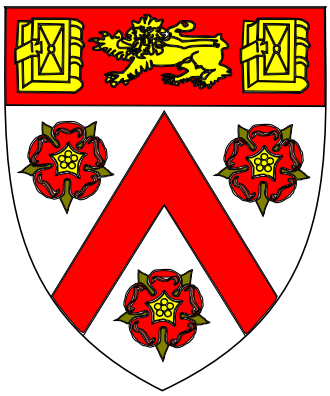
Topics in Computing with Quantum Oracles and Higher-Dimensional Many-Body Systems

Imdad Sajjad Badruddin Sardharwalla
Trinity College, University of Cambridge



June 2017

This dissertation is submitted for the degree of
Doctor of Philosophy

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text.

It is not substantially the same as any that I have submitted, or is being concurrently submitted, for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my dissertation has already been submitted, or is being concurrently submitted, for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text.

It does not exceed the prescribed word limit for the relevant Degree Committee.

Preface

The material of this thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text or detailed below.

The work in Sections 2.3–2.5 and Chapter 3 was done jointly with T. Cubitt, A. Harrow and N. Linden, resulting in a paper under review in *Phys. Rev. Lett. (arXiv:1602.07963 [quant-ph])*. T. Cubitt, A. Harrow and N. Linden conceived the idea for qubit operators (Section 2.3) and the application to the Solovay-Kitaev Theorem (Chapter 3), and were somewhat involved in writing the manuscript. I performed all of the calculations for the paper and independently developed the generalisation in Section 2.4.

The work in Chapters 5 and 6 and Appendix A was done jointly with S. Strelchuk and R. Jozsa, resulting in a paper published in *QIC Vol.17 No.7&8 (2017) (arXiv:1609.01600 [quant-ph])*. S. Strelchuk and I jointly conceived the idea of developing a quantum analogue of the classical COND oracle, and S. Strelchuk and R. Jozsa assisted in writing the manuscript. I performed all of the calculations for the paper and independently developed the idea for Chapter 6.

The work in Chapter 8 and Appendix B was done jointly with S. Strelchuk and R. Jozsa, resulting in a paper soon to be submitted for publication. S. Strelchuk and I jointly conceived the idea for the paper, and S. Strelchuk and R. Jozsa assisted in writing the manuscript. I performed all of the calculations for the paper, and independently developed the idea for Section 8.4.

Abstract

Since they were first envisioned, quantum computers have oft been portrayed as devices of limitless power, able to perform calculations in a mere instant that would take current computers years to determine. This is, of course, not the case. A huge amount of effort has been invested in trying to understand the limits of quantum computers—under which circumstances they outperform classical computers, how large a speed-up can be gained, and what draws the distinction between quantum and classical computing. In this Ph.D. thesis, I investigate a few intriguing properties of quantum computers involving quantum oracles and classically-simulatable quantum circuits.

In Part I I study the notion of black-box unitary operations, and procedures for effecting the inverse operation. Part II looks at how quantum oracles can be used to test properties of probability distributions, and Part III considers classes of quantum circuits that can be simulated efficiently on a classical computer.

In more detail, Part I studies procedures for inverting black-box unitary operations. Known techniques are generally limited in some way, often requiring ancilla systems, working only for restricted sets of operators, or simply being too inefficient. We develop a novel procedure without these limitations, and show how it can be applied to lift a requirement of the Solovay-Kitaev theorem, a landmark theorem of quantum compiling.

Part II looks at property testing for probability distributions, and in particular considers a special type of access known as the *conditional oracle*. The classical conditional oracle was developed by Canonne et al. in 2015 and subsequently greatly explored. We develop a quantum version of this oracle, and show that it has advantages over the classical process. We use this oracle to develop an algorithm that decides whether or not a mixed state is fully mixed.

In Part III we study classically-simulatable quantum circuits in more depth. Two well-known classes are Clifford circuits and matchgate circuits, which we briefly review. Using these as inspiration, we use the Jordan-Wigner transform to develop new classes of non-trivial quantum circuits that are also classically simulatable.

Acknowledgements

Although it is my name that appears on the title page, there are many others who have contributed greatly to this thesis.

Firstly, I would like to offer my gratitude to my supervisor Richard Jozsa for the unwavering support he has given me throughout these past years. His uncannily accurate insights and sound guidance have been invaluable to me in my research, and his encouraging and calm manner has made the whole experience thoroughly enjoyable.

I would also like to extend my thanks to Sergii Strelchuk, who has been a wonderfully supportive friend, and with whom I have collaborated on much of this work. His valuable advice and perpetually kind words have helped to keep me on the path to completion.

I am very grateful to Toby Cubitt, Aram Harrow and Noah Linden, with whom I very much enjoyed collaborating. I have learnt much from their vast knowledge of quantum computation and expertise in presenting results in a clear and engaging manner.

I acknowledge the discussions with present and former members of the Centre for Quantum Information and Foundations: Johannes Bausch, Nilanjana Datta, Terry Farrelly, Berry Groisman, Will Matthews and Ashley Montanaro.

I am grateful to EPSRC for the financial support I have received through the course of my Ph.D.

My deep-felt thanks also go to David Hall and Walter Morgan for their dedicated support and encouragement.

And finally, I feel greatly indebted to my family and Stephanie Maw, who not only carefully proof-read this thesis, but whose continual support and love have carried me through the past four years to where I am today. It is to them that I dedicate this Ph.D. thesis.

The outline of results

This thesis is presented in three parts: *Quantum Gate Inversion*, *Quantum Distribution Testing*, and *Classical Simulation of Quantum Circuits*. Here I shall give a brief description of the results that are developed in each of these parts.

Quantum Gate Inversion

In Chapter 2 we consider methods for effecting the inverse of an unknown unitary operation U. We begin by discussing two known, but limited, procedures for achieving this, *Quantum Process Tomography* and *Quantum State Refocussing*. In Sections 2.3 and 2.4, currently under review in *Phys. Rev. Lett. (arXiv:1602.07963 [quant-ph])*, we develop a novel procedure for the 'in-line' inversion of an arbitrary unitary operator. In this scenario the operator U is provided in the form of an unlimited number of 'black-boxes' and our system is restricted such that all control unitaries are required to act on a single system with the state space of U. We determine a sequence of quantum gates (unitary operators) that inverts U with arbitrarily small error and failure probability.

Section 2.3 demonstrates this method for a qubit black-box operator. The first step is based on Concatenated Dynamical Decoupling, a refocussing technique described in Section 2.2.2, which is analysed in detail. The remaining steps serve to 'move' U around the space of unitary operators and into a region in which the first step can be applied. The final sequence length is logarithmic in the inverse error and inverse polynomial in the failure probability. Section 2.4 aims to generalise these ideas to unitary operators acting on states of dimension d, rather than 2.

In Chapter 3 we discuss the Solovay-Kitaev theorem and its limitations. A *universal gate set* is a finite set of gates for which any quantum operation can be approximated (to arbitrary accuracy) by a sequence of gates from this set. Informally, the Solovay-Kitaev Theorem states that these sequences are 'short'. One drawback of this theorem, however, is that it requires the inverse of each gate in the universal gate set to also be part of the set. In Section 3.2 we develop an 'inverse-free' version of the Solovay-Kitaev Theorem by approximating the inverse operators using the work in Section 2.4.

Quantum Distribution Testing

In Chapter 4 we introduce the notion of distribution testing, and describe recent work involving the classical conditional oracles, COND and PCOND, defined by C. Canonne, D. Ron and R. Servedio [CRS15]. These authors demonstrated that even the simplest conditional oracles could provide substantial speed-ups in classical distribution testing.

In Chapter 5, published in *QIC Vol.17 No.7&8 (2017) (arXiv:1609.01600 [quant-ph])*, we define quantum analogues of these oracles, called QCOND and PQCOND, and show that significant speed-ups are also possible in quantum distribution testing. In addition, our algorithms outperform their classical counterparts. The problems we consider are:

1. *Uniformity Test:* Given a distribution D and a promise that D is either the uniform distribution $\mathcal{A}$ or $|D - \mathcal{A}| \geq \epsilon$, where $|\cdot|$ is the L_1-norm, decide which of the options holds.

2. *Identity Test:* Given a fixed distribution D^* and a promise that either $D = D^*$ or $|D - D^*| \geq \epsilon$, decide which of the options holds.

3. *Equivalence Test:* Given two distributions $D^{(1)}$ and $D^{(2)}$ and a promise that either $D^{(1)} = D^{(2)}$ or $|D^{(1)} - D^{(2)}| \geq \epsilon$, decide which of the options holds.

4. *Distance from uniformity:* Given a distribution D and the uniform distribution $\mathcal{A}$, estimate $\hat{d} = |D - \mathcal{A}|$.

The query complexities for these problems with the standard quantum sampling oracle QSAMP and the classical PCOND oracle are listed in Table 1, with our new results given in the last column. The notation $\tilde{O}(f(N,\epsilon))$ denotes $O(f(N,\epsilon)\log^k f(N,\epsilon))$ for some k, i.e. logarithmic factors are hidden.

In Section 5.5, we show that a slight modification of the PQCOND oracle will allow for efficient testing of whether a boolean function is 'balanced' or ϵ-far from balanced.

Chapter 6, also published in *QIC Vol.17 No.7&8 (2017)* (*arXiv:1609.01600 [quant-ph])*, is concerned with testing mixedness [MdW13] of a quantum state, that is, deciding whether an n-dimensional quantum state ρ is the fully mixed state or is ϵ-far from it. We describe a novel type of access to ρ that is based on the PQCOND oracle, and

Task	**Standard quantum oracle (**QSAMP**)**	PCOND **oracle** [CRS15]	PQCOND **oracle** *[Chapter 5]*
Uniformity Test	$O\left(\frac{N^{1/3}}{\epsilon^{4/3}}\right)$ [BHH11]	$\tilde{O}\left(\frac{1}{\epsilon^2}\right), \Omega\left(\frac{1}{\epsilon^2}\right)$	$\tilde{O}\left(\frac{1}{\epsilon}\right)$
Identity Test	$\tilde{O}\left(\frac{N^{1/3}}{\epsilon^5}\right)$ [CFMdW10]	$\tilde{O}\left[\left(\frac{\log N}{\epsilon}\right)^4\right]$	$\tilde{O}\left[\left(\frac{\log N}{\epsilon}\right)^3\right]$
Equivalence Test	$\tilde{O}\left(\frac{N^{1/2}}{\epsilon^{3/2}}\right)$ [Mon15]	$\tilde{O}\left[\left(\frac{\log^2 N}{\epsilon^7}\right)^3\right]$	$\tilde{O}\left[\left(\frac{\log^2 N}{\epsilon^7}\right)^2\right]$
Distance from uniformity	$\tilde{O}\left(\frac{N^{1/2}}{\epsilon^{3/2}}\right)$ [Mon15]	$\tilde{O}\left(\frac{1}{\epsilon^{20}}\right)$	$\tilde{O}\left(\frac{1}{\epsilon^{13}}\right)$

Table 1: Query complexity for property testing problems using three different access models: the standard quantum oracle (QSAMP), the PCOND oracle, and our PQCOND oracle.

develop an algorithm that solves the decision problem using $\tilde{O}(n/\epsilon)$ queries. We additionally give a proof (see Appendix A.2), subject to a small conjecture, that $\tilde{O}(\sqrt{n}/\epsilon)$ queries are sufficient.

Classical Simulation of Quantum Circuits

In this part we consider and construct non-trivial quantum circuits that can be efficiently simulated by a classical computer. Perhaps some of the most well-known examples of these are circuits comprising Clifford gates (the Gottesman-Knill Theorem [Got98, NC10, Got97]), and those comprising matchgates (Valiant's Theorem [Val02, TD02, Joz08, JM08]). We discuss both of these constructions in Chapter 7. In addition, we use the connection between matchgates and the Jordan-Wigner transform [TD02] as motivation for the work in Chapter 8 (soon to be submitted for publication).

There we consider a novel method for constructing classically-simulatable circuits by generalising the Jordan-Wigner transform to 2-dimensional lattices. Such circuits are generated whole, rather than comprising specific sets of gates. Naïve attempts to generalise the Jordan-Wigner transform to higher-dimensional lattices map fermionic Hamiltonians with local terms to spin Hamiltonians with non-local terms. We therefore make use of a method based on that given in [VC05] to show how nearest-

neighbour fermionic Hamiltonians on a 2-dimensional lattice may be mapped to a local spin Hamiltonian on a similar lattice. This spin Hamiltonian can be broken into commuting parts and turned into a local, 3-dimensional quantum circuit, for which the output probabilities can be calculated from the partition function of the original system.

Furthermore, we show how the Jordan-Wigner transform can be generalised to deal with multiple flavours of fermions at each site on the lattice. We use this generalisation to construct circuits that are related to the Hubbard model.

In Section 8.5 we extend this mapping and present a classical technique for computing the thermodynamic properties of the original, 2-dimensional fermionic system by transforming it into a 3-dimensional pseudo-classical system and applying the Metropolis-Hastings algorithm.

Contents

Chapter 1

Introduction

1.1 Technical background

This thesis assumes that the reader is familiar with the basic concepts of quantum computation and classical probability theory. For a general introduction to quantum computation, see [NC10]. A useful reference on probability theory is [GW14].

We now present some more specific technical details that we make use of later on in this thesis. We include some basic details of probability theory for completeness.

1.1.1 Weyl operators and the $\mathfrak{su}(d)$ Lie algebra

Weyl operators [Wey27] are a specific higher-dimensional generalisation of the Pauli operators that exist in 2-dimensional systems.

Consider a d-dimensional quantum system, a *qudit*, with basis $\{|0\rangle, \ldots, |d-1\rangle\}$. Up to a phase factor, the operators acting on the quantum system can be described by the (d^2-1)-dimensional Lie algebra $\mathfrak{su}(d)$ with corresponding Lie group $\mathcal{SU}(d)$.

Let $\{\kappa_t\}_{t=0}^{d^2-2}$ be a basis for $\mathfrak{su}(d)$. It is well-known [Pfe03] that all κ_t are traceless and anti-Hermitian. We now introduce the Weyl operators:

Definition 1.1.1 (Weyl operators in d dimensions [Wey27])**.** *σ_1 and σ_3 are defined by their actions on the computational basis states:*

$$\sigma_1 |x\rangle = |(x+1) \mod d\rangle, \quad \sigma_3 |x\rangle = \omega^x |x\rangle,$$

where $\omega = \exp(2\pi i/d)$ is a primitive dth root of unity.

Setting $\boldsymbol{a} = (a_1, a_2)$ with $a_1, a_2 \in [d] \coloneqq \{0, \ldots, d-1\}$, we define $\sigma_{\boldsymbol{a}} \coloneqq {\sigma_3}^{a_1}{\sigma_1}^{a_2}$.

The following properties of the $\sigma_{\boldsymbol{a}}$ and κ_t are important to Part I:

Proposition 1.1.2 (Properties of the $\sigma_{\boldsymbol{a}}$ and κ_t)**.**

1. *The $\sigma_{\boldsymbol{a}}$'s form an orthogonal basis with respect to the Hilbert-Schmidt inner product*[1] *for $GL(d,\mathbb{C})$. More specifically, they satisfy $\mathrm{Tr}(\sigma_{\boldsymbol{a}}^\dagger \sigma_{\boldsymbol{b}}) = d\delta_{\boldsymbol{ab}}$. Note, in addition, that by setting $\boldsymbol{a} = \boldsymbol{0}$, we have that $\mathrm{Tr}(\sigma_{\boldsymbol{b}}) = 0$ for $\boldsymbol{b} \neq \boldsymbol{0}$.*
2. *$\sigma_{\boldsymbol{a}}\sigma_{\boldsymbol{b}} = \sigma_{\boldsymbol{b}}\sigma_{\boldsymbol{a}}\omega^{[\boldsymbol{a},\boldsymbol{b}]}$, where $\omega = \exp(2\pi i/d)$ and $[\boldsymbol{a},\boldsymbol{b}]$ is the symplectic inner product*[2]*.*
3. $\sum_{\boldsymbol{a}\in[d]^2} \omega^{[\boldsymbol{a},\boldsymbol{b}]} = d^2\delta_{\boldsymbol{b}\boldsymbol{0}}$.
4. $\sum_{\boldsymbol{a}\in[d]^2} \sigma_{\boldsymbol{a}}\sigma_{\boldsymbol{b}}\sigma_{\boldsymbol{a}}^\dagger = d^2\delta_{\boldsymbol{b}\boldsymbol{0}}\mathbb{1}$.
5. $\sum_{\boldsymbol{a}\in[d]^2} \sigma_{\boldsymbol{a}}\kappa_t\sigma_{\boldsymbol{a}}^\dagger = 0 \ \ \forall t$.

Proof.

Let $\boldsymbol{a} = (a_1, a_2)$ and $\boldsymbol{b} = (b_1, b_2)$, with $a_1, a_2, b_1, b_2 \in [d]$, and set $\omega = \exp(2\pi i/d)$.

1.

$$\begin{aligned}
\mathrm{Tr}(\sigma_{\boldsymbol{a}}^\dagger\sigma_{\boldsymbol{b}}) &= \mathrm{Tr}({\sigma_1}^{-a_2}{\sigma_3}^{-a_1}{\sigma_3}^{b_1}{\sigma_1}^{b_2}) \\
&= \mathrm{Tr}({\sigma_3}^{b_1-a_1}{\sigma_1}^{b_2-a_2}) \\
&= \sum_{j\in[d]} \langle j|\, {\sigma_3}^{b_1-a_1}{\sigma_1}^{b_2-a_2}\, |j\rangle \\
&= \sum_{j\in[d]} \omega^{(a_1-b_1)j} \underbrace{\langle j|j+b_2-a_2 \mod d\rangle}_{=\delta_{a_2,b_2}} \\
&= \delta_{a_2,b_2} \underbrace{\sum_{j\in[d]} \omega^{(a_1-b_1)j}}_{=d\delta_{a_1,b_1}} \\
&= d\delta_{\boldsymbol{ab}}
\end{aligned}$$

2. Note that

$$\sigma_3\sigma_1\,|j\rangle = \sigma_3\,|j+1 \mod d\rangle = \omega^{j+1}\,|j+1 \mod d\rangle = \omega^{j+1}\sigma_1\,|j\rangle = \omega\sigma_1\sigma_3\,|j\rangle\,,$$

and hence that

$$\sigma_3\sigma_1 = \omega\sigma_1\sigma_3. \tag{1.1}$$

Then

$$\sigma_{\boldsymbol{a}}\sigma_{\boldsymbol{b}} = {\sigma_3}^{a_1}{\sigma_1}^{a_2}{\sigma_3}^{b_1}{\sigma_1}^{b_2} = \omega^{a_1b_2-a_2b_1}{\sigma_3}^{b_1}{\sigma_1}^{b_2}{\sigma_3}^{a_1}{\sigma_1}^{a_2} = \omega^{[\boldsymbol{a},\boldsymbol{b}]}\sigma_{\boldsymbol{b}}\sigma_{\boldsymbol{a}},$$

where the second equality is due to eq. (1.1).

[1] $\langle A, B\rangle := \mathrm{Tr}(A^\dagger B)$ [GG81]

[2] $[\boldsymbol{a},\boldsymbol{b}] := a_1b_2 - a_2b_1$, where $\boldsymbol{a} = (a_1,a_2)^T$ and $\boldsymbol{b} = (b_1,b_2)^T$

3.
$$\sum_{\boldsymbol{a}\in[d]^2}\omega^{[\boldsymbol{a},\boldsymbol{b}]}=\underbrace{\left(\sum_{a_1\in[d]}\omega^{a_1b_2}\right)}_{=d\delta_{b_2,0}}\underbrace{\left(\sum_{a_2\in[d]}\omega^{-a_2b_1}\right)}_{=d\delta_{b_1,0}}=d^2\delta_{\boldsymbol{b}\boldsymbol{0}}$$

4.
$$\begin{aligned}\sum_{\boldsymbol{a}\in[d]^2}\sigma_{\boldsymbol{a}}\sigma_{\boldsymbol{b}}\sigma_{\boldsymbol{a}}^\dagger&=\sum_{\boldsymbol{a}\in[d]^2}\sigma_{\boldsymbol{b}}\omega^{[\boldsymbol{a},\boldsymbol{b}]}&&\text{(Property 2)}\\&=d^2\delta_{\boldsymbol{b}\boldsymbol{0}}\mathbb{1}.&&\text{(Property 3)}\end{aligned}$$

5. From Property 1, we can write
$$\kappa_t=\sum_{\boldsymbol{b}\in[d]^2}(\kappa_t)_{\boldsymbol{b}}\sigma_{\boldsymbol{b}},$$
where $(\kappa_t)_{\boldsymbol{b}}\in\mathbb{C}$ for $\boldsymbol{b}\in[d]^2$. Taking the trace of both sides (remembering that κ_t is traceless) immediately gives
$$0=\sum_{\boldsymbol{b}\in[d]^2}(\kappa_t)_{\boldsymbol{b}}\operatorname{Tr}(\sigma_{\boldsymbol{b}})=(\kappa_t)_{\boldsymbol{0}}\operatorname{Tr}(\sigma_{\boldsymbol{0}})=d(\kappa_t)_{\boldsymbol{0}},$$
and hence we conclude that $(\kappa_t)_{\boldsymbol{0}}=0$.

Now,
$$\begin{aligned}\sum_{\boldsymbol{a}\in[d]^2}\sigma_{\boldsymbol{a}}\kappa_t\sigma_{\boldsymbol{a}}^\dagger&=\sum_{\boldsymbol{a}\in[d]^2}\sigma_{\boldsymbol{a}}\left(\sum_{\boldsymbol{b}\in[d]^2}(\kappa_t)_{\boldsymbol{b}}\sigma_{\boldsymbol{b}}\right)\sigma_{\boldsymbol{a}}^\dagger\\&=\sum_{\boldsymbol{a}\in[d]^2}\sigma_{\boldsymbol{a}}\left(\sum_{\boldsymbol{b}\in[d]^2,\boldsymbol{b}\neq\boldsymbol{0}}(\kappa_t)_{\boldsymbol{b}}\sigma_{\boldsymbol{b}}\right)\sigma_{\boldsymbol{a}}^\dagger\\&=\sum_{\boldsymbol{b}\in[d]^2,\boldsymbol{b}\neq\boldsymbol{0}}(\kappa_t)_{\boldsymbol{b}}\left(\sum_{\boldsymbol{a}\in[d]^2}\sigma_{\boldsymbol{a}}\sigma_{\boldsymbol{b}}\sigma_{\boldsymbol{a}}^\dagger\right),\end{aligned}$$
and the result follows from Property 4.

□

1.1.2 Classical probability theory

The following definitions and formulae can be found in any basic course on probability theory (e.g. [GW14]).

Expectation

Given a discrete random variable X, which takes values $x \in \zeta$ with probability $p_\zeta(x)$, the *expectation* or *mean* of X is defined to be

$$\mathbb{E}(X) := \sum_{x \in \zeta} x p_\zeta(x).$$

If X is a continuous random variable over the domain $\zeta \subseteq \mathbb{R}$ with probability density function $p_\zeta(x)$ (i.e. $p_\zeta(x) \geq 0 \ \forall x \in \zeta$; $\int_\zeta p_\zeta(x)\ dx = 1$), then

$$\mathbb{E}(X) := \int_\zeta x p_\zeta(x)\ dx.$$

Note that if X and Y are two random variables distributed over the domain ζ and $\alpha, \beta \in \mathbb{C}$, then

$$\mathbb{E}(\alpha X + \beta Y) = \alpha \mathbb{E}(X) + \beta \mathbb{E}(Y). \tag{1.2}$$

Variance and Covariance

The *variance* is a measure of the spread of a random variable, and is defined to be

$$\mathrm{Var}(X) := \mathbb{E}\left((X - \mathbb{E}(X))^2\right) = \mathbb{E}(X^2) - \mathbb{E}(X)^2 \geq 0. \tag{1.3}$$

Given two random variables X and Y distributed over the same domain Ω, the *covariance* of X and Y is a measure of the linear dependence of the variables, and is defined by

$$\mathrm{Cov}(X, Y) := \mathbb{E}\left((X - \mathbb{E}(X))(Y - \mathbb{E}(Y))\right) = \mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y)$$

and may be less than 0. Note that $\mathrm{Cov}(X, X) = \mathrm{Var}(X)$.

Suppose that we have n random variables, $X_1, \ldots X_n$, distributed over the domain Ω. Then

$$\begin{aligned}
\mathrm{Var}\left(\sum_{i=1}^n X_i\right) &= \mathbb{E}\left(\left(\sum_{i=1}^n X_i\right)^2\right) - \left(\mathbb{E}\left(\sum_{i=1}^n X_i\right)\right)^2 \\
&= \sum_{i,j=1}^n \left(\mathbb{E}(X_i X_j) - \mathbb{E}(X_i)\mathbb{E}(X_j)\right) \\
&= \sum_{i=1}^n \mathrm{Var}(X_i) + \sum_{i \neq j} \mathrm{Cov}(X_i, X_j).
\end{aligned} \tag{1.4}$$

The Chebyshev Inequality

If the variance of a random variable X is small, one would expect that a sample from X is likely to be near $\mathbb{E}(X)$ (since variance measures the spread from the mean). The Chebyshev inequality [Tch67] quantifies this as follows:

$$\mathbb{P}\left[|X - \mathbb{E}(X)| > \epsilon\right] \leq \frac{\mathrm{Var}(X)}{\epsilon^2}, \tag{1.5}$$

provided that $\mathrm{Var}(X) < \infty$.

1.1.3 Fermionic operators and the 1-dimensional Jordan-Wigner transform

In this section we describe the fermionic creation and annihilation operators and derive the Jordan-Wigner transform.

We consider a 1-dimensional chain of sites numbered $1, \ldots, N$. Each site i can hold a fermion with associated creation and annihilation operators $a_i^\dagger$ and a_i obeying the canonical commutation relations (CCRs) [VC05]

$$\{a_i^\dagger, a_j^\dagger\} = \{a_i, a_j\} = 0, \quad \{a_i^\dagger, a_j\} = \delta_{ij}\mathbb{1}. \tag{1.6}$$

Let $|\Omega\rangle$ be the normalised vacuum state, and write

$$|\alpha\rangle \equiv |\alpha_1, \alpha_2, \ldots, \alpha_N\rangle := \left(a_1^\dagger\right)^{\alpha_1} \left(a_2^\dagger\right)^{\alpha_2} \cdots \left(a_N^\dagger\right)^{\alpha_N} |\Omega\rangle, \tag{1.7}$$

where $\alpha_i \in \{0, 1\}$. The CCRs in eq. (1.6) can easily be used to show that

$$\langle\alpha|\alpha'\rangle = \delta_{\alpha\alpha'}, \tag{1.8}$$

and thus that $\{|\alpha\rangle\}_{\alpha\in\{0,1\}^N}$ is a basis for the fermionic space. Hence we can associate the state $|\alpha\rangle$ on N fermions with equivalent state $|\alpha\rangle$ on N qubits.

We can now determine the form of the $a_i^\dagger$ and a_i operators within the spin paradigm. By calculating $\langle\alpha| a_i^\dagger |\alpha'\rangle$ using the CCRs in eq. (1.6), we find that

$$\begin{aligned} a_i^\dagger &= Z_1 \cdots Z_{i-1} S_i^+ \\ a_i &= Z_1 \cdots Z_{i-1} S_i^-, \end{aligned} \tag{1.9}$$

where R_i effects the R operator on the ith qubit and acts as the identity elsewhere, $S^\pm = \frac{1}{2}(X \pm iY)$, and X, Y and Z are the standard one-qubit Pauli operators. Eq. (1.9)

is known as the Jordan-Wigner transform, and its exact form depends on the ordering of the operators chosen in eq. (1.7).

Here we also define the Majorana fermions [MM06, BK02], $c_i := (a_i + a_i^\dagger)$ and $d_i := i(a_i - a_i^\dagger)$. In the spin paradigm, these are represented as

$$\begin{aligned} c_i &= Z_1 \cdots Z_{i-1} X_i \\ d_i &= Z_1 \cdots Z_{i-1} Y_i. \end{aligned} \tag{1.10}$$

Using the CCRs in eq. (1.6), we find that c_i and d_j obey the commutation relations

$$\{c_i, c_j\} = \{d_i, d_j\} = 2\delta_{ij}\mathbb{1}, \quad \{c_i, d_j\} = 0. \tag{1.11}$$

Part I

Quantum Gate Inversion

Chapter 2

Inversion of black-box unitary operators

2.1 Introduction

A 'black-box' operator is a quantum process for which we do not have a specification. Unitary black-box operators are commonly employed in quantum computation, where the task is often to determine whether or not they satisfy a given property. Black-boxes can be effected by the unknown evolution of a state over time, by a quantum oracle, or by a sequence of gates unknown to us. The *constant-balanced problem* describes a black-box comprising an oracle for a function f that is either constant or balanced (see Problem 5.5.1), and one must determine which is the case. This can be decided by the Deutsch-Jozsa algorithm [CEMM98, DJ92] with only one use of the black-box. Another example is when one is presented with a black-box U effecting an unknown (classically-described) circuit, and one must decide whether or not U acts almost as the identity on all states. This problem was shown [JWB03] to be QMA-complete.

Here, we consider the problem of whether or not a black-box unitary U can be inverted, i.e. given access to as many copies of U as we need, can we implement U^{-1}? Aside from being an interesting question in its own right, there are two particularly useful applications of such a result, both of which we explore:

- **Refocussing:** A quantum state evolves over time according to the system's inherent Hamiltonian. The black-box U would describe the evolution over one time-step. Being able to invert U would mean that the evolution could be cancelled out. (see Section 2.2.2)
- **Solovay-Kitaev theorem:** The Solovay-Kitaev theorem, a landmark theorem in quantum computing, requires access to the inverse operations of a set of unitary operators. Being able to invert these operators directly would mean that this requirement could be removed. (see Chapter 3)

We shall discuss a number of different methods that may be used to effect U^{-1}. First, we look at *quantum process tomography*, a procedure that uses additional ancilla systems to gain a complete description of the operator, after which the matrix can be classically inverted and subsequently implemented. We then consider *refocussing techniques*, which solve a related problem in a different area of physics. Both methods of refocussing that we study are 'on the fly' protocols that require no ancillas. These techniques, however, are limited in that they are applicable only to restricted sets of operators U. We subsequently, therefore, derive a universal procedure to refocus any unitary U to arbitrary accuracy. We develop a protocol to generate a sequence of unitary operations $\{R_1, \ldots, R_n\}$, independent of U, such that

$$R_1 U R_2 U \cdots U R_n \approx U^{-1}. \tag{2.1}$$

with high probability. More precisely, $\|R_1 U R_2 U \cdots U R_n - U^{-1}\| \leq \epsilon$, where the number n of control unitaries R only needs to scale as $n = O(\log^2(1/\epsilon))$ if U is a qubit operator.

2.2 Known inversion techniques

2.2.1 Quantum process tomography

Here we describe a technique presented in [NC10], Section 8.4.2. *Quantum process tomography* is a procedure that completely specifies an operator U (to within a given error). Once this specification has been deduced, U^{-1} can be easily calculated and subsequently implemented.

To understand quantum process tomography, we must first discuss *quantum state tomography*. To simplify the analysis, we shall work with a single qubit, although the idea is generalisable to many d-dimensional qudits.

The method works intuitively as follows:

- **Quantum state tomography:** copies of a density matrix state ρ are measured in several different bases to deduce a classical description of the state;
- **Quantum process tomography:** one acts with the operator U on a known state, and the output is evaluated using quantum state tomography. This process is repeated with a full basis of states to gain a complete specification of U.

We now analyse quantum state tomography in more detail. Suppose that we have many copies of a qubit density matrix ρ. Since the matrices $\frac{\mathbb{1}}{\sqrt{2}}, \frac{X}{\sqrt{2}}, \frac{Y}{\sqrt{2}}, \frac{Z}{\sqrt{2}}$ form an orthonormal basis (where X, Y and Z are the 2-dimensional Pauli operators) with respect to the Hilbert-Schmidt inner product (see Section 1.1.1), we can write

$$\rho = \frac{1}{2}\operatorname{Tr}(\rho)\mathbb{1} + \frac{1}{2}\operatorname{Tr}(\rho X)X + \frac{1}{2}\operatorname{Tr}(\rho Y)Y + \frac{1}{2}\operatorname{Tr}(\rho Z)Z.$$

Expressions like $\operatorname{Tr}(\rho A)$ can be interpreted as the expectation of the observable A given the state ρ. Thus, for example, $\operatorname{Tr}(\rho Z)$ can be determined by repeatedly measuring the observable Z, and averaging the result. The central limit theorem (Section 8 in [GW14]) can be used to determine how accurate this estimate is. Repeating this process for each of the four observables, we can determine ρ to arbitrary accuracy with high probability.

This procedure can be extended to quantum process tomography in the following way. Suppose that U is a unitary operator on one qubit. Choose the pure states $|\psi_1\rangle, \ldots, |\psi_4\rangle$ so that the corresponding density matrices $|\psi_1\rangle\langle\psi_1|, \ldots, |\psi_4\rangle\langle\psi_4|$ form a basis for the space of (2×2) matrices. One such choice is

$$|\psi_1\rangle = |0\rangle, \quad |\psi_2\rangle = |1\rangle, \quad |\psi_3\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |\psi_4\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle).$$

U can be applied to each of these states respectively, gaining $\rho_1 = U|\psi_1\rangle\langle\psi_1|U^\dagger$, $\ldots$, $\rho_4 = U|\psi_4\rangle\langle\psi_4|U^\dagger$, which can be estimated using quantum state tomography. This gives a complete classification of the operator U, since for any density matrix $\rho = \alpha_1|\psi_1\rangle\langle\psi_1| + \cdots + \alpha_4|\psi_4\rangle\langle\psi_4|$, $\alpha \in \mathbb{C}$, we have that $U\rho U^\dagger = \alpha_1\rho_1 + \cdots + \alpha_4\rho_4$. Through classical processing the matrix form of U can be deduced.

The method extends trivially if U is acting on a d-dimensional space, for which d^2 pure states are required.

Finally, the matrix can be inverted to deduce U^{-1}.

Other variants on standard quantum process tomography exist, such as ancilla-assisted quantum tomography and entanglement-assisted quantum tomography [ABJ+03].

2.2.2 Refocussing techniques

It is a common problem in fields such as nuclear magnetic resonance (NMR) and quantum information processing that the state of a system will evolve over time.

This evolution is tied to the Hamiltonian governing the system, which is often time-independent. Refocussing techniques work by applying certain radio-frequency pulses to the system at pre-determined times that cancel out this evolution, thus 'refocussing' the state.

