip errors occurring over many steps. Fortunately, the results presented in the previous chapters do not indicate the presence of bit-flip errors (discussed next), although the DTQWs have not been implemented for a very large number of steps. Therefore, this is a potential source of error that should be taken into account when extending these binary-encoded walks further.

This leads us to the second point of discussion, that is the variations in fidelity depending on time of execution, or in other words, anomalous results. This is discussed in detail APPENDIX A.6.

In this work, careful consideration has been taken to present results that do contain such obvious anomalous results, where possible. In practice this has meant repeated execution of quantum circuits on both the real devices and simulator. The simulator also being susceptible to possible anomalous results, as suggested in Chapter 11, Figure 11.6 c). This variation in fidelity and introduction of anomalous results presents a major obstacle for implementing high-fidelity quantum computation, not just for quantum field theory simulation but in general, and needs to be addressed by providers of these services going forward.

Let us now consider the performance of the full-noise strength custom model versus the IBM noise model, and how the two are comparable, with the custom noise model slightly outperforming the latter as in Chapter 11 Section 11.2. This is surprising considering the simplicity of the custom noise model, one possible explanation for the outperformance of the IBM noise model is that the custom noise model was designed specifically for these DTQWs. In other words, the tuning of the numbers that correspond to the gate errors are optimised for this particular set of circuits. Furthermore, the inclusion of 4 and 5-qubit gate errors, in the custom noise model, will also contribute to the increased accuracy in fidelity. However, this is not to say the custom noise model always outperforms the IBM noise model, as this was not the case in Chapter 10 (Custom Noise Models). Perhaps the most interesting point to note regarding this, is the relative simplicity of the IBM noise model, only consisting of depolarising errors and thermal relaxation errors. Furthermore, it lacks multiple (3,4,5 etc.) qubit gate errors as well as other possible error channels. It is possible that IBM intends the user to build on their noise model, and add additional gate errors, error channels, etc. However, this would be a somewhat unusual approach for modelling their own devices. Nevertheless, it can be said that the IBM noise model performs adequately.

Throughout this work, the Hellinger fidelity, using the definition given by IBM, has been used as a benchmark for how accurate a quantum computation is. The qualitative meaning of the fidelity has also been repeated multiple times. Currently, there is no standard definition for fidelity as a measure of accuracy, in other words the exact definition varies from author to author and therefore the qualitative meaning changes. This presents an inherent difficulty when making comparisons between different systems that are using different definitions. Furthermore, by changing the definition of fidelity, a computation can be made to look more or less accurate than it would otherwise be. Say, for example, a relatively high-fidelity (with a specific definition) result, may not necessarily equate to easily human-readable (as in discernable by eye) results. As has been the convention in this work, high-fidelity has implied strong correlation to the ideal result, and in the case of quantum walks this has meant histograms (probability distributions) that appear as would be expected of an accurate depiction of walker position probabilities, in a quantum walk. The issue of fidelity definitions is one with the field of study at large, and in time a suitable standard shall have to be established. However, the results of this work are self-consistent, meaning they are compared to each other, and this has not been an issue, but this is a point that should be noted in general when considering fidelity.

Despite the previous issues regarding fidelity and variations in it, it is clear that the fundamental issues lie with the hardware itself. That is to say, there shall have to be an improvement in this current generation of transmons before further attempts at larger, more sophisticated quantum field theory simulations are made. The topic of superconducting quantum computation is one that is very active, and improvements on current hardware are constantly being made [66,67]. One possible solution involves changing the material transmons are constructed from in order to achieve better coherence times. [68]. Another area of improvement is the read-out of states, and possibly even utilising higher energy levels within the transmon potential [69]. Another proposal is improvement regarding the implementation of quantum circuits [70], this is a particularly pragmatic approach and one that could yield the most immediate results. The rapid development of the qiskit software has already been discussed in Chapter 9, but as has been seen, the hardware has a lot to catch-up on and hence solutions such as alternative circuit implementations and low-level programming offer the most immediate solutions to poor performing quantum computations.

In order for effective quantum field theory simulation to be done, using digital quantum computers, multiple targets will need to be met. Firstly, a reduction in current noise levels of at least, approximately 94%, is needed to efficiently simulate a sufficiently large DTQW. Secondly, a much greater number of qubits within the quantum register of the device will be required, of the order of thousands or even millions, in order to simulate more advanced quantum field theories. As in this paradigm at least, more qubits equates to more sites in the walk and more space-time positions. Finally, with greater qubit numbers, better connectivity shall be required. Currently this is not an issue, however for larger circuits especially with multiple-qubit gates this presents a scaling challenge.

12.1 Further Work

There are many further areas to explore regarding this particular method of quantum field theory simulation. Firstly, is the further development of the theory regarding discrete-time quantum walks to simulate quantum field theories. This work has examined the simplest case in the form of a Dirac field. In order to develop this into a simulation that is more representative of quantum field theory in general, multiple additional factors shall have to be considered. Firstly, is the addition of multiple particles, i.e. walkers in a quantum walk, that is a fundamental ingredient of QFT. As well as the incorporation of fermionic anti-commutation relations and interacting (above quadratic term) theories. Furthermore, a possible next step would be to introduce a simple potential into this system, similar to [44, 71] and find a way of encoding that into a quantum walk.

This leads directly into another area of further research. The encoding of the quantum walk into the quantum computer. Finding another encoding could further expand the geometry of the simulation. Currently, only one space (and time) dimension has been considered, and the difference between a cycle graph and a line can be trivial at times. However, for greater dimensions this geometry (encoding) can become non-trivial and even lead to quantum walks in exotic systems, such as triangular or honeycomb lattices [47, 48], which may have potential applications in subjects such as quantum field theory for condensed matter physics.

Furthermore, the variation in fidelity in time is a topic that requires further study, as there are many factors that remain unknown in the creation of anomalous results. For example, is there a pattern to fluctuations in fidelity, if so, over what time period? In this work it has been implicitly assumed that these fluctuations occur at random. However there is currently not enough data to decisively draw this conclusion. Moreover, do all IBM quantum computers exhibit this fluctuation in fidelity, and by the same amount? IBM has many more backends available that have not been used in this work. This would be a relatively straightforward line of further work that could be implemented in the near future, as the same DTQWs can be used to obtain more data. In addition, there is also the behaviour of the simulator to consider and anomalous results that arise in that system, and crucially why they may occur.