In the two techniques presented below, the Hamiltonian is considered to be time-independent, and the interval between the pulses is constant. As a result, the evolution of the system between consecutive pulses can be described by a unitary operator $U = e^{-iH}$, where H is the Hamiltonian governing the system's dynamics. The n pulses that are applied can be described by unitary operators $R_1, \ldots, R_n$ (independent of U). As a result, the evolution of the system can be described by the quantum circuit $R_1 U R_2 U \cdots U R_n U$. The aim of the refocussing technique is to eliminate the evolution caused by U, so that

$$R_1 U R_2 U \cdots U R_n U \approx \mathbb{1}.$$

Of course, this is quickly rearranged to give

$$R_1 U R_2 U \cdots U R_n \approx U^{-1}.$$

Spin Echo

Spin echo is a technique often employed within the field of NMR, and is able to correct for evolution caused by Hamiltonians of a particular form, using just two pulses [Hah50, FM98].

Suppose we consider a system consisting of a single qubit, with a governing Hamiltonian of the form $H_z = \alpha X + \beta Y$, where $\alpha, \beta \in \mathbb{R}$. Then

$$U = e^{-iH_z}.$$

Using the Pauli operator commutation relations (Exercise 2.4.1 in [NC10]), we see that $ZH_zZ = -H_z$, and hence that

$$ZUZ = U^{-1} \implies ZUZU = \mathbb{1}.$$

This method can be generalised, noting that if the anti-commutator $\{H, \sigma\} = 0$ for some operator σ, then

$$\sigma U \sigma = U^{-1} \implies \sigma U \sigma U = \mathbb{1},$$

where $U = e^{-iH}$.

In particular, X pulses will refocus a Hamiltonian $H_x = \beta Y + \gamma Z$, and Y pulses will refocus a Hamiltonian $H_y = \alpha X + \gamma Z$, where $\alpha, \beta, \gamma \in \mathbb{R}$.

Dynamical Decoupling

A general Hamiltonian acting on a single-qubit system is of the form

$$H = \alpha X + \beta Y + \gamma Z,$$

where $\alpha, \beta, \gamma \in \mathbb{R}$. If the Hamiltonian is unknown, spin echo will not be effective in refocussing the state, and a different technique must be employed.

This is a common problem encountered in quantum information processing, where an unwanted and unknown always-on evolution leads to a coupling between two initially isolated systems [ÁSS12, D$^+$00, KL05, VKL99, WS07, YWL11, ZWL14, KL11].

Dynamical decoupling encompasses a set of perturbative methods, similar to spin echo, that involve applying several control pulses to the combined system over a period of time to dynamically eliminate the coupling [KL11].

Here we focus on a technique known as *Concatenated Dynamical Decoupling* (CDD) [YWL11, WS07, KL11, KL05, ÁSS12], a natural extension of spin echo, and for which the interval between pulses is constant.

Since H can be written as a linear combination of Hamiltonians of the form of H_x, H_y and H_z, CDD considers a sequence formed from the spin echo refocussing pulses for each of these Hamiltonians [ÁSS12]:

$$\mathrm{CDD}_1 = (XUX)(YUY)(ZUZ)U = -XUZUXUZU.$$

By using Property 4 of Proposition 1.1.2, this sequence can be shown to give $\mathbb{1}$ to first order in H when expanded as a power series.

We then recursively concatenate the sequence to eliminate higher-order terms in H:

$$\mathrm{CDD}_N = -X\,\mathrm{CDD}_{N-1}\,Z\,\mathrm{CDD}_{N-1}\,X\,\mathrm{CDD}_{N-1}\,Z\,\mathrm{CDD}_{N-1}\,.$$

To eliminate terms of order up to (and including) N, the number of pulses required is then $n = 4^N$ [KL05, KL11].

For such sequences to be practical for refocussing a system, $\|H\|$ must be small. In Section 2.3.1 we place bounds on $\|H\|$ in an attempt to understand the effectiveness of this method. If $\|H\|$ is 'small enough', however, we produce a sequence

$$R_1 U R_2 U \cdots U R_n U \approx \mathbb{1} \implies R_1 U R_2 U \cdots U R_n \approx U^{-1}.$$

2.3 Universal in-line inversion of qubit operators

We describe here our procedure to invert a black-box unitary operator acting on a single qubit.

Any unitary operation $U \in \mathcal{SU}(2)$ may be written in the form $U = e^{-iH}$, where the Hamiltonian H is of the form $H = \boldsymbol{h} \cdot \boldsymbol{\sigma}$, where $\boldsymbol{\sigma} = (X, Y, Z)$ is the vector of Pauli matrices, and $\boldsymbol{h} \in \mathbb{R}^3$.

We introduce the function (also seen in Section 2.2.2)

$$f(U) := XUXYUYZUZU = (-X)UZUXUZU. \tag{2.2}$$

As noted previously, this can be shown to give $\mathbb{1}$ to first order in H when expanded as a power series. Thus1 we expect that for U within a certain distance of $\mathbb{1}$, the recursive application of f will reduce this distance. This forms the basis of CDD (see Section 2.2.2), and one of the stages of our procedure.

Outside of this region, what can be said about f? It is clear that f does not necessarily reduce the distance to the identity, as it has several fixed points and cycles. For example, the unitary operator $\frac{1-i}{2}\left(\begin{smallmatrix} 1 & i \\ -1 & i \end{smallmatrix}\right)$ is a fixed point, and $\left(\frac{1-i}{2}\left(\begin{smallmatrix} i & i \\ -1 & 1 \end{smallmatrix}\right), \frac{1-i}{2}\left(\begin{smallmatrix} 1 & -1 \\ i & i \end{smallmatrix}\right)\right)$ is a two-cycle. This provides strong motivation for developing a randomised, rather than deterministic, protocol for refocussing.

Note that f can be expressed in the form of eq. (2.1) as $f(U) = (-X)UZUXUZU$.

The analysis for the one-qubit case can be computed explicitly, and we do so in the following three stages:

1. In terms of a chosen measure of distance, we lower bound the size of the neighbourhood of $\mathbb{1}$ for which an application of f reduces the distance to $\mathbb{1}$. We shall call this the *shrinking region*. This is the crux of CDD, which can only be applied within this region.

2. We find other points in $\mathcal{SU}(2)$ that are mapped exactly to $\mathbb{1}$ under a single application of f, and hence (by continuity of f) determine regions that are mapped into the shrinking region. We call these *jumping regions*.

3. We apply certain random operations to our unitary and lower bound the probability of moving it into one of the jumping regions. We call these *random conjugations*.

2.3.1 Bounding the shrinking region

We start by proving the following well-known proposition:

Proposition 2.3.1. *Any $U \in \mathcal{SU}(2)$ can be written in the form*

$$\begin{aligned} &U = a\mathbb{1} + ibX + icY + idZ, \\ &a^2 + b^2 + c^2 + d^2 = 1, \qquad a, b, c, d \in \mathbb{R}. \end{aligned} \tag{2.3}$$

Proof. Any unitary operation $U \in \mathcal{SU}(2)$ may be written in the form $U = e^{i\boldsymbol{u}\cdot\boldsymbol{\sigma}}$ [Pfe03], where $\boldsymbol{u} = (u_1, u_2, u_3) \in \mathbb{R}^3$, and $\boldsymbol{\sigma} = (X, Y, Z)$. Since $(\boldsymbol{u}\cdot\boldsymbol{\sigma})^2 = |\boldsymbol{u}|^2\mathbb{1}$, we see that if $\boldsymbol{u} \neq \mathbf{0}$,

$$U = e^{i\boldsymbol{u}\cdot\boldsymbol{\sigma}} = (\cos|\boldsymbol{u}|)\mathbb{1} + i(\sin|\boldsymbol{u}|)(\hat{\boldsymbol{u}}\cdot\boldsymbol{\sigma})$$

where $\hat{\boldsymbol{u}} = \boldsymbol{u}/|\boldsymbol{u}| = (\hat{u}_1, \hat{u}_2, \hat{u}_3)$ is a normalised vector. Letting $a = \cos|\boldsymbol{u}|, b = (\sin|\boldsymbol{u}|)\hat{u}_1, c = (\sin|\boldsymbol{u}|)\hat{u}_2$, and $d = (\sin|\boldsymbol{u}|)\hat{u}_3$, we arrive at eq. (2.3). □

Now, using the Hilbert-Schmidt norm $\|A\| \coloneqq \sqrt{\mathrm{Tr}(A^\dagger A)}$, we define the *distance* between U and $\mathbb{1}$ to be

$$\varepsilon_0 \coloneqq \|U - \mathbb{1}\| = 2\sqrt{1-a}.$$

A straightforward matrix multiplication then tells us that

$$f(U) = (1 - 8b^2d^2)\mathbb{1} + i\cdot 8abd^2X + i\cdot 4bd(d^2-1)Y + i\cdot(-8bcd^2)Z$$

and hence that the distance between $f(U)$ and $\mathbb{1}$ is

$$\varepsilon_1 \coloneqq \|f(U) - \mathbb{1}\| = 2\sqrt{8}|bd| \leq \sqrt{8}(b^2 + d^2) \leq \sqrt{8}(1 - a^2) \leq \sqrt{2}\varepsilon_0^2, \tag{2.4}$$

where the second inequality follows from eq. (2.3).

If ε_m is the distance from $\mathbb{1}$ after m applications of f, then repeated application of eq. (2.4) implies that $\varepsilon_m \leq \sqrt{2}^{2^m-1}\varepsilon_0^{2^m}$. Choosing $\varepsilon_0 \leq 1/2$ gives us doubly-exponential convergence towards $\mathbb{1}$ as m increases, that is,

$$\varepsilon_m \leq 2^{-\frac{1}{2}(2^m+1)}. \tag{2.5}$$

We thus define the shrinking region to be $a = 1 - \frac{\varepsilon_0^2}{4} \geq 15/16$. This is represented in Figure 2.1 as region A.

2.3.2 Bounding the jumping regions

We saw previously that $\varepsilon_1 = 2\sqrt{8}|bd|$. To ensure that $f(U)$ is inside the shrinking region A, we require that $\varepsilon_1 \leq 1/2$. Let us denote the 'jumping region' by $J \equiv f^{-1}(A)$ and thus observe that J is the set of U with $2\sqrt{8}|bd| \leq 1/2 \implies |bd| \leq 1/\sqrt{128}$; see Figure 2.1.

2.3.3 Bounding the probability of landing in a jumping region after applying a random conjugation

We now write U in the form $U = a\mathbb{1} + i(\boldsymbol{u} \cdot \boldsymbol{\sigma})$, where $\boldsymbol{u} = (b, c, d)$ and $\boldsymbol{\sigma} = (X, Y, Z)$. The operation we apply is conjugation by an operator $R = \boldsymbol{r} \cdot \boldsymbol{\sigma}$, where $\boldsymbol{r}$ is a real unit vector, and R is unitary. Then

$$U' := RUR^\dagger = a\mathbb{1} + i\boldsymbol{u'} \cdot \boldsymbol{\sigma}.$$

where $\boldsymbol{u'} = [2(\boldsymbol{r} \cdot \boldsymbol{u})\boldsymbol{r} - \boldsymbol{u}] = (b', c', d')$. This transformation has two important properties:

- The distance from $\mathbb{1}$ is invariant, i.e. $\|U - \mathbb{1}\| = \|U' - \mathbb{1}\|$. This ensures that the unitary can never leave the shrinking region once inside it; and
- $\boldsymbol{u'}$ is the rotation of $\boldsymbol{u}$ by π about the vector $\boldsymbol{r}$. Thus choosing $\boldsymbol{r}$ to point in a uniformly random direction (according to the spherical measure on S^2) ensures that $\boldsymbol{u'}$ also points in a similarly uniformly random direction (with $|\boldsymbol{u'}| = |\boldsymbol{u}|$). In Figure 2.1, this would be represented by a reflection of the sphere in a plane containing $\mathbb{1}$ along the a axis.

We now lower bound the probability that U' is in a jumping region. To do so, we write $\boldsymbol{u'}$ in spherical co-ordinates: $\boldsymbol{u'} = (b', c', d')_{\text{cart}} = (|\boldsymbol{u}|, \theta, \phi)_{\text{sph}}$. The jumping region J corresponds to the unitaries with

$$|\boldsymbol{u}|^2 |\cos(\theta)\sin(\theta)\cos(\phi)| \leq \frac{1}{\sqrt{128}}.$$

Recall that $\theta \in [0, \pi]$, $\phi \in [0, 2\pi)$ are drawn uniformly at random from the sphere, while $|\boldsymbol{u}|$ depends on U. To eliminate this dependence we can bound

$$\begin{aligned}\mathbb{P}[U' \in J] &\geq \mathbb{P}\left[|\cos(\theta)\sin(\theta)\cos(\phi)| \leq 1/\sqrt{128}\right] \\ &\approx 0.271\cdots\end{aligned}$$

The constant $0.271\cdots$ can be obtained by numerical integration, and for notational convenience we will use $\mathbb{P}[U' \in J] \geq 1/4$.

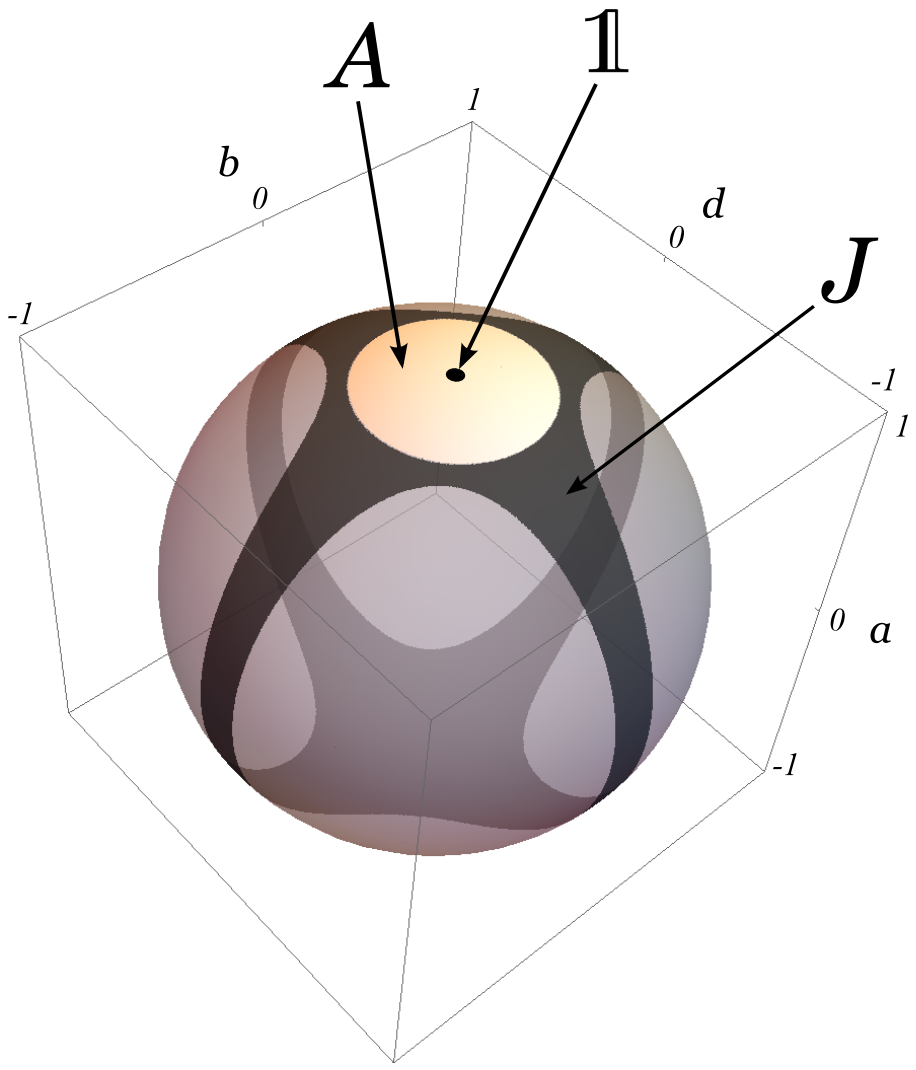


Figure 2.1: Universal refocussing for $U \in \mathcal{SU}(2)$. For illustration we set $c = 0$ in eq. (2.3) so that the surface of the sphere represents the remaining part of $\mathcal{SU}(2)$. A represents the shrinking region, with $U = \mathbb{1}$ marked at its center point. $J \equiv f^{-1}(A)$ is the jumping region, for which $|bd| \leq 1/\sqrt{128}$. The action of a random conjugation $R = \boldsymbol{r} \cdot \boldsymbol{\sigma}$ (where, for this illustration, $\boldsymbol{r} = (r_1, 0, r_3)$) is to reflect the sphere in a plane along the a axis containing $\mathbb{1}$, leaving the distance to $\mathbb{1}$ invariant.

2.3.4 Tying it all together

We now introduce the function $g(U) = (\boldsymbol{r}\cdot\boldsymbol{\sigma})U(\boldsymbol{r}\cdot\boldsymbol{\sigma})^\dagger$, where each application of g chooses a unit direction vector $\boldsymbol{r}$ uniformly at random according to the spherical measure on S^2. Consider $(f\circ g)^{\circ l}$, i.e. f and g composed l times. In order to enter a jumping region with probability $\geq 1-\eta$ we require

$$l \geq \frac{\log_2(1/\eta)}{\log_2(4/3)}.$$

Once in the shrinking region, we require a further m steps to get within $\epsilon := \epsilon_{l+m}$ distance of the identity, where

$$m \geq \log_2\log_2\left(\frac{1}{\sqrt{2}\epsilon}\right) + 1.$$

Combining these and introducing the function $F(U) \coloneqq (f\circ g)^{\circ k}$, we see that if

$$k \geq \frac{\log_2(1/\eta)}{\log_2(4/3)} + \log_2\log_2\left(\frac{1}{\sqrt{2}\epsilon}\right) + 1, \tag{2.6}$$

U will be mapped to within ϵ distance of $\mathbb{1}$ with probability $\geq 1-\eta$.

Expanding $F(U)$ gives a pulse sequence of the form

$$F(U) = R_1 U R_2 \cdots R_n U R_{n+1}.$$

In order to produce a sequence of the form of eq. (2.1), we conjugate by R_{n+1}, since

$$\|(R_{n+1}R_1)UR_2\cdots R_nU - \mathbb{1}\| = \|R_{n+1}(F(U)-\mathbb{1})R_{n+1}^\dagger\| = \|F(U)-\mathbb{1}\|.$$

The number of pulses n required for the full refocussing function F is the same as the number of uses of U, which is 4^k. Thus we see that the number of pulses is bounded by

$$n = 4^k \leq \frac{16}{\eta^5}\log_2^2\left(\frac{1}{\sqrt{2}\epsilon}\right). \tag{2.7}$$

The multiplicative factor of 16 comes from the fact that k may need to be rounded up to the nearest integer greater than the RHS of eq. (2.6). In addition, we have rounded the power of $1/\eta$ up from $2/\log_2(4/3) \approx 4.82$ to 5.

2.4 Universal in-line inversion of d-dimensional qudit operators

Though the basic idea of the one-qubit case generalises to d dimensions, it is more difficult to determine the jumping regions, and not at all clear that random conjugations can even bring arbitrary d-dimensional unitary operations close to these jumping regions. However, we will show there exist jumping regions that can be reached from any unitary operator.

2.4.1 Bounding the d-dimensional shrinking region

We introduce the mapping $f : \mathcal{SU}(d) \to \mathcal{SU}(d)$, defined by

$$f(U) = \prod_{\boldsymbol{a}\in[d]^2} \sigma_{\boldsymbol{a}} U \sigma_{\boldsymbol{a}}^\dagger. \tag{2.8}$$

Using the operator norm[1], we define the *distance* between an operator W and $\mathbb{1}$ to be $\|W - \mathbb{1}\|$. We write $U = \mathbb{1} + \delta U = e^{-iH}$, where iH can be expressed as a linear combination of κ_t's, the generators for the Lie algebra $\mathfrak{su}(d)$, which are defined in Section 1.1.1. Furthermore, we impose that $\|\delta U\| \leq 1/2$.

We have that $H = i\log(\mathbb{1} + \delta U)$, and hence the Mercator series[2] gives us that

$$\|H\| \leq \sum_{k=1}^{\infty} \frac{\|\delta U\|^k}{k} \leq \sum_{k=1}^{\infty} \frac{(1/2)^{k-1}}{k} \|\delta U\| < \frac{1}{1-1/2}\|\delta U\| = 2\|\delta U\| \leq 1. \tag{2.9}$$

With f defined as in eq. (2.8) and writing $U = \mathbb{1} + \delta U$, we see that

$$f(U) = \prod_{\boldsymbol{a}\in[d]^2} \sigma_{\boldsymbol{a}}(\mathbb{1} + \delta U)\sigma_{\boldsymbol{a}}^\dagger = \mathbb{1} + \sum_{\boldsymbol{a}\in[d]^2} \sigma_{\boldsymbol{a}} \delta U \sigma_{\boldsymbol{a}}^\dagger + \sum_{\boldsymbol{a}<\boldsymbol{b}} \sigma_{\boldsymbol{a}} \delta U \sigma_{\boldsymbol{a}}^\dagger \cdot \sigma_{\boldsymbol{b}} \delta U \sigma_{\boldsymbol{b}}^\dagger + \cdots,$$

where $<$ is an ordering on the set $[d]^2$ (the exact form of the ordering is irrelevant). As an example, we could have $\boldsymbol{a} < \boldsymbol{b} \Leftrightarrow da_1 + a_2 < db_1 + b_2$, for $\boldsymbol{a} = (a_1, a_2), \boldsymbol{b} = (b_1, b_2) \in [d]^2$.

After moving the $\mathbb{1}$ to the left-hand side, we take the operator norm of both sides and

[1] $\|A\| = \sup_{|\psi\rangle\in\mathbb{C}^d, \||\psi\rangle\|=1} \|A|\psi\rangle\|$

[2] If $\|A\| < 1$, $\log(\mathbb{1} + A) = \sum_{k=1}^{\infty} (-1)^{k+1} A^k/k$

use the triangle inequality and sub-multiplicative property[3] to deduce that

$$\begin{aligned}\|f(U)-\mathbb{1}\| &\leq \left\|\sum_{\boldsymbol{a}\in[d]^2}\sigma_{\boldsymbol{a}}\delta U\sigma_{\boldsymbol{a}}^\dagger\right\| + \binom{d^2}{2}\|\delta U\|^2 + \binom{d^2}{3}\|\delta U\|^3+\cdots \\ &= \left\|\sum_{\boldsymbol{a}\in[d]^2}\sigma_{\boldsymbol{a}}\delta U\sigma_{\boldsymbol{a}}^\dagger\right\| + \sum_{j=2}^{d^2}\binom{d^2}{j}\|\delta U\|^j.\end{aligned}$$

Since $\|\delta U\|\leq 1/2$, $\|\delta U\|^j\leq\|\delta U\|^2$ for $j\geq 2$. Thus

$$\begin{aligned}\|f(U)-\mathbb{1}\| &\leq \left\|\sum_{\boldsymbol{a}\in[d]^2}\sigma_{\boldsymbol{a}}\delta U\sigma_{\boldsymbol{a}}^\dagger\right\| + \left[\binom{d^2}{2}+\cdots+\binom{d^2}{d^2}\right]\|\delta U\|^2 \\ &= \left\|\sum_{\boldsymbol{a}\in[d]^2}\sigma_{\boldsymbol{a}}\delta U\sigma_{\boldsymbol{a}}^\dagger\right\| + (2^{d^2}-d^2-1)\|\delta U\|^2.\end{aligned}$$

Now consider the first term on the right-hand side, and recall that $U=e^{-iH}$. Hence

$$\begin{aligned}\left\|\sum_{\boldsymbol{a}\in[d]^2}\sigma_{\boldsymbol{a}}\delta U\sigma_{\boldsymbol{a}}^\dagger\right\| &= \left\|\sum_{\boldsymbol{a}\in[d]^2}\sigma_{\boldsymbol{a}}(e^{-iH}-\mathbb{1})\sigma_{\boldsymbol{a}}^\dagger\right\| \\ &= \left\|\sum_{\boldsymbol{a}\in[d]^2}\sigma_{\boldsymbol{a}}\left(\sum_{k=1}^{\infty}\frac{(-iH)^k}{k!}\right)\sigma_{\boldsymbol{a}}^\dagger\right\| \\ &\leq \left\|\sum_{\boldsymbol{a}\in[d]^2}\sigma_{\boldsymbol{a}}(-iH)\sigma_{\boldsymbol{a}}^\dagger\right\| + \sum_{k=2}^{\infty}\frac{1}{k!}\left\|\sum_{\boldsymbol{a}\in[d]^2}\sigma_{\boldsymbol{a}}(-iH)^k\sigma_{\boldsymbol{a}}^\dagger\right\| \\ &\leq \left(\sum_{k=2}^{\infty}\frac{1}{k!}\right)\|H\|^2d^2 \\ &\leq 4d^2(e-2)\|\delta U\|^2,\end{aligned}$$

where the third line follows from the triangle inequality. The first term in the third line is 0 by Property 5 in Proposition 1.1.2. The fourth line then follows by the triangle inequality, the sub-multiplicative property, and the fact that $\|H\|<1$ (from eq. (2.9)). The final line follows from eq. (2.9). Hence we discover that

$$\|f(U)-\mathbb{1}\|\leq(2^{d^2}+d^2(4e-9)-1)\|\delta U\|^2<\alpha\|\delta U\|^2, \tag{2.10}$$

where $\alpha=2^{d^2+1}$.

We now define the shrinking region to be

$$\|U-\mathbb{1}\|=\epsilon_0\leq 1/(2\alpha). \tag{2.11}$$

[3] $\|AB\|\leq\|A\|\|B\|$ for all $d\times d$ complex matrices A,B

If ε_m is the distance from $\mathbb{1}$ after m applications of f, then the repeated application of eq. (2.10) implies that

$$\varepsilon_m < 2^{-2^m}/\alpha. \tag{2.12}$$

2.4.2 Finding and bounding the d-dimensional jumping regions

As before, we can write $U = e^{-iH}$, where $iH \in \mathfrak{su}(d)$. We show below that if U (and hence H) is diagonal, $f(U) = \mathbb{1}$. Thus the jumping regions include the neighbourhoods of all diagonal unitaries.

Property 1 of Proposition 1.1.2 allows us to write

$$iH = \sum_{\boldsymbol{a}\in[d]^2, \boldsymbol{a}\neq\boldsymbol{0}} \lambda_{\boldsymbol{a}}\sigma_{\boldsymbol{a}}, \tag{2.13}$$

where $\lambda_{\boldsymbol{a}} \in \mathbb{R}\ \forall \boldsymbol{a}$, and $\boldsymbol{a} = 0$ is excluded from the sum because $iH \in \mathfrak{su}(d)$ is traceless. In addition, if U is diagonal, we have that iH is diagonal, and thus the only non-zero $\lambda_{\boldsymbol{a}}$'s are those corresponding to diagonal $\sigma_{\boldsymbol{a}}$'s (i.e. $\boldsymbol{a} = (a_1, 0)$).

From eq. (2.8), we have

$$f(U) = \prod_{\boldsymbol{c}\in[d]^2} \sigma_{\boldsymbol{c}} U \sigma_{\boldsymbol{c}}^\dagger = \prod_{\boldsymbol{c}\in[d]^2} \exp(-\Lambda_{\boldsymbol{c}}),$$

where

$$\Lambda_{\boldsymbol{c}} = \sigma_{\boldsymbol{c}}(iH)\sigma_{\boldsymbol{c}}^\dagger = \sum_{\boldsymbol{a}\neq\boldsymbol{0}} \omega^{[\boldsymbol{c},\boldsymbol{a}]}\lambda_{\boldsymbol{a}}\sigma_{\boldsymbol{a}},$$

in which we have used eq. (2.13) and Property 2 of Proposition 1.1.2 to deduce the final equality. Note that the non-zero terms of the sum are diagonal (recall that they correspond to $\boldsymbol{a}$'s of the form $\boldsymbol{a} = (a_1, 0)$), and hence all $\Lambda_{\boldsymbol{c}}$ commute. Thus using Property 3 of Proposition 1.1.2, we see that

$$f(U) = \exp\left(-\sum_{\boldsymbol{c}} \Lambda_{\boldsymbol{c}}\right) = \exp\left(-\sum_{\boldsymbol{a}\neq\boldsymbol{0}} \underbrace{\left(\sum_{\boldsymbol{c}} \omega^{[\boldsymbol{c},\boldsymbol{a}]}\right)}_{=0} \lambda_{\boldsymbol{a}}\sigma_{\boldsymbol{a}}\right) = \mathbb{1}.$$

$f(U)$ is a product of operators, containing d^2 instances of U. The hybrid argument, Theorem 2 in [Vaz98], then implies that

$$\|f(U) - f(V)\| \leq d^2\|U - V\|. \tag{2.14}$$

Suppose that we have a W such that $f(W) = \mathbb{1}$, and define $W' = W(\mathbb{1} + \delta W)$. Eq. (2.14) then gives

$$\|f(W') - \mathbb{1}\| \leq d^2\|\delta W\|.$$

Thus to ensure that $f(W')$ is in the shrinking region, we must have, from eq. (2.11), that

$$\|\delta W\| \leq \delta \coloneqq \frac{1/(2\alpha)}{d^2} = \frac{1}{2\alpha d^2}. \tag{2.15}$$

2.4.3 Bounding the probability of landing in a d-dimensional jumping region after applying a random conjugation

Here we conjugate U with a Haar random unitary $V \in \mathcal{SU}(d)$ (i.e. uniformly random with respect to the Haar measure [Haa33]) and bound the probability that the resulting operator is close to diagonal, and thus in a jumping region. Conjugation is a useful operation to apply since

$$\|VUV^\dagger - \mathbb{1}\| = \|V(U-\mathbb{1})V^\dagger\| = \|U-\mathbb{1}\|$$

and thus, as in the one-qubit case, it leaves the distance from the identity invariant.

We note that there is at least one good choice of V: let V_0 be a unitary such that $V_0 U V_0^\dagger$ is diagonal. While $V = V_0$ has zero probability, we argue that there is a non-negligible probability that V will be close to V_0. We choose a unitary operator $V \in \mathcal{SU}(d)$ uniformly at random according to the Haar measure [Haa33], and lower-bound the probability that it is close to V_0, where $V_0 \in \mathcal{SU}(d)$ and $V_0 U V_0^\dagger$ is diagonal.

We first note that $\mathbb{P}[\|V - V_0\| \leq \delta]$ is independent of V_0, and so wlog we consider $V_0 = \mathbb{1}$. Consider the map $\exp : \mathfrak{su}(d) \to \mathcal{SU}(d)$, and let $B_r = \{s \in \mathfrak{su}(d) : \|s\| \leq r\}$, for $r \leq \pi$. Note that $\exp(B_r)$ is a ball around $\mathbb{1}$ in $\mathcal{SU}(d)$ of radius $|\exp(ir) - 1| = 2\sin(r/2)$. Thus the pre-image of the ball of radius δ is B_ν with

$$\nu = 2\arcsin(\delta/2) \tag{2.16}$$

Now, the volume of B_r is $\mathrm{vol}(B_r) = cr^{d^2-1}$, where c is dependent upon d, and we are using the Euclidean metric on $\mathfrak{su}(d)$.

Lemma 4 in [Sza97] provides the result

$$\frac{4}{10}\|s\| \leq \|\exp(s) - \mathbb{1}\| \leq \|s\|$$

for $s \in \mathfrak{su}(d)$; the upper bound holds for all s, and the lower bound holds for $\|s\| \leq \pi/4$. Thus

- if $\nu \leq \pi/4$, then $\mathrm{vol}(\exp(B_\nu)) \geq \frac{4}{10}\mathrm{vol}(B_\nu) = \frac{4}{10}c\nu^{d^2-1}$; and

- since $\exp(B_\pi) = \mathcal{SU}(d)$, we have that $\mathrm{vol}(\mathcal{SU}(d)) \leq \mathrm{vol}(B_\pi) = c\pi^{d^2-1}$.

Hence the probability that a random operator $V \in \mathcal{SU}(d)$ is within distance δ from $\mathbb{1}$ (or any other V_0) is lower bounded by

$$\mathbb{P}[\|V - V_0\| \leq \delta] \geq \frac{4}{10}\left(\frac{\nu}{\pi}\right)^{d^2-1}.$$

In addition, eq. (2.16) implies that

$$\nu = 2\arcsin(\delta/2) \geq \delta,$$

hence we arrive at

$$\mathbb{P}[\|V - V_0\| \leq \delta] \geq \frac{4}{10}\left(\frac{\delta}{\pi}\right)^{d^2-1} \geq \left(\frac{\delta}{10}\right)^{d^2-1}. \tag{2.17}$$

2.4.4 Summary of the d-dimensional case

We summarise the results below:

1. Given $U \in \mathcal{SU}(d)$, the shrinking region is defined (from eq. (2.11)) by $\varepsilon_0 \leq 1/(2\alpha)$, where $\alpha = 2^{d^2+1}$. Within this region, f provides doubly-exponential convergence to $\mathbb{1}$. More specifically, (from eq. (2.12)) we have that $\varepsilon_m < 2^{-2^m}/\alpha$.
2. The jumping regions include $W(\mathbb{1} + \delta W)$, where W is diagonal, and (from eq. (2.15)) $\|\delta W\| \leq \delta = 1/(2\alpha d^2)$
3. Applying a random conjugation gives us (from eq. (2.17)) a probability of at least $p := (\delta/10)^{d^2-1}$ of landing in a jumping region.

As in the one-qubit case, we now introduce the function $g(U) = VUV^\dagger$, where each application of g chooses a unitary V uniformly at random according to the Haar measure on $\mathcal{SU}(d)$. Consider the function $F(U) = (f \circ g)^{\circ k}$, i.e. f and g composed k times. Following identical logic to the qubit case, we deduce that if

$$k \geq \frac{\log_2(1/\eta)}{\log_2(1/(1-p))} + \log_2\log_2\left(\frac{1}{\alpha\epsilon}\right) + 1, \tag{2.18}$$

where $\epsilon \leq \epsilon_0 \leq 1/(2\alpha)$, then U will be mapped to within ϵ distance of $\mathbb{1}$ with probability $\geq 1 - \eta$. As before, F can then be trivially expanded in the form of eq. (2.1) to give the required function.

The number of pulses n required for the full refocussing function F is the same as the number of uses of U, which is d^{2k}. Thus we see that the number of pulses looks

like

$$n = d^{2k} \leq d^2 \left(\frac{1}{\eta}\right)^{2^{O(d^4)}} \left(\log_2\left(\frac{1}{\epsilon}\right) - d^2 - 1\right)^{2\log_2 d},$$

where the multiplicative factor of d^2 comes from the fact that k may need to be rounded up to the nearest integer greater than the RHS of eq. (2.18). For fixed d, we see that this is similar to eq. (2.7) from the one-qubit case. With increasing d, we see that the dependence on ϵ increases only modestly (owing to the decrease in size of the shrinking region), but the number of steps required to maintain the probability of success, $1 - \eta$, increases doubly-exponentially in the Hilbert-space dimension.

2.5 Final remarks and open questions

As described in Section 2.2.2, refocussing techniques are generally limited in that they can only be applied to restricted sets of unitary operators. The procedures that we developed in Sections 2.3 and 2.4 can also be used to refocus states, and have the advantage that they can be applied to all unitary operators. Chapter 3 additionally explores the relevance of these results to the Solovay-Kitaev Theorem.