Bibliography

- [1] R. P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 1982.
- [2] D. Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proc. R. Soc. Lond.*, 1985.
- [3] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. *Proceedings: Mathematical and Physical Sciences*, 439(1907):553–558, 1992.
- [4] E. Bernstein and U. Vazirani. Quantum complexity theory. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '93, page 11–20, New York, NY, USA, 1993. Association for Computing Machinery.
- [5] D. Simon. On the power of quantum computation. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 116–123, 1994.
- [6] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, Oct 1997.
- [7] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, page 212–219, New York, NY, USA, 1996. Association for Computing Machinery.
- [8] S. Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, 1996.
- [9] J. W. Pan, K. Mattle, D. Bouwmeester, et al. Experimental quantum teleportation. *Nature*, 390(6660):575–579, Dec 1997.
- [10] D. G. Cory, M. D. Price, W. Maas, E. Knill, R. Laflamme, W. H. Zurek, T. F. Havel, and S. S. Somaroo. Experimental quantum error correction. *Phys. Rev. Lett.*, 81:2152–2155, Sep 1998.
- [11] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang. Experimental realization of shor’s quantum factoring algorithm using nuclear magnetic resonance. *Nature*, 414(6866):883–887, Dec 2001.
- [12] L. Tian, R. Blatt, and P. Zoller. Scalable ion trap quantum computing without moving ions. *The European Physical Journal D*, 32(2):201–208, Nov 2004.

- [13] C. Rigetti, J. M. Gambetta, S. Poletto, B. L. T. Plourde, J. M. Chow, A. D. Córcoles, J. A. Smolin, S. T. Merkel, J. R. Rozen, G. A. Keefe, et al. Superconducting qubit in a waveguide cavity with a coherence time approaching 0.1 ms. *Physical Review B*, 86(10), Sep 2012.
- [14] J. Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, Aug 2018.
- [15] D. Zhang. Exact solution for three-dimensional ising model. *Symmetry*, 13(10), 2021.
- [16] K. G. Wilson. The renormalization group: Critical phenomena and the kondo problem. *Rev. Mod. Phys.*, 47:773–840, Oct 1975.
- [17] K. G. Wilson. Ab Initio Quantum Chemistry: A Source of Ideas for Lattice Gauge Theorists. *Nucl. Phys. B Proc. Suppl.*, 17:82–92, 1990.
- [18] E. Learmann. Recent results from lattice qcd simulations. *Nuclear Physics A*, 610:1–12, 1996. Quark Matter '96.
- [19] S. Aoki, T. Hatsuda, and N. Ishii. Theoretical Foundation of the Nuclear Force in QCD and Its Applications to Central and Tensor Forces in Quenched Lattice QCD Simulations. *Progress of Theoretical Physics*, 123(1):89–128, 01 2010.
- [20] S. Schaefer, R. Sommer, and F. Virotta. Critical slowing down and error analysis in lattice qcd simulations. *Nuclear Physics B*, 845(1):93–119, 2011.
- [21] S. P. Jordan, K. S. Lee, and J. Preskill. Quantum algorithms for quantum field theories. *Science*, 336(6085):1130–1133, 2012.
- [22] K. Marshall, R. Pooser, G. Siopsis, and C. Weedbrook. Quantum simulation of quantum field theory using continuous variables. *Phys. Rev. A*, 92:063825, Dec 2015.
- [23] J. Preskill. Simulating quantum field theory with a quantum computer, 2018.
- [24] A. H. Moosavian and S. Jordan. Faster quantum algorithm to simulate fermionic quantum field theory. *Physical Review A*, 98(1):012332, 2018.
- [25] N. Klco, M. J. Savage, and J. R. Stryker. Su(2) non-abelian gauge field theory in one dimension on digital quantum computers. *Physical Review D*, 101(7), Apr 2020.
- [26] D. Tong. Quantizing the dirac field. *Quantum Field Theory Lecture Notes*, 5, 2006.
- [27] N. Shenvi, J. Kempe, and K. B. Whaley. Quantum random-walk search algorithm. *Physical Review A*, 67(5), May 2003.
- [28] A. Ambainis. Quantum random walks - new method for designing quantum algorithms. In *SOFSEM*, 2008.
- [29] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman. Exponential algorithmic speedup by a quantum walk. *Proceedings of the thirty-fifth ACM symposium on Theory of computing - STOC '03*, 2003.

- [30] A. M. Childs. Universal computation by quantum walk. *Phys. Rev. Lett.*, 102:180501, May 2009.
- [31] N. B. Lovett, S. Cooper, M. Everitt, M. Trevers, and V. Kendon. Universal quantum computation using the discrete-time quantum walk. *Physical Review A*, 81(4), Apr 2010.
- [32] E. W. Montroll and G. H. Weiss. Random walks on lattices. ii. *Journal of Mathematical Physics*, 6(2):167–181, 1965.
- [33] E. Farhi and S. Gutmann. Quantum computation and decision trees. *Physical Review A*, 58(2):915–928, Aug 1998.
- [34] R. P. Feynman and A. R. Hibbs. *Quantum Mechanics and Path Integrals*. McGraw-Hill, emended edition, 2005.
- [35] T. Jacobson and L. Schulman. Quantum stochastics: the passage from a relativistic to a non-relativistic path integral. *Journal of Physics A*, 17:375–383, 1984.
- [36] W. E. Boyce and R. C. Prima. *Elementary Differential Equations and Boundary Value Problems*. John Wiley and Sons, 9th edition, 2008.
- [37] T. A. Jacobson. Spinor chain path integral for the dirac electron. 1983.
- [38] A. Kull and R. Treumann. On the path integral of the relativistic electron. *International Journal of Theoretical Physics*, 38:1423–1428, 1999.
- [39] N. Konno, T. Namiki, and T. Soshi. Symmetry of distribution for the one-dimensional hadamard walk. *Interdisciplinary Information Sciences*, 10(1):11–22, 2004.
- [40] N. Konno. Quantum random walks in one dimension. *Quantum Information Processing*, 1(5):345–354, 2002.
- [41] P. L. Knight, E. Roldán, and J. E. Sipe. Quantum walk on the line as an interference phenomenon. *Physical Review A*, 68(2), Aug 2003.
- [42] P. Blanchard and M.-O. Hongler. Quantum random walks and piecewise deterministic evolutions. *Phys. Rev. Lett.*, 92:120601, Mar 2004.
- [43] C. M. Chandrashekar, S. Banerjee, and R. Srikanth. Relationship between quantum walks and relativistic quantum mechanics. *Phys. Rev. A*, 81:062340, Jun 2010.
- [44] G. Di Molfetta, M. Brachet, and F. Debbasch. Quantum walks in artificial electric and gravitational fields. *Physica A: Statistical Mechanics and its Applications*, 397:157–168, Mar 2014.
- [45] G. Di Molfetta, L. Honter, B. B. Luo, T. Wada, and Y. Shikano. Massless dirac equation from fibonacci discrete-time quantum walk. *Quantum Studies: Mathematics and Foundations*, 2(3):243–252, Apr 2015.