One may ask whether it is possible to have sequences where $\eta = 0$. Here, the randomness is important to our analysis. Moreover, the function f contains fixed points and cycles of various orders, and the random conjugations serve to break free of these. Indeed, we conjecture that there are cycles of all orders. However, it may be possible to avoid the random conjugations completely, as numerical simulations strongly suggest these cycles form a zero-measure subset of $\mathcal{SU}(d)$, and that the only stable fixed point of f is $\mathbb{1}$. This is also noted in [BG07], where it is observed that 'a few iterations [of f] are sufficient to reach a good approximation to the identity'. We leave rigorous proof of these conjectures as an interesting open problem.

In this chapter we have developed techniques for inverting a black-box unitary operator. A future research direction might be to understand what other functions of these operators is it possible to effect in a similar way. For example, would it be possible to effect an arbitrary power of a black-box unitary operator? Or perhaps a controlled version of the operator?

Chapter 3

Efficient Gate Approximation

3.1 The Solovay-Kitaev Theorem

Given a set of unitary operators, a *gate set*, one might wonder what operators can be generated from products of those in the gate set. For example, the qubit gates X and Z generate (up to phases) the Pauli group, $\{\mathbb{1}, X, Y, Z\}$.

For any unitary operator U, if an arbitrarily close approximation to U can be generated by the gate set, then the set is called a *universal gate set* (see Section 4.5 of [NC10]). An example of such a gate set is the set comprising the 1-qubit gates

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad R_{\pi/4} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix},$$

and the 2-qubit gate

$$C_X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Naturally, one might ask how many of these gates are required to approximate any given U, and whether this depends on the gate set that is being used. One of the central results in quantum compiling—the Solovay-Kitaev theorem [NC10, KSV02]—states that a universal quantum gate set *that includes inverse gates* can simulate any other universal gate set to arbitrary precision ϵ, with at most $O(\log^{3.97}(1/\epsilon))$ overhead [NC10, DN05]. This is fundamental to the theory of quantum circuits and to practical quantum computation, as it shows that any universal gate set can simulate any other with low overhead.

In a circuit of size L we can think of ϵ as $O(1/L)$, so changing from one universal gate set to another would increase the number of gates to at most $O(L \log^{3.97}(L))$. How-

ever, when inverse gates are not included, all known variants of the Solovay-Kitaev theorem [DN05, KSV02] fail. The only previously known method of approximating the inverse of a gate U was to wait until a member of the sequence $U, U^2, U^3, \ldots$ approximated U^{-1}, which in general required overhead $1/\epsilon^{d^2-1}$, i.e. $1/\epsilon^3$ for qubits. Thus, a circuit of size L would turn into poly(L) gates, which is a large enough overhead to overwhelm the polynomial speedup from algorithms such as Grover's.

More formally, the theorem is stated as follows:

Theorem 3.1.1 (Solovay-Kitaev Theorem)**.** *Let $\mathcal{G} \subseteq \mathcal{SU}(2)$ be a quantum gate set such that $\mathcal{G} \cup \mathcal{G}^\dagger$ is universal in $\mathcal{SU}(2)$, where $\mathcal{G}^\dagger := \{V^\dagger : V \in \mathcal{G}\}$. For any $\epsilon > 0$ and any $U \in \mathcal{SU}(2)$, there is an efficient classical algorithm that constructs a sequence of gates $V_L \cdots V_1 V_0$ with $V_i \in \mathcal{G} \cup \mathcal{G}^\dagger$ and $L = O(\log^{3.97}(1/\epsilon))$ such that $\|V_L \cdots V_1 V_0 - U\| \leq \epsilon$, where $\|\cdot\|$ is the Hilbert-Schmidt norm.*

For a proof of the theorem, we refer the reader to Appendix 3 of [NC10].

Note that the norm used in [NC10] is the trace norm, whereas we are using the Hilbert-Schmidt norm. But these are equivalent up to an unimportant factor of $\sqrt{2}$.

Remark: $\mathcal{G} \cup \mathcal{G}^\dagger$ is universal $\Leftrightarrow$ $\mathcal{G}$ is universal, as any element $V^\dagger \in \mathcal{G}^\dagger$ can be approximated to arbitrary accuracy by an integer power of $V \in \mathcal{G}$, as described above. In addition, the theorem can be scaled up from $\mathcal{SU}(2)$ to $\mathcal{SU}(d)$ for $d \geq 2$ [DN05], but at a cost that is exponential in d.

3.2 An 'inverse-free' Solovay-Kitaev Theorem

An application of our universal in-line inversion result is to extend the Solovay-Kitaev theorem to the case when inverse gates are not included.

By using our refocussing result to efficiently approximate inverse gates, we obtain a new inverse-free version of the theorem: Any universal quantum gate set that includes the Pauli operators (or Weyl operators for qudits) can simulate any other universal gate set to arbitrary precision ϵ, with at most $O(\log^{5.97}(1/\epsilon))$ overhead.

Definition 3.2.1 (η-net [NC10])**.** *A set S is an* η-net *for a set T if every element of T is within a distance η of an element of S.*

The following is the key lemma, using part of our refocussing result to show that inverses can be approximated efficiently:

Lemma 3.2.2. *Let Δ be a $\frac{1}{2}$-net for $\mathcal{SU}(2)$, and let $\mathcal{P} = \{\mathbb{1}, X, Y, Z\}$ be the set of Pauli operators (2-dimensional Weyl operators).*

For any ϵ and any $U \in \mathcal{SU}(2)$, there is an efficient classical algorithm that constructs a product of unitary operators $g_\epsilon(U)$ from the set $\Delta \cup \mathcal{P}$, of length $O(\log^2(1/\epsilon))$, for which $\|g_\epsilon(U) - U^\dagger\| \leq \frac{1}{\sqrt{2}}\epsilon$.

Proof. Since Δ is a $\frac{1}{2}$-net for $\mathcal{SU}(2)$, there exists a $W \in \Delta$ with $\|U^\dagger - W\| \leq \frac{1}{2}$, and hence $\|\mathbb{1} - WU\| \leq \frac{1}{2}$. Thus WU is in the shrinking region.

Let $f : \mathcal{SU}(2) \to \mathcal{SU}(2)$ be the mapping defined in eq. (2.2). By eq. (2.5), $\|f^m(WU) - \mathbb{1}\| \leq 2^{-\frac{1}{2}(2^m+1)}$. Setting $m = \log_2 \log_2(1/\epsilon^2)$, we have $\|f^m(WU) - \mathbb{1}\| \leq \frac{1}{\sqrt{2}}\epsilon$.

Now, $f^m(WU)$ is a sequence of unitary operators of the form

$$R_1 WUR_2 WU \cdots R_{L-1} WUR_L WU,$$

where the R_i are Pauli operators. By removing the trailing U from this sequence to form the sequence

$$g_\epsilon(U) := R_1 WUR_2 WU \cdots WUR_L W,$$

we see that $\|g_\epsilon(U) - U^\dagger\| = O(\epsilon)$ by unitary invariance of the norm.

$f^m(WU)$ has length $3 \times 4^m = O(\log^2(1/\epsilon))$, hence $g_\epsilon(U)$ also has length $O(\log^2(1/\epsilon))$. □

Putting Theorem 3.1.1 and Lemma 3.2.2 together, we obtain the inverse-free Solovay-Kitaev theorem:

Theorem 3.2.3 (Inverse-free Solovay-Kitaev)**.** *Let $\mathcal{G} \subseteq \mathcal{SU}(2)$ be a universal quantum gate set for $\mathcal{SU}(2)$ such that $\mathcal{P} \subseteq \mathcal{G}$, where $\mathcal{P} = \{\mathbb{1}, X, Y, Z\}$ is the set of Pauli operators.*

For any $\epsilon > 0$ and given any $U \in \mathcal{SU}(2)$, there is an efficient classical algorithm that constructs a sequence of gates $V_L \cdots V_1 V_0$ with $V_i \in \mathcal{G}$ and $L = O(\log^{5.97}(1/\epsilon))$ such that $\|V_L \cdots V_1 V_0 - U\| \leq \epsilon$.

Proof. We wish to apply Lemma 3.2.2 to elements of $\mathcal{G}^\dagger := \{V^\dagger : V \in \mathcal{G}\}$.

As $\mathcal{G}$ is universal, we can generate a $\frac{1}{2}$-net, denoted Δ, from constant-length products of operators from $\mathcal{G}$. One can see that constant-length products are sufficient as follows:

Given a set of unitary operators $\mathcal{U} = \{U_1, ..., U_N\}$, let us define

$$w(\mathcal{U}) := \max_{V \in \mathcal{SU}(2)} \min_{U \in \mathcal{U}} \|V - U\|,$$

$$v(L) := w(\{\text{set of all products of operators from } \mathcal{G} \text{ of length } L\}).$$

Clearly, $v(L) \leq v(L-1)$. In addition, since $\langle \mathcal{G} \rangle$ is dense in $\mathcal{SU}(2)$, $\lim_{L\to\infty} v(L) = 0$. In other words: for all $\delta > 0$ there exists an L such that $v(L) < \delta$.

Now, since $\mathcal{G}$ contains the Pauli operators, Lemma 3.2.2 allows us to construct a $O(\log^2(1/\epsilon))$-length product $g_\epsilon(V)$ of operators from $\mathcal{G}$ such that $\|g_\epsilon(V) - V^\dagger\| \leq \frac{1}{\sqrt{2}}\epsilon$.

Theorem 3.1.1 lets us construct a product of gates $\tilde{V}_K \cdots \tilde{V}_1 \tilde{V}_0$ with $\tilde{V}_i \in \mathcal{G} \cup \mathcal{G}^\dagger$ and $K = O(\log^{3.97}(1/\epsilon))$ such that $\|\tilde{V}_K \cdots \tilde{V}_1 \tilde{V}_0 - U\| \leq \epsilon/2$. We construct a new product of gates $V_L \cdots V_1 V_0$ with $V_i \in \mathcal{G}$ by replacing each $\tilde{V}_i \in \mathcal{G}^\dagger \setminus \mathcal{G}$ with $g_{\epsilon/(\sqrt{2}K)}(\tilde{V}_i^\dagger)$ (where $\tilde{V}_i^\dagger \in \mathcal{G}$).

Thus $\|V_L \cdots V_1 V_0 - \tilde{V}_K \cdots \tilde{V}_1 \tilde{V}_0\| \leq \epsilon/2$, and hence $\|V_L \cdots V_1 V_0 - U\| \leq \epsilon$.

Since we have replaced at most $K = O(\log^{3.97}(1/\epsilon))$ gates, we see that

$$L = K \cdot O(\log^2(\sqrt{2}K/\epsilon)) = O(\log^{5.97}(1/\epsilon)).$$

□

Remark: Theorem 3.2.3 is easily extended to $\mathcal{SU}(d)$ by using the d-dimensional generalisation of Theorem 3.1.1 [DN05], replacing the Pauli operators with the d-dimensional Weyl operators, and making use of the definition and bound from eq. (2.8) and eq. (2.12) respectively.

Part II

Quantum Distribution Testing

Chapter 4

Setting the scene: classical probability distribution testing

4.1 Introduction

The world is quickly moving towards a reality that is shaped by big data. In a statistics-fuelled environment, one of the most important challenges faced is inferring information about properties of large datasets as efficiently as possible. In particular, it would be desirable to determine properties of collected data without examining the entire datasets, which is typically infeasible. This can be formalised as the task of *property testing*: determining whether an object has a certain property, or is 'far' from having that property, ideally minimising the number of inspections of it. There has been an explosive growth in recent years in this field [GGR98, Gol10, BBM12], and particularly in the sub-field of *distribution testing*, in which one seeks to learn information about a data set by drawing samples from an associated probability distribution.

A few simple questions that one might ask could be:

- Are two probability distributions independent? For example, is the probability that a person likes movie B affected by whether or not they like movie A?
- Does a lottery machine really choose balls uniformly at random?
- Does the distribution of children's heights in the UK match the known distribution of children's heights in France?
- Are the distributions of the weights of adults the same, or different, for two randomly chosen countries?

Many of these problems, and others, have been extensively studied in the classi-

cal [BFR+10, VV11, CRS15, CR14, DKN15, GMV06, CDVV14, CFMdW10] literature, and near-optimal bounds have often been placed on the number of queries required to solve the respective problems.

We discuss these problems in more detail in Section 4.3.

4.2 Preliminaries & Notation

Let D be a discrete probability distribution over a finite set $[N] := \{0, 1, \ldots, N-1\}$, where $D(i) \geq 0$ is the weight of the element $i \in [N]$. Furthermore, if $S \subseteq [N]$, then $D(S) = \sum_{i \in S} D(i)$ is the weight of the set S. If $D(S) > 0$, define D_S to be the conditional distribution, i.e. $D_S(i) := D(i)/D(S)$ if $i \in S$ and $D_S(i) = 0$ if $i \notin S$.

The distance between two distributions $D^{(1)}$ and $D^{(2)}$ over $[N]$ is defined by the L_1-norm: $|D^{(1)} - D^{(2)}| = \sum_{i \in [N]} |D^{(1)}(i) - D^{(2)}(i)|$. We say that $D^{(1)}$ is ϵ-far from $D^{(2)}$ if $|D^{(1)} - D^{(2)}| \geq \epsilon$.

Algorithms for classical distribution testing often make use of the following two types of classical 'oracle'.

Definition 4.2.1 (Classical Sampling Oracle [CRS15])**.** *Given a probability distribution D over $[N]$, we define the* classical sampling oracle SAMP_D *as follows: each time SAMP_D is queried, it returns a single $i \in [N]$, where the probability that element i is returned is $D(i)$.*

Definition 4.2.2 (Classical Evaluation Oracle [CRS15])**.** *Given a probability distribution D over $[N]$, we define the* classical evaluation oracle EVAL_D *as follows:* EVAL_D *returns $D(i)$ when queried with argument $i \in [N]$.*

The complexity of algorithms for classical distribution testing is usually described by the total number of queries made to the SAMP oracle and queries to EVAL_D are ignored, as access to the EVAL oracle implies that we already 'know' the distribution. We typically present such complexities using big-O notation. Furthermore, the notation $\tilde{O}(f(N, \epsilon))$ denotes $O(f(N, \epsilon) \log^k f(N, \epsilon))$ for some k, i.e. logarithmic factors are hidden. Generally, the algorithms work with bounded probability, i.e. they output the correct answer with probability, say, at least $2/3$. This can be boosted arbitrarily close to 1 by repeating the test several times and taking a majority.

A Naïve Approach

It is quick to show (using a Chernoff bound—eq. (1) in [CRS15]) that up to an additive error ϵ, the weights of *all* elements $i \in [N]$ can be determined using just $O(1/\epsilon^2)$

queries to the SAMP_D oracle. However, for the problems we describe below, the additive error is required to be ϵ/N, and thus the number of queries required to solve these problems is naïvely $O(N^2/\epsilon^2)$.

Much work has been put into finding 'sub-linear' algorithms that can solve these problems using fewer than $O(N)$ queries, and we mention these in the discussion below.

4.3 Examples

Independence Test: Predicting movie preferences.

Suppose we had a large enough amount of data about two movies, A and B, in order to access the joint probability distribution D describing how many people watch these movies on any given day. One would like to find out if watching movie A affects whether or not a person will watch movie B. More generally, we ask: is D a product of two independent distributions, or are viewings of movie A correlated with viewings of movie B? That is, we must decide between:

- D is independent; i.e. D is a product of two distributions, $D = D^{(A)} \times D^{(B)}$; or
- D is ϵ-far from independent; i.e. it is ϵ-far from every product distribution.

If we assume that $D^{(A)}$ and $D^{(B)}$ are distributions over $[N]$ and $[M]$ respectively (with $M \leq N$), then naïve solutions to this problem, such as the Kolmogorov-Smirnoff test (see [BFF$^+$01]), require more than NM queries. The authors of [BFF$^+$01] develop an algorithm that uses only $\tilde{O}(N^{2/3}M^{1/3}\,\mathrm{poly}(\epsilon^{-1}))$ queries.

Uniformity Test: Lottery machine. A *gravity pick lottery machine* works as follows [CFGM16]: N balls, numbered $1, \ldots, N$, are dropped into a spinning machine, and after a few moments a ball is released. One might wish to determine whether or not such a machine is fair, i.e. whether or not a ball is released uniformly at random. We must decide between the following two options:

- The lottery machine is fair and outputs i with probability $1/N$;
- The lottery machine is ϵ-far from uniform.

Using the naïve approach described in Section 4.2 this would require $O(N^2/\epsilon^2)$ queries. However, a different algorithm is presented in [BFF$^+$01] (Theorem 17) that solves

this problem with $\tilde{O}(\sqrt{N}/\epsilon^4)$ queries, which is essentially optimal (see Section 6.3 of [BHH11]).

Identity Test: Height distribution of the UK population. Suppose that we have a detailed description of the distribution D^* (i.e. access to EVAL_{D^*}) governing the heights of children in France, and we suspect that the heights of children in the UK follow the same distribution. If we call the latter distribution D, we need to decide between

- $D = D^*$; and
- D is ϵ-far from D^*.

This can be solved naïvely using $O(N^2/\epsilon^2)$ queries to the SAMP_D oracle to approximate the weight of each element $i \in [N]$ to additive error $\epsilon/(2N)$, and N calls to the EVAL_{D^*} oracle. In [BFF+01], the authors give an algorithm to solve this problem using $\tilde{O}(\sqrt{N}\,\mathrm{poly}(\epsilon^{-1}))$ queries to SAMP_D (and some queries to EVAL_{D^*}).

Equivalence Test: Weight distribution of two countries. Suppose that we choose two countries (at random) and decide to test whether the distributions of the weights of adults are equal. Let $D^{(1)}$ be the distribution of adults' weights in the first country, and $D^{(2)}$ be the distribution in the second. We must decide between

- $D^{(1)} = D^{(2)}$; and
- $D^{(1)}$ is ϵ-far from $D^{(2)}$.

Once again this can be solved naïvely using $O(N^2/\epsilon^2)$ queries to $\mathsf{SAMP}_{D^{(1)}}$ and $\mathsf{SAMP}_{D^{(2)}}$ by approximating both $D^{(1)}$ and $D^{(2)}$ as we did for D in the previous example. In [BFR+10], the authors present an algorithm to solve this problem using $\tilde{O}(N^{2/3}\epsilon^{-8/3})$ queries to the oracles.

4.4 Conditional sampling

The *classical conditional sampling oracle* (COND) [ACK14, CRS15, CFGM16] grants access to a distribution D such that one can draw samples not only from D, but also from D_S, the conditional distribution of D restricted to an arbitrary subset S of the domain. Such oracle access reveals a separation between the classical query complexity of identity testing (i.e. whether an unknown distribution D is the same as some known distribution D^*), which takes a constant number of queries, and equivalence testing

(i.e. whether two unknown distributions D_1 and D_2 are the same), which requires $\tilde{O}((\log\log N)/\epsilon^5)$ queries, where N is the size of the domain [ACK14, FJO+15].

Definition 4.4.1 (Classical Conditional Sampling Oracle [CRS15])**.** *Given a probability distribution D over $[N]$ and a 'query subset' $S \subseteq [N]$ such that $D(S) > 0$, we define the* classical conditional sampling oracle COND_D *as follows: each time COND_D is queried with query set S, it returns a single $i \in [N]$, where the probability that element i is returned is $D_S(i)$.*

In other words, given such a set S, COND_D effects SAMP_{D_S}. We note here that COND_D encompasses SAMP_D, since a query to COND_D with $S = [N]$ effects SAMP_D.

In addition, we define the PCOND_D 'pairwise-COND' oracle, described in [CRS15], which is a simplification of the COND_D oracle, only accepting query subsets S of cardinality 2 or N.

4.4.1 Improved algorithms

In this section we consider the PCOND oracle, the simplest type of COND oracle. The results we present were developed in [CRS15], and greatly improve those given in Section 4.3.

Uniformity Test. Given a distribution D over $[N]$, one must decide between

- $D = \mathcal{A}$, where $\mathcal{A}$ is the uniform distribution, i.e. $\mathcal{A}(i) = 1/N$; or
- $|D - \mathcal{A}| \geq \epsilon$

There is an algorithm that decides between these two options using $\tilde{O}(1/\epsilon^2)$ (and in fact $\Omega(1/\epsilon^2)$) PCOND_D queries. To provide an understanding of the intuition behind this algorithm, a simpler procedure requiring $\tilde{O}(1/\epsilon^4)$ queries to the PCOND oracle is presented in Section A.1.

Identity Test. Given a distribution D and a 'known' distribution D^* (i.e. we have access to EVAL_{D^*}) over $[N]$, one must decide between

- $D = D^*$; or
- $|D - D^*| \geq \epsilon$

There is an algorithm that decides between these two options using $\tilde{O}\left[\left(\frac{\log N}{\epsilon}\right)^4\right]$ PCOND_D queries.

Equivalence Test. Given two distributions $D^{(1)}$ and $D^{(2)}$ over $[N]$, one must decide between

- $D^{(1)} = D^{(2)}$; or
- $|D^{(1)} - D^{(2)}| \geq \epsilon$

There is an algorithm that decides between these two options using $\tilde{O}\left[\left(\frac{\log^2 N}{\epsilon^2}\right)^3\right]$ $\mathsf{PCOND}_{D^{(1)}}$ and $\mathsf{PCOND}_{D^{(2)}}$ queries.

As can be seen from these results, even the PCOND oracle provides significant advantages over the standard classical sampling oracle. Even more drastic improvements on these complexities can be derived by adopting the full COND oracle, which can be seen in [CRS15, FJO$^+$15, ACK14].

4.4.2 Motivation for a conditional oracle

But does the COND oracle represent a natural model of access? Let us consider a few practical examples, some of which are explored in [FJO$^+$15, CRS15].

Bacterial growth. Controlled bacterial growth could be restricted in certain ways by changing environmental factors or by introducing a range of chemicals. This would only allow cells with particular characteristics to survive, thus conditioning the output distribution of the experiment. The COND oracle perhaps is a first step into understanding scenarios like these in more detail.

Lottery machine. We return to the example given in Section 4.3, in which one must decide if a lottery machine is fair. In this example, access to a COND oracle is equivalent to being able to choose which balls are allowed into the spinner. The number of queries required by the uniformity test algorithm given in Section 4.4.1 is independent of N, i.e. independent of the size of the lottery machine. This, therefore, is a practical method to efficiently test the fairness of lottery machines of any size.

Cloud-based computation. Cloud computing has become versatile and ubiquitous resource in recent years, allowing huge amounts of computation to be performed on dedicated and powerful servers, while the user works at a comparatively weak computer. As an example, cloud computing is currently used to provide users with cheap computers with the ability to play resource-heavy video games by off-loading all of the intensive computations to another server.

Suppose that a server has access to a set of data describing a probability distribution, and is powerful enough to instantly execute associated COND and PCOND queries. A user at a desktop computer could use this cloud-based scenario to ascertain properties of the distribution. Moreover, several individual users could have concurrent access to the data.

Chapter 5

Distribution testing using quantum algorithms

5.1 Introduction

Quantum computers are often described as being 'infinitely' powerful machines, able to instantly compute the solution to any problem. While this is far from the truth, they do exhibit advantages over classical computers in many scenarios. The field of quantum property testing involves using quantum computers to solve the decision problems described in Chapter 4, and is a new and exciting area of research [BHH11, MdW13, Mon15].

In Chapter 4 we introduced the classical sampling oracle (SAMP) and the classical conditional sampling oracle (COND) and discussed how they may be used to solve various distribution-testing problems. In this chapter we will introduce natural quantum versions of the SAMP and COND oracles. More specifically, we will introduce the PQCOND oracle—a quantum analogue of PCOND oracle—and study its computational power.

Consider the example of the lottery machine, first introduced in Section 4.3. The distribution testing algorithm must decide between the following options:

- The lottery machine is fair and outputs i with probability $1/N$;
- The lottery machine is ϵ-far from uniform.

As we noted in Section 4.3, the classical algorithms to decide this require

- $\tilde{O}(N^{1/2}/\epsilon^4)$ queries using the SAMP oracle;
- $\tilde{O}(1/\epsilon^2)$ queries using the PCOND oracle. In fact, in [CRS15] it was shown that $\Omega(1/\epsilon^2)$ queries to the PCOND oracle (and COND oracle) are required.

Using the quantum versions of these oracles, however, we find that there are algorithms that require

- $O(N^{1/3}/\epsilon^{4/3})$ queries using the QSAMP oracle [BHH11];
- $\tilde{O}(1/\epsilon)$ queries using the PQCOND oracle (see Section 5.4).

In this chapter we will develop quantum algorithms using the PQCOND oracle that are more efficient than their PCOND counterparts. We additionally show how these algorithms can be applied to property testing of Boolean functions. In Chapter 6, using these procedures, we develop an algorithm for testing whether or not a quantum state ρ is the fully mixed state.

5.2 Preliminaries & Notation

One might wonder how to define a quantum analogue of the classical sampling oracle SAMP. SAMP works by returning a random sample from the distribution each time it is accessed. What would this mean in a quantum setting? And can we define a quantum oracle with which we can take advantage of superposition?

The quantum sampling oracle (QSAMP) was defined in [BHH11], and this natural generalisation is derived in the following way: We first start by examining the classical sampling oracle in a different light. Suppose that we have a distribution D over $[N]$, and some specified integer $T \in \mathbb{N}$. We assume that D can be represented by a map $O_D : [T] \to [N]$, such that the weight $D(i)$ of any element $i \in [N]$ is proportional to the number of elements in the pre-image of i, i.e. $D(i) = |\{t \in [T] : O_D(t) = i\}|/T$. In other words, O_D labels the elements of $[T]$ by $i \in [N]$, and the $D(i)$ are the frequencies of these labels, and are thus all rational with denominator T. Then if $t \in [T]$ is chosen uniformly at random, $O_D(t)$ is a sample from the distribution D.

This notion of an oracle, which now requires an input $t \in [T]$, can be elevated to a quantum oracle in the usual way, resulting in the following definition:

Definition 5.2.1 (Quantum Sampling Oracle [BHH11])**.** *Given a probability distribution D over $[N]$, let $T \in \mathbb{N}$ be some specified integer, and assume that D can be represented by a mapping $O_D : [T] \to [N]$ such that for any $i \in [N]$, $D(i) = |\{t \in [T] : O_D(t) = i\}|/T$.*

Then each query to the quantum sampling oracle QSAMP_D *applies the unitary operation U_D, described by its action on basis states:*

$$U_D |t\rangle |\beta\rangle = |t\rangle |\beta + O_D(t) \bmod N\rangle .$$

In particular,

$$U_D |t\rangle |0\rangle = |t\rangle |O_D(t)\rangle .$$

As an example, note that querying with a uniformly random $t \in [T]$ in the first register will result in $i \in [N]$ in the second register with probability $D(i)$.

In Chapter 4, we also defined the classical conditional sampling oracle, COND_D. We are now ready to define our quantum analogue of this oracle, the *quantum conditional sampling oracle*, QCOND, and the related oracle PQCOND.

Definition 5.2.2 (Quantum Conditional Sampling Oracle)**.** *Given a probability distribution D over $[N]$, let $T \in \mathbb{N}$ be some specified integer, and assume that there exists a mapping $O_D : \mathcal{P}([N]) \times [T] \to [N]$, where $\mathcal{P}([N])$ is the power set of $[N]$, such that for any $S \subseteq [N]$ with $D(S) > 0$ and any $i \in [N]$, $D_S(i) = |\{t \in [T] : O_D(S,t) = i\}|/T$.*

Then each query to the quantum conditional sampling oracle QCOND_D *applies the unitary operation U_D, defined below.*

U_D acts on 3 registers:

- *The first consists of N qubits, whose computational basis states label the 2^N possible query sets S;*
- *The second consists of* $\log T$ *qubits that describe an element of $[T]$; and*
- *The third consists of* $\log N$ *qubits to store the output, an element of $[N]$.*

The action of the oracle on basis states is

$$U_D |S\rangle |t\rangle |\beta\rangle = |S\rangle |t\rangle |\beta + O_D(S,t) \bmod N\rangle .$$

In particular,

$$U_D |S\rangle |t\rangle |0\rangle = |S\rangle |t\rangle |O_D(S,t)\rangle .$$

An illustrative example of how the QCOND_D oracle works is given in Figure 5.1.

Remark: Note that querying QCOND_D with query set $S = [N]$ is equivalent to a query to QSAMP_D.

Remark: In the above definition we have made two key assumptions: the first is that the $D(i)$ are rational values; and the second is that O_D exists, which requires that the values $D_S(i)$ are consistent for different subsets $S \subseteq N$. These are strong promises on D and perhaps rather restrictive. However, by making T sufficiently large (requiring only $\log T$ qubits), we can approximate any probability distribution closely enough that the algorithms discussed in the remainder of this part of the thesis can still be applied (since the value of T does not affect the number of queries).

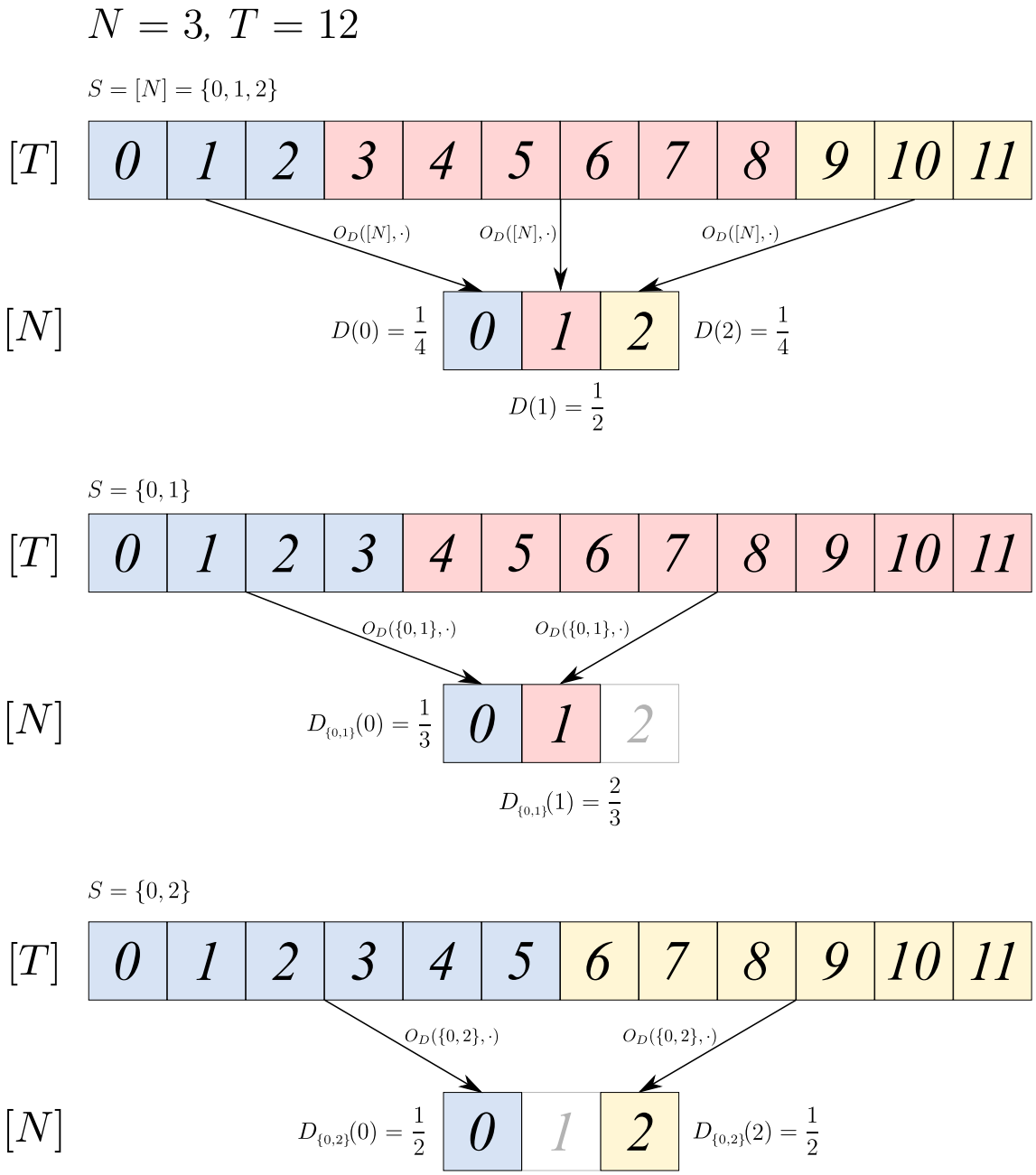


Figure 5.1: A diagram illustrating the O_D function described in the definition of the QCOND oracle, Definition 5.2.2, in the case where $N = 3$, $T = 12$ and $D(0) = 1/4$, $D(1) = 1/2$, $D(2) = 1/4$. In the topmost diagram, choosing t uniformly at random from $[T]$ and calculating $O_D([N], t)$ will give an element $i \in [N]$ with probability $D(i)$. For the lower two diagrams, calculating $O_D(S, t)$ (where $S = \{0, 1\}$ and $S = \{0, 2\}$ respectively) will give an element $i \in [N]$ with probability $D_S(i)$.

Definition 5.2.3 (Pairwise Quantum Conditional Sampling Oracle)**.** *The* PQCOND_D *oracle is equivalent to the* QCOND_D *oracle, with the added requirement that the query set* S *must satisfy* $|S| = 2$ *or* N*, i.e. the distribution can only be conditioned over pairs of elements or the whole set.*

5.3 Efficient comparison of conditional probabilities

In this section we first prove a lemma to improve the dependency on failure probability for a general probabilistic algorithm. We subsequently use this result to prove our main technical tool, the QCOMPARE algorithm, which compares conditional probabilities of a distribution, and is crucial to our improved property tests.

5.3.1 Improving dependence on success probability

The following lemma (described in [Mon15]) provides a general method for improving the dependence between the number of queries and the failure probability of the algorithm.

Lemma 5.3.1. *Suppose an algorithm (classical or quantum)* $\textsc{Alga}(\xi, \epsilon, \delta)$ $(\epsilon > 0, \delta \in (0,1])$ *outputs an (additive) approximation to* $f(\xi) \in \mathbb{R}$*. More formally, suppose it outputs* $\tilde{f}(\xi)$ *such that* $\mathbb{P}[|\tilde{f}(\xi) - f(\xi)| \leq \epsilon] \geq 1 - \delta$ *using* $M(\xi, \epsilon, \delta)$ *queries to a classical/quantum oracle, for some function* M*.*

Then there exists an algorithm $\textsc{Alg}\mathrm{B}(\xi, \epsilon, \delta)$ *that makes* $\Theta(M(\xi, \epsilon, \frac{1}{10}) \log(1/\delta))$ *queries to the same oracle and outputs* $\tilde{f}(\xi)$ *such that* $\mathbb{P}[|\tilde{f}(\xi) - f(\xi)| \leq \epsilon] \geq 1 - \delta$*, i.e. the dependence of the number of queries on the success probability can be taken to be* $\log(1/\delta)$*.*

Proof. We first state the procedure for $\textsc{Alg}\mathrm{B}(\xi, \epsilon, \delta)$ $(\epsilon > 0, \delta \in (0,1])$.

1. Run $\textsc{Alga}(\xi, \epsilon, \frac{1}{10})$ m times, where $m = \Theta(\log(1/\delta))$ (and such that m is even). Denote the outputs as $\tilde{f}_1, \ldots \tilde{f}_m$, labelled such that $\tilde{f}_1 \leq \cdots \leq \tilde{f}_m$.
2. Output $\tilde{f}_{m/2}$.

Consider $\textsc{Alga}(\xi, \epsilon, \frac{1}{10})$, and let E_1 be the event that $|\tilde{f}(\xi) - f(\xi)| \leq \epsilon$, which is equivalent to the event that $\tilde{f}(\xi) \in [f(\xi) - \epsilon, f(\xi) + \epsilon]$. Then we have that $\mathbb{P}[E_1] \geq \frac{9}{10}$.

Let Y be a random variable that takes the value 1 if E_1 occurs during a run of $\textsc{Alga}(\xi, \epsilon, \frac{1}{10})$, and 0 otherwise. Let $Y_1, \ldots, Y_m \sim Y$ be i.i.d. random variables. Let E_2 be the event that at least $\frac{8}{10}m$ of the Y_i output 1 (i.e. the event that E_1 occurs at least $\frac{8}{10}m$ times).

Using a Chernoff bound (here we use eq. (1) in [CRS15]), it is easy to see that $\mathbb{P}[E_2] \geq 1 - \exp(-\frac{1}{50}m)$.

Setting m to be a multiple of 2 such that $m \geq 50\log(1/\delta)$ then gives $\mathbb{P}[E_2] \geq 1 - \delta$.

Thus, we see that, with probability at least $1 - \delta$, Step 1 results in $\tilde{f}_1 \leq \cdots \leq \tilde{f}_m$ such that $|\tilde{f}_i - f(\xi)| \leq \epsilon$ for at least $\frac{8}{10}m$ values of $i \in \{1, \ldots, m\}$. Henceforth we assume that E_2 occurs. Now consider $\tilde{f}_{m/2}$. Suppose $\tilde{f}_{m/2} \notin [f(\xi) - \epsilon, f(\xi) + \epsilon]$. Then one of the two following statements must hold:

- $\tilde{f}_{m/2} < f(\xi) - \epsilon$. Since $\tilde{f}_1 \leq \cdots \leq \tilde{f}_{m/2}$, we have that $\tilde{f}_1, \ldots, \tilde{f}_{m/2} \notin [f(\xi) - \epsilon, f(\xi) + \epsilon]$, which contradicts the statement of E_2;
- $\tilde{f}_{m/2} > f(\xi) + \epsilon$. Since $\tilde{f}_{m/2} \leq \cdots \leq \tilde{f}_m$, we have that $\tilde{f}_{m/2}, \ldots, \tilde{f}_m \notin [f(\xi) - \epsilon, f(\xi) + \epsilon]$, which contradicts the statement of E_2.

Hence we conclude that $\tilde{f}_{m/2} \in [f(\xi) - \epsilon, f(\xi) + \epsilon]$. □

Remark: It is worth noting that the method used in the above proof could be applied to algorithms with multiplicative error, amongst others.

We additionally make use of Theorem 5 of [BHH11], which we recast below in our notation.

Theorem 5.3.2 (Theorem 5 of [BHH11])**.** *There exists a quantum algorithm* $\textsc{EstProb}(D, S, M)$ *that has* QSAMP *access to a distribution* D *over* $[N]$ *and takes as input a set* $S \subset [N]$ *and an integer* M*. The algorithm makes exactly* M *queries to the* QSAMP_D *oracle and outputs* $\tilde{D}(S)$*, an approximation to* $D(S)$*, such that* $\mathbb{P}[|\tilde{D}(S) - D(S)| \leq \epsilon] \geq 1 - \delta$ *for all* $\epsilon > 0$ *and* $\delta \in (0, 1/2]$ *satisfying*

$$M \geq \frac{c}{\delta} \max\left(\frac{\sqrt{D(S)}}{\epsilon}, \frac{1}{\sqrt{\epsilon}}\right),$$

where $c = O(1)$ *is some constant.*

Applying Lemma 5.3.1 to Theorem 5.3.2 gives an exponential improvement, from $1/\delta$ to $\log(1/\delta)$, in the dependence on the failure probability. This is summarised in the theorem below. Note that this changes the requirement that $\delta \in (0, 1/2]$ in Theorem 5.3.2 to $\delta \in (0, 1]$ in Theorem 5.3.3.

Theorem 5.3.3. *There exists a quantum algorithm* $\textsc{AddEstProb}(D, S, M)$ *that has* QSAMP *access to a distribution* D *over* $[N]$ *and takes as input a set* $S \subset [N]$ *and an integer* M*. The algorithm makes exactly* M *queries to the* QSAMP_D *oracle and outputs* $\tilde{D}(S)$*, an approximation to* $D(S)$*, such that* $\mathbb{P}[|\tilde{D}(S) - D(S)| \leq \epsilon] \geq 1 - \delta$ *for all* $\epsilon > 0$ *and* $\delta \in (0, 1]$ *satisfying*

$$M \geq c\log(1/\delta) \max\left(\frac{\sqrt{D(S)}}{\epsilon}, \frac{1}{\sqrt{\epsilon}}\right),$$

where $c = O(1)$ is some constant.

A multiplicative version of Theorem 5.3.3 follows straightforwardly:

Theorem 5.3.4. *There exists a quantum algorithm* MULESTPROB(D, S, M) *that has* QSAMP *access to a distribution D over $[N]$ and takes as input a set $S \subset [N]$ and an integer M. The algorithm makes exactly M queries to the* QSAMP_D *oracle and outputs $\tilde{D}(S)$, an approximation to $D(S)$, such that $\mathbb{P}[\tilde{D}(S) \in [1-\epsilon, 1+\epsilon]D(S)] \geq 1-\delta$ for all $\epsilon, \delta \in (0,1]$ satisfying*

$$M \geq \frac{c \log(1/\delta)}{\epsilon \sqrt{D(S)}},$$

where $c = O(1)$ is some constant.

Access to the QCOND_D oracle effectively gives us access to the oracle QSAMP_{D_S} for any $S \subseteq [N]$, and this allows us to produce stronger versions of Theorems 5.3.3 and 5.3.4:

Theorem 5.3.5. *There exists a quantum algorithm* ADDESTPROBQCOND(D, S, R, M) *that has* QCOND *access to a distribution D over $[N]$ and takes as input a set $S \subseteq [N]$ with $D(S) > 0$, a subset $R \subset S$ and an integer M. The algorithm makes exactly M queries to the* QCOND_D *oracle and outputs $\tilde{D}_S(R)$, an approximation to $D_S(R)$, such that $\mathbb{P}[|\tilde{D}_S(R) - D_S(R)| \leq \epsilon] \geq 1-\delta$ for all $\epsilon > 0$ and $\delta \in (0,1]$ satisfying*

$$M \geq c \log(1/\delta) \max\left(\frac{\sqrt{D_S(R)}}{\epsilon}, \frac{1}{\sqrt{\epsilon}}\right),$$

where $c = O(1)$ is some constant.

Theorem 5.3.6. *There exists a quantum algorithm* MULESTPROBQCOND(D, S, R, M) *that has* QCOND *access to a distribution D over $[N]$ and takes as input a set $S \subseteq [N]$ with $D(S) > 0$, a subset $R \subset S$ and an integer M. The algorithm makes exactly M queries to the* QCOND_D *oracle and outputs $\tilde{D}_S(R)$, an approximation to $D_S(R)$, such that $\mathbb{P}[\tilde{D}_S(R) \in [1-\epsilon, 1+\epsilon]D_S(R)] \geq 1-\delta$ for all $\epsilon, \delta \in (0,1]$ satisfying*

$$M \geq \frac{c \log(1/\delta)}{\epsilon \sqrt{D_S(R)}},$$

where $c = O(1)$ is some constant.

5.3.2 The QCOMPARE algorithm

An important routine used in many classical distribution testing protocols (see [CRS15]) is the COMPARE function, which outputs an estimate of the ratio $r_{X,Y} := D(Y)/D(X)$ of the weights of two disjoint subsets $X, Y \subset [N]$ over D. As stated in Section 3.1

of [CRS15], if X and Y are disjoint, $D(X \cup Y) > 0$, and $1/K \leq r_{X,Y} \leq K$ for some integer $K \geq 1$, the routine outputs $\tilde{r}_{X,Y} \in [1-\eta, 1+\eta] r_{X,Y}$ with probability at least $1-\delta$ using only $O(K \log(1/\delta)/\eta^2)$ COND_D queries. Surprisingly, the number of queries is independent of N, the size of the domain of the distribution.

Here we introduce a procedure called QCOMPARE that makes use of the QCOND_D oracle and subsequent quantum operations to perform the same function as COMPARE, whilst using only $O(\sqrt{K}\log(1/\delta)/\eta)$ queries. As the COMPARE algorithm cannot be trivially adapted to take advantage of the QCOND oracle, we devise a new algorithm for the QCOMPARE procedure, constructed from calls to ADDESTPROBQCOND and MULESTPROBQCOND. The proof of this algorithm does, however, make use of some of the ideas involved in the proof for the COMPARE procedure.

Algorithm 1 QCOMPARE$(D, X, Y, \eta, K, \delta)$

Input: QCOND access to a probability distribution D over $[N]$, disjoint subsets $X, Y \subset [N]$ such that $D(X \cup Y) > 0$, 'range' parameter $K \geq 1$, 'distance' parameter $\eta \in (0, \frac{3}{8K})$, and 'failure probability' $\delta \in (0, 1]$.

1. Set $M = O\left(\frac{\sqrt{K}\log(1/\delta)}{\eta}\right)$.
2. Set $\tilde{w}_+(X) =$ ADDESTPROBQCOND$(D, X \cup Y, X, M)$.
3. Set $\tilde{w}_+(Y) =$ ADDESTPROBQCOND$(D, X \cup Y, Y, M)$.
4. Set $\tilde{w}_\times(X) =$ MULESTPROBQCOND$(D, X \cup Y, X, M)$.
5. Set $\tilde{w}_\times(Y) =$ MULESTPROBQCOND$(D, X \cup Y, Y, M)$.
6. Check that $\tilde{w}_+(X) \leq \frac{3K}{3K+1} - \frac{\eta}{3}$. If the check fails, return Low and exit.
7. Check that $\tilde{w}_+(Y) \leq \frac{3K}{3K+1} - \frac{\eta}{3}$. If the check fails, return High and exit.
8. Return $\tilde{r}_{X,Y} = \frac{\tilde{w}_\times(Y)}{\tilde{w}_\times(X)}$.

Output: One of the following: a value $\tilde{r}_{X,Y} > 0$, the string High or the string Low.

Theorem 5.3.7. *Given the input as described,* QCOMPARE *(Algorithm 1) outputs* Low, High, *or a value* $\tilde{r}_{X,Y} > 0$, *and satisfies the following:*

1. *If* $1/K \leq r_{X,Y} \leq K$, *then with probability at least* $1-\delta$ *the procedure outputs a value* $\tilde{r}_{X,Y} \in [1-\eta, 1+\eta] r_{X,Y}$;
2. *If* $r_{X,Y} > K$ *then with probability at least* $1-\delta$ *the procedure outputs either* High *or a value* $\tilde{r}_{X,Y} \in [1-\eta, 1+\eta] r_{X,Y}$;
3. *If* $r_{X,Y} < 1/K$ *then with probability at least* $1-\delta$ *the procedure outputs either* Low *or a value* $\tilde{r}_{X,Y} \in [1-\eta, 1+\eta] r_{X,Y}$.

The procedure performs $O\left(\frac{\sqrt{K}\log(1/\delta)}{\eta}\right)$ QCOND_D *queries on the set* $X \cup Y$ *via use of* ADDESTPROBQCOND *and* MULESTPROBQCOND.

Proof. We prove this case-by-case. As in the proof of the COMPARE procedure [CRS15],

we introduce the shorthand $w(X) \coloneqq D_{X\cup Y}(X) = D(X)/D(X\cup Y), w(Y) \coloneqq D_{X\cup Y}(Y) = D(Y)/D(X\cup Y)$ and note that $r_{X,Y} = w(Y)/w(X)$. In addition, since $w(X) + w(Y) = 1$, it is straightforward to show the following inequalities for a constant $T \geq 1$:

$$\begin{aligned}
r_{X,Y} \geq \frac{1}{T} &\implies w(X) \leq \frac{T}{T+1},\ w(Y) \geq \frac{1}{T+1} \\
r_{X,Y} \leq \frac{1}{T} &\implies w(X) \geq \frac{T}{T+1},\ w(Y) \leq \frac{1}{T+1} \\
r_{X,Y} \geq T &\implies w(X) \leq \frac{1}{T+1},\ w(Y) \geq \frac{T}{T+1} \\
r_{X,Y} \leq T &\implies w(X) \geq \frac{1}{T+1},\ w(Y) \leq \frac{T}{T+1}
\end{aligned} \tag{5.1}$$

The strict versions of these inequalities also hold true.

1. $\boldsymbol{1/K \leq r_{X,Y} \leq K}$

   In this case we wish our algorithm to output $\tilde{r}_{X,Y} \in [1-\eta, 1+\eta] r_{X,Y}$.

   From eq. (5.1), we immediately have that

   $$\frac{1}{K+1} \leq w(X), w(Y) \leq \frac{K}{K+1}. \tag{5.2}$$

   Steps 2 and 3 use AddEstProbQCond to estimate $w(X)$ and $w(Y)$ to within additive error $\eta/3$ with probability at least $1-\delta/4$. As stated in Theorem 5.3.5, this requires

   $$O\left(\max\left(\frac{\sqrt{w(X)}}{\eta}, \frac{1}{\sqrt{\eta}}\right)\log(1/\delta)\right) = O\left(\frac{\log(1/\delta)}{\eta}\right)$$

   queries to QCOND_D, where the equality is due to the fact that $w(X) \leq 1$, and thus M (defined in Algorithm 1) queries suffice.

   Step 4 uses MulEstProbQCond to estimate $w(X)$ to within multiplicative error $\eta/3$ with probability at least $1-\delta/4$. From Theorem 5.3.6, we clearly require

   $$O\left(\frac{\log(1/\delta)}{\eta\sqrt{w(X)}}\right) = O\left(\frac{\sqrt{K}\log(1/\delta)}{\eta}\right)$$

   queries to QCOND_D in order to achieve these, where the equality is due to eq. (5.2), and thus M queries suffice. Step 5 requires the same number of queries.

   With a combined probability of at least $1-\delta$, Steps 2–5 all pass, and produce the following values:

   $$\begin{aligned}
   \tilde{w}_+(X) &\in [w(X) - \eta/3, w(X) + \eta/3], \\
   \tilde{w}_+(Y) &\in [w(Y) - \eta/3, w(Y) + \eta/3], \\
   \tilde{w}_\times(X) &\in [1-\eta/3, 1+\eta/3] w(X), \\
   \tilde{w}_\times(Y) &\in [1-\eta/3, 1+\eta/3] w(Y).
   \end{aligned}$$

From eq. (5.2), we see that

$$\tilde{w}_+(X), \tilde{w}_+(Y) \leq \frac{K}{K+1} + \frac{\eta}{3} < \frac{3K}{3K+1} - \frac{\eta}{3},$$

where the final inequality is due to the algorithm's requirement that $\frac{\eta}{3} < \frac{1}{8K}$.

Thus, the checks in Steps 6 and 7 pass, and Step 8 gives us

$$\tilde{r}_{X,Y} \in [1-\eta, 1+\eta] r_{X,Y}.$$

2. $\boldsymbol{K < r_{X,Y}}$

This is split into two sub-cases.

(a) $\boldsymbol{3K < r_{X,Y}}$

In this case we wish our algorithm to output High.

From eq. (5.1) we have that

$$w(X) < \frac{1}{3K+1}, \;\; w(Y) > \frac{3K}{3K+1}. \tag{5.3}$$

As in Case 1, Steps 2 and 3 allow us to gain

$$\tilde{w}_+(X) \in [w(X) - \eta/3, w(X) + \eta/3],$$
$$\tilde{w}_+(Y) \in [w(Y) - \eta/3, w(Y) + \eta/3],$$

with combined probability at least $1 - \delta/2$. (We henceforth assume that we have gained such values.)

Using eq. (5.3) it is easy to show that $\tilde{w}_+(X) < \frac{3K}{3K+1} - \frac{\eta}{3}$ and that $\tilde{w}_+(Y) > \frac{3K}{3K+1} - \frac{\eta}{3}$. Hence the check in Step 6 passes, but the check in Step 7 fails, and the algorithm outputs High and exits.

(b) $\boldsymbol{K < r_{X,Y} \leq 3K}$

In this case we wish our algorithm to either output High *or output* $\tilde{r}_{X,Y} \in [1-\eta, 1+\eta] r_{X,Y}$.

From eq. (5.1), we have that

$$\frac{1}{3K+1} \leq w(X) < \frac{1}{K+1}, \;\; \left(\frac{1}{3K+1} <\right) \frac{K}{1+K} < w(Y) \leq \frac{3K}{3K+1}. \tag{5.4}$$

Thus, with $O(\sqrt{K}\log(1/\delta)/\eta)$ queries, as in Case 1, we gain

$$\tilde{w}_+(X) \in [w(X) - \eta/3, w(X) + \eta/3],$$
$$\tilde{w}_+(Y) \in [w(Y) - \eta/3, w(Y) + \eta/3],$$
$$\tilde{w}_\times(X) \in [1 - \eta/3, 1 + \eta/3] w(X),$$
$$\tilde{w}_\times(Y) \in [1 - \eta/3, 1 + \eta/3] w(Y),$$

with combined probability at least $1 - \delta$. (We henceforth assume that we have gained such values.)

Using eq. (5.4), we see that $\tilde{w}_+(X) < \frac{3K}{3K+1} - \frac{\eta}{3}$, and thus Step 6 will pass.

Assuming the check in Step 7 passes, Step 8 will output $\tilde{r}_{X,Y} \in [1-\eta, 1+\eta] r_{X,Y}$.

However, given the upper bound for $w(Y)$ in eq. (5.4), it is possible to have $\tilde{w}_+(Y) > \frac{3K}{3K+1} - \frac{\eta}{3}$, causing the check in Step 7 to fail and the algorithm to output High.

3. $\boldsymbol{r_{X,Y} < 1/K}$

   This is split into two sub-cases.

   (a) $\boldsymbol{r_{X,Y} < 1/(3K)}$

   This is equivalent to the condition that $3K < r_{Y,X}$, and thus follows the same argument as Case 2a, with X and Y interchanged and an output of Low instead of High.

   (b) $\boldsymbol{1/(3K) \leq r_{X,Y} < 1/K}$

   This is equivalent to the condition that $K < r_{Y,X} \leq 3K$, and thus follows the same argument as Case 2b, with X and Y interchanged and an output of Low instead of High.

□

5.4 Property testing of probability distributions

We now apply our results to obtain new algorithms for a number of property testing problems.

Corollary 5.4.1. *Let $\mathcal{A}^{(N)}$ be the uniform distribution on $[N]$ (i.e. $\mathcal{A}^{(N)}(i) = 1/N, i \in [N]$). Given* PQCOND *access to a probability distribution D over $[N]$, there exists an algorithm that uses $\tilde{O}(1/\epsilon)$ PQCOND_D queries and decides with probability at least $2/3$ whether*

- $|D - \mathcal{A}^{(N)}| = 0$ *(i.e. $D = \mathcal{A}^{(N)}$) (the algorithm outputs* Equal*), or*
- $|D - \mathcal{A}^{(N)}| \geq \epsilon$ *(the algorithm outputs* Far*),*

provided that it is guaranteed that one of these is true.

Proof. We replace the calls to COMPARE with the corresponding calls to QCOMPARE in Algorithm 4 of [CRS15]. For this method, calls to QCOMPARE only require condi-

tioning over pairs of elements, and hence the PQCOND_D oracle may be used instead of QCOND_D. □

Remark: The corresponding classical algorithm (Algorithm 4 in [CRS15]) uses $\tilde{O}(1/\epsilon^2)$ PCOND_D queries. The authors also show (Section 4.2 of [CRS15]) that any classical algorithm making COND_D queries must use $\Omega(1/\epsilon^2)$ queries to solve this problem with bounded probability. Thus the above quantum algorithm is quadratically more efficient than *any* classical COND algorithm.

To provide an understanding of the intuition behind Corollary 5.4.1, a simpler algorithm requiring $\tilde{O}(1/\epsilon^3)$ queries to the PQCOND oracle is presented in Section A.1.

Corollary 5.4.2. *Given the full specification of a probability distribution D^* (i.e. a* known *distribution) and* PQCOND *access to a probability distribution D, both over $[N]$, there exists an algorithm that uses $\tilde{O}\left(\frac{\log^3 N}{\epsilon^3}\right)$ PQCOND_D queries and decides with probability at least $2/3$ whether*

- $|D - D^*| = 0$ *(i.e. $D = D^*$), or*
- $|D - D^*| \geq \epsilon$,

provided that it is guaranteed that one of these is true.

Proof. We replace the calls to COMPARE with the corresponding calls to QCOMPARE in Algorithm 5 of [CRS15]. □

Remark: The corresponding classical algorithm (Algorithm 5 in [CRS15]) uses $\tilde{O}\left(\frac{\log^4 N}{\epsilon^4}\right)$ PCOND_D queries.

Corollary 5.4.3. *Given* PQCOND *access to probability distributions $D^{(1)}$ and $D^{(2)}$ over $[N]$, there exists an algorithm that decides, with probability at least $2/3$, whether*

- $|D^{(1)} - D^{(2)}| = 0$ *(i.e. $D^{(1)} = D^{(2)}$), or*
- $|D^{(1)} - D^{(2)}| \geq \epsilon$,

provided that it is guaranteed that one of these is true. The algorithm uses $\tilde{O}\left(\frac{\log^4 N}{\epsilon^{14}}\right)$ $\mathsf{PQCOND}_{D^{(1)}}$ and $\mathsf{PQCOND}_{D^{(2)}}$ queries.

Proof. We replace the calls to COMPARE with the corresponding calls to QCOMPARE in Algorithm 9 of [CRS15]. □

Remark: The corresponding classical algorithm (Algorithm 9 in [CRS15]) uses $\tilde{O}\left(\frac{\log^6 N}{\epsilon^{21}}\right)$ $\mathsf{PCOND}_{D^{(1)}}$ and $\mathsf{PCOND}_{D^{(2)}}$ queries.

Corollary 5.4.4. *Given* PQCOND *access to a probability distribution D over $[N]$, there exists an algorithm that uses $\tilde{O}(1/\epsilon^{13})$ queries and outputs a value $\hat{d}$ such that $|\hat{d} - |D - \mathcal{A}^{(N)}|| = O(\epsilon)$.*

Proof. We replace the calls to COMPARE with the corresponding calls to QCOMPARE in Algorithm 11 of [CRS15]. In addition, we trivially replace all queries to the SAMP_D oracle with queries to PQCOND_D with query set $[N]$. □

Remark: The corresponding classical algorithm (Algorithm 11 in [CRS15]) uses $\tilde{O}(1/\epsilon^{20})$ queries.

5.5 Property testing of Boolean functions

The results in Section 5.4 can be applied to test properties of Boolean functions. One challenge in the field of cryptography is determining whether or not a given boolean function is 'balanced'. We present an algorithm to solve this problem with a constant number of PQCOND queries.

Consider a function $f : \{0,1\}^n \to \{0,1\}^m$, for $n, m \in \mathbb{N}$ with $n \geq m$. If $m = 1$, we might consider the following problem:

Problem 5.5.1 (Constant-balanced problem)**.** *Given $f : \{0,1\}^n \to \{0,1\}$, decide whether*

- *f is a balanced function, i.e. $|\{x \in \{0,1\}^n : f(x) = 0\}| / 2^n = |\{x \in \{0,1\}^n : f(x) = 1\}| / 2^n = \frac{1}{2}$, or*
- *f is a constant function, i.e. $f(x) = 0 \ \forall x \in \{0,1\}^n$ or $f(x) = 1 \ \forall x \in \{0,1\}^n$,*

provided that it is guaranteed that f satisfies one of these conditions.

With standard quantum oracle access to f, this problem can be solved exactly with one query, through use of the Deutsch-Jozsa algorithm [CEMM98, DJ92]. Consider the following extension of this problem:

Problem 5.5.2. *Given $f : \{0,1\}^n \to \{0,1\}$, write $F_i := |\{x \in \{0,1\}^n : f(x) = i\}| / 2^n$. Decide whether*

- *f is a balanced function, i.e. $F_0 = F_1 = \frac{1}{2}$, or*
- *f is ϵ-far from balanced, i.e. $\left|F_0 - \frac{1}{2}\right| + \left|F_1 - \frac{1}{2}\right| = 2\left|F_0 - \frac{1}{2}\right| \geq \epsilon$,*

provided that it is guaranteed that f satisfies one of these conditions.

This problem can be solved classically with bounded probability by querying f $O(1/\epsilon^2)$ times to estimate F_0 to error $\epsilon/3$, described as the 'naïve approach' in Section 4.2.

Now we consider an even more general problem:

Problem 5.5.3. *Given $f : \{0,1\}^n \to \{0,1\}^m$, write $F_i := |\{x \in \{0,1\}^n : f(x) = i\}| / 2^n$. Decide whether*

- *f is a balanced function, i.e. $F_i = \frac{1}{2^m} \quad \forall i \in \{0,1\}^m$, or*
- *f is ϵ-far from balanced, i.e. $\sum_{i \in \{0,1\}^m} \left| F_i - \frac{1}{2^m} \right| \geq \epsilon$,*

provided that it is guaranteed that f satisfies one of these conditions.

By allowing PQCOND access to f, this can be solved in $\tilde{O}(1/\epsilon)$ queries. In what sense do we allow PQCOND access to f? We relate f to a probability distribution by setting $N = 2^m$, $D(i) = F_i$ (i.e. D is the probability distribution formed from the image of f), and using the definition of $D_S(i)$ given at the start of Section 5.2. The problem is now a question of uniformity testing, and is solved by an application of the algorithm presented in Corollary 5.4.1. Using the standard quantum oracle QSAMP, this problem requires $\Omega(2^{m/3})$ queries (a lower bound for uniformity testing given in [CFMdW09, BHH11]).

The problem does not naturally lend itself to the classical COND model, as our solution makes use of the mapping O_D (see Definition 5.2.1). Using the standard classical sampling oracle SAMP, this problem requires $\Omega(2^{m/2})$ queries [BFF$^+$01].

Chapter 6

Mixedness Testing

6.1 Introduction

The problem of deciding if a quantum state ρ of dimension n is maximally-mixed or ϵ-far from it is a natural question within the framework of Quantum Spectrum Testing. In the standard model, where one measures a number of copies of ρ, it is found that $\Theta(n/\epsilon^2)$ measurements are needed [OW15]. It is of interest to study whether the PQCOND model, a more powerful model, can improve the complexity of solving this problem, without trivialising it.

Formally, the Mixedness problem is stated as follows: Given access to copies of a quantum state ρ of dimension n and a constant $\epsilon > 0$, it is promised that one of the following holds:

- $\|\rho - \mathbb{1}/n\|_1 = 0$, i.e. ρ is the maximally-mixed state; or
- $\|\rho - \mathbb{1}/n\|_1 \geq \epsilon$, i.e. ρ is ϵ-far from the maximally-mixed state,

where $\|\cdot\|_1$ is the trace norm[1]. Decide which is the case.

In this chapter we prove the following theorem about the complexity of the Mixedness problem:

Theorem 6.1.1. *Given an n-dimensional quantum state $\rho \in \mathbb{C}^n \times \mathbb{C}^n$ and a basis $\mathcal{B} = \{|b_i\rangle\}_{i\in[n]}$ where n is even, let $D_{[n]}^{(\rho,\mathcal{B})}$ be the probability distribution over $[n]$ defined by*

$$D_{[n]}^{(\rho,\mathcal{B})}(i) := \mathrm{Tr}(\rho\,|b_i\rangle\,\langle b_i|\,) = \langle b_i|\,\rho\,|b_i\rangle\,.$$

For any $\epsilon > 0$, there exists an algorithm that solves the Mixedness problem for ρ with probability at least $2/3$ using $\tilde{O}(n/\epsilon)$ PQCOND queries to $D_{[n]}^{(\rho,\mathcal{B})}$ (where each query may involve a different $\mathcal{B}$). The algorithm outputs MaximallyMixed *if ρ is the maximally-mixed state, and* NotMaximallyMixed *otherwise.*

[1] For an $(n \times n)$ matrix A, $\|A\|_1 = \mathrm{Tr}\sqrt{AA^\dagger} = \sum_{i\in[n]} a_i$, where the a_i are the singular values of A.

We additionally prove and make use of the following theorem and lemmas, which may be of independent interest.

Theorem 6.1.2. *Given an n-dimensional quantum state $\rho \in \mathbb{C}^n \times \mathbb{C}^n$ satisfying*

$$\|\rho - \mathbb{1}/n\| \geq \epsilon$$

for some $\epsilon > 0$ and where n is even, we choose a basis $\mathcal{B} = \{|b_i\rangle\}_{i\in[n]}$ uniformly at random (i.e. if $\{|\tilde{b}_i\rangle\}_{i\in[n]}$ is a fixed basis, then we choose $W \in \mathcal{U}(n)$ uniformly at random according to the Haar measure, and set $|b_i\rangle = W|\tilde{b}_i\rangle \ \forall i$).

Define

$$\delta^{(\mathcal{B})} := \left| D_{[n]}^{(\rho,\mathcal{B})} - \mathcal{A}^{(n)} \right|, \tag{6.1}$$

where $\mathcal{A}^{(n)}$ is the uniform distribution over $[n]$.

Then

$$\mathbb{P}\left[\delta^{(\mathcal{B})} \geq \frac{\min(1,\epsilon)}{8\sqrt{n}}\right] \geq \frac{1}{8\sqrt{n}}.$$

Lemma 6.1.3. *Define*

$$M^{(d)} := \max_{\sigma\in Sym([n])} \left| d_{\sigma(0)} - d_{\sigma(1)} + d_{\sigma(2)} - \cdots - d_{\sigma(n-1)} \right|,$$

where n is even, $Sym([n])$ is the symmetric group on $[n]$, and $d = (d_0, d_1, \ldots, d_{n-1}) \in \mathbb{R}^n$ for $n \geq 2$ and satisfies

$$\sum_{i\in[n]} d_i = 0, \qquad \sum_{i\in[n]} |d_i| = \eta.$$

Then $M^{(d)} \geq \frac{1}{2}\eta$.

Lemma 6.1.4. *Let*

$$T_n := \{(v_0, \ldots, v_{n-1}) : v_i \in [0,1], \textstyle\sum_{i\in[n]} v_i = 1\}$$

be the probability simplex in n dimensions with associated probability measure

$$dV := (n-1)!\ \delta\left[1 - \sum_{i\in[n]} v_i\right]\ dv_0 \cdots dv_{n-1}.$$

Then the quantity

$$\begin{aligned} E_n &:= \mathbb{E}(|v_0 - v_1 + v_2 - \cdots - v_{n-1}|) \\ &= \int_{T_n} |v_0 - v_1 + v_2 - \cdots - v_{n-1}|\ dV \end{aligned} \tag{6.2}$$

has the lower bound

$$E_n \geq \frac{1}{2\sqrt{n}},$$

and for large n,

$$E_n \sim \sqrt{\frac{2}{\pi}}\frac{1}{\sqrt{n}}.$$

Algorithm 2 MAXIMALLYMIXEDSTATETEST(ρ)

Input: PQCOND access to a probability distribution $D_{[n]}^{(\rho,\mathcal{B})}$ over $[n]$ for any $\mathcal{B}$, as described in Theorem 6.1.1, and parameter ϵ. Set $l = 128 \log n$.

1. Choose $k = 32\sqrt{n}$ bases $\mathcal{B}_1, \ldots \mathcal{B}_k$ uniformly at random.
2. For each $j = 1, \ldots, k$, run the algorithm given in Corollary 5.4.1 on the distribution $D_{[n]}^{(\rho,\mathcal{B}_j)}$ l times with distance parameter $\frac{\min(1,\epsilon)}{8\sqrt{n}}$, returning $u_j = 1$ if at least $\frac{1}{2}l$ of the runs return Far, and $u_j = 0$ otherwise.
3. If $u_j = 0$ $\forall j$, output MaximallyMixed, otherwise output NotMaximallyMixed.

Output: Either the string MaximallyMixed or the string NotMaximallyMixed.

A sub-linear algorithm

From numerical calculations, it is clear that $\tilde{O}(\sqrt{n}/\epsilon)$ queries (rather than $\tilde{O}(n/\epsilon)$ queries, as stated in Theorem 6.1.1) will be sufficient to solve the Mixedness problem. In Appendix A.2 we give a proof of this result, barring a small conjecture that is currently open.

6.2 Proof of Theorem 6.1.1

The full algorithm is set out in Algorithm 2.

Firstly, we note that in Step 2, each run of the algorithm given in Corollary 5.4.1 requires $\tilde{O}(\sqrt{n}/\epsilon)$ PQCOND queries if $\epsilon \leq 1$, and hence in total Algorithm 2 requires

$$\tilde{O}\left(kl\frac{\sqrt{n}}{\epsilon}\right) = \tilde{O}\left(\frac{n}{\epsilon}\right)$$

PQCOND queries, as claimed.

Now, let $\mathcal{A}^{(n)}$ be the uniform distribution over $[n]$, and introduce $\delta^{(\mathcal{B})} := \left| D_{[n]}^{(\rho,\mathcal{B})} - \mathcal{A}^{(n)} \right|$, as in eq. (6.1). Let $k = 32\sqrt{n}$ and $l = 128 \log n$ as specified in Algorithm 2. We analyse each case of the algorithm separately.

The case when $\|\rho - \mathbb{1}/n\|_1 = 0$

It is easy to see that $\delta^{(\mathcal{B})} = 0$ for any basis $\mathcal{B}$, and hence that each run of Corollary 5.4.1 in Step 2 will output Equal with probability at least $\frac{2}{3}$. By using a Chernoff bound (eq. (1) in [CRS15]), it follows that for each j, $\mathbb{P}[u_j = 1] \leq e^{-l/18}$. By the union bound[2], we

[2]For a countable set of events $A_1, A_2, \ldots$, we have that $\mathbb{P}\left[\bigcup_i A_i\right] \leq \sum_i \mathbb{P}[A_i]$.

have that $\mathbb{P}\left[\text{at least one of the } u_j \text{ is } 1\right] \leq ke^{-l/18} \leq \frac{1}{3}$, and hence the algorithm will output MaximallyMixed in Step 3 with probability at least $\frac{2}{3}$.