- [46] P. Arnault, G. Di Molfetta, M. Brachet, and F. Debbasch. Quantum walks and non-abelian discrete gauge theory. *Physical Review A*, 94(1), Jul 2016.
- [47] P. Arrighi, G. Di Molfetta, I. Márquez-Martín, and A. Pérez. Dirac equation as a quantum walk over the honeycomb and triangular lattices. *Physical Review A*, 97(6), Jun 2018.
- [48] P. Arrighi, G. D. Molfetta, I. Márquez-Martín, and A. Pérez. From curved spacetime to spacetime-dependent local unitaries over the honeycomb and triangular quantum walks, 2018.
- [49] G. di Molfetta and F. Debbasch. Discrete-time quantum walks: Continuous limit and symmetries. *Journal of Mathematical Physics*, 53(12):123302, Dec 2012.
- [50] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 10th edition, 2010.
- [51] J. Koch, T. M. Yu, J. Gambetta, A. A. Houck, D. I. Schuster, J. Majer, A. Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf. Charge-insensitive qubit design derived from the cooper pair box. *Phys. Rev. A*, 76:042319, Oct 2007.
- [52] W. Oliver and P. Welander. Materials in superconducting quantum bits. *Mrs Bulletin*, 38:816–825, 2013.
- [53] A. Blais, J. Gambetta, A. Wallraff, D. I. Schuster, S. M. Girvin, M. H. Devoret, and R. J. Schoelkopf. Quantum-information processing with circuit quantum electrodynamics. *Phys. Rev. A*, 75:032329, Mar 2007.
- [54] <https://qiskit.org/textbook/ch-quantum-hardware/calibrating-qubits-pulse.html>, Accessed: 29-05-2021.
- [55] F. Yan. *Coherence Characterization with a Superconducting Flux Qubit through NMR Approaches*. PhD thesis, Massachusetts Institute of Technology, 6 2013.
- [56] E. L. Hahn. Spin echoes. *Phys. Rev.*, 80:580–594, Nov 1950.
- [57] https://qiskit.org/documentation/tutorials/simulators/1.aer_provider.html, Accessed: 26-08-2021.
- [58] https://github.com/qiskit-community/qiskit-community-tutorials/blob/master/terra/qis_adv/quantum_walk.ipynb, Accessed: 25-04-2021.
- [59] B. L. Douglas and J. B. Wang. Efficient quantum circuit implementation of quantum walks, 2009.
- [60] https://qiskit.org/documentation/stubs/qiskit.quantum_info.hellinger_fidelity.htm, Accessed: 25-06-2021.

- [61] <https://qiskit.org/documentation/apidoc/transpiler.html>,
Accessed: 25-04-2021
- [62] <https://qiskit.org/documentation/stubs/qiskit.providers.aer.noise.NoiseModel.html>
`#qiskit.providers.aer.noise.NoiseModel.from_backend`,
Accessed: 24-08-2021.
- [63] H. Xiao-Chuan, F. Lan-Tian, L. Yu-Xuan, Z. Lan-Xuan, S. Jung-Feng, and Z. Yong-Sheng. Experimental observations of 1d quantum walks in a limited region. *Quantum Information Processing*, 18, 2019.
- [64] A. Politi, M. J. Cryan, J. G. Rarity, S. Yu, and J. L. O'Brien. Silica-on-silicon waveguide quantum circuits. *Science*, 320(5876):646–649, May 2008.
- [65] R. Matjeschk, C. Schneider, M. Enderlein, T. Huber, H. Schmitz, J. Glueckert, and T. Schaetz. Experimental simulation and limitations of quantum walks with trapped ions. *New Journal of Physics*, 14:035012, 2012.
- [66] M. Ware. *Flux-tunable superconducting transmons for quantum information processing*. PhD thesis, Syracuse University, 5 2015.
- [67] H.-L. Huang, D. Wu, D. Fan, and X. Zhu. Superconducting quantum computing: A review, 2020.
- [68] A. P. M. Place, L. V. H. Rodgers, P. Mundada, B. M. Smitham, M. Fitzpatrick, Z. Leng, A. Premkumar, J. Bryon, A. Vrajitoarea, S. Sussman, and et al. New material platform for superconducting transmon qubits with coherence times exceeding 0.3 milliseconds. *Nature Communications*, 12(1), Mar 2021.
- [69] C. Wang, M.-C. Chen, C.-Y. Lu, and J.-W. Pan. Optimal readout of superconducting qubits exploiting high-level states. *Fundamental Research*, 1(1):16–21, 2021.
- [70] E. Wilson, F. Mueller, L. Bassman, and C. Iancu. Empirical evaluation of circuit approximations on noisy quantum devices, 2021.
- [71] G. D. Molfetta and F. Debbasch. Discrete-time quantum walks in random artificial gauge fields, 2015.

Appendix

A.0 Spinors

A spinor is a 4-component column matrix. Crucially, it is *not* a vector and does not behave like one.

Consider the following analogy in order to develop some intuition about (2-d) spinors. A vector represents the magnitude and direction of some quantity. It can be represented on the Cartesian plane by an arrow pointing in some direction. By extension, a spinor can be thought of as a flag pole and the orientation of the flag on the pole is equivalent to the direction component of a vector.