The case when $\|\rho - \mathbb{1}/n\|_1 \geq \epsilon$

From Theorem 6.1.2, we know that if the basis $\mathcal{B}$ is chosen uniformly at random,

$$\mathbb{P}\left[\delta^{(\mathcal{B})} \geq \frac{\min(1,\epsilon)}{8\sqrt{n}}\right] \geq \frac{1}{8\sqrt{n}}.$$

Suppose we choose k different bases $\mathcal{B}_1, \ldots, \mathcal{B}_k$ uniformly at random. We call $\mathcal{B}$ 'good' if $\delta^{(\mathcal{B})} \geq \frac{\min(1,\epsilon)}{8\sqrt{n}}$. Let $K(k)$ represent the event that at least one of $\mathcal{B}_1, \ldots, \mathcal{B}_k$ is 'good'. Then

$$\mathbb{P}[K(k)] \geq 1 - \left(1 - \frac{1}{8\sqrt{n}}\right)^k \geq 1 - \frac{1}{e^4} \geq \frac{49}{50}.$$

Suppose now that $\mathcal{B}_{k^*}$ is 'good'. Each time the algorithm given in Corollary 5.4.1 is performed on $D_{[n]}^{(\rho,\mathcal{B}_{k^*})}$ in Step 2, it will output Far with probability at least $\frac{2}{3}$. Using a Chernoff bound (eq. (1) in [CRS15]), we see that with probability at least $1 - e^{-l/18} > \frac{99}{100}$ we have $u_{k^*} = 1$, and the algorithm outputs NotMaximallyMixed in Step 3.

Combining these two events, we see that the probability that Algorithm 2 outputs NotMaximallyMixed in Step 3 is at least $\frac{49}{50} \cdot \frac{99}{100} > \frac{2}{3}$. □

Remark: We may consider what the complexity would be if we allowed access only to the QSAMP oracle, rather than the PQCOND oracle. In this case, instead of Corollary 5.4.1, Step 2 would make use of the essentially optimal uniformity testing algorithm in [BHH11], the complexity of which is given in Table 1. This would result in an algorithm to solve the Mixedness problem with a total complexity of $\tilde{O}(n^{3/2}/\epsilon^{4/3})$.

6.3 Proof of Theorem 6.1.2

First, note that

$$\delta^{(\mathcal{B})} = \left|D_{[n]}^{(\rho,\mathcal{B})} - \mathcal{A}^{(n)}\right| = \sum_{i\in[n]} |\langle b_i| \Delta |b_i\rangle|.$$

Now let $\tilde{\mathcal{B}} = \left\{\left|\tilde{b}_i\right\rangle\right\}_{i\in[n]}$ be the eigenbasis of Δ, and let $d_i := \left\langle \tilde{b}_i\right| \Delta \left|\tilde{b}_i\right\rangle$, $i \in [n]$ be the eigenvalues. Thus, $\Delta = \sum_{i\in[n]} d_i \left|\tilde{b}_i\right\rangle \left\langle \tilde{b}_i\right|$. Note that $\operatorname{Tr}\Delta = \sum_{i\in[n]} d_i = 0$, and also $\eta := \|\rho - \mathbb{1}/n\|_1 = \|\Delta\|_1 = \sum_{i\in[n]} |d_i| \geq \epsilon$.

Now suppose we choose a basis $\mathcal{B} = \{|b_i\rangle\}_{i\in[n]}$ uniformly at random, i.e. we choose $W \in \mathcal{U}(n)$ uniformly at random according to the Haar measure, and set $|b_i\rangle = W\left|\tilde{b}_i\right\rangle$. Then

$$\delta^{(\mathcal{B})} = \sum_{i\in[n]} |\langle b_i| \Delta |b_i\rangle| = \sum_{i\in[n]} \left| \sum_{j\in[n]} |W_{ji}|^2 d_j \right|,$$

where $W_{ji} \coloneqq \left\langle \tilde{b}_j \right| W \left|\tilde{b}_i\right\rangle$.

The triangle inequality gives

$$\begin{aligned} \delta^{(\mathcal{B})} &\geq \left| \sum_{j\in[n]} \left(\sum_{i\in[n]} |W_{ji}|^2 \right) d_j \right| = \left| \sum_{j\in[n]} d_j \right| = 0, \\ \delta^{(\mathcal{B})} &\leq \sum_{j\in[n]} \left(\sum_{i\in[n]} |W_{ji}|^2 \right) |d_j| = \eta. \end{aligned} \tag{6.3}$$

Let $v_j^{(i)} = |W_{ji}|^2$, introduce the vector $V^{(i)} = (v_0^{(i)}, \ldots, v_{n-1}^{(i)})$, and write $d = (d_0, \ldots, d_{n-1})$. Then

$$\delta^{(\mathcal{B})} = \sum_{i\in[n]} |V^{(i)} \cdot d|.$$

We now make use of Sykora's theorem [Sỳk74, DDJB14], which states that if W is chosen uniformly at random according to the Haar measure on $\mathcal{U}(n)$, then the vector $V^{(i)}$, for any i, is uniformly distributed over the probability simplex

$$T_n = \{(v_0, \ldots, v_{n-1}) : v_i \in [0,1],\ \textstyle\sum_{i\in[n]} v_i = 1\}.$$

Since all of the $V^{(i)}$'s have the same distribution, we see from eq. (1.2) that

$$\mathbb{E}\left(\delta^{(\mathcal{B})}\right) = n\mathbb{E}(|V \cdot d|), \tag{6.4}$$

where V is a generic $V^{(i)}$.

The following lemma allows us to relate a lower bound on $\mathbb{E}\left(\delta^{(\mathcal{B})}\right)$ to a lower bound on $\mathbb{P}[\delta^{(\mathcal{B})} \geq \lambda]$, for some λ.

Lemma 6.3.1.

$$\mathbb{P}\left[\delta^{(\mathcal{B})} \geq \lambda\right] \geq \frac{1}{\eta}\left(\mathbb{E}\left(\delta^{(\mathcal{B})}\right) - \lambda\right)$$

Proof. Let $p = p(\mu)$ be the probability density function for $\delta^{(\mathcal{B})}$. As noted in eq. (6.3),

$0 \leq \delta^{(\mathcal{B})} \leq \eta$. Thus, for $\lambda \in [0, \eta]$ we can write

$$\begin{aligned}\mathbb{E}\left(\delta^{(\mathcal{B})}\right) &= \int_0^\eta \mu p(\mu)\, d\mu \\ &= \int_0^\lambda \mu p(\mu)\, d\mu + \int_\lambda^\eta \mu p(\mu)\, d\mu \\ &\leq \int_0^\lambda \lambda p(\mu)\, d\mu + \int_\lambda^\eta \eta p(\mu)\, d\mu \\ &\leq \lambda + \eta \mathbb{P}\left[\delta^{(\mathcal{B})} \geq \lambda\right].\end{aligned}$$

Rearranging the inequality gives the result. □

We now write $\mathbb{E}(|V \cdot d|)$ as an integral over the probability simplex T_n. We have

$$\begin{aligned}\mathbb{E}(f(V)) &= \int_{T_n} f(V) dV \\ &= (n-1)! \int_{v_0=0}^1 \cdots \int_{v_{n-1}=0}^1 \delta(1 - \textstyle\sum_{i\in[n]} v_i) f(V)\, dv_0 \cdots dv_{n-1}. \end{aligned} \tag{6.5}$$

where $dV = (n-1)!\, \delta(1 - \sum_{i\in[n]} v_i)\, dv_0 \cdots dv_{n-1}$ is the normalised measure on T_n, defined so that $\mathbb{E}(1) = 1$, and $\delta(\cdots)$ is the Dirac delta.

Note that the integral expression for $\mathbb{E}(|V \cdot d|) = \mathbb{E}(|v_0 d_0 + \cdots + v_{n-1} d_{n-1}|)$ is completely symmetric in the v_i's (and hence in the d_i's). Thus, if σ is a permutation on $[n]$, we have that

$$\mathbb{E}(|v_0 d_0 + \cdots + v_{n-1} d_{n-1}|) = \mathbb{E}(|v_0 d_{\sigma(0)} + \cdots + v_{n-1} d_{\sigma(n-1)}|).$$

Using this observation, we can write

$$\begin{aligned}&\mathbb{E}(|v_0 d_0 + \cdots + v_{n-1} d_{n-1}|) \\ =& \frac{1}{n}\Big[\mathbb{E}(|v_0 d_{\sigma(0)} + \cdots + v_{n-1} d_{\sigma(n-1)}|) + \mathbb{E}(|v_0 d_{\sigma(1)} + \cdots + v_{n-1} d_{\sigma(0)}|) \\ &\quad + \mathbb{E}(|v_0 d_{\sigma(2)} + \cdots + v_{n-1} d_{\sigma(1)}|) + \cdots + \mathbb{E}(|v_0 d_{\sigma(n-1)} + \cdots + v_{n-1} d_{\sigma(n-2)}|)\Big] \\ =& \frac{1}{n}\Big[\mathbb{E}(|v_0 d_{\sigma(0)} + \cdots + v_{n-1} d_{\sigma(n-1)}|) + \mathbb{E}(|-v_0 d_{\sigma(1)} - \cdots - v_{n-1} d_{\sigma(0)}|) \qquad (6.6) \\ &\quad + \mathbb{E}(|v_0 d_{\sigma(2)} + \cdots + v_{n-1} d_{\sigma(1)}|) + \cdots + \mathbb{E}(|-v_0 d_{\sigma(n-1)} - \cdots - v_{n-1} d_{\sigma(n-2)}|)\Big] \\ \geq& \frac{1}{n}\mathbb{E}\Big[|v_0(d_{\sigma(0)} - d_{\sigma(1)} + \cdots - d_{\sigma(n-1)}) + v_1(d_{\sigma(1)} - d_{\sigma(2)} + \cdots - d_{\sigma(0)}) \qquad (6.7) \\ &\quad + v_2(d_{\sigma(2)} - d_{\sigma(3)} + \cdots - d_{\sigma(1)}) + \cdots v_{n-1}(d_{\sigma(n-1)} - d_{\sigma(0)} + \cdots - d_{\sigma(n-2)})|\Big] \\ =& \frac{1}{n}\left|d_{\sigma(0)} - d_{\sigma(1)} + d_{\sigma(2)} - \cdots - d_{\sigma(n-1)}\right| \mathbb{E}(|v_0 - v_1 + v_2 - \cdots - v_{n-1}|),\end{aligned}$$

where in eq. (6.6) minus signs are added inside every other expectation (note that n is even), and eq. (6.7) is derived using the triangle inequality.

Since σ was an arbitrary permutation, we can instead write

$$\mathbb{E}(|V \cdot d|) \geq \frac{1}{n} M^{(d)} E_n,$$

where

$$M^{(d)} \coloneqq \max_{\sigma \in Sym([n])} \left| d_{\sigma(0)} - d_{\sigma(1)} + d_{\sigma(2)} - \cdots - d_{\sigma(n-1)} \right|,$$
$$E_n \coloneqq \mathbb{E}(|v_0 - v_1 + v_2 - \cdots - v_{n-1}|)$$

and $Sym([n])$ is the symmetric group on $[n]$. Thus

$$\mathbb{E}\left(\delta^{(\mathcal{B})}\right) \geq M^{(d)} E_n \geq \frac{\eta}{4\sqrt{n}}, \tag{6.8}$$

where the final inequality is due to Lemmas 6.1.3 and 6.1.4.

Use of Lemma 6.3.1 immediately tells us that

$$\mathbb{P}\left[\delta^{(\mathcal{B})} \geq \lambda\right] \geq \frac{1}{4\sqrt{n}} - \frac{\lambda}{\eta}.$$

Setting $\lambda = \frac{\min(1,\epsilon)}{8\sqrt{n}}$ and recalling that $\epsilon \leq \eta$ gives

$$\mathbb{P}\left[\delta^{(\mathcal{B})} \geq \frac{\min(1,\epsilon)}{8\sqrt{n}}\right] \geq \frac{1}{4\sqrt{n}} - \frac{\min(1,\epsilon)}{8\eta\sqrt{n}} \geq \frac{1}{4\sqrt{n}} - \frac{1}{8\sqrt{n}} = \frac{1}{8\sqrt{n}}.$$

□

6.4 Proof of Lemma 6.1.3

Let D^+ be the set of non-negative d_i's, labelled such that $d_0^+ \geq d_1^+ \geq \cdots$, and similarly let D^- be the set of negative d_i's, labelled such that $d_0^- \leq d_1^- \leq \cdots$. w.l.o.g. suppose $|D^-| \geq |D^+|$.

Let $|D^+| = \frac{n}{2} - k$, where $k \leq \frac{n}{2}$. Thus $|D^-| = \frac{n}{2} + k$. Note that $\sum_i d_i^+ = -\sum_i d_i^- = \frac{1}{2}\eta$.

We now define σ so that the following statements are true:

- $d_{\sigma(1)} = d_0^-, d_{\sigma(3)} = d_1^-, \ldots, d_{\sigma(n-1)} = d_{\frac{n}{2}-1}^-$;
- $d_{\sigma(0)} = d_0^+, d_{\sigma(2)} = d_1^+, \ldots, d_{\sigma(n-2k-2)} = d_{\frac{n}{2}-k-1}^+$;
- $d_{\sigma(n-2k)}, d_{\sigma(n-2k+2)}, \ldots, d_{\sigma(n-2)}$ can be filled with the remaining members of D^-.

Then

- $d_{\sigma(0)} + d_{\sigma(2)} + \cdots + d_{\sigma(n-2k-2)} = \frac{1}{2}\eta$;

- $d_0^-, \ldots, d_{\frac{n}{2}-1}^- \leq d_{\frac{n}{2}-1}^- \implies -d_{\sigma(1)} - d_{\sigma(3)} - \cdots - d_{\sigma(n-1)} \geq -\frac{n}{2} d_{\frac{n}{2}-1}^-$;
- $d_{\frac{n}{2}}^-, \ldots, d_{\frac{n}{2}+k-1}^- \geq d_{\frac{n}{2}-1}^- \implies d_{\sigma(n-2k)} + d_{\sigma(n-2k+2)} + \cdots + d_{\sigma(n-2)} \geq k d_{\frac{n}{2}-1}^-$.

Hence

$$\left| d_{\sigma(0)} - d_{\sigma(1)} + d_{\sigma(2)} - \cdots - d_{\sigma(n)} \right| \geq \left| \frac{1}{2}\eta + \left(k - \frac{n}{2}\right) d_{\frac{n}{2}-1}^- \right| \geq \frac{1}{2}\eta,$$

where the final inequality follows since $k \leq \frac{n}{2}$ and $d_{\frac{n}{2}-1}^- < 0$.

Thus $M^{(d)} \geq \frac{1}{2}\eta$. □

6.5 Proof of Lemma 6.1.4

To evaluate E_n we will make use of the Hermite-Genocchi Theorem (Theorem 3.3 in [Atk08]), which relates integrals over the probability simplex to associated divided differences.

The divided difference of n points $(x_0, f(x_0)), \ldots, (x_{n-1}, f(x_{n-1}))$ is defined by

$$f[x_0, \ldots, x_{n-1}] := \sum_{j \in [n]} \frac{f(x_j)}{\prod_{k \neq j}(x_j - x_k)}, \tag{6.9}$$

where limits are taken if any of the x_j are equal. It can be shown that for repeated points (see Exercise 4.6.6 in [Sch02])

$$f[\underbrace{x_0, \ldots, x_0}_{(r_0+1) \text{ times}}, \underbrace{x_1, \ldots, x_1}_{(r_1+1) \text{ times}}, x_2, \ldots, x_{n-1}] = \frac{1}{r_0! r_1!} \frac{\partial^{r_0+r_1}}{\partial x_0^{r_0} \partial x_1^{r_1}} f[x_0, x_1, x_2, \ldots, x_{n-1}], \tag{6.10}$$

where $x_0, \ldots, x_{n-1} \in \mathbb{R}$ are distinct.

Now, the Hermite-Genocchi Theorem states that

$$f[x_0, \ldots, x_{n-1}] = \frac{1}{(n-1)!} \int_{T_n} f^{(n-1)}(v_0 x_0 + \cdots + v_{n-1} x_{n-1}) \; dV,$$

where $dV = (n-1)! \; \delta(1 - \sum_{i \in [n]} v_i) \; dv_0 \cdots dv_{n-1}$.

In order to evaluate E_n, we set $f^{(n-1)}(\xi) = (n-1)!|\xi|$. Thus

$$f(\xi) = \begin{cases} \frac{1}{n}\xi^n & \xi \geq 0 \\ -\frac{1}{n}\xi^n & \xi < 0 \end{cases}$$

and $E_n = f[1, -1, 1, -1, \ldots, 1, -1]$.

Let $m = \frac{n}{2} - 1$ (i.e. $n = 2m + 2$). Then by eq. (6.10) we have that

$$E_{2m+2} = \frac{1}{m!^2} \, \partial_0^m \partial_1^m f[x_0, x_1]|_{x_0=-1, x_1=1},$$

where we have used the notation $\partial_i \equiv \frac{\partial}{\partial x_i}$.

In the neighbourhood of $x_0 = -1, x_1 = 1$, we have (by eq. (6.9))

$$f[x_0, x_1] = -\frac{1}{2m+2}\frac{x_0^{2m+2} + x_1^{2m+2}}{x_0 - x_1},$$

and thus

$$E_{2m+2} = -\frac{1}{2m+2}\frac{1}{m!^2}\, A|_{x_0=-1, x_1=1}\,, \tag{6.11}$$

where

$$A = \partial_0^m \partial_1^m \left(\frac{x_0^{2m+2} + x_1^{2m+2}}{x_0 - x_1}\right).$$

We see that

$$\begin{aligned} A &= \partial_1^m \partial_0^m \left(\frac{x_0^{2m+2}}{x_0 - x_1}\right) - \partial_0^m \partial_1^m \left(\frac{x_1^{2m+2}}{x_1 - x_0}\right) \\ &= \partial_1^m \partial_0^m \left(\frac{x_0^{2m+2}}{x_0 - x_1}\right) - (\textit{same term with } x_0 \textit{ and } x_1 \textit{ interchanged}). \end{aligned} \tag{6.12}$$

We use the Leibniz product rule[3] to deduce that

$$\partial_0^m \left(x_0^n \left(\frac{1}{x_0 - x_1}\right)\right) = \sum_{k=0}^{m} \binom{m}{k} \left[\frac{(2m+2)!}{(2m+2-k)!} x_0^{2m+2-k}\right] \left[\frac{(-1)^{m-k}}{(x_0 - x_1)^{m+1-k}}(m-k)!\right],$$

and hence that the first term in eq. (6.12) is

$$\begin{aligned} &\partial_1^m \partial_0^m \left(x_0^n \left(\frac{1}{x_0 - x_1}\right)\right) \\ &= \sum_{k=0}^{m} \binom{m}{k} \left[\frac{(2m+2)!}{(2m+2-k)!} x_0^{2m+2-k}\right] \left[\frac{(-1)^{m-k}}{(x_0 - x_1)^{2m+1-k}}(2m-k)!\right] \\ &= (2m+2)!(-1)^m \sum_{k=0}^{m} \binom{m}{k} \frac{(-1)^k (2m-k)!}{(2m+2-k)!} \frac{x_0^{2m+2-k}}{(x_0 - x_1)^{2m+1-k}} \\ &= (2m+2)!(-1)^m (x_0 - x_1) \sum_{k=0}^{m} \binom{m}{k} \frac{(-1)^k}{(2m+2-k)(2m+1-k)} \left(\frac{x_0}{x_0 - x_1}\right)^{2m+2-k}. \end{aligned}$$

Substituting this into eq. (6.12) and setting $x_0 = -1, x_1 = 1$ gives

$$A|_{x_0=-1, x_1=1} = -4(2m+2)!(-1)^m \sum_{k=0}^{m} \binom{m}{k} \frac{(-1)^k}{(2m+2-k)(2m+1-k)} \left(\frac{1}{2}\right)^{2m+2-k}.$$

Now set

$$B = (-1)^m \sum_{k=0}^{m} \binom{m}{k} \frac{(-1)^k}{(2m+2-k)(2m+1-k)} \gamma^{2m+2-k}$$

[3] $\partial^m(uv) = \sum_{k=0}^{m} \binom{m}{k} \partial^k(u) \partial^{m-k}(v)$

so that

$$A|_{x_0=-1,x_1=1} = -4(2m+2)!B|_{\gamma=\frac{1}{2}}. \tag{6.13}$$

Next, note that

$$\frac{\partial^2 B}{\partial \gamma^2} = (-1)^m \sum_{k=0}^{m} \binom{m}{k} (-1)^k \gamma^{2m-k} = \gamma^m \sum_{k=0}^{m} \binom{m}{k} (-\gamma)^{m-k} = \gamma^m (1-\gamma)^m,$$

and thus

$$\begin{aligned} B|_{\gamma=\frac{1}{2}} &= \int_{z=0}^{\frac{1}{2}} \int_{\alpha=0}^{z} \alpha^m (1-\alpha)^m \ d\alpha \, dz + C \\ &= \int_{z=0}^{\frac{1}{2}} B_z(m+1, m+1) \ dz + C, \end{aligned}$$

where $B_z(p,q) = \int_0^z \alpha^{p-1}(1-\alpha)^{q-1} \ d\alpha$ is the incomplete Beta function. By setting $m = 0$ it is easy to deduce that $C = 0$.

Now, the indefinite integral of the incomplete Beta function is

$$\int B_z(p,q) \ dz = zB_z(p,q) - B_z(p+1,q),$$

and hence we deduce that

$$\begin{aligned} B|_{\gamma=\frac{1}{2}} &= \frac{1}{2} B_{\frac{1}{2}}(m+1, m+1) - B_{\frac{1}{2}}(m+2, m+1) \\ &= \frac{1}{2} \int_0^{\frac{1}{2}} \alpha^m (1-\alpha)^m d\alpha - \int_0^{\frac{1}{2}} \alpha^{m+1} (1-\alpha)^m d\alpha \\ &= \frac{1}{2} \left[\int_0^{\frac{1}{2}} \alpha^m (1-\alpha)^m \underbrace{(1-2\alpha)}_{=(1-\alpha)-\alpha} \ d\alpha \right] \\ &= \frac{1}{2} \int_0^{\frac{1}{2}} (\alpha^m (1-\alpha)^{m+1} - \alpha^{m+1}(1-\alpha)^m) \ d\alpha \\ &= \frac{1}{2(m+1)} \int_0^{\frac{1}{2}} \frac{d(\alpha^{m+1}(1-\alpha)^{m+1})}{d\alpha} \ d\alpha \\ &= \frac{1}{2(m+1)} [\alpha^{m+1}(1-\alpha)^{m+1}]_0^{1/2} \\ &= \frac{1}{2^{2m+3}(m+1)}. \end{aligned}$$

Substituting this into eq. (6.13) and subsequently into eq. (6.11), we get

$$\begin{aligned} E_{2m+2} &= -\frac{1}{2m+2} \frac{1}{m!^2} \cdot -4(2m+2)! \cdot \frac{1}{2^{2m+3}(m+1)} \\ &= \frac{(2m+1)!}{2^{2m+1} m!^2 (m+1)} \\ &= \frac{2m+1}{m+1} \cdot \frac{(2m)!}{m!^2} \cdot \frac{1}{2^{2m+1}}. \end{aligned} \tag{6.14}$$

Stirling's formula [Rob55] tells us that for $m \geq 1$

$$\sqrt{2\pi}m^{m+\frac{1}{2}}e^{-m} < m! < \sqrt{2\pi}m^{m+\frac{1}{2}}e^{-m}e^{\frac{1}{12}}, \tag{6.15}$$

and thus

$$\frac{(2m)!}{m!^2} > \frac{\sqrt{2\pi}(2m)^{2m+\frac{1}{2}}e^{-2m}}{2\pi m^{2m+1}e^{-2m}e^{\frac{1}{6}}} = \frac{2^{2m}e^{-\frac{1}{6}}}{\sqrt{m\pi}}.$$

Since $\frac{2m+1}{m+1} \geq \frac{3}{2}$, eq. (6.14) tells us that

$$E_{2m+2} > \frac{3e^{-\frac{1}{6}}}{4\sqrt{\pi}} \cdot \frac{1}{\sqrt{m}}.$$

Replacing m with $\frac{n}{2} - 1$, we deduce that

$$E_n > \frac{3e^{-\frac{1}{6}}}{4\sqrt{\pi}} \cdot \frac{1}{\sqrt{\frac{n}{2}-1}} > \frac{3e^{-\frac{1}{6}}}{4\sqrt{\pi}} \cdot \frac{1}{\sqrt{\frac{n}{2}}} = \frac{3e^{-\frac{1}{6}}}{2\sqrt{2\pi}} \cdot \frac{1}{\sqrt{n}} > \frac{1}{2\sqrt{n}}. \tag{6.16}$$

The case when $m = 0$ is easily dealt with through direct calculation using eq. (6.14), giving $E_2 = \frac{1}{2}$. Hence we conclude that eq. (6.16) holds for all positive, even n.

Remark: Using the asymptotic form of Stirling's formula (the lower bound of eq. (6.15)), it can be easily shown that $E_n \sim \sqrt{\frac{2}{\pi}}\frac{1}{\sqrt{n}}$ for large n. □

6.6 Final remarks and open questions

Quantum conditional oracles give us new insights into the kinds of information that are useful for testing properties of distributions. In addition, they are able to demonstrate separations in query complexity between a number of problems, thereby providing interesting new perspectives on information without trivialising the set-up. We now mention some open questions.

Group testing and pattern matching are further important areas to which our notion of a quantum conditional oracle could be applied. The structure of questions commonly considered there suggest that the use of the PQCOND oracle would decrease the query complexity dramatically for many practically relevant problems compared to the best known quantum and classical algorithms [Por09, DBGV05, ABRdW15, Bon15].

In our algorithms, we have made particular use of the PQCOND oracle, the quantum analogue of the PCOND oracle. It is noted in [CRS15] that the unrestricted COND oracle offers significant advantages over the PCOND oracle for many problems, and it is possible that similar improvements could be achieved for some quantum algorithms through use of the unrestricted QCOND oracle.

The algorithm that we present for quantum spectrum testing (Algorithm 2) chooses several bases $\mathcal{B}_1, \ldots, \mathcal{B}_k$ independently and uniformly at random. It remains open,

however, whether or not a more adaptive approach to choosing bases will yield an algorithm requiring fewer queries.

Our definition of the spectrum testing problem in Chapter 6 made use of the trace norm, $\|\cdot\|_1$. One might wonder how the query complexity would be affected if the problem were defined with a different norm, such as the operator norm[4], $\|\cdot\|_\infty$. Numerical simulations and limited analysis suggest that the probability of picking a 'good' basis $\mathcal{B}$ tends to 1 as $n \to \infty$, and hence that the number of queries required to distinguish between the two options would be independent of n. We leave the proof of this conjecture as an open question.

[4]For an $(n \times n)$ matrix A, $\|A\|_\infty = \max_{i\in[n]} a_i$, where the a_i are the singular values of A.

Part III

Classical Simulation of Quantum Circuits

Chapter 7

Drawing the line between classical and quantum—what do we know?

7.1 Introduction

One can easily show that all classical processes can be simulated on a quantum computer with little or no overhead, but whether or not quantum processes can be efficiently simulated on a classical computer is still very much an active area of research. Even the question of whether or not quantum computers are more 'powerful' than classical computers remains a mystery, although it is widely suspected that they are. Thus, a natural area of research is determining what gives quantum computers their 'power'.

It is known that without large-scale entanglement, quantum circuits are efficiently classically simulatable. But is entanglement a sufficient condition for making circuits more powerful? It is relatively straightforward to find a counter example to show that this is not the case. So what does draw the line between classical and quantum computation? One approach that has been used to investigate this is determining classes of non-trivial (highly entangling) quantum circuits that can be efficiently simulated on classical computers, and understanding what must be added to allow for full quantum computing.

Two well-known results in this area are for circuits built from Clifford gates, and those built from matchgates (see Section 7.2). Both of these gate sets produce non-trivial quantum circuits for which the output probabilities can be efficiently calculated on a classical computer.

The Jordan-Wigner transform (see Section 1.1.3) is a very useful tool developed to deal with fermionic systems. Surprisingly, it is the crux of a beautiful proof of the classical simulability of matchgates [TD02, JM08, JMS15]. This begs the question of whether a similar or generalised transform can be used to develop other classes of simulatable circuits. We explore this idea in Appendix B.1.

7.2 Known classes of classically-simulatable circuits

The Gottesman-Knill Theorem

The Gottesman-Knill Theorem [Got98, Joz08] is a landmark result that shows that output probabilities of circuits comprising Clifford gates (and subject to some boundary conditions) are efficiently simulatable on a classical computer. The description of the theorem that we give here can also be found in [Joz08].

We define the one-qubit gates

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix},$$

and the two-qubit gate

$$CZ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}.$$

Arbitrary circuits on n qubits comprising these gates are called *Clifford* operations.

Theorem 7.2.1 (Gottesman-Knill Theorem (slightly modified))**.** *Consider a polynomial-sized circuit on n-qubits comprising H, P and CZ gates (i.e. a Clifford circuit). Provided that:*

- *The input is a product state; and*
- *The output is a Z measurement (i.e. measurement in the computational basis) of any single qubit;*

then the output is efficiently classically-simulatable (i.e. the output probabilities can be calculated to k digits in $\mathrm{poly}(n, k)$ *time).*

A simple proof of this theorem is given in [Joz08].

The authors of a recent paper [VFGE12] even show that for a wide range of inputs the state of the system can be tracked through a discrete phase space as the circuit progresses, providing a hidden variable model for such computations.

Matchgates

Matchgates, introduced by L. Valiant [Val02], are another (non-trivial) class of gates for which circuits built from these gates can be efficiently simulated on a classical computer.

A matchgate is defined to be any two-qubit matrix $G(A, B)$ of the form

$$G(A,B) = \begin{pmatrix} p & 0 & 0 & q \\ 0 & w & x & 0 \\ 0 & y & z & 0 \\ r & 0 & 0 & s \end{pmatrix},$$

where

$$A = \begin{pmatrix} p & q \\ r & s \end{pmatrix}, \quad B = \begin{pmatrix} w & x \\ y & z \end{pmatrix},$$

and $A, B \in \mathcal{U}(2)$ and $\det A = \det B$.

The structure is such that $G(A, B)$ essentially acts with A on the even-parity subspace (spanned by $|00\rangle$ and $|11\rangle$), and acts with B on the odd-parity subspace (spanned by $|01\rangle$ and $|10\rangle$).

Valiant's theorem is as follows:

Theorem 7.2.2 (Valiant's Theorem (slightly modified))**.** *Consider a polynomial-sized circuit on n-qubits comprising $G(A, B)$ gates* acting on nearest-neighbour qubits only*. Provided that:*

- *The input is a product state; and*
- *The output is a Z measurement (i.e. measurement in the computational basis) of any single qubit;*

then the output is efficiently classically-simulatable (i.e. the output probabilities can be calculated to k digits in $\text{poly}(n, k)$ *time).*

The proof given in [JM08] makes use of the connection between matchgates and the Jordan-Wigner transform for Majorana fermions (see Section 1.1.3), a relationship first presented in [TD02].

In this proof, $c_1, d_1, \ldots, c_n, d_n$ are relabelled to $b_1, b_2, \ldots, b_{2n}$ respectively. In addition, let $\mathcal{L}_2$ be the span of $\{b_i\, b_j\}_{i,j=1}^{2n}$, that is, the quadratic span of the b_i's. The crux of the proof is the statement that the commutator of any two elements of $\mathcal{L}_2$ is also in $\mathcal{L}_2$. More formally,

$$[b_i\, b_j\, , b_k b_l\,] \in \mathcal{L}_2.$$

It is through this surprising property that the classical simulability of the circuit arises. In Appendix B.1, we explore the idea of whether or not there exist qutrit or general qudit Jordan-Wigner-type representations that have the same property, and provide evidence that such representations do not exist.

Chapter 8

Beyond the line: generating classically-simulatable quantum circuits in higher dimensions

8.1 Introduction

Fermionic systems describe a diverse range of complex physical phenomena, ranging from quantum dots to low-temperature effects, such as the quantum Hall effect and superfluidity. The study of their properties through mathematical physics, and more recently through computer science, has revealed new, intriguing connections between the respective fields.

There exist a wide number of tools for studying fermionic systems [Suz93, ID91], one of the simplest being the Jordan-Wigner transform [JW28] (see Section 1.1.3). This represents a simple, yet powerful, method for investigating the behaviour of fermionic models by mapping them directly to spins. It dramatically simplifies the representations of Hamiltonians for some of the most ubiquitous statistical models—such as the Ising and XY models in a transverse magnetic field—by transforming them into exactly solvable systems of non-interacting fermions [LM13]. This spin-fermion mapping, coupled with the correspondence between the resulting spin system and a class of quantum computations, has led to a surprising connection between 1-dimensional local non-interacting fermionic systems and classically-simulatable quantum circuits comprised of *matchgates* acting on qubits [TD02, Joz08, JM08, JMS15, Tsv07].

Little is known about relating generic local fermionic systems to quantum computation, current examples being confined to chains of spin-1/2 particles in one dimension [TD02, JM08, JMS15, BC14]. Such a link depends on two transformations: (a) the existence of a new (locality-preserving) spin-fermionic mapping, and (b) the correspondence between the resulting local quantum spin Hamiltonian and a local quantum circuit. It would give rise to novel, physically-motivated classes of circuits acting on qudits that are classically simulatable and which would help us better understand the properties of the underlying fermionic systems, giving insight into their compu-

tational power.

A natural way to proceed in the search for the former transformation is to naïvely extend the Jordan-Wigner transform beyond spin-1/2 chains to lattices of higher dimension. Such generalisations unfortunately suffer from a failure to preserve the locality of the interactions. However, more complex generalisations [VC05, Bal05] are able to retain locality, at the expense of an increase in the number of lattice sites.

In this chapter we present a systematic approach for relating general models of non-interacting fermions on two-dimensional (and higher-dimensional) lattices to quantum circuits. The latter can be efficiently classically simulated whenever the original Hamiltonian is solvable. Furthermore, we show how this mapping may be applied to the Hubbard model.

We additionally show how these same models of fermions may be related to pseudo-classical spin models on higher-dimensional lattices, and subsequently how classical algorithms may be used to derive their thermodynamic properties.

In more detail:

- In Section 8.2 we discuss how the Jordan-Wigner transformation maps a local fermionic system on a 2-dimensional lattice to a non-local spin model. By making use of tools from [VC05], we develop a mapping to a spin model that retains locality.
- In Section 8.3 we show how these local spin models may be mapped to a quantum circuit, for which the output probabilities are related to the partition function of the original fermionic system. If the original Hamiltonian is solvable, i.e. there is a closed form for its partition function, then the corresponding quantum circuit is efficiently classically simulatable.
- As an example, in Section 8.4 we apply this mapping to the Hubbard model [Tas98a, Tas98b, Lie04, Jar92, FK90], which describes the movements of electrons in a solid. In order to do this, we introduce a new, generalised version of the Jordan-Wigner transform that maps multi-flavour fermions to spins.
- In Section 8.5 we extend this mapping and present a classical technique for computing the thermodynamic properties of the original, 2-dimensional fermionic system by transforming it into a 3-dimensional pseudo-classical system and applying the Metropolis-Hastings algorithm [MRR$^+$53, Has70]. In order to achieve this, we make use of the quantum-classical mapping [HKM15] to relate the local quantum spin models, generated in Sections 8.2 and 8.3, to local pseudo-classical spin models on a higher-dimensional lattice.

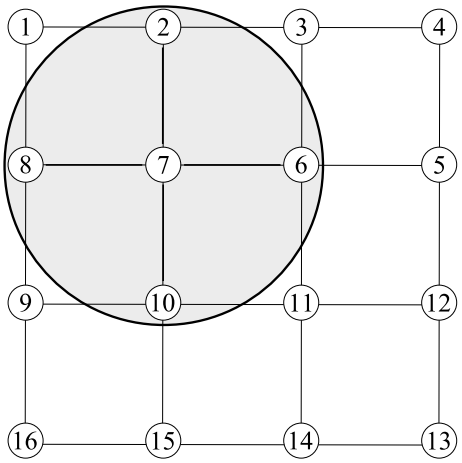


Figure 8.1: An operator that acts non-trivially on sites 2, 6, 7, 8 and 10 is a distance-1 (nearest-neighbour) operator about site 7.

8.2 A local Jordan-Wigner transform on a 2-dimensional lattice

In this section we explain how to map a local fermionic Hamiltonian to a local spin Hamiltonian, using a method based on the work in [VC05]. In [VC05], the authors additionally require that the ground state of the Hamiltonian is preserved by the mapping. This is not necessary for our result, and we adapt and simplify the procedure to remove this requirement.

Let us first define our notion of locality: a distance-k Hamiltonian.

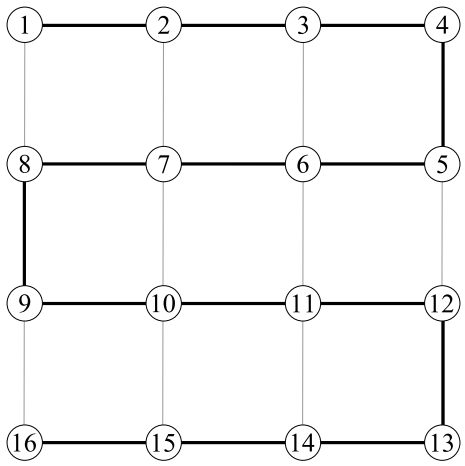
Definition 8.2.1. *Given a d-dimensional regular lattice, we define an operator O to be distance-k if there exists a site i such that O only acts non-trivially on sites of taxicab distance at most k from site i.*

See Figure 8.1 for an example.

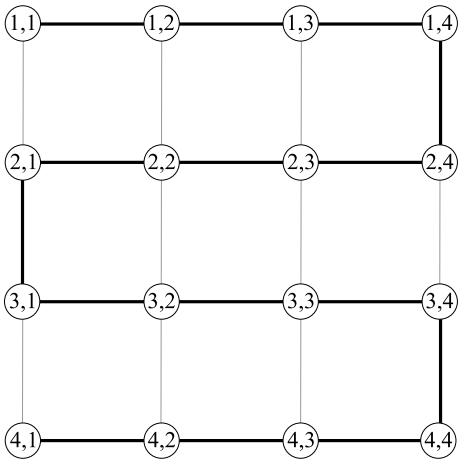
Remark: The above definition can be easily modified to account for irregular lattices; however, we have insisted upon regular lattices for simplicity. See Section 8.6.

Non-locality of the Jordan-Wigner transform on a 2-dimensional lattice

Consider again the fermionic system on a 1-dimensional chain described in Section 1.1.3. A non-interacting, distance-k fermionic Hamiltonian H governing the system would consist of two types of terms: the single-site terms $n_i = a_i^\dagger a_i$, and the hopping terms $(a_i^\dagger a_j + a_j^\dagger a_i)$. Using the Jordan-Wigner transform (eq. (1.9)), these can be



(a) Enumeration of sites on a 2D square lattice that is treated as a 1D chain



(b) Enumeration of sites on a 2D square lattice
that is treated as a 2D grid

Figure 8.2: Enumerating sites on a 2D lattice in two different ways. (a) treats the sites as a 1D chain, which is important for applying the Jordan-Wigner transform as derived in eq. (1.9); (b) treats the sites as a 2D grid, which is important for describing our local transform in eq. (8.4).

written in terms of spin operators:

$$\begin{aligned} n_i &= \frac{1}{2}(Z + \mathbb{1}_2)_i \\ a_i^\dagger a_j + a_j^\dagger a_i &= -\frac{1}{2}(X_i X_j + Y_i Y_j) Z_{i+1} \cdots Z_{j-1} \quad (i < j). \end{aligned}$$

It is clear from the above representations that a Hamiltonian that is distance-k in the fermionic representation remains distance-k in the spin representation.

Let us now consider a fermionic system over a 2-dimensional lattice, where we number the sites as in Figure 8.2a. If the Hamiltonian is distance-1 (i.e. allows single-site terms and nearest-neighbour hopping terms only), then it could contain vertical hopping terms such as $(a_9^\dagger a_{16} + a_{16}^\dagger a_9)$. These terms do not remain local when written in the spin representation. For example, the $(a_9^\dagger a_{16} + a_{16}^\dagger a_9)$ term is equivalent to $-\frac{1}{2}(X_9 X_{16} + Y_9 Y_{16}) Z_{10} \cdots Z_{15}$, which is now a distance-3 term on the lattice, and this degree of locality is dependent upon the width of the lattice (that is, if this lattice had a width of Q, the term would be approximately distance-$Q/2$ in the spin representation).

We now construct a new Hamiltonian H' that encompasses the dynamics of H, but is also local in the spin representation.

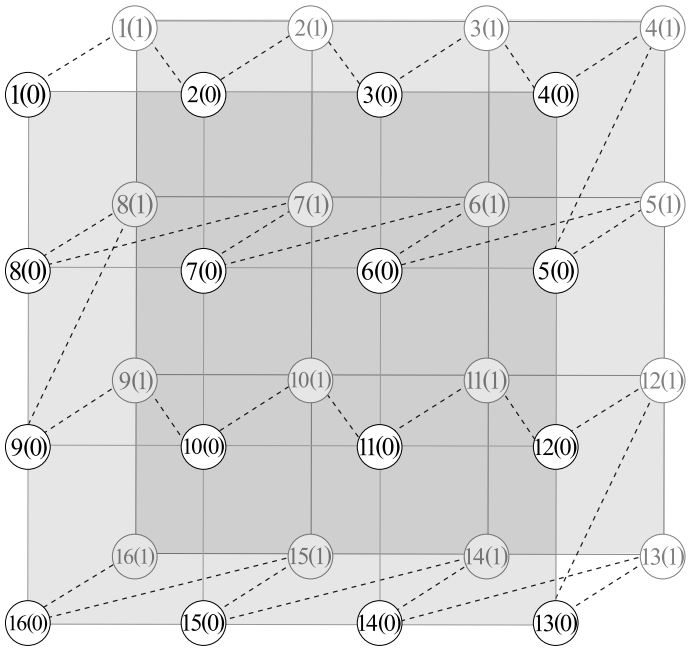


Figure 8.3: The original '0-layer' of sites with an additional 1-layer of corresponding sites. The dotted line illustrates our ordering of the sites.

8.2.1 Construction of a new Hamiltonian that retains locality

We add another 'layer' to our lattice so that for each existing site i (which we now call $i(0)$), there is a new, neighbouring site $i(1)$, and we order the sites as $1(0)$, $1(1)$, $2(0)$, $2(1)$, $\ldots$, $N(0)$, $N(1)$, as shown in Figure 8.3. We define states 'restricted' to the both the original layer (0-layer) and the additional layer (1-layer) of sites respectively by

$$|\alpha\rangle_0 \coloneqq \left(a^\dagger_{1(0)}\right)^{\alpha_1} \cdots \left(a^\dagger_{N(0)}\right)^{\alpha_N} |\Omega\rangle$$
$$|\gamma\rangle_1 \coloneqq \left(a^\dagger_{1(1)}\right)^{\gamma_1} \cdots \left(a^\dagger_{N(1)}\right)^{\gamma_N} |\Omega\rangle ,$$

where $\alpha = (\alpha_1, \ldots, \alpha_N)$ and $\gamma = (\gamma_1, \ldots, \gamma_N)$.

For these states it is unclear what is meant by the tensor product $\otimes$, and so here we explicitly define the bilinear product operator $\odot$. If $|\varphi\rangle = A_\varphi |\Omega\rangle$ and $|\xi\rangle = A_\xi |\Omega\rangle$ are two states on the full lattice, where A_φ and A_ξ are products of creation and annihilation operators ($a^\dagger_{i(\sigma)}$ and $a_{j(\sigma)}$ respectively), then $|\varphi\rangle \odot |\xi\rangle$ is defined by

$$|\varphi\rangle \odot |\xi\rangle \coloneqq A_\varphi A_\xi |\Omega\rangle .$$

Let the set $\Lambda(1)$ contain ordered pairs $\langle i(1), j(1)\rangle$ of sites in the 1-layer corresponding to directed edges pointing vertically down the lattice, as shown in Figure 8.4. We define an operator $P_{i(1)\,j(1)}$ for each pair $\langle i(1), j(1)\rangle \in \Lambda(1)$ by

$$P_{i(1)\,j(1)} = \theta_{i(1)\,j(1)} i c_{i(1)} d_{j(1)},$$

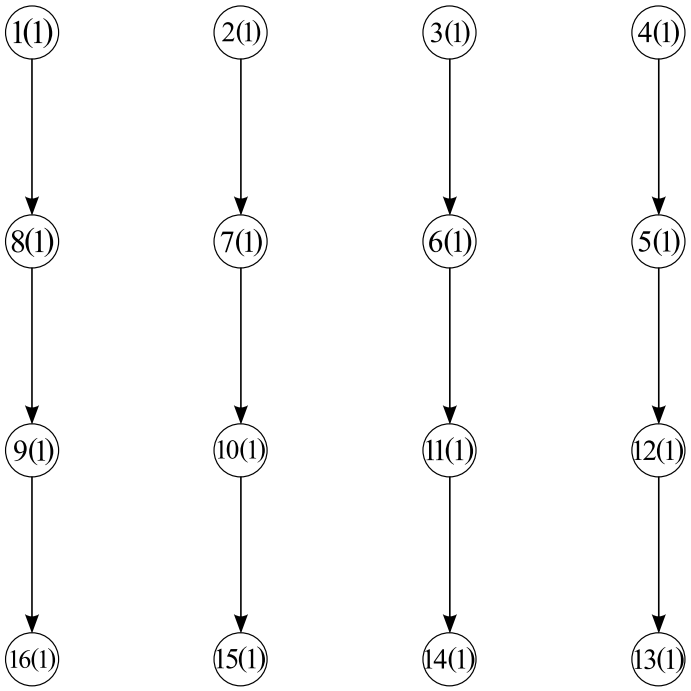


Figure 8.4: The $P_{i(1)\,j(1)}$'s correspond to the directed edges on the 1-layer that point vertically down the lattice.

where $c_{i(1)} = (a_{i(1)} + a^\dagger_{i(1)})$ and $d_{i(1)} = i(a_{i(1)} - a^\dagger_{i(1)})$ are Majorana fermions (see Section 1.1.3), and the value of $\theta_{i(1)\,j(1)} = \pm 1$ is to be determined.

Using eq. (1.11), it is straightforward to derive the following three useful properties of the $P_{i(1)\,j(1)}$:

1. $(P_{i(1)\,j(1)})^2 = \mathbb{1} \implies P_{i(1)\,j(1)}$ has eigenvalues of ± 1.
2. $[P_{i(1)\,j(1)}, P_{k(1)\,l(1)}] \propto \delta_{j(1)l(1)} c_{k(1)} c_{i(1)} - \delta_{i(1)k(1)} d_{j(1)} d_{l(1)} = 0$, since $\langle i(1), j(1)\rangle$, $\langle k(1), l(1)\rangle \in \Lambda(1)$ (we know that $(i(1) = k(1)) \Leftrightarrow (j(1) = l(1))$ and that $(c_{m(1)})^2 = (d_{m(1)})^2 = \mathbb{1}$).
3. $[P_{i(1)\,j(1)}, a^{(\dagger)}_{k(0)}] = 0$ since $k(0)$ is in the 0-layer.

Point 2 implies that the $P_{i(1)\,j(1)}$ have a common eigenbasis over the 1-layer. We choose an element of this eigenbasis, $|\chi\rangle_1$, and fix the values $\theta_{i(1)\,j(1)}$ appropriately for each $P_{i(1)\,j(1)}$ so that

$$P_{i(1)\,j(1)} |\chi\rangle_1 = |\chi\rangle_1 \quad \forall \{i(1), j(1)\} \in \Lambda(1),$$

which is possible owing to Point 1.

In addition, we can use Point 3 to show that

$$\begin{aligned} P_{i(1)\,j(1)} \left|\alpha\right\rangle_0 \odot \left|\gamma\right\rangle_1 &= P_{i(1)\,j(1)} \left[\left(a^\dagger_{1(0)}\right)^{\alpha_1} \cdots \left(a^\dagger_{N(0)}\right)^{\alpha_N}\right] \left[\left(a^\dagger_{1(1)}\right)^{\gamma_1} \cdots \left(a^\dagger_{N(1)}\right)^{\gamma_N}\right] \left|\Omega\right\rangle \\ &= \left[\left(a^\dagger_{1(0)}\right)^{\alpha_1} \cdots \left(a^\dagger_{N(0)}\right)^{\alpha_N}\right] \left[P_{i(1)\,j(1)} \left(a^\dagger_{1(1)}\right)^{\gamma_1} \cdots \left(a^\dagger_{N(1)}\right)^{\gamma_N}\right] \left|\Omega\right\rangle \\ &= \left|\alpha\right\rangle_0 \odot \left(P_{i(1)\,j(1)} \left|\gamma\right\rangle_1\right). \end{aligned}$$

Hence it is clear that

$$P_{i(1)\,j(1)} \left|\varphi\right\rangle_0 \odot \left|\chi\right\rangle_1 = \left|\varphi\right\rangle_0 \odot \left|\chi\right\rangle_1 \tag{8.1}$$

for any state $\left|\varphi\right\rangle_0$ on the 0-layer.

Similarly, since the original Hamiltonian H consists only of operators on indices in the 0-layer, we immediately have that

$$H\left(\left|\varphi\right\rangle_0 \odot \left|\xi\right\rangle_1\right) = \left(H\left|\varphi\right\rangle_0\right) \odot \left|\xi\right\rangle_1. \tag{8.2}$$

We now construct a new Hamiltonian H' from H by altering the vertical hopping terms as follows:

$$a^\dagger_{i(0)} a_{j(0)} + a^\dagger_{j(0)} a_{i(0)} \mapsto (a^\dagger_{i(0)} a_{j(0)} + a^\dagger_{j(0)} a_{i(0)}) P_{i(1)\,j(1)}$$

where i is the site directly above j on the lattice.

It is clear from eq. (8.1) and eq. (8.2) that

$$H'\left(\left|\varphi\right\rangle_0 \odot \left|\chi\right\rangle_1\right) = \left(H\left|\varphi\right\rangle_0\right) \odot \left|\chi\right\rangle_1 \tag{8.3}$$

and thus the local fermionic system described by H follows the same dynamics as the 0-layer of sites of the system described by H'. Now it remains to be shown that H' is local in the spin representation.

Locality of H' in the spin representation

After ordering the sites as $1(0), 1(1), 2(0), 2(1), \ldots$, the relevant form of eq. (1.9) is

$$\begin{aligned} a^\dagger_{i(0)} &= Z_{1(0)} Z_{1(1)} \cdots Z_{(i-1)(0)} Z_{(i-1)(1)} S^+_{i(0)} \\ a_{i(0)} &= Z_{1(0)} Z_{1(1)} \cdots Z_{(i-1)(0)} Z_{(i-1)(1)} S^-_{i(0)} \\ c_{i(1)} &= Z_{1(0)} Z_{1(1)} \cdots Z_{i(0)} X_{i(1)} \\ d_{i(1)} &= Z_{1(0)} Z_{1(1)} \cdots Z_{i(0)} Y_{i(1)}. \end{aligned}$$

This allows us to determine the spin representations of the terms in H', and we find that the problematic string of Z operators that arose for vertical hopping terms is

cancelled off by a string of Z operators created by the additional $P_{i(1)\,j(1)}$ operators. We can show this explicitly:

$$\begin{aligned} n_{i(0)} &= \frac{1}{2}(Z + \mathbb{1}_2)_{i(0)} \\ a^\dagger_{i(0)}a_{j(0)} + a^\dagger_{j(0)}a_{i(0)} &= -\frac{1}{2}(X_{i(0)}X_{j(0)} + Y_{i(0)}Y_{j(0)})Z_{i(1)}Z_{(i+1)(0)}\cdots Z_{(j-1)(1)} \\ P_{i(1)\,j(1)} &= \theta_{i(1)\,j(1)}Y_{i(1)}Z_{(i+1)(0)}Z_{(i+1)(1)}\cdots Z_{j(0)}Y_{j(1)} \\ (a^\dagger_{i(0)}a_{j(0)} + a^\dagger_{j(0)}a_{i(0)})P_{i(1)\,j(1)} &= -\frac{1}{2}\theta_{i(1)\,j(1)}(Y_{i(0)}X_{j(0)} - X_{i(0)}Y_{j(0)})X_{i(1)}Y_{j(1)}. \end{aligned}$$

So far we have consistently labelled the sites with a single integer as if they were part of a 1-dimensional chain (as in Figure 8.2a). However, in order to determine the full form of H', we now consider the sites to be points on a grid, labelling each site by its co-ordinate, as in Figure 8.2b.

Allowing only distance-1 (nearest-neighbour) hopping terms, we see that H consists of three types of terms:

$$n_{(i,j)(0)}; \quad a^\dagger_{(i,j)(0)}a_{(i,j+1)(0)} + a^\dagger_{(i,j+1)(0)}a_{(i,j)(0)}; \quad a^\dagger_{(i,j)(0)}a_{(i+1,j)(0)} + a^\dagger_{(i+1,j)(0)}a_{(i,j)(0)}.$$

In H', these respectively become:

$$\begin{gathered} n_{(i,j)(0)}; \quad a^\dagger_{(i,j)(0)}a_{(i,j+1)(0)} + a^\dagger_{(i,j+1)(0)}a_{(i,j)(0)}; \\ (a^\dagger_{(i,j)(0)}a_{(i+1,j)(0)} + a^\dagger_{(i+1,j)(0)}a_{(i,j)(0)})P_{(i,j)(1)\,(i+1,j)(1)}, \end{gathered}$$

which, in the spin representation, respectively have the form

$$\begin{gathered} \frac{1}{2}(Z + \mathbb{1}_2)_{(i,j)(0)}; \quad -\frac{1}{2}(X_{(i,j)(0)}X_{(i,j+1)(0)} + Y_{(i,j)(0)}Y_{(i,j+1)(0)})Z_{(i,j)(1)}; \\ -\frac{1}{2}\theta_{(i,j)(1)\,(i+1,j)(1)}(Y_{(i,j)(0)}X_{(i+1,j)(0)} - X_{(i,j)(0)}Y_{(i+1,j)(0)})X_{(i,j)(1)}Y_{(i+1,j)(1)}, \end{gathered} \tag{8.4}$$

which are all at most distance-2 terms.

Hence the dynamics of the local fermionic Hamiltonian H acting on N sites can be simulated by a local spin Hamiltonian H' acting on $2N$ qubits.

8.2.2 Summary

This section summarises the results thus far:

Suppose that the original fermionic lattice (the 0-layer) has height P and width Q (i.e. $i \in \{1,\ldots,P\}, j \in \{1,\ldots,Q\}$), where for ease of notation we assume P and Q are

even, with dynamics governed by a distance-1 Hamiltonian H of the form

$$H = \mu \sum_{\substack{1\le i\le P\\ 1\le j\le Q}} n_{(i,j)(0)} + \sum_{\substack{1\le i\le P\\ 1\le j\le Q-1}} \mu_{ij}^{H}(a^{\dagger}_{(i,j)(0)}a_{(i,j+1)(0)} + a^{\dagger}_{(i,j+1)(0)}a_{(i,j)(0)}) + \sum_{\substack{1\le i\le P-1\\ 1\le j\le Q}} \mu_{ij}^{V}(a^{\dagger}_{(i,j)(0)}a_{(i+1,j)(0)} + a^{\dagger}_{(i+1,j)(0)}a_{(i,j)(0)}).$$

In general, μ could be site-dependent, but for simplicity we treat it as a constant.

We then increase the number of lattice sites by introducing an additional site $i(1)$ for each 0-layer site $i(0)$, and introduce the Hamiltonian H', with

$$H' = J \sum_{\substack{1\le i\le P\\ 1\le j\le Q}} (Z + \mathbb{1}_2)_{(i,j)(0)} + \sum_{\substack{1\le i\le P\\ 1\le j\le Q-1}} J_{ij}^{H}(X_{(i,j)(0)}X_{(i,j+1)(0)} + Y_{(i,j)(0)}Y_{(i,j+1)(0)})Z_{(i,j)(1)} + \sum_{\substack{1\le i\le P-1\\ 1\le j\le Q}} J_{ij}^{V}(Y_{(i,j)(0)}X_{(i+1,j)(0)} - X_{(i,j)(0)}Y_{(i+1,j)(0)})X_{(i,j)(1)}Y_{(i+1,j)(1)}, \tag{8.5}$$

where $J = \frac{1}{2}\mu$, $J_{ij}^{H} = -\frac{1}{2}\mu_{ij}^{H}$ and $J_{ij}^{V} = -\frac{1}{2}\theta_{(i,j)(1)\,(i+1,j)(1)}\mu_{ij}^{V}$. H' is then a distance-2 Hamiltonian, whose action on the 0-layer is identical to the original action of H.

8.3 Classically-simulatable quantum circuits from quantum lattice models

Here we show how to map the quantum partition function $\mathcal{Z}$ of the fermionic Hamiltonian H to the output probabilities of a quantum circuit. If the original Hamiltonian is solved (i.e. there is a closed form for $\mathcal{Z}$), then the resultant circuit is efficiently classically simulatable.

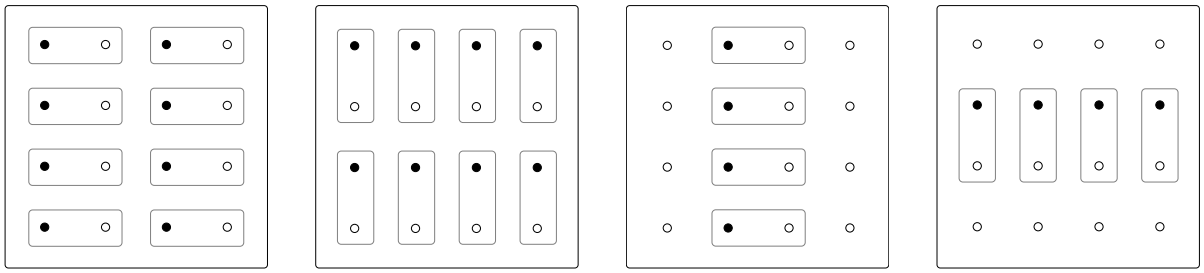
The quantum partition function $\mathcal{Z}$ is defined by

$$\mathcal{Z} := \mathrm{Tr}_0\left[e^{-\beta H}\right] = \sum_{\alpha\in\{0,1\}^{PQ}} \langle\alpha|_0\, e^{-\beta H}\, |\alpha\rangle_0 = \sum_{\alpha\in\{0,1\}^{PQ}} \left(\langle\alpha|_0 \odot \langle\chi|_1\right) e^{-\beta H'} \left(|\alpha\rangle_0 \odot |\chi\rangle_1\right), \tag{8.6}$$

where the final equality is due to eq. (8.3).

Each of the terms in H' (see eq. (8.5)) is distance-2, and they can be separated into four sets such that all interactions in each set are disjoint, as can be seen in Figure 8.5. Hence

$$H' = \sum_{(i,j)\in a_H} H_{ij}^{H} + \sum_{(i,j)\in a_V} H_{ij}^{V} + \sum_{(i,j)\in b_H} H_{ij}^{H} + \sum_{(i,j)\in b_V} H_{ij}^{V},$$



(a) $a_H = \{(1, 1), (1, 3), (2, 1), (2, 3), (3, 1), (3, 3), (4, 1), (4, 3)\}$

(b) $a_V = \{(1, 1), (1, 2), (1, 3), (1, 4), (3, 1), (3, 2), (3, 3), (3, 4)\}$

(c) $b_H = \{(1, 2), (2, 2), (3, 2), (4, 2)\}$

(d) $b_V = \{(2, 1), (2, 2), (2, 3), (2, 4)\}$

Figure 8.5: All possible distance-1 interactions on the lattice, broken up into four sets inside each of which all of the interactions are disjoint. The sets a_H, a_V, b_H, b_V contain the co-ordinates of the points marked as full black disks on (a), (b), (c) and (d) respectively, where the site in the top-left corner has co-ordinate $(1, 1)$, and in these diagrams, the site in the top-right corner has co-ordinate $(1, 4)$.

where

$$H_{ij}^H = \begin{cases} \frac{1}{2}J(Z + \mathbb{1}_2)_{(i,j)(0)} + \frac{1}{4}J(Z + \mathbb{1}_2)_{(i,j+1)(0)} \\ \quad + J_{ij}^H(X_{(i,j)(0)}X_{(i,j+1)(0)} + Y_{(i,j)(0)}Y_{(i,j+1)(0)})Z_{(i,j)(1)} & j = 1, \\ \frac{1}{4}J(Z + \mathbb{1}_2)_{(i,j)(0)} + \frac{1}{4}J(Z + \mathbb{1}_2)_{(i,j+1)(0)} \\ \quad + J_{ij}^H(X_{(i,j)(0)}X_{(i,j+1)(0)} + Y_{(i,j)(0)}Y_{(i,j+1)(0)})Z_{(i,j)(1)} & 2 \leq j \leq Q - 1, \\ \frac{1}{4}J(Z + \mathbb{1}_2)_{(i,j)(0)} & j = Q, \end{cases}$$

$$H_{ij}^V = \begin{cases} \frac{1}{2}J(Z + \mathbb{1}_2)_{(i,j)(0)} + \frac{1}{4}J(Z + \mathbb{1}_2)_{(i+1,j)(0)} \\ \quad + J_{ij}^V(Y_{(i,j)(0)}X_{(i+1,j)(0)} - X_{(i,j)(0)}Y_{(i+1,j)(0)})X_{(i,j)(1)}Y_{(i+1,j)(1)} & i = 1, \\ \frac{1}{4}J(Z + \mathbb{1}_2)_{(i,j)(0)} + \frac{1}{4}J(Z + \mathbb{1}_2)_{(i,j+1)(0)} \\ \quad + J_{ij}^V(Y_{(i,j)(0)}X_{(i+1,j)(0)} - X_{(i,j)(0)}Y_{(i+1,j)(0)})X_{(i,j)(1)}Y_{(i+1,j)(1)} & 2 \leq i \leq P - 1, \\ \frac{1}{4}J(Z + \mathbb{1}_2)_{(i,j)(0)} & i = P. \end{cases}$$

For ease of notation, we define

$$M_{ij}^R = \exp\left(-\frac{\beta}{n}H_{ij}^R\right), R \in \{H, V\} \tag{8.7}$$

$$T_\tau^R = \exp\left(-\frac{\beta}{n}\sum_{(i,j)\in\tau_R} H_{ij}^R\right) = \prod_{(i,j)\in\tau_R} M_{ij}^R, \quad \tau \in \{a, b\}, \tag{8.8}$$

where n is any non-negative integer, and the second equality in eq. (8.8) follows because the sets τ_R ensure that the H_{ij}^R all act on disjoint pairs of lattice sites, and hence commute.

We now make use of the Suzuki-Trotter expansion, which states that

$$\exp\left(A_0 + \cdots + A_{m-1}\right) = \left[\exp\left(\frac{1}{n}A_0\right)\cdots\exp\left(\frac{1}{n}A_{m-1}\right)\right]^n + O\left(\frac{1}{n}\right)$$

for any (even non-commuting) operators A_i, provided that $n \geq \max_i \|A_i\|$ [Suz76].

From eq. (8.6), we thus have

$$\mathcal{Z} \sim \sum_{\alpha\in\{0,1\}^{PQ}} \left(\langle\alpha|_0 \odot \langle\chi|_1\right) \cdot \Pi \cdot \left(|\alpha\rangle_0 \odot |\chi\rangle_1\right). \tag{8.9}$$

where

$$\begin{aligned}\Pi &\coloneqq \left(T_a^H T_a^V T_b^H T_b^V\right)^n \\ &= \prod_{q=0}^{n-1}\left[\left(\prod_{(i,j)\in a_H} M_{ij}^H\right)\left(\prod_{(i,j)\in a_V} M_{ij}^V\right)\left(\prod_{(i,j)\in b_H} M_{ij}^H\right)\left(\prod_{(i,j)\in b_V} M_{ij}^V\right)\right].\end{aligned} \tag{8.10}$$

We include q in the above expression to illustrate how it relates to Figure 8.7.

If we now consider each pair of qubit sites $(i,j)(0)$ and $(i,j)(1)$ to be a single 4-dimensional qudit site, we can view Π as a circuit, displayed in Figure 8.7. In order for Π to represent a realisable quantum circuit, however, M_{ij}^R must be unitary for all i, j, R, which occurs only when β is imaginary (see eq. (8.7)).

If the original quantum system is solved (i.e. there is a closed form for $\mathcal{Z}$) for *imaginary* β, then this circuit is efficiently simulatable on a classical computer, in the sense that the trace over the 0-system can be calculated, as in eq. (8.9).

Remark: A similar result is obtained in [VdNDRB09], in which classical lattice models are mapped to classically-simulatable quantum circuits. The authors additionally note that β may be complex in order for the constructed gates to be unitary.

An alternative view

Using the ideas of Knill and Laflamme [KL98], the circuit in Figure 8.6 essentially calculates the trace over the 0-system of Π. More specifically, the output of the circuit is 0 with probability $(\frac{1}{2} + \frac{1}{2^{PQ}}\mathrm{Re}\ \mathcal{Z})$. Using the state $\frac{1}{2}(|0\rangle - i|1\rangle)(\langle 0| + i\langle 1|)$ on the first line similarly produces output 0 with probability $(\frac{1}{2} + \frac{1}{2^{PQ}}\mathrm{Im}\ \mathcal{Z})$. A closed form for $\mathcal{Z}$ thus implies that we can efficiently calculate the outcome probabilities of these circuits.

If no closed form for $\mathcal{Z}$ is known, however, quantum process tomography (see Section 2.2.1) can be used on these circuits to determine $\mathcal{Z}$ to arbitrary accuracy.

We now examine the form of the gates that make up Π in more detail.

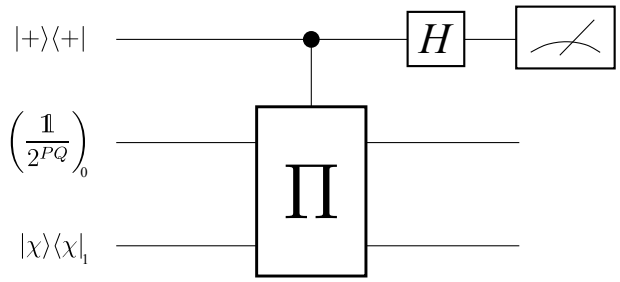


Figure 8.6: A circuit based on the 'one clean qubit' work by Knill and Laflamme [KL98], for which the output probabilities are dependent upon $\mathcal{Z}$, and the relationship between Π and $\mathcal{Z}$ is defined in eq. (8.9). In particular, the probability of the output being 0 is $(\frac{1}{2} + \frac{1}{2^{PQ}}\mathrm{Re}\ \mathcal{Z})$. The first line of the circuit is a single qubit, whereas the second and third lines consist of PQ qubits each. The middle line is set to the fully mixed state.

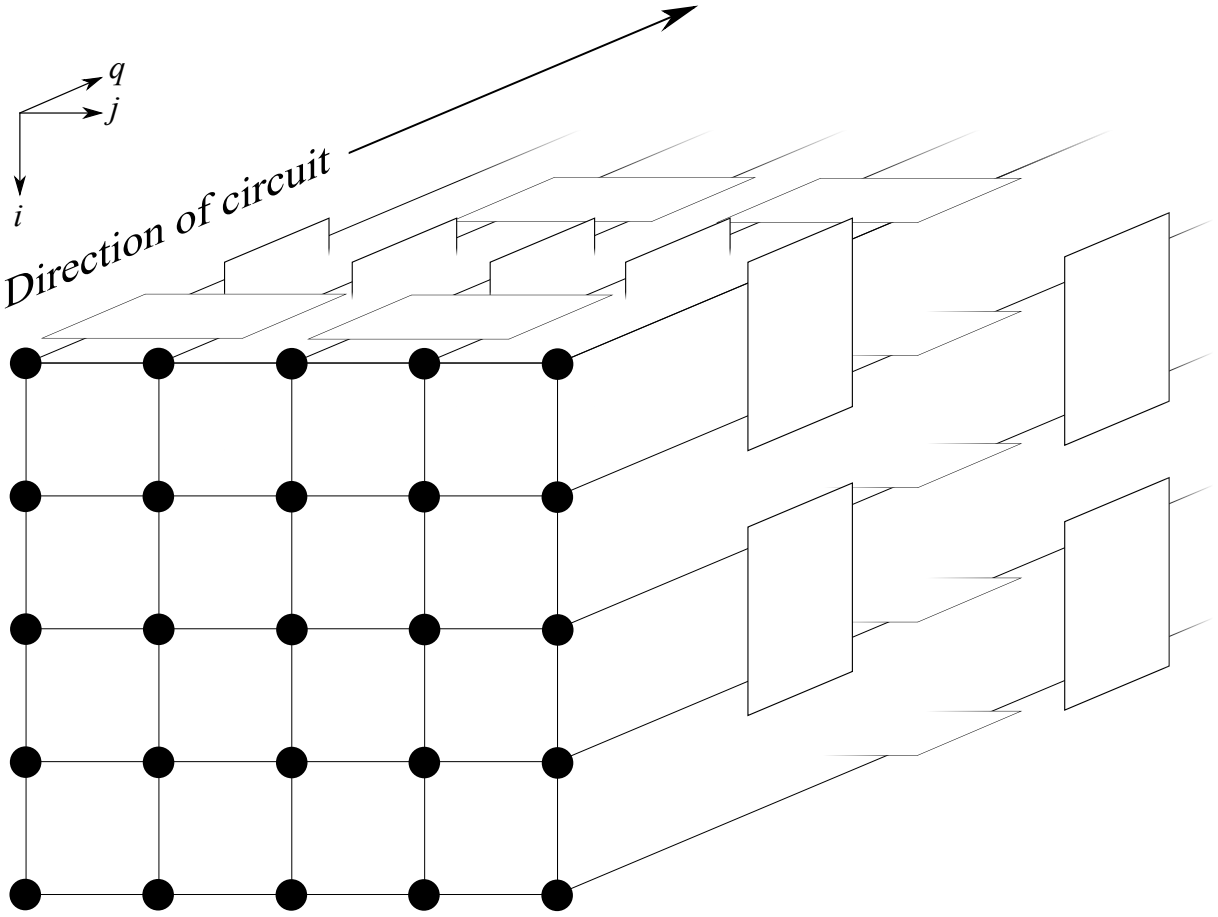


Figure 8.7: The resultant circuit after applying the mapping in Section 8.3. There are PQ lines, each of which represents a 4-dimensional qudit. The gates shown in the circuit are the individual local unitary operators given in eq. (8.10).