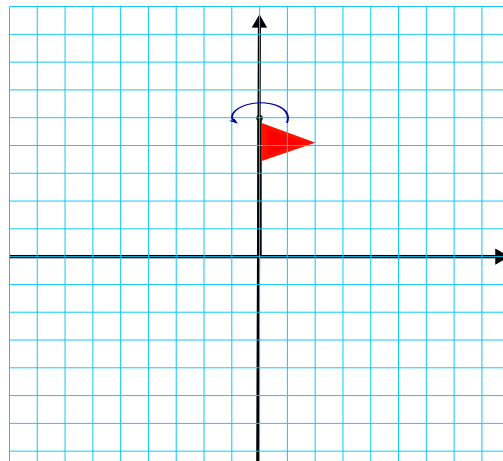


Figure A.1: Illustration of a flag in a 2-d plane, used as a loose analogy for a spinor. The flag itself can rotate about the pole (indicated by the dark blue arrow) and the pole itself can change direction.

This orientation of the flag is the key property of a spinor that makes it such a useful construct. Naturally, this means it lends itself well to describe spin-1/2 particles.

Spinors have some unusual properties, one of which being that in order for it to return to its original position, a spinor must undergo a rotation of 720° (as opposed to 360°).

Technically speaking, there is a slight distinction between Dirac spinors and bi-spinors. However, for the purposes of this text the terms are used interchangeably. The strict definition is: a Dirac spinor is a bi-spinor in the Dirac convention for the γ matrices.

A.1 Feynman Path Integral and Propagator

Here, the connection between the propagator and Feynman path integral shall be motivated.

Consider the expectation value of some operator \hat{A} in quantum mechanics

$$\langle \psi | \hat{A} | \psi \rangle. \quad (\text{A.1})$$

This can be generalised to include time-dependence in the form of a transition amplitude

$$\langle q_f | \hat{U}(t_f, t_i) | q_i \rangle \text{ with } t_f > t_i, \quad (\text{A.2})$$

where $\hat{U}(t_f, t_i)$ is the time evolution operator, q_f and q_i are the final and initial position states respectively. It is this transition amplitude that is also known as the propagator or Feynman's kernel.

The modulus squared of the propagator gives the probability of finding the particle in question at position q_f , at a final time t_f , given that it started at a position q_i at time t_i . So it is a useful quantity to know - it contains the dynamics of the system.

Consider calculating the transition amplitude for a state $|\psi_i\rangle$ at time t_i that evolves to some state $\langle \psi_f |$ at time t_f . Re-iterating that $t_f > t_i$.

$$\langle \psi_f | \hat{U}(t_f, t_i) | \psi_i \rangle = \int dq_i dq_f \langle \psi_f | q_f \rangle \langle q_f | \hat{U}(t_f, t_i) | q_i \rangle \langle q_i | \psi_i \rangle. \quad (\text{A.3})$$

Identifying that $\langle \psi_f | q_f \rangle = \psi^*(q_f)$ and $\langle q_i | \psi_i \rangle = \psi(q_i)$.

Feynman's path integral method shall now be used to compute this quantity.

In quantum mechanics, to find the amplitude of a given outcome, we sum over all ways a process can take place. As a qualitative description of the path integral method, consider the famous double-slit experiment.

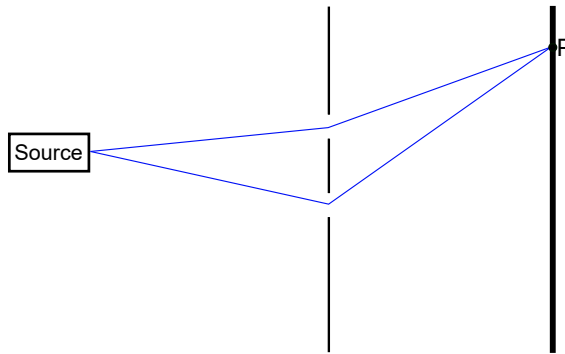


Figure A.2: Double-slit experiment set-up. A source produces identical particles that travel through a screen with two slits, the particles can travel through either of the slits and arrive at some point P on the final screen which is a detector.

Figure A.2 shows a basic version of the double-slit experiment. When the particles encounter the slits their path changes until they arrive at some final position on the detector. As a thought experiment, an additional screen with slits could be added, and another one, and another. There could be many, or even an infinite number of screens added with many or an infinite number of slits, between the source and the detector. Figure A.3 illustrates this.

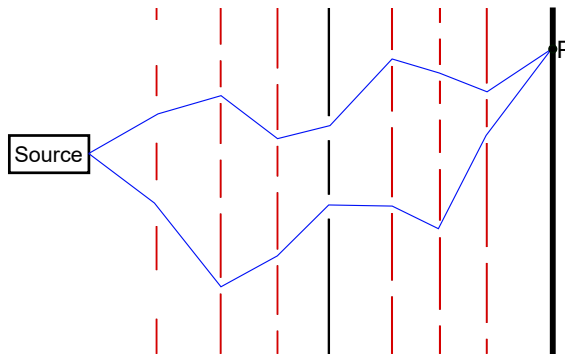


Figure A.3: Double-slit experiment with additional screens and slits (red).

As can be seen the path for the particles now changes as there are more slits to encounter. This addition of screens/slits is in essence the principle of a Feynman path integral. By adding more of them, this acts as a form of discretisation for the particle's path, where you are summing over all paths the particle can take. The more you add, the more the sum represents an integral and thus the more you refine your approximation of the possible paths of the particle.

For our purposes, a time-independent Hamiltonian shall be used to compute the propagator, i.e.

$$\hat{U}(t_1, t_2) = e^{-i\frac{\hat{H}}{\hbar}(t_1-t_2)}. \quad (\text{A.4})$$

The first step is to slice time into infinitesimal intervals:

$$\hat{U}(t_2, t_3)\hat{U}(t_3, t_4) = \hat{U}(t_2, t_4) \quad (\text{A.5})$$

from causality.