Evaluation of M_{ij}^R when $R = H$ (and $2 \leq j \leq Q-1$)

From the definition of M_{ij}^R (eq. (8.7)), we have

$$\begin{aligned}
M_{ij}^H &= \exp\left(-\frac{\beta}{n} H_{ij}^H\right) \\
&= \exp\left(-\frac{\beta}{n}\left(\frac{1}{4}J(Z\otimes \mathbb{1}_2^{\otimes 2} + \mathbb{1}_2\otimes Z\otimes \mathbb{1}_2 + 2\mathbb{1}_2^{\otimes 3}) + J_{ij}^H(X\otimes X\otimes Z + Y\otimes Y\otimes Z)\right)\otimes \mathbb{1}_2\right) \\
&= \exp\left(-\frac{\beta}{n}\left(\frac{1}{4}J(Z\otimes \mathbb{1}_2^{\otimes 2} + \mathbb{1}_2\otimes Z\otimes \mathbb{1}_2 + 2\mathbb{1}_2^{\otimes 3}) + J_{ij}^H(X\otimes X\otimes Z + Y\otimes Y\otimes Z)\right)\right)\otimes \mathbb{1}_2 \\
&= \left(\begin{array}{c|c|c|c} A\mathbb{1}_2 & 0 & 0 & 0 \\ \hline 0 & C_{ij}^H\mathbb{1}_2 & B_{ij}^H Z & 0 \\ \hline 0 & B_{ij}^H Z & C_{ij}^H\mathbb{1}_2 & 0 \\ \hline 0 & 0 & 0 & \mathbb{1}_2 \end{array}\right)\otimes \mathbb{1}_2,
\end{aligned}$$

where each block is a 2×2 matrix, and

$$\begin{aligned}
A &= e^{-J\beta/n}, \\
B_{ij}^R &= -e^{-J\beta/2n}\sinh\left(2J_{ij}^R\beta/n\right), \\
C_{ij}^R &= e^{-J\beta/2n}\cosh\left(2J_{ij}^R\beta/n\right).
\end{aligned}$$

Evaluation of M_{ij}^R when $R = V$ (and $2 \leq i \leq P-1$)

Again from the definition of M_{ij}^R (eq. (8.7)), we have

$$\begin{aligned}
M_{ij}^V &= \exp\left(-\frac{\beta}{n} H_{ij}^V\right) \\
&= \exp\left(-\frac{\beta}{n}\left(\frac{1}{4}J(Z\otimes \mathbb{1}_2^{\otimes 3} + \mathbb{1}_2\otimes Z\otimes \mathbb{1}_2^{\otimes 2} + 2\mathbb{1}_2^{\otimes 4}) + J_{ij}^V(Y\otimes X\otimes X\otimes Y - X\otimes Y\otimes X\otimes Y)\right)\right) \\
&= \left(\begin{array}{c|c|c|c} A\mathbb{1}_4 & 0 & 0 & 0 \\ \hline 0 & C_{ij}^V\mathbb{1}_4 & iB_{ij}^V X\otimes Y & 0 \\ \hline 0 & iB_{ij}^V X\otimes Y & C_{ij}^V\mathbb{1}_4 & 0 \\ \hline 0 & 0 & 0 & \mathbb{1}_4 \end{array}\right),
\end{aligned} \tag{8.11}$$

where each block is a 4×4 matrix.

8.4 An example: the Hubbard Model

In this section we explain how the above mapping may be applied to the Hubbard model [Tas98a, Tas98b, Lie04, Jar92, FK90], a well-studied model that attempts to approximate the behaviour of electrons in a solid by ignoring all but the shortest-range interactions. It is described by a Hamiltonian of the form [Tas98b]

$$H = -\sum_{\langle i,j\rangle,\sigma} t_{ij} a_i(\sigma)^\dagger a_j(\sigma) + \sum_i U_i n_i(1) n_i(-1) - \mu \sum_i (n_i(1) + n_i(-1))$$

where $\langle i,j\rangle$ denotes the set of nearest neighbours i and j, $\sigma \in \{1,-1\}$ is the 'flavour' of the fermion, $n_i(\sigma) := a_i(\sigma)^\dagger a_i(\sigma)$ is the number operator, and t_{ij}, U_i, and μ are real constants.

Unlike the model described in Section 8.2, we deal here with *two* flavours of fermions. There are two natural cases for investigation:

1. the flavours of fermions are exclusive, i.e. the site can be empty, contain a fermion of flavour 1, or contain a fermion of flavour -1 (note that this implies that the second term in the Hamiltonian is 0);
2. the flavours of fermions can co-exist, i.e. the site can be empty, contain a fermion of flavour 1, contain a fermion of flavour -1, or contain both a fermion of flavour 1 and a fermion of flavour -1.

We focus here on case 1 (although the methods used can also be applied to case 2) and develop a generalisation of the Jordan-Wigner transform for multiple flavours of fermions.

The $a_i(\sigma)$ must obey the following relations [Tas98b]:

$$\{a_i(\sigma)^\dagger, a_j(\sigma')^\dagger\} = \{a_i(\sigma), a_j(\sigma')\} = 0,$$
$$\{a_i(\sigma)^\dagger, a_j(\sigma')\} = \delta_{ij}\delta_{\sigma\sigma'}\mathbb{1},$$

where $\sigma, \sigma' \in \{1,-1\}$. In addition, we require the exclusivity of fermion flavours. This is a restriction on the state space of the fermions, but here we simplify our analysis by imposing the condition on the operators instead:

$$a_i(\sigma)^\dagger a_i(\sigma')^\dagger = 0.$$

For ease of notation, let us define $a_i(0)^{(\dagger)} := \mathbb{1}$.

If we write

$$|\alpha\rangle = |\alpha_1, \ldots, \alpha_N\rangle := \left[a_1(\alpha_1)^\dagger\right] \cdots \left[a_N(\alpha_N)^\dagger\right] |\Omega\rangle$$

with $\alpha_i \in \{1, 0, -1\}$, then, as in Section 1.1.3, it is straightforward to determine that $\langle\alpha|\alpha'\rangle = \delta_{\alpha\alpha'}$.

In addition, for $\sigma \in \{1, -1\}$, the anti-commutation relations imply that

$$\begin{aligned} a_i(\sigma)|\alpha\rangle &= a_i(\sigma)\left(\left[a_1(\alpha_1)^\dagger\right]\cdots\left[a_i(\alpha_i)^\dagger\right]\cdots\left[a_N(\alpha_N)^\dagger\right]|\Omega\rangle\right) \\ &= \begin{cases} (-1)^{\left|\{j:\alpha_j\neq 0, j<i\}\right|}|\alpha'\rangle & \text{if } \alpha_i = \sigma \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{8.12}$$

where $\alpha'_j = \alpha_j$ for $j \neq i$ and $\alpha'_i = 0$.

If we now write

$$|1\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad |0\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad |-1\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

then eq. (8.12) implies that in the spin paradigm,

$$a_i(\sigma) = \Phi_1 \cdots \Phi_{i-1} S_i(\sigma),$$

where

$$\Phi = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \quad S(1) = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad S(-1) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$

This construction is easily generalised to arbitrarily many fermionic flavours, giving a natural generalisation of the Jordan-Wigner transform. The case of two flavours yields a much more straightforward generalisation of the transform for spin-1 particles introduced in [BO01].

For ease of notation, let us also define, for any 2×2 matrix $Q = \left(\begin{smallmatrix} Q_{00} & Q_{01} \\ Q_{10} & Q_{11} \end{smallmatrix}\right)$,

$$Q(1) = \begin{pmatrix} Q_{00} & Q_{01} & 0 \\ Q_{10} & Q_{11} & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad Q(-1) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & Q_{00} & Q_{01} \\ 0 & Q_{10} & Q_{11} \end{pmatrix}.$$

Extending the technique given in Section 8.2.1, where a second '1-layer' of fermions was added to the lattice, we now add two extra layers of fermions to the lattice, the 1-layer and the (-1)-layer. We order the sites as $1(0)$, $1(1)$, $1(-1)$, $2(0)$, $2(1)$, $2(-1)$, $\ldots$, $N(0)$, $N(1)$, $N(-1)$. To aid notation, we additionally define $\Phi_{i(\sigma)\to j(\sigma')}$ to be the operator that effects Φ on all sites between $i(\sigma)$ and $j(\sigma')$ (exclusive). In particular, $\Phi_{i(0)\to i(1)} = \mathbb{1}$ and $\Phi_{i(0)\to i(-1)} = \Phi_{i(1)}$.

We now continue as in Section 8.2.1 to obtain analogues of the $P_{i(1)\,j(1)}$ operators. The Majorana fermions (see Section 1.1.3) are

$$c_{i(\sigma)}(\sigma) \coloneqq a_{i(\sigma)}(\sigma) + a_{i(\sigma)}(\sigma)^\dagger, \quad d_{i(\sigma)}(\sigma) \coloneqq i(a_{i(\sigma)}(\sigma) - a_{i(\sigma)}(\sigma)^\dagger).$$

As before, let $\Lambda(\sigma)$ be the set of all directed edges on the σ-layer pointing vertically down the lattice (see Figure 8.4). We define the $P_{i(\sigma)\,j(\sigma)}(\sigma)$ operators for each pair $\langle i(\sigma), j(\sigma)\rangle \in \Lambda(\sigma)$ to be

$$P_{i(\sigma)\,j(\sigma)}(\sigma) = \theta_{i(\sigma)\,j(\sigma)} i c_{i(\sigma)}(\sigma) d_{i(\sigma)}(\sigma)$$

where the $\theta_{i(\sigma)\,j(\sigma)} = \pm 1$ are to be determined.

Similar to Section 8.2.1, we find that

$$[P_{i(\sigma)\,j(\sigma)}(\sigma), P_{k(\sigma')\,l(\sigma')}(\sigma')] = [P_{i(\sigma)\,j(\sigma)}(\sigma), a_{m(0)}(\sigma)^{(\dagger)}] = 0,$$

for $\sigma, \sigma' \in \{1, -1\}$.

Now, $(P_{i(\sigma)\,j(\sigma)}(\sigma))^2 = \mathbb{1}_{i(\sigma)}(\sigma)\mathbb{1}_{j(\sigma)}(\sigma)$. Thus,

- in the $\{|0\rangle, |1\rangle\}$-spanned subspace, $P_{i(1)\,j(1)}(1)$ has eigenvalues ± 1. By fixing $\theta_{i(1)\,j(1)}$ appropriately, we can find a state $|\chi\rangle_1$ such that

$$P_{i(1)\,j(1)}(1)\,|\chi\rangle_1 = |\chi\rangle_1 \quad \forall\ \langle i(1), j(1)\rangle \in \Lambda(1)$$

- in the $\{|0\rangle, |-1\rangle\}$-spanned subspace, $P_{i(-1)\,j(-1)}(-1)$ has eigenvalues ± 1. By fixing $\theta_{i(-1)\,j(-1)}$ appropriately, we can find a state $|\xi\rangle_{-1}$ such that

$$P_{i(-1)\,j(-1)}(-1)\,|\xi\rangle_{-1} = |\xi\rangle_{-1} \quad \forall\ \langle i(-1), j(-1)\rangle \in \Lambda(-1)$$

As before, we find that

$$P_{i(\sigma)\,j(\sigma)}(\sigma)(|\varphi\rangle_0 \odot |\chi\rangle_1 \odot |\xi\rangle_{-1}) = |\varphi\rangle_0 \odot |\chi\rangle_1 \odot |\xi\rangle_{-1},$$
$$H(|\varphi\rangle_0 \odot |\zeta_1\rangle_1 \odot |\zeta_2\rangle_{-1}) = H(|\varphi\rangle_0) \odot |\zeta_1\rangle_1 \odot |\zeta_2\rangle_{-1}.$$

We construct a new Hamiltonian H' from H by altering the vertical hopping terms as follows:

$$a_{i(0)}(\sigma)^\dagger a_{j(0)}(\sigma) + a_{j(0)}(\sigma)^\dagger a_{i(0)}(\sigma) \mapsto (a_{i(0)}(\sigma)^\dagger a_{j(0)}(\sigma) + a_{j(0)}(\sigma)^\dagger a_{i(0)}(\sigma)) P_{i(\sigma)\,j(\sigma)}(\sigma).$$

From the above equations, it follows that

$$H'\left(|\varphi\rangle_0 \odot |\chi\rangle_1 \odot |\xi\rangle_{-1}\right) = H\left(|\varphi\rangle_0\right) \odot |\chi\rangle_1 \odot |\xi\rangle_{-1},$$

and thus the local fermionic system described by H follows the same dynamics as the 0-layer sites of the system described by H'. In addition, H' is local in the spin representation.

H consists of only three types of terms:

$$n_{(i,j)(0)}(\sigma); \quad a_{(i,j)(0)}(\sigma)^\dagger a_{(i,j+1)(0)}(\sigma) + a_{(i,j+1)(0)}(\sigma)^\dagger a_{(i,j)(0)}(\sigma);$$
$$a_{(i,j)(0)}(\sigma)^\dagger a_{(i+1,j)(0)}(\sigma) + a_{(i+1,j)(0)}(\sigma)^\dagger a_{(i,j)(0)}(\sigma).$$

In H', these respectively become:

$$n_{(i,j)(0)}(\sigma); \quad a_{(i,j)(0)}(\sigma)^\dagger a_{(i,j+1)(0)}(\sigma) + a_{(i,j+1)(0)}(\sigma)^\dagger a_{(i,j)(0)}(\sigma);$$
$$\left(a_{(i,j)(0)}(\sigma)^\dagger a_{(i+1,j)(0)}(\sigma) + a_{(i+1,j)(0)}(\sigma)^\dagger a_{(i,j)(0)}(\sigma)\right) P_{(i,j)(\sigma)\,(i+1,j)(\sigma)}(\sigma),$$

which, in the spin representation, have the form

$$\frac{1}{2}(Z(\sigma) + \mathbb{1}_2(\sigma))_{(i,j)(0)};$$
$$\frac{1}{2}\left(X_{(i,j)(0)}(\sigma)X_{(i,j+1)(0)}(\sigma) + Y_{(i,j)(0)}(\sigma)Y_{(i,j+1)(0)}(\sigma)\right)\Phi_{(i,j)(1)}\Phi_{(i,j)(-1)};$$
$$-\frac{1}{2}\theta_{(i,j)(\sigma)\,(i+1,j)(\sigma)}(X_{(i,j)(0)}(\sigma)Y_{(i+1,j)(0)}(\sigma) - Y_{(i,j)(0)}(\sigma)X_{(i+1,j)(0)}(\sigma))\cdot$$
$$X_{(i,j)(\sigma)}(\sigma)Y_{(i+1,j)(\sigma)}(\sigma)\Phi_{(i,j)(0)\to(i,j)(\sigma)}\Phi_{(i+1,j)(0)\to(i+1,j)(\sigma)}.$$

This gives a final Hamiltonian

$$\begin{aligned}
H' = J &\sum_{\substack{\sigma, 1\le i\le P,\\ 1\le j\le Q}} (Z(\sigma) + \mathbb{1}_2(\sigma))_{(i,j)(0)} \\
&+ \sum_{\substack{\sigma, 1\le i\le P,\\ 1\le j\le Q-1}} J_{ij}^H \left(X_{(i,j)(0)}(\sigma)X_{(i,j+1)(0)}(\sigma) + Y_{(i,j)(0)}(\sigma)Y_{(i,j+1)(0)}(\sigma)\right)\Phi_{(i,j)(1)}\Phi_{(i,j)(-1)} \\
&+ \sum_{\substack{\sigma, 1\le i\le P-1,\\ 1\le j\le Q}} J_{ij}^V (X_{(i,j)(0)}(\sigma)Y_{(i+1,j)(0)}(\sigma) - Y_{(i,j)(0)}(\sigma)X_{(i+1,j)(0)}(\sigma))\cdot \\
&\qquad X_{(i,j)(\sigma)}(\sigma)Y_{(i+1,j)(\sigma)}(\sigma)\Phi_{(i,j)(0)\to(i,j)(\sigma)}\Phi_{(i+1,j)(0)\to(i+1,j)(\sigma)}
\end{aligned}$$

for some real constants J, J_{ij}^H, J_{ij}^V.

If we write

$$H_{ij}^H = \begin{cases}
\sum_{\sigma=\pm1} \Big(\frac{1}{2}J(Z(\sigma) + \mathbb{1}_2(\sigma))_{(i,j)(0)} + \frac{1}{4}J(Z(\sigma) + \mathbb{1}_2(\sigma))_{(i,j+1)(0)} \\
\quad + J_{ij}^H(X_{(i,j)(0)}(\sigma)X_{(i,j+1)(0)}(\sigma) + Y_{(i,j)(0)}(\sigma)Y_{(i,j+1)(0)}(\sigma))\Phi_{(i,j)(1)}\Phi_{(i,j)(-1)}\Big) & j = 1, \\
\sum_{\sigma=\pm1} \Big(\frac{1}{4}J(Z(\sigma) + \mathbb{1}_2(\sigma))_{(i,j)(0)} + \frac{1}{4}J(Z(\sigma) + \mathbb{1}_2(\sigma))_{(i,j+1)(0)} \\
\quad + J_{ij}^H(X_{(i,j)(0)}(\sigma)X_{(i,j+1)(0)}(\sigma) + Y_{(i,j)(0)}(\sigma)Y_{(i,j+1)(0)}(\sigma))\Phi_{(i,j)(1)}\Phi_{(i,j)(-1)}\Big) & 2 \le j \le Q-1, \\
\sum_{\sigma=\pm1} \frac{1}{4}J(Z(\sigma) + \mathbb{1}_2(\sigma))_{(i,j)(0)} & j = Q,
\end{cases}$$

$$H_{ij}^V = \begin{cases} \sum_{\sigma=\pm 1} \Big(\frac{1}{2} J(Z(\sigma) + \mathbb{1}_2(\sigma))_{(i,j)(0)} + \frac{1}{4} J(Z(\sigma) + \mathbb{1}_2(\sigma))_{(i+1,j)(0)} \\ \qquad + J_{ij}^V (X_{(i,j)(0)}(\sigma) Y_{(i+1,j)(0)}(\sigma) - Y_{(i,j)(0)}(\sigma) X_{(i+1,j)(0)}(\sigma)) \cdot \\ \qquad X_{(i,j)(\sigma)}(\sigma) Y_{(i+1,j)(\sigma)}(\sigma) \Phi_{(i,j)(0)\to(i,j)(\sigma)} \Phi_{(i+1,j)(0)\to(i+1,j)(\sigma)} \Big) & i = 1, \\ \sum_{\sigma=\pm 1} \Big(\frac{1}{4} J(Z(\sigma) + \mathbb{1}_2(\sigma))_{(i,j)(0)} + \frac{1}{4} J(Z(\sigma) + \mathbb{1}_2(\sigma))_{(i+1,j)(0)} \\ \qquad + J_{ij}^V (X_{(i,j)(0)}(\sigma) Y_{(i+1,j)(0)}(\sigma) - Y_{(i,j)(0)}(\sigma) X_{(i+1,j)(0)}(\sigma)) \cdot \\ \qquad X_{(i,j)(\sigma)}(\sigma) Y_{(i+1,j)(\sigma)}(\sigma) \Phi_{(i,j)(0)\to(i,j)(\sigma)} \Phi_{(i+1,j)(0)\to(i+1,j)(\sigma)} \Big) & 2 \le i \le P-1, \\ \sum_{\sigma=\pm 1} \frac{1}{4} J(Z(\sigma) + \mathbb{1}_2(\sigma))_{(i,j)(0)} & i = P, \end{cases}$$

we see that

$$H' = \sum_{(i,j)\in a_H} H_{ij}^H + \sum_{(i,j)\in a_V} H_{ij}^V + \sum_{(i,j)\in b_H} H_{ij}^H + \sum_{(i,j)\in b_V} H_{ij}^V.$$

Setting M_{ij}^R and T_τ^R as in eq. (8.7) and eq. (8.8) respectively, and Π as in eq. (8.10), gives a local quantum circuit with output probabilities dependent on $\mathcal{Z}$, the partition function of the original system. If $\mathcal{Z}$ is known, the corresponding circuits are efficiently classically simulatable. If $\mathcal{Z}$ is unknown, however, quantum process tomography (see Section 2.2.1) can be used on the circuit to estimate $\mathcal{Z}$ to any desired accuracy.

8.5 An aside: computing thermodynamic properties of quantum lattice models

This section deviates from the main focus of the chapter to illustrate another useful property of the mapping that we derived in Sections 8.2 and 8.3. Here we show how thermodynamic properties of quantum fermionic lattice models may be computed using classical algorithms.

A result that we make use of is the so-called *quantum-classical mapping* [HKM15], a technique for transforming a quantum partition function (of the form $\mathrm{Tr}[e^{-\beta H_{qu}}]$) [CL00] for a d-dimensional system into a classical partition function (of the form $\sum_s e^{-\beta' H_{cl}(s)}$) for a $(d+1)$-dimensional system. The mapping is not exact, but produces an approximation that can be made arbitrarily close.

There are two parts to our analysis:

1. Combining the quantum-classical mapping with the results from Sections 8.2 and 8.3, we derive a transformation that maps a local quantum fermionic lattice model in 2 dimensions to a local pseudo-classical spin lattice model in 3 dimensions.

2. We subsequently show how the Metropolis-Hastings Algorithm [MRR$^+$53, Has70, CG95] may be applied to this system to derive thermodynamic properties of the original quantum lattice model.

8.5.1 Mapping the quantum lattice model to a pseudo-classical lattice model

Here we apply the quantum-classical mapping to the Hamiltonian H' given in Section 8.3, though it is easily generalised to the Hubbard model presented in Section 8.4.

Returning to eq. (8.9), we note that $\{|\alpha\rangle_0 \odot |\alpha'\rangle_1\}_{\alpha,\alpha' \in \{0,1\}^{PQ}}$ forms a basis for the full fermionic space (over both the 0-layer and the 1-layer), which is easily verified using the CCRs as we did with eq. (1.7) and eq. (1.8). We can therefore write the identity operator on the whole space as

$$\mathbb{1} = \sum_{s \in \{0,1\}^{2PQ}} |s\rangle \langle s| ,$$

where $s = (s_0, s_1)$ and $|s\rangle = |s_0\rangle_0 \odot |s_1\rangle_1$.

Inserting a copy of the identity operator between each pair of terms gives

$$\begin{aligned} \mathcal{Z} \sim \sum_{\substack{\alpha \in \{0,1\}^{PQ}, \\ s^0,\ldots,s^{4n} \in \{0,1\}^{2PQ}}} \Bigg[& \left(\langle\alpha|_0 \odot \langle\chi|_1\right) |s^0\rangle \cdot \\ & \prod_{q=0}^{n-1} \left\langle s^{4q} \right| T_a^H \left| s^{4q+1} \right\rangle \left\langle s^{4q+1} \right| T_a^V \left| s^{4q+2} \right\rangle \left\langle s^{4q+2} \right| T_b^H \left| s^{4q+3} \right\rangle \left\langle s^{4q+3} \right| T_b^V \left| s^{4q+4} \right\rangle \cdot \\ & \left\langle s^{4n} \right| \left(|\alpha\rangle_0 \odot |\chi\rangle_1\right) \Bigg] . \end{aligned}$$

Writing $U_\tau^{t,R} := \left\langle s^t \right| T_\tau^R \left| s^{t+1} \right\rangle$, we see that

$$\mathcal{Z} \sim \sum_{\substack{\alpha \in \{0,1\}^{PQ}, \\ s^0,\ldots,s^{4n} \in \{0,1\}^{2PQ}}} \underbrace{\left[\left(\langle\alpha|_0 \odot \langle\chi|_1\right) \left|s^0\right\rangle \right]}_{\text{lattice boundary term}} \underbrace{\left[\prod_{q=0}^{n-1} U_a^{4q,H} U_a^{4q+1,V} U_b^{4q+2,H} U_b^{4q+3,V} \right]}_{\text{main lattice term}} \underbrace{\left[\left\langle s^{4n} \right| \left(|\alpha\rangle_0 \odot |\chi\rangle_1\right) \right]}_{\text{lattice boundary term}} . \tag{8.13}$$

Remark: The *lattice boundary terms* are so called because they only affect the front ($t = 0$, see Figure 8.8) and back ($t = 4n$, see Figure 8.8) of our now 3-dimensional lattice. The *main lattice term* contains the interactions that run throughout the lattice. See Figure 8.8 for more details.

We shall now evaluate each term of the expression in eq. (8.13), writing it in the form of an exponential.

The main lattice term

Here we consider the term

$$\prod_{q=0}^{n-1} U_a^{4q,H} U_a^{4q+1,V} U_b^{4q+2,H} U_b^{4q+3,V}.$$

Since the interactions in each of the $U_\tau^{t,R}$ are disjoint, we can write

$$U_\tau^{t,R} = \prod_{(i,j)\in\tau_R} W_{ij}^{t,R},$$

where

$$W_{ij}^{t,R} = \left\langle S_{ij}^{t,R} \right| M_{ij}^R \left| S_{ij}^{t+1,R} \right\rangle, \tag{8.14}$$

and

$$\left|S_{ij}^{t,H}\right\rangle = \left|s^t_{(i,j)(0)} \quad s^t_{(i,j+1)(0)} \quad s^t_{(i,j)(1)} \quad s^t_{(i,j+1)(1)}\right\rangle,$$
$$\left|S_{ij}^{t,V}\right\rangle = \left|s^t_{(i,j)(0)} \quad s^t_{(i+1,j)(0)} \quad s^t_{(i,j)(1)} \quad s^t_{(i+1,j)(1)}\right\rangle.$$

Note that as we no longer require M_{ij}^R to be unitary, we choose β to be real (see eq. (8.7)).

We now write $|S\rangle = |S_1 \;\; S_2 \;\; S_3 \;\; S_4\rangle$ and $|S'\rangle = |S'_1 \;\; S'_2 \;\; S'_3 \;\; S'_4\rangle$, and let $\mathcal{S}_{ij}^R = \{(S,S') : \langle S| M_{ij}^R |S'\rangle \neq 0\}$. Then from eq. (8.14) we see that

$$W_{ij}^{t,R} = \Delta_{ij}^{t,R} \exp\left(E_{ij}^{t,R}\right),$$

where

$$\Delta_{ij}^{t,R} := \sum_{(S,S')\in\mathcal{S}_{ij}^R} \delta_{S_{ij}^{t,R},\,S}\;\delta_{S_{ij}^{t+1,R},\,S'}$$
$$E_{ij}^{t,R} := \sum_{(S,S')\in\mathcal{S}_{ij}^R} \log\left(\langle S| M_{ij}^R |S'\rangle\right) \delta_{S_{ij}^{t,R},\,S}\;\delta_{S_{ij}^{t+1,R},\,S'},$$

with $\log z$ defined by its principal branch (i.e. for $z \in \mathbb{C}$, $\log z = \log|z| + i\arg z$, where $\arg z \in (-\pi, \pi]$).

This is easily verified by noting that $\Delta_{ij}^{t,R}$ restricts us to the non-zero elements of M_{ij}^R, and that the δ's in $E_{ij}^{t,R}$ ensure that the sum collapses to one term.

Notice that thus far our spins have taken values in $\{0,1\}$, as we started with a quantum system. Now that we are dealing with a classical system of spins, however, it is more useful to define spins to take values in $\{1,-1\}$. We define $\bar{s}^t_{(i,j)(\sigma)} = 1 - 2s^t_{(i,j)(\sigma)}$,

where $s^t_{(i,j)(\sigma)} \in \{0,1\}$, to be the value of our classical spin. The δ-terms in the exponent can now be written as interactions between these classical spins. For example,

$$\begin{aligned}\delta_{S^{t,H}_{ij},\,1101} &= \delta_{s^t_{(i,j)(0)},\,1}\;\delta_{s^t_{(i,j+1)(0)},\,1}\;\delta_{s^t_{(i,j)(1)},\,0}\;\delta_{s^t_{(i,j+1)(1)},\,1}\\ &= s^t_{(i,j)(0)}\;s^t_{(i,j+1)(0)}\;\left(1-s^t_{(i,j)(1)}\right)\;s^t_{(i,j+1)(1)}\\ &= \frac{1}{16}\left(1-\bar{s}^t_{(i,j)(0)}\right)\left(1-\bar{s}^t_{(i,j+1)(0)}\right)\left(1+\bar{s}^t_{(i,j)(1)}\right)\left(1-\bar{s}^t_{(i,j+1)(1)}\right)\end{aligned}$$

As a more complete example, the term in the exponent corresponding to the element in the top-left corner of the matrix in eq. (8.11) is

$$\begin{aligned}(\log A)\left(1+\bar{s}^t_{(i,j)(0)}\right)\left(1+\bar{s}^t_{(i+1,j)(0)}\right)\left(1+\bar{s}^t_{(i,j)(1)}\right)\left(1+\bar{s}^t_{(i+1,j)(1)}\right)\cdot\\ \left(1+\bar{s}^{t+1}_{(i,j)(0)}\right)\left(1+\bar{s}^{t+1}_{(i+1,j)(0)}\right)\left(1+\bar{s}^{t+1}_{(i,j)(1)}\right)\left(1+\bar{s}^{t+1}_{(i+1,j)(1)}\right),\end{aligned}$$

which is an 8-degree, distance-3 classical interaction. In general, each of these interactions will similarly be 8-degree and distance-3.

In addition, δ-terms in $\Delta^{t,R}_{ij}$ can be straightforwardly adjusted to take account of the new classical spin values. For example, a term of the form $\delta_{S^1,\,S^2}$, where $\left|S^i\right\rangle = \left|S^i_1\;\;S^i_2\;\;S^i_3\;\;S^i_4\right\rangle$, is equivalent to $\delta_{\bar{S}^1,\,\bar{S}^2}$, where $\left|\bar{S}^i\right\rangle = \left|\bar{S}^i_1\;\;\bar{S}^i_2\;\;\bar{S}^i_3\;\;\bar{S}^i_4\right\rangle$.

If we now write $\Theta^{t,R}_\tau := \prod_{(i,j)\in\tau_R}\Delta^{t,R}_{ij}$ and $K^{t,R}_\tau := \sum_{(i,j)\in\tau_R}E^{t,R}_{(i,j)}$, we see that

$$U^{t,R}_\tau = \Theta^{t,R}_\tau e^{K^{t,R}_\tau}.$$

We can therefore write the main lattice term of eq. (8.13) as

$$\begin{aligned}&\prod_{q=0}^{n-1}U_a^{4q,H}U_a^{4q+1,V}U_b^{4q+2,H}U_b^{4q+3,V}\\ &=\left[\prod_{q=0}^{n-1}\Theta_a^{4q,H}\Theta_a^{4q+1,V}\Theta_b^{4q+2,H}\Theta_b^{4q+3,V}\right]\exp\left(\sum_{q=0}^{n-1}\left(K_a^{4q,H}+K_a^{4q+1,V}+K_b^{4q+2,H}+K_b^{4q+3,V}\right)\right).\end{aligned}$$

Remark: In applying this procedure to M^H_{ij}, one can either expand the tensor product to get a 16×16 matrix, or one can note that we can work with the 8×8 matrix and include the condition of $\delta_{\bar{s}^t_{(i,j+1)(1)},\,\bar{s}^{t+1}_{(i,j+1)(1)}}$.

A note on reducing the range of locality and degree of the interaction term

In some cases the interaction can be made more local, and the degree can be reduced. For example, consider the top left 4×4 block in eq. (8.11). The δ-terms in $\Delta^{t,V}_{ij}$ restrict us to the diagonal of this block, and hence it would be sufficient to have

$$(\log A)\left(1+\bar{s}^t_{(i,j)(0)}\right)\left(1+\bar{s}^t_{(i+1,j)(0)}\right)\left(1+\bar{s}^{t+1}_{(i,j)(0)}\right)\left(1+\bar{s}^{t+1}_{(i+1,j)(0)}\right),$$

as the relevant term in the exponent, as it references the block as a whole, rather than through individual elements. This interaction term is now 4-degree and distance-2 (rather than 8-degree and distance-3).

The lattice boundary terms

Writing $|\chi\rangle_1 = \sum_{\alpha'} \chi_{\alpha'} |\alpha'\rangle_1$, we see that the boundary term at the front of the lattice in eq. (8.13) is

$$\left(\langle\alpha|_0 \odot \langle\chi|_1\right) \left|s^0\right\rangle = \sum_{\alpha' \in \{0,1\}^{PQ}} \chi^*_{\alpha'} \left(\langle\alpha|_0 \odot \langle\alpha'|_1\right) \left(\left|s^0_0\right\rangle_0 \odot \left|s^0_1\right\rangle_1\right) = \chi^*_{\bar{s}^0_1} \delta_{\alpha,\bar{s}^0_0},$$

where we have changed to classical spins as we did for the main lattice term (i.e. if $s = (s_1, s_2, s_3, \dots)$, then $\bar{s} = (\bar{s}_1, \bar{s}_2, \bar{s}_3, \dots)$).