Equation (A.2) becomes

$$\langle q_f | \hat{U}(t_f, t_i) | q_i \rangle = \langle q_f | \hat{U}(t_f, t_N) \dots \hat{U}(t_2, t_1) \hat{U}(t_1, t_2) | q_i \rangle. \quad (\text{A.6})$$

The time intervals are taken to be ϵ :

$$t_n - t_{n-1} = \epsilon = \frac{t_f - t_i}{N-1} > 0. \quad (\text{A.7})$$

Inserting a complete set of states

$$\int_{-\infty}^{+\infty} dq_n |q_n\rangle \langle q_n| = 1 \quad n = 1, \dots, N \quad (\text{A.8})$$

between each pair of \hat{U} s.

$$\langle q_f | \hat{U}(t_f, t_i) | q_i \rangle = \left[\prod_{n=1}^N \int_{-\infty}^{+\infty} dq_n \right] \langle q_f | \hat{U}(t_f, t_N) | q_N \rangle \langle q_N | \dots | q_2 \rangle \langle q_2 | \hat{U}(t_2, t_1) | q_1 \rangle \langle q_1 | \hat{U}(t_1, t_i) | q_i \rangle. \quad (\text{A.9})$$

The aim is to compute the infinitesimal amplitude,

$$\langle q_{n+1} | \hat{U}(t_{n+1} = t_n + \epsilon, t_n) | q_n \rangle = \langle q_{n+1} | e^{-i\frac{\hat{H}}{\hbar}\epsilon} | q_n \rangle. \quad (\text{A.10})$$

Let us assume the Hamiltonian takes a standard form with the kinetic term $K(\hat{p})$ dependent only on the momentum and the potential term $V(\hat{q})$ dependent only on the position,

$$\hat{H} = K(\hat{p}) + V(\hat{q}). \quad (\text{A.11})$$

The exponentials can be split using the Campbell-Baker-Hausdorff formula, such that

$$e^{-\frac{i(K(\hat{p})+V(\hat{q}))}{\hbar}\epsilon} = e^{-\frac{iK(\hat{p})}{\hbar}\epsilon} e^{-\frac{iV(\hat{q})}{\hbar}\epsilon} e^{\frac{\epsilon^2[K(\hat{p}, V(\hat{q}))]}{2\hbar^2}} + \dots \quad (\text{A.12})$$

taking ϵ to 0, sub-leading terms are ignored.

There are now two exponentials that act on the states, since the kinetic terms depend on \hat{p} ,

a complete set of momentum eigenstates are introduced.

$$\begin{aligned} \langle q_{n+1} | \hat{U}(t_{n+1} = t_n + \epsilon, t_n) | q_n \rangle &= \int_{-\infty}^{+\infty} dp_{n+1} \langle q_{n+1} | p_{n+1} \rangle \langle p_{n+1} | e^{-\frac{iK(\hat{p})}{\hbar}\epsilon} e^{-\frac{iV(\hat{q})}{\hbar}\epsilon} | q_n \rangle \\ &= \int_{-\infty}^{+\infty} dp_{n+1} \langle q_{n+1} | p_{n+1} \rangle \langle p_{n+1} | e^{-\frac{iK(p_{n+1})}{\hbar}\epsilon} e^{-\frac{iV(q_n)}{\hbar}\epsilon} \end{aligned} \quad (\text{A.13})$$

$$\int_{-\infty}^{+\infty} \frac{dp_{n+1}}{2\pi\hbar} \exp\left(-\frac{i\epsilon}{\hbar} (K(p_{n+1}) + V(q_n)) + \frac{i\epsilon}{\hbar} p_{n+1} \frac{q_{n+1} - q_n}{\epsilon}\right) \quad (\text{A.14})$$

where the term in the middle, $(K(p_{n+1}) + V(q_n))$ is identified with $H(q_n, p_{n+1})$.

Defining $q_0 = q_i$ and $q_{N+1} = q_f$ and plugging into (A.9)

$$\langle q_f | \hat{U}(t_f, t_i) | q_i \rangle = \int \left[\prod_{n=1}^N dq_n \right] \left[\prod_{n=1}^{N+1} \frac{dp_n}{2\pi\hbar} \right] \exp\left(\sum_{n=0}^N \frac{iH(q_n, p_{n+1})}{\hbar} \epsilon + \frac{i\epsilon}{\hbar} \frac{q_{n+1} - q_n}{\epsilon} p_{n+1}\right). \quad (\text{A.15})$$

Taking $N \rightarrow \infty$ and therefore $\epsilon \rightarrow 0$ gives the Feynman path integral in phase space

$$\langle q_f | \hat{U}(t_f, t_i) | q_i \rangle = \int_{q(t_i)=q_i, q(t_f)=q_f} \mathcal{D}'q \mathcal{D}p e^{-\frac{i}{\hbar} \int_{t_i}^{t_f} (H(q,p) - pq) dt} \quad (\text{A.16})$$

where $\mathcal{D}'q \mathcal{D}p$ is the measure,

$$\left[\prod_{n=1}^N dq_n \right] \left[\prod_{n=1}^{N+1} \frac{dp_n}{2\pi\hbar} \right]$$

with the prime meaning there is one less q integral than p .

In order to get the path integral in position space, assume a standard kinetic term

$$K(\hat{p}) = \frac{\hat{p}^2}{2m}. \quad (\text{A.17})$$

Then the p integral in equation (A.14) is Gaussian

$$\int_{-\infty}^{+\infty} \frac{dp_{n+1}}{2\pi\hbar} e^{-\frac{i\epsilon}{\hbar} \left(\frac{p_{n+1}^2}{2m} + V(q_n) - p_{n+1} \frac{q_{n+1} - q_n}{\epsilon} \right)} \quad (\text{A.18})$$

$$= \int_{-\infty}^{+\infty} \frac{dp'_{n+1}}{2\pi\hbar} e^{-\frac{i\epsilon}{\hbar} \left(\frac{(p'_{n+1})^2}{2m} + V(q_n) - m \frac{(q_{n+1} - q_n)^2}{2\epsilon^2} \right)} \quad (\text{A.19})$$

$$= \sqrt{\frac{m}{2\pi i \hbar \epsilon}} e^{-\frac{i\epsilon}{\hbar} \left(V(q) - m \frac{(q_{n+1} - q_n)^2}{2\epsilon^2} \right)}. \quad (\text{A.20})$$

This gives,

$$\langle q_f | \hat{U}(t_f, t_i) | q_i \rangle = \left[\prod_{n=1}^N dq_n \sqrt{\frac{m}{2\pi i \hbar \epsilon}} \right] \sqrt{\frac{m}{2\pi i \hbar \epsilon}} \exp \left(\sum_{n=0}^N \frac{i\epsilon}{\hbar} \left(\frac{m}{2} \left(\frac{q_{n+1} - q_n}{\epsilon} \right)^2 - V(q) \right) \right) \quad (\text{A.21})$$

where the exponent is the action $S(q)$ as $N \rightarrow \infty$ ($\epsilon = 0$).

$$S(q) = \int_{t_i}^{t_f} dt L(q, \dot{q}) = \int_{t_i}^{t_f} dt \left(\frac{m}{2} \dot{q}^2 - V(q) \right), \quad (\text{A.22})$$

L is the Lagrangian.

This gives the Feynman path integral in configuration space¹

$$\langle q_f | \hat{U}(t_f, t_i) | q_i \rangle = \int_{q(t_i)=q_i, q(t_f)=q_f} \mathcal{D}q e^{\frac{i}{\hbar} S(q)} \quad (\text{A.23})$$

where again, $\mathcal{D}q$ is the measure

$$\mathcal{D}q = \left[\prod_{n=1}^N dq_n \sqrt{\frac{m}{2\pi i \hbar \epsilon}} \right] \sqrt{\frac{m}{2\pi i \hbar \epsilon}}. \quad (\text{A.24})$$

Thus the propagator (transition amplitude) is obtained as a Feynman path integral.

¹A configuration space is the vector space of the generalised coordinates of a system.

A.2 Device Topologies

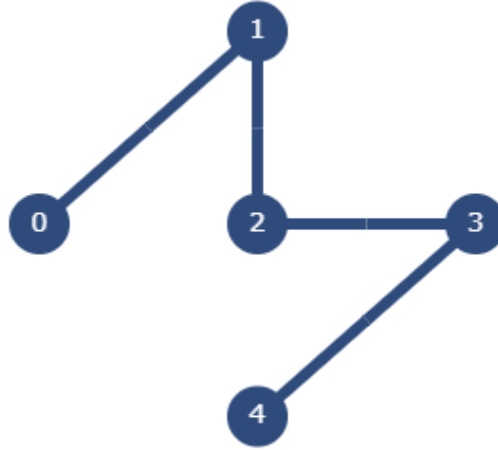


Figure A.4: Topology (connectivity) of qubits in `ibmq_santiago`. This has the same topology as a straight line.

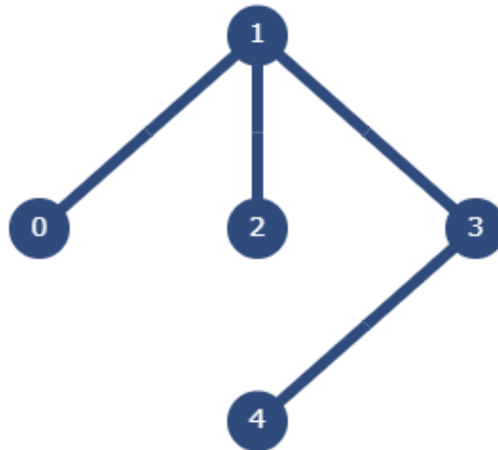


Figure A.5: Topology (connectivity) of qubits in `ibmq_quito`. This has the same topology as a “T” shape.

A.3 8-node Quantum Walk Circuit

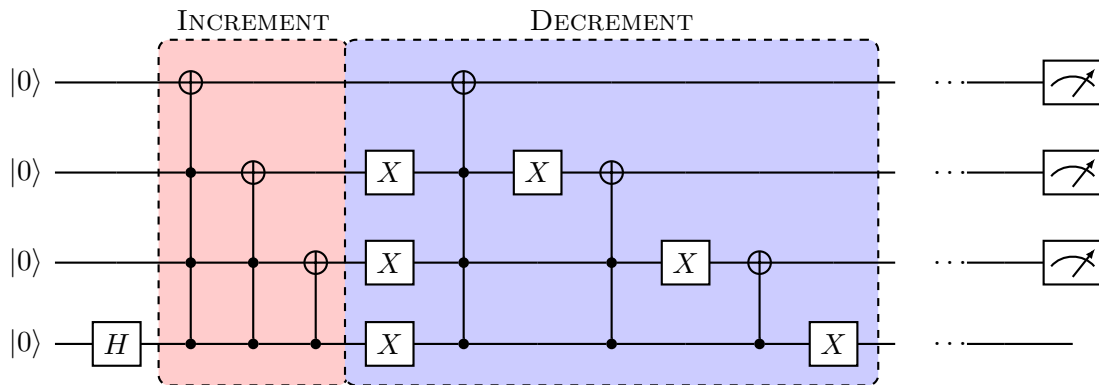


Figure A.6: Quantum circuit of [58]. Figure depicts one step of the walk. Increment and Decrement gate sets are repeated for the desired number of steps and qubits 1-3 are used to determine the position by evaluating the value of the register.

A.4 16-node Quantum Walk Circuit

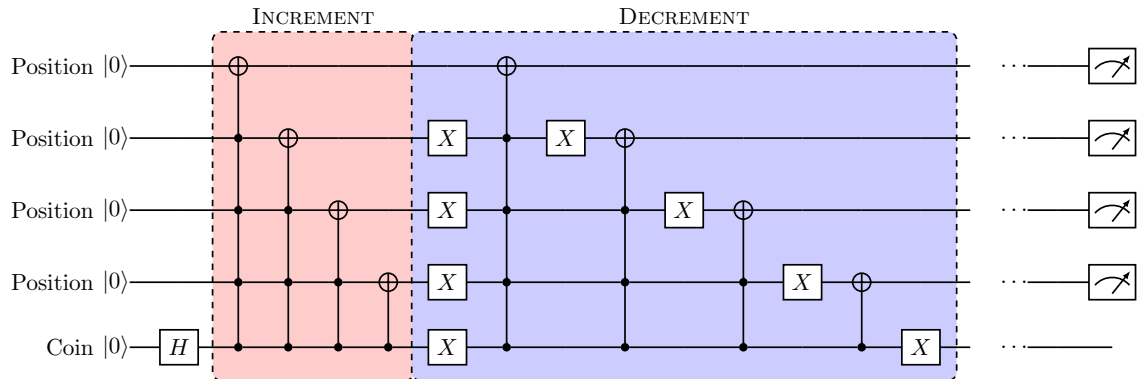


Figure A.7: Quantum circuit for the 16-node DTQW using 5 qubits. Figure depicts one step of the walk. Qubits 1-4 are used to evaluate the position of the walker and qubit 5 acts as the coin.

A.5 Fock states and Coherent states

A Fock state is a representation for a state with a well-defined but variable number of particles. Fock states inhabit a Fock space.

In layman's terms, a Fock state can be motivated by the following. Ordinarily, when considering a system in Hilbert space, the state of each particle is considered individually, i.e. *what state is this particle in?* Whereas in Fock space, the suitable question to ask is *how many particles are in a particular state?* This becomes a useful rephrasing when dealing with many and a varying number of particles within a given system.

For a more rigorous description, let us consider a quantised Hamiltonian describing an electromagnetic field in the mode l .

$$\hat{H}_l = \hbar\omega_l \left(\hat{a}_l \hat{a}_l^\dagger + \frac{1}{2} \right) \quad (\text{A.25})$$

with

$$[\hat{a}_l, \hat{a}_l^\dagger] = 1 \quad (\text{A.26})$$

where \hat{a}_l and \hat{a}_l^\dagger are the annihilation and creation operators respectively.

The Schrödinger equation for this system is

$$\hat{H}_l |\phi_{n_l}\rangle = E_{n_l} |\phi_{n_l}\rangle. \quad (\text{A.27})$$

The index n_l characterises each eigensolution.

This system is simply a quantum harmonic oscillator (QHO) and so the solutions (eigenvalues, eigenvectors etc.) are well-known.

The eigenvalues of \hat{H} are

$$E_{n_l} = \hbar\omega_l \left(n_l + \frac{1}{2} \right) \quad \text{with } n_l = 0, 1, 2, \dots \quad (\text{A.28})$$

The eigenstates are related by

$$\hat{a}_l |\phi_{n_l}\rangle = \sqrt{n_l} |\phi_{n_l-1}\rangle, \quad (\text{A.29})$$

$$\hat{a}_l^\dagger |\phi_{n_l}\rangle = \sqrt{n_l + 1} |\phi_{n_l+1}\rangle. \quad (\text{A.30})$$

The lowest energy state, i.e. the ground state must satisfy

$$\hat{a}_l |\phi_{n_l=0}\rangle = 0. \quad (\text{A.31})$$

With this information it is possible to generate all other states. Using the above relation and the action of the annihilation operator on the eigenstate of $n_l = 0$, as well as normalising gives

$$|\phi_{n_l}\rangle = \frac{(\hat{a}_l^\dagger)^{n_l}}{\sqrt{n_l!}} |\phi_{n_l=0}\rangle. \quad (\text{A.32})$$

In other words, we have obtained a complete basis for the state-space.

Note that eigenstates of the Hamiltonian are also eigenstates of the number operator \hat{N} , where

$$\hat{N}_l \doteq \hat{a}_l^\dagger \hat{a}_l \quad (\text{A.33})$$

and

$$\hat{N}_l |\phi_{n_l}\rangle = n_l |\phi_{n_l}\rangle. \quad (\text{A.34})$$

The states $|\phi_{n_l}\rangle$ are often denoted as $|n_l\rangle$ and are known as Fock states or number states.

The Fock state $|n_l\rangle$ has a well-defined energy which is given by

$$E_{n_l} = \hbar\omega_l \left(n_l + \frac{1}{2} \right), \quad (\text{A.35})$$

$$E_0 = \frac{\hbar\omega_l}{2}, \quad (\text{A.36})$$

$$E_{n_l} - E_0 = n_l \hbar\omega_l. \quad (\text{A.37})$$

A coherent state is the quantum mechanical equivalent of a classical monochromatic electromagnetic wave. In the number state representation, coherent states are given by

$$|\alpha\rangle = e^{-|\alpha|^2/2} \sum_{n=0}^{\infty} \frac{\alpha^n}{\sqrt{n!}} |n\rangle \quad (\text{A.38})$$

where $|n\rangle$ are the Fock states and α is a dimensionless complex number that specifies the complex field amplitude in photon number units. Coherent states are not orthogonal to each other nor are they eigenstates of the Hamiltonian.

A.6 Fidelity over 8 days in August 2021

Due to the previously mentioned sensitivity of transmons to environmental factors (see Chapter 7), this can lead to variations in the amount of noise and even errors occurring during certain executions. Figure A.8 in shows the fidelity for an 8 step, 16-node DTQW, run on `ibmq_santiago` (against the `qasm_simulator`) for 8 days in August 2021. As can be seen, the fidelity is relatively constant for most days, with the error bars overlapping with the average fidelity (0.35) for 5 out of the 8 days, with the final day (Tuesday 10/08/21) being slightly above this average. However, there are two notable exceptions, Wednesday 04/08/21, where the fidelity has a large decrease in magnitude to a value of 0.19, and Thursday 05/08/21 where the fidelity is considerably higher than other days, with a value of 0.43. However, recall the meaning of the fidelity for this work, a value of 0.5 shows no correlation and a value of 0 indicates anti-correlation. So, whilst it has been established that the backend performs poorly, the interesting case is for the fidelity on Wednesday 04/08/21. Figure A.9 in shows the results for each day presented as histograms, panel b) shows the result for Wednesday 04/08/21. It is clear to see this drop in fidelity when compared with the other panes. The interesting effect to note is the clustering of states on this day. This is the cause of the slightly anti-correlated fidelity and the underlying physical cause is assumed to be (multiple) bit-flip errors, as mentioned in the conclusion. However, this is not to say there could not be other sources of errors that have yet to be explored and lie beyond the scope of this current work. For example, the effect of variations in temperature inside the dilution refrigerators (that house the quantum computers), on results such as these. Furthermore, this effect may have occurred on other days, such as Sunday 08/08/21 and Monday 09/08/21, but it is hard to conclusively discern this clustering from the random noise, particularly for the latter day.

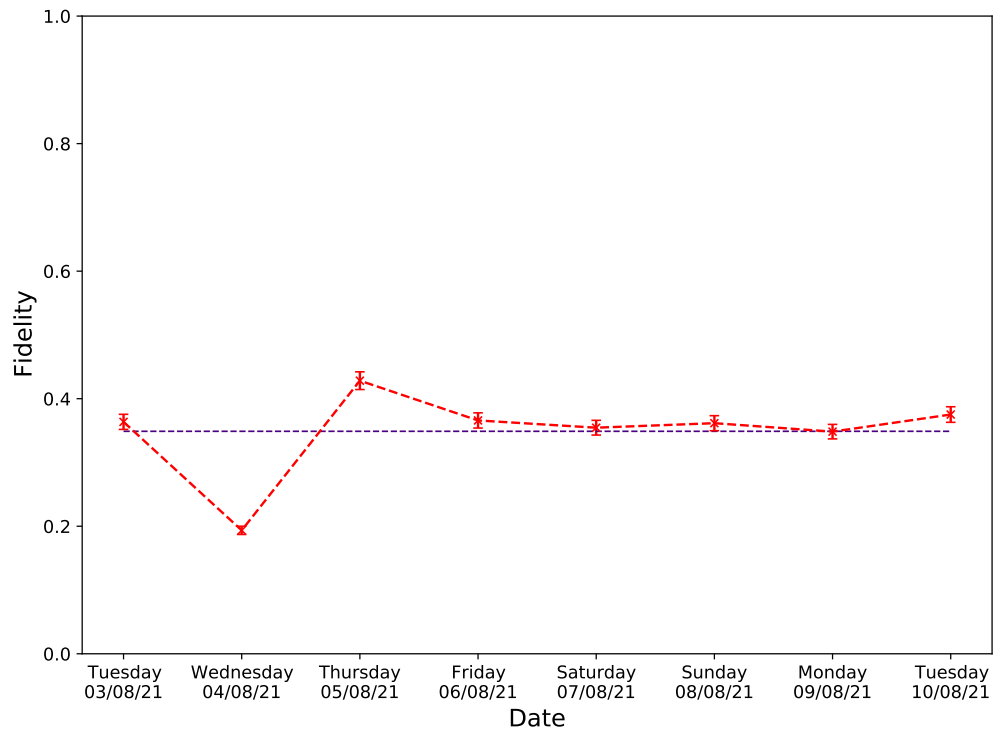


Figure A.8: Fidelity for 8 step, 16-node DTQW (red) run over 8 days in August 2021. Magenta dashed-line indicates the average fidelity over those 8 days. Error bars represent standard error. Dates are formatted dd/mm/yy.

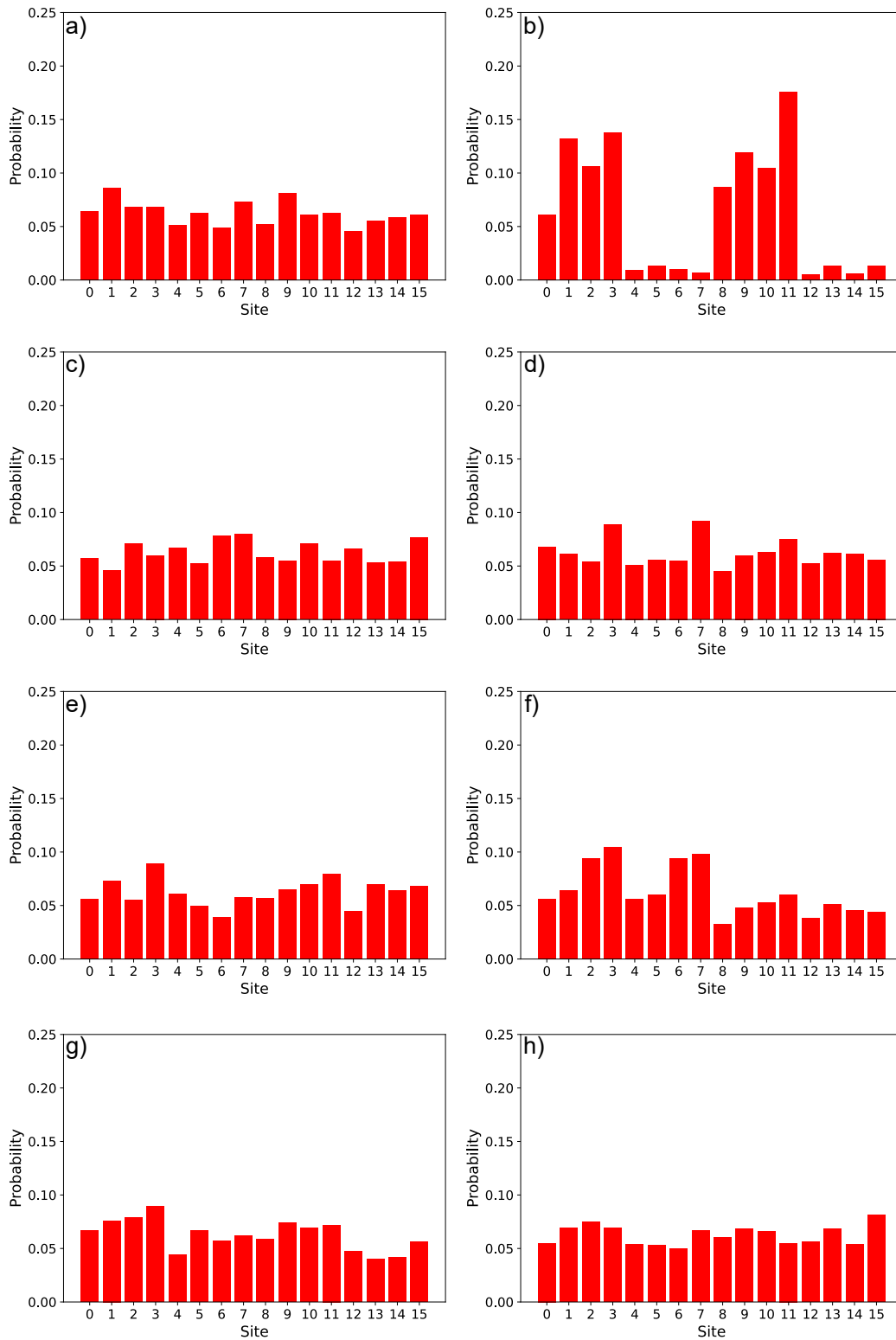


Figure A.9: Corresponding probability distributions for the 8 step, 16-node DTQW run over 8 days in August 2021, run on Santiago. Panels a) to h) correspond to dates 3 to 10 of August 2021 respectively.