Similarly, the boundary term at the back of the lattice gives

$$\left\langle s^{4n}\right| \left(|\alpha\rangle_0 \odot |\chi\rangle_1\right) = \sum_{\alpha' \in \{0,1\}^{PQ}} \chi_{\alpha'} \left(\left\langle s^{4n}_0\right|_0 \odot \left\langle s^{4n}_1\right|_1\right) \left(|\alpha\rangle_0 \odot \left|\alpha'\right\rangle_1\right) = \chi_{\bar{s}^{4n}_1} \delta_{\alpha,\bar{s}^{4n}_0}.$$

The pseudo-classical spin lattice model

Substituting the main lattice and lattice boundary terms into eq. (8.13) gives

$$\begin{aligned}
\mathcal{Z} \sim & \sum_{\substack{\bar{s}^0,\dots,\bar{s}^{4n} \\ \in\{-1,1\}^{2PQ}}} \delta_{\bar{s}^0_0,\bar{s}^{4n}_0} \chi^*_{\bar{s}^0_1} \chi_{\bar{s}^{4n}_1} \left[\prod_{q=0}^{n-1} \Theta_a^{4q,H} \Theta_a^{4q+1,V} \Theta_b^{4q+2,H} \Theta_b^{4q+3,V}\right] \cdot \\
& \qquad \exp\left[\sum_{q=0}^{n-1} \left(K_a^{4q,H} + K_a^{4q+1,V} + K_b^{4q+2,H} + K_b^{4q+3,V}\right)\right] \\
= & \sum_{\substack{\bar{s}^0,\dots,\bar{s}^{4n} \\ \in\{-1,1\}^{2PQ}}} \delta_{\bar{s}^0_0,\bar{s}^{4n}_0} \left[\prod_{q=0}^{n-1} \Theta_a^{4q,H} \Theta_a^{4q+1,V} \Theta_b^{4q+2,H} \Theta_b^{4q+3,V}\right] \cdot \\
& \qquad \exp\left[\sum_{q=0}^{n-1} \left(K_a^{4q,H} + K_a^{4q+1,V} + K_b^{4q+2,H} + K_b^{4q+3,V}\right) + \log\left(\chi^*_{\bar{s}^0_1} \chi_{\bar{s}^{4n}_1}\right)\right] \\
= & \sum_{\substack{\bar{s}^0,\dots,\bar{s}^{4n} \\ \in\{-1,1\}^{2PQ}}} \Delta(\bar{s}^0,\dots,\bar{s}^{4n}) e^{-H_{cl}(\bar{s}^0,\dots,\bar{s}^{4n})}
\end{aligned} \tag{8.15}$$

where

$$\begin{aligned}
\Delta(\bar{s}^0,\dots,\bar{s}^{4n}) &:= \delta_{\bar{s}^0_0,\bar{s}^{4n}_0} \prod_{q=0}^{n-1} \Theta_a^{4q,H} \Theta_a^{4q+1,V} \Theta_b^{4q+2,H} \Theta_b^{4q+3,V} \in \{0,1\} \\
H_{cl}(\bar{s}^0,\dots,\bar{s}^{4n}) &:= -\sum_{q=0}^{n-1} \left(K_a^{4q,H} + K_a^{4q+1,V} + K_b^{4q+2,H} + K_b^{4q+3,V}\right) - \log(\chi^*_{\bar{s}^0_1} \chi_{\bar{s}^{4n}_1}),
\end{aligned}$$

with $\log z$ once again defined by its principal branch.

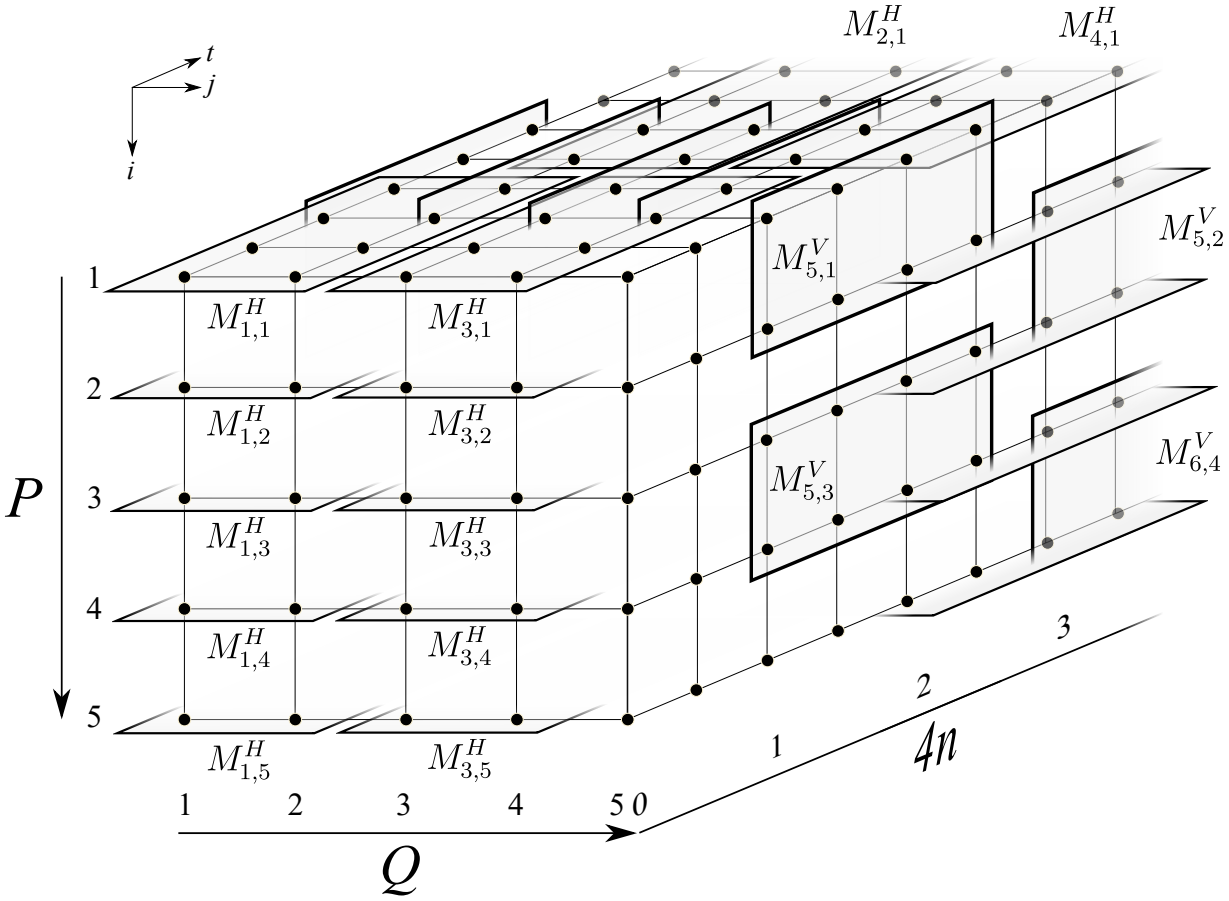


Figure 8.8: The resultant lattice after applying the quantum-classical mapping. Each site is illustrated by a black circle. The local interactions in eq. (8.15) are represented by grey boxes that enclose the relevant sites. Note that the co-ordinates in the i and j directions start at 1, while the co-ordinates in the t direction start at 0.

This is in the form of a classical partition function on a 3-dimensional lattice: two dimensions are present from the quantum lattice (height P, width Q), but with an additional third dimension arising from breaking the Hamiltonian into $4n$ pieces in eq. (8.9) and eq. (8.10) (see Figure 8.8). However, since H_{cl} contains complex terms arising from the logarithms, this Hamiltonian does not represent a real classical system, and hence we describe it as a *pseudo-classical* spin lattice model.

Note that H_{cl} is not only dependent upon the spins at each of the sites, but also on the value of β from the definition of the quantum partition function in eq. (8.6). In addition, H_{cl} is made from distance-3 terms, and each of these terms is an interaction between at most 8 of the spins in the lattice.

8.5.2 Computing thermodyamic properties using the Metropolis-Hastings algorithm

We now wish to make use of the mapping derived above to compute thermodynamic properties of the original quantum system. Let us first rewrite the partition function (eq. (8.15)) as

$$\mathcal{Z} \sim \sum_{\substack{\bar{s}^0,\ldots,\bar{s}^{4n}:\\ \Delta(\bar{s}^0,\ldots,\bar{s}^{4n})=1}} e^{-H_{cl}(\bar{s}^0,\ldots,\bar{s}^{4n})}, \tag{8.16}$$

which is possible since the function $\Delta(\bar{s}^0,\ldots,\bar{s}^{4n})$ defines the set of allowed spin configurations in the system.

The average energy of the quantum system $\langle E\rangle$, for example, is defined to be

$$\langle E\rangle_{\bar{\beta}} = -\frac{1}{\mathcal{Z}}\left.\frac{\partial \mathcal{Z}}{\partial \beta}\right|_{\beta=\bar{\beta}}.$$

Using eq. (8.16), we see that

$$\langle E\rangle_{\bar{\beta}} \sim \sum_{\substack{\bar{s}^0,\ldots,\bar{s}^{4n}:\\ \Delta(\bar{s}^0,\ldots,\bar{s}^{4n})=1}} \left(\left.\frac{\partial H_{cl}(\bar{s}^0,\ldots,\bar{s}^{4n})}{\partial \beta}\right|_{\beta=\bar{\beta}}\right)\left(\frac{1}{\mathcal{Z}}e^{-H_{cl}(\bar{s}^0,\ldots,\bar{s}^{4n})}\right)_{\beta=\bar{\beta}}, \tag{8.17}$$

which is a quantity that can be evaluated using the Metropolis-Hastings Algorithm.

The Metropolis-Hastings algorithm

The Metropolis-Hastings Algorithm [MRR$^+$53, Has70, CG95] employs a Monte-Carlo method to efficiently approximate some intractable probability distributions. It can be adapted to calculate values like $\langle E\rangle_{\bar{\beta}}$ above.

Below we give a description of the algorithm [MRR$^+$53, Has70, CG95].

Let us suppose that a system can take configurations $\sigma \in \Omega$, and that we wish to evaluate the quantity

$$\langle A\rangle = \sum_{\sigma\in\Omega} A(\sigma)\left(\frac{1}{\mathcal{Z}}W(\sigma)\right), \quad \mathcal{Z} = \sum_{\sigma\in\Omega} W(\sigma),$$

where we assume that $\sigma \in \Omega \Rightarrow W(\sigma) \neq 0$.

The idea behind the algorithm is the formation of a random walk process through configurations of the system, after which the probability of arriving at a particular configuration σ is approximately $W(\sigma)/\mathcal{Z}$. Several σ's are obtained in this way, $A(\sigma)$ is calculated for each, and the approximation of $\langle A\rangle$ is calculated as the average of

these values. The way in which the random walk process occurs makes the algorithm particularly suited to evaluating thermodynamic variables in Statistical Physics, which are generally only strongly dependent on a small number of different configurations.

The random walk is effected as follows. Given the current configuration σ and two probability distributions $\{Q_{\sigma\sigma'}\}_{\sigma'}$ and $\{R_{\sigma\sigma'}\}_{\sigma'}$,

- a new configuration σ' is picked with respect to the distribution $\{Q_{\sigma\sigma'}\}_{\sigma'}$;
- the walk either proceeds to σ' (with probability $R_{\sigma\sigma'}$), or remains at σ (otherwise).

Hence the total probability of moving from σ to σ' is $P_{\sigma\sigma'} = Q_{\sigma\sigma'}R_{\sigma\sigma'}$, which we may treat as the transition matrix for the Markov chain formed by the random walk process. Note that it is essential that Q and R are designed so that the Markov chain is ergodic in Ω (i.e. wherever the random walk starts, any configuration may be reached at some point with non-zero probability).

We aim for the equilibrium distribution of our random walk to be π, where $\pi_\sigma = \frac{1}{Z}W(\sigma)$, and hence

$$\pi P = \pi. \tag{8.18}$$

This allows P a lot of freedom, and most implementations of the algorithm impose the following condition, known as the *Detailed Balance Condition* [CG95], to simplify the analysis:

$$\pi_\sigma P_{\sigma\sigma'} = \pi_{\sigma'} P_{\sigma'\sigma} \tag{8.19}$$

for all $\sigma, \sigma' \in \Omega$. This is consistent with eq. (8.18), since summing over σ' gives

$$\pi_\sigma \underbrace{\sum_{\sigma'\in\Omega} P_{\sigma\sigma'}}_{=1} = \underbrace{\sum_{\sigma'\in\Omega} \pi_{\sigma'} P_{\sigma'\sigma}}_{=(\pi P)_\sigma} \Rightarrow \pi P = \pi.$$

We can then use eq. (8.19) to write

$$\frac{R_{\sigma\sigma'}}{R_{\sigma'\sigma}} = \frac{\pi_{\sigma'}}{\pi_\sigma} \cdot \frac{Q_{\sigma'\sigma}}{Q_{\sigma\sigma'}} = \frac{W(\sigma')}{W(\sigma)} \cdot \frac{Q_{\sigma'\sigma}}{Q_{\sigma\sigma'}} =: \lambda_{\sigma\sigma'}.$$

This equation is satisfied by choosing

$$R_{\sigma\sigma'} = \min(1, \lambda_{\sigma\sigma'}).$$

The algorithm then runs as follows:

1. Pick a starting configuration σ, and set $n = 0$ and $A = 0$.

2. Pick σ' according to the distribution $\{Q_{\sigma\sigma'}\}_{\sigma'}$.
3. Accept σ' with probability $R_{\sigma\sigma'} = \min(1, \lambda_{\sigma\sigma'})$. If it is accepted, then set $\sigma = \sigma'$.
4. Repeat Steps 2 and 3 several times to ensure that we are close to the equilibrium distribution.
5. $n = n + 1$; $A = A + A(\sigma)$.
6. Repeat from Step 2 as many times as desired.
7. $\langle A \rangle \approx \frac{1}{n} A$

It is common to repeat the entire algorithm for several starting configurations.

Application of the Metropolis-Hastings algorithm to $\mathcal{Z}$

In order to relate the above algorithm to our expression for $\langle E \rangle_{\bar{\beta}}$ in eq. (8.17), we set $W(\sigma) = e^{-H_{cl}(\sigma)}\Big|_{\beta=\bar{\beta}}$ and $A(\sigma) = [\partial H_{cl}(\sigma)/\partial\beta]_{\beta=\bar{\beta}}$, and we define Ω such that $\sigma = (\bar{s}^0, \ldots, \bar{s}^{4n}) \in \Omega$ if and only if $\Delta(\bar{s}^0, \ldots, \bar{s}^{4n}) = 1$.

Choosing the distribution $Q_{\sigma\sigma'}$ is more complicated. While many implementations of the Metropolis-Hastings algorithm change one of the spins at random, it is not clear in this case that this would lead to an ergodic process (owing to our irregular state space Ω). We therefore adopt the following procedure to pick a new configuration σ' given a configuration σ:

1. Choose an integer $n_s \in \{0, \ldots, N\}$ with probability $1/(2^{n_s+1}(1 - 2^{-N-1}))$, where $N = 2PQ(4n + 1)$ is the total number of spins in the system.
2. Select, uniformly at random, a subset of n_s spins from the system and flip the signs of all of these spins, denoting the resultant configuration σ'.
3. If σ' is a valid configuration of the system (i.e. $\Delta(\sigma') = 1$), then choose σ' as the new configuration (i.e. set $\sigma = \sigma'$). Otherwise, the walk remains at σ.

$Q_{\sigma\sigma'}$ is then easily calculable for $\sigma \neq \sigma'$, and clearly forms an ergodic system, as it is possible (while unlikely) to move from any given spin configuration to any other valid spin configuration in one step.

The Metropolis-Hastings algorithm then allows us to calculate $\langle E \rangle_{\bar{\beta}}$ to any desired accuracy.

8.6 Final remarks and open questions

The procedures described in this chapter are easily generalised to fermionic systems on higher-dimensional lattices, with arbitrarily many flavours of fermions. The circuits and classical lattices that result from these mappings will have one additional dimension, owing to the use of the Suzuki-Trotter expansion.

In addition, it is not necessary for the fermionic system to exist on a regular lattice, as is specified in Definition 8.2.1. This definition can be relaxed to allow for sites at arbitrary points in space, or even to allow for arbitrary graphs in which the sites are vertices and the edges denote that there is a non-negligible interaction between the connected sites.

While the connection between matchgates and the Jordan-Wigner transform [TD02, JM08, JMS15] was the initial motivation for developing classically-simulatable quantum circuits in this way, the resultant circuits differ greatly from those formed from matchgates. In particular, our construction outputs circuits that are generated whole, rather than comprising specific sets of gates. In addition, the property of the Jordan-Wigner transform for one-flavour fermions that leads to the classical simulability of matchgates (see eq. (B.1)) is lost in our general transform for multiple flavours. It is an open question whether there is a generalisation that retains this property, and we explore the idea in Appendix B.1.

In order to ensure that a local, non-interacting fermionic Hamiltonian was mapped to a local spin Hamiltonian, we doubled the number of sites in the lattice (for the Hubbard model, we tripled the number of sites). This begs the question: how large must the auxiliary space be in order to retain locality throughout the mapping? We leave this as an open question.

Concluding remarks

The research topics addressed in this thesis are varied and were drawn from a wide range of interesting questions that can be asked about quantum computation.

The first part of the thesis aimed to gain a deeper understanding of our ability to manipulate unknown quantum operations, and focussed on different procedures for inverting such processes. We presented a detailed study of a novel technique for the in-line inversion of unitary operators, firstly for those acting on qubits, and more generally for those acting on d-dimensional qudit operators. This procedure was shown to have applications both in the refocussing of quantum states, and in the proof of an 'inverse-free' version of the Solovay-Kitaev Theorem.

The second part was motivated by startling results released by Canonne et al. in 2012, on the power of conditional oracles in classical distribution testing. Surprisingly, even the most restricted conditional oracle was able to significantly reduce the number of queries necessary to test several properties. Indeed, in one example, the number of queries was so diminished that it was found to be completely independent of the size of the domain of the distribution. We defined natural quantum analogues of the conditional oracles and investigated their power, developing an algorithm to test whether or not a state was fully mixed, and up to a small conjecture, proved that this algorithm could be sub-linear.

The third part was inspired by the remarkable link between matchgates and the Jordan-Wigner transform [TD02, JM08], sparking an investigation into whether a generalised transform could lead to higher-dimensional analogues of matchgates. While we provided evidence that this is not possible, we were able to develop generalised transforms that could be used to construct other classically-simulatable quantum circuits. Furthermore, we demonstrated the existence of a relationship between local, non-interacting fermionic systems and local, classical spin systems.

The fascinating problems studied in this thesis have given rise to several intriguing open questions and ideas. I look forward to exploring these new pathways and directions in my future research.

Appendix A

Quantum Distribution Testing

A.1 An $\tilde{O}(1/\epsilon^4)$-query PCOND algorithm for testing uniformity

This section aims to provide an understanding of the intuition behind the improved uniformity testing algorithm in Section 4.4.1 and Corollary 5.4.1. We present a simpler and slightly weaker algorithm for uniformity testing requiring $\tilde{O}(1/\epsilon^4)$ queries to the PCOND *oracle, or $\tilde{O}(1/\epsilon^3)$ queries to the* PQCOND *oracle. The* PCOND *version of the algorithm is presented in [CRS15], though here we give a more in-depth derivation.*

Let $\mathcal{A}^{(N)}$ be the uniform distribution on $[N]$ (i.e. $\mathcal{A}^{(N)}(i) = 1/N, i \in [N]$). Given PCOND access to a probability distribution D over $[N]$, we wish to decide (with high probability) whether

- $|D - \mathcal{A}^{(N)}| = 0$ (i.e. $D = \mathcal{A}^{(N)}$), or
- $|D - \mathcal{A}^{(N)}| \geq \epsilon$,

provided that it is guaranteed that one of these is true.

Suppose that the latter option is true, i.e. D is ϵ-far from uniform.

We now partition our domain into two sets: elements of weight at least $1/N$; and elements of weight less than $1/N$. More formally, we define

$$H \coloneqq \left\{h \in [N] : D(h) \geq \frac{1}{N}\right\}, \quad L \coloneqq \left\{l \in [N] : D(l) < \frac{1}{N}\right\}$$

Proposition A.1.1.

$$\sum_{h \in H} \left(D(h) - \frac{1}{N}\right) = \sum_{l \in L} \left(\frac{1}{N} - D(l)\right) \geq \frac{\epsilon}{2}$$

Proof. First, note that $\sum_{i\in[N]} D(i) = 1$ and thus

$$\begin{aligned}
0 &= \sum_{i\in[N]} \left(D(i) - \frac{1}{N}\right) \\
&= \sum_{h\in H} \left(D(h) - \frac{1}{N}\right) + \sum_{l\in L} \left(D(l) - \frac{1}{N}\right) \\
&= \sum_{h\in H} \left(D(h) - \frac{1}{N}\right) - \sum_{l\in L} \left(\frac{1}{N} - D(l)\right)
\end{aligned}$$

and the equality follows.

Since D is ϵ-far from uniform, we have that

$$\begin{aligned}
\epsilon &\leq \sum_{i\in[N]} \left|D(i) - \frac{1}{N}\right| \\
&= \sum_{h\in H} \left|D(h) - \frac{1}{N}\right| + \sum_{l\in L} \left|D(l) - \frac{1}{N}\right| \\
&= 2\sum_{h\in H} \left|D(h) - \frac{1}{N}\right| = 2\sum_{l\in L} \left|\frac{1}{N} - D(l)\right|
\end{aligned}$$

and the inequality follows. □

We define the 'significantly heavy' and 'significantly light' sets

$$\begin{aligned}
H' &:= \left\{h \in [N] : D(h) \geq \frac{1}{N} + \frac{\epsilon}{4N}\right\} \subseteq H, \\
L' &:= \left\{l \in [N] : D(h) < \frac{1}{N} - \frac{\epsilon}{4N}\right\} \subseteq L
\end{aligned}$$

Now,

$$\begin{aligned}
\frac{\epsilon}{2} &\leq \sum_{h\in H} \left(D(h) - \frac{1}{N}\right) \\
&= \sum_{h\in H'} \left(D(h) - \frac{1}{N}\right) + \sum_{h\in H\setminus H'} \underbrace{\left(D(h) - \frac{1}{N}\right)}_{<\frac{\epsilon}{4N}} \\
&< D(H') - \frac{|H'|}{N} + \frac{\epsilon}{4N}(\underbrace{|H|}_{\leq N} - |H'|) \\
&\leq D(H') + \frac{\epsilon}{4} - \left(\frac{|H'|}{N} + \frac{\epsilon|H'|}{4N}\right) \\
&\leq D(H') + \frac{\epsilon}{4},
\end{aligned}$$

and hence

$$D(H') > \frac{\epsilon}{2} - \frac{\epsilon}{4} = \frac{\epsilon}{4}.$$

And,

$$\begin{aligned}
\frac{\epsilon}{2} &\leq \sum_{l\in L}\left(\frac{1}{N}-D(l)\right)\\
&= \sum_{l\in L'}\left(\frac{1}{N}-D(l)\right)+\sum_{l\in L\setminus L'}\underbrace{\left(\frac{1}{N}-D(l)\right)}_{\leq\frac{\epsilon}{4N}}\\
&\leq \frac{|L'|}{N}-D(L')+\frac{\epsilon}{4N}(\underbrace{|L|}_{\leq N}-|L'|)\\
&\leq \frac{|L'|}{N}+\frac{\epsilon}{4}-\left(D(L')+\frac{\epsilon|L'|}{4N}\right)\\
&\leq \frac{|L'|}{N}+\frac{\epsilon}{4},
\end{aligned}$$

and thus

$$|L'| \geq \frac{N\epsilon}{4}.$$

We can obtain an element of L' with high probability by sampling from $\mathsf{SAMP}_{\mathcal{A}^{(N)}}$ $O(1/\epsilon)$ times, and we can obtain an element of H' with high probability by sampling from SAMP_D $O(1/\epsilon)$ times. These elements will have a multiplicative difference of at least $\frac{1/N+\epsilon/(4N)}{1/N-\epsilon(4N)} \geq 1+\frac{\epsilon}{2}$, which can be detected with high probability by using the COMPARE procedure with parameters, say, $\eta = \epsilon/100$ and $K = 2$, requiring $\tilde{O}(1/\epsilon^2)$ PCOND_D queries.

Since there will be $O(1/\epsilon^2)$ pairs to test, and each use of COMPARE requires $\tilde{O}(1/\epsilon^2)$ queries, the overall sample complexity of the algorithm will be $\tilde{O}(1/\epsilon^4)$.

In addition, by replacing the COMPARE procedure with the QCOMPARE procedure, we can instantly reduce the sample complexity of this algorithm to $\tilde{O}(1/\epsilon^3)$.

A.2 A sub-linear algorithm for Mixedness Testing

Conjecture A.2.1. *Given an n-dimensional quantum state $\rho \in \mathbb{C}^n \times \mathbb{C}^n$ and a basis $\mathcal{B} = \{|b_i\rangle\}_{i\in[n]}$ where n is even, let $D^{(\rho,\mathcal{B})}_{[n]}$ be the probability distribution over $[n]$ such that $D^{(\rho,\mathcal{B})}_{[n]}(i) \coloneqq \mathrm{Tr}(\rho\,|b_i\rangle\,\langle b_i|) = \langle b_i|\,\rho\,|b_i\rangle$.*

For any $\epsilon > 0$, there exists an algorithm that solves the Mixedness problem for ρ with probability at least $2/3$ using $\tilde{O}(\sqrt{n}/\epsilon)$ PQCOND queries to $D^{(\rho,\mathcal{B})}_{[n]}$ (where each query may involve a different $\mathcal{B}$). The algorithm outputs MaximallyMixed *if ρ is the maximally-mixed state, and* NotMaximallyMixed *otherwise.*

In order to prove the above conjecture, we make use of the Chebyshev inequality (see eq. (1.5)), and analyse $\mathbb{E}\left(\delta^{(\mathcal{B})}\right)$ and $\mathrm{Var}\left(\delta^{(\mathcal{B})}\right)$ more closely than in Section 6.3.

We first prove the following useful proposition:

Proposition A.2.2.

$$\int_{T_n} v_0^{\alpha_0-1}\cdots v_{n-1}^{\alpha_{n-1}-1}dV = (n-1)!\frac{\Gamma(\alpha_0)\cdots\Gamma(\alpha_{n-1})}{\Gamma(\alpha_0+\cdots+\alpha_{n-1})}.$$

Proof. First, note that

$$\begin{aligned}\int_0^a x^{p-1}(1-x)^{q-1}dx &= a^{p+q-2}\int_0^a \left(\frac{x}{a}\right)^{p-1}\left(1-\frac{x}{a}\right)^{q-1}dx\\ &= a^{p+q-1}\int_0^1 y^{p-1}(1-y)^{q-1}dy\\ &= a^{p+q-1}B(p,q),\end{aligned} \tag{A.1}$$

where $B(p,q)$ is the Beta function.

Now,

$$\begin{aligned}&\int_{T_n} v_0^{\alpha_0-1}\cdots v_{n-1}^{\alpha_{n-1}-1}dV\\ &= (n-1)!\int_{v_0=0}^1\cdots\int_{v_{n-1}=0}^1 v_0^{\alpha_0-1}\cdots v_{n-1}^{\alpha_{n-1}-1}\delta(1-\textstyle\sum_{i\in[n]}v_i)\,dv_0\cdots dv_{n-1}\\ &= (n-1)!\int_{v_0=0}^1\int_{v_1=0}^{1-v_0}\cdots\int_{v_{n-2}=0}^{1-\sum_{i=0}^{n-3}v_i} v_0^{\alpha_0-1}\cdots v_{n-2}^{\alpha_{n-2}-1}\left(1-\sum_{i=0}^{n-2}v_i\right)^{\alpha_{n-1}-1}dv_0\cdots dv_{n-2}\\ &= (n-1)!\int_{v_0=0}^1 v_0^{\alpha_0-1}\int_{v_1=0}^{1-v_0} v_1^{\alpha_1-1}\cdots\underbrace{\left[\int_{v_{n-2}=0}^{1-\sum_{i=0}^{n-3}v_i} v_{n-2}^{\alpha_{n-2}-1}\left(1-\sum_{i=0}^{n-2}v_i\right)^{\alpha_{n-1}-1}dv_{n-2}\right]}_{(1-\sum_{i=0}^{n-3}v_i)^{\alpha_{n-2}+\alpha_{n-1}-1}B(\alpha_{n-2},\alpha_{n-1})}dv_{n-3}\cdots dv_0,\end{aligned}$$

where, in the last line, we have made use of eq. (A.1). Repeated use of this identity gives

$$\begin{aligned}&\int_{T_n} v_0^{\alpha_0-1}\cdots v_{n-1}^{\alpha_{n-1}-1}dV\\ &= (n-1)!B(\alpha_0,\alpha_1+\cdots+\alpha_{n-1})B(\alpha_1,\alpha_2+\cdots+\alpha_{n-1})\cdots B(\alpha_{n-2},\alpha_{n-1}).\end{aligned}$$

The result follows from use of the identity for the Beta function $B(p,q)=\frac{\Gamma(p)\Gamma(q)}{\Gamma(p+q)}$. □

Analysing $\mathbb{E}\left(\delta^{(\mathcal{B})}\right)$

Recall that n is even, and let $2\le k\le n$ be even.

By slightly modifying the approach used in eq. (6.6) and eq. (6.7) (by cycling only the first k indices), we arrive at

$$\mathbb{E}(|V\cdot d|) \ge \frac{1}{k}\left[\max_{\sigma\in Sym([n])}|d_{\sigma(0)}-d_{\sigma(1)}+d_{\sigma(2)}-\cdots-d_{\sigma(k-1)}|\right]\mathbb{E}(|v_0-v_1+v_2-\cdots-v_{k-1}|). \tag{A.2}$$

We now evaluate the quantity $E_{k,n} \coloneqq \mathbb{E}(|v_0 - v_1 + v_2 - \cdots - v_{k-1}|)$.

Recall that $E_n \coloneqq \mathbb{E}(|v_0 - v_1 + v_2 - \cdots - v_{n-1}|)$ (defined in eq. (6.2)), and note that $E_{k,n} \neq E_k$, as the expectation is still over n variables.

Now, from eq. (6.5) we have

$$\begin{aligned} E_{k,n} &= (n-1)! \int_{v_0=0}^{1} \cdots \int_{v_{n-1}=0}^{1} \delta\left(1 - \sum_{i=0}^{n-1} v_i\right) |v_0 - v_1 + v_2 - \cdots - v_{k-1}| \, dv_{n-1} \cdots dv_0 \\ &= (n-1)! \int_{v_{n-1}=0}^{1} \int_{v_{n-2}=0}^{1-v_{n-1}} \cdots \int_{v_1=0}^{1=\sum_{i=2}^{n-1} v_i} |v_0 - v_1 + v_2 - \cdots - v_{k-1}| \, dv_{n-1} \cdots dv_0, \end{aligned}$$

where, in the second line, v_0 is defined to be $1 - \sum_{i=1}^{n-1} v_i$, as it is no longer a variable in the integration. Here, if we let $\lambda = \sum_{i=k}^{n-1} v_i$, then $v_0 = \lambda - \sum_{i=1}^{k-1} v_i$, and

$$E_{k,n} = (n-1)! \int_{v_{n-1}=0}^{1} \cdots \int_{v_k=0}^{1-\sum_{i=k+1}^{n-1} v_i} \Lambda_{\lambda,k} \, dv_k \cdots dv_{n-1},$$

where

$$\begin{aligned} \Lambda_{\lambda,k} &\coloneqq \int_{v_{k-1}=0}^{\lambda} \int_{v_{k-2}=0}^{\lambda - v_{k-1}} \cdots \int_{v_1=0}^{\lambda - \sum_{i=2^{k-1}} v_i} |v_0 - v_1 + v_2 - \cdots - v_{k-1}| \, dv_1 \cdots dv_{k-1} \\ &= \lambda^k \int_{w_{k-1}=0}^{1} \int_{w_{k-2}=0}^{1-w_{k-1}} \cdots \int_{w_1=0}^{1-\sum_{i=2^{k-1}} w_i} |w_0 - w_1 + w_2 - \cdots - w_{k-1}| \, dw_1 \cdots dw_{k-1}, \end{aligned}$$

where we have performed the substitution $w_i = \frac{v_i}{\lambda}$. Thus

$$\begin{aligned} \Lambda_{\lambda,k} &= \lambda^k \int_{w_{k-1}=0}^{1} \cdots \int_{w_1=0}^{1} \delta\left(1 - \sum_{i=0}^{k-1} v_i\right) |w_0 - w_1 + w_2 - \cdots - w_{k-1}| \, dw_1 \cdots dw_{k-1} \\ &= \frac{\lambda^k}{(k-1)!} E_k, \end{aligned}$$

from the definition in eq. (6.2). Hence

$$E_{k,n} = \frac{(n-1)! E_k}{(k-1)!} \int_{v_{n-1}=0}^{1} \cdots \int_{v_k=0}^{1-\sum_{i=k+1}^{n-1} v_i} \lambda^k \, dv_k \cdots dv_{n-1}.$$

By writing v_n in place of λ, we can rewrite this as (where the dummy indices have been altered)

$$\begin{aligned} E_{k,n} &= \frac{(n-1)! E_k}{(k-1)!} \cdot \frac{1}{(n-k)!} \int_{T_{n-k+1}} v_0^k \, dV \\ &= \frac{(n-1)! E_k}{(k-1)!} \cdot \frac{k!}{n!}, \end{aligned}$$

by using Proposition A.2.2. Thus,

$$E_{k,n} = \frac{k}{n} E_k \geq \frac{\sqrt{k}}{2n}, \tag{A.3}$$

where we have used the bound derived in eq. (6.16).

We now wish to find a lower bound for $\max_{\sigma \in Sym([n])} |d_{\sigma(0)} - d_{\sigma(1)} + d_{\sigma(2)} - \cdots - d_{\sigma(k-1)}|$.

Wlog assume that $|d_0| \geq |d_1| \geq \cdots$, and define $S := \sum_{i=0}^{k-1} |d_i|$. The approach used in Section 6.4 then yields

$$\max_{\sigma \in Sym([n])} \left| d_{\sigma(0)} - d_{\sigma(1)} + d_{\sigma(2)} - \cdots - d_{\sigma(k-1)} \right| \geq \frac{S}{2}.$$

Combining this result with eq. (A.2), eq. (A.3) and eq. (6.4), we find that

$$\mathbb{E}\left(\delta^{(\mathcal{B})}\right) \geq \frac{S}{4\sqrt{k}} \tag{A.4}$$

for any k, and so we choose k to maximise the RHS. Note that if $k = n$ then $S = \eta$, and we regain the same bound as in eq. (6.16).

Analysing Var $\left(\delta^{(\mathcal{B})}\right)$

From the definition of the variance, we see that

$$\begin{aligned}
\mathrm{Var}\left(\delta^{(\mathcal{B})}\right) &= \mathrm{Var}\left(\sum_{i\in[n]} |V^{(i)} \cdot d|\right) \\
&= \sum_{i\in[n]} \mathrm{Var}\left(|V^{(i)} \cdot d|\right) + \sum_{i,j\in[n];i\neq j} \mathrm{Cov}\left(|V^{(i)} \cdot d|, |V^{(j)} \cdot d|\right) \\
&= n\,\mathrm{Var}\left(|V \cdot d|\right) + n(n-1)\,\mathrm{Cov}\left(|V^{(1)} \cdot d|, |V^{(2)} \cdot d|\right) \\
&\leq n\mathbb{E}\left(|V \cdot d|^2\right) + n(n-1)\,\mathrm{Cov}\left(|V^{(1)} \cdot d|, |V^{(2)} \cdot d|\right),
\end{aligned} \tag{A.5}$$

where the second line follows from eq. (1.4), the third line follows since each of the $|V^{(i)} \cdot d|$'s has the same marginal distribution, and the final line follows from eq. (1.3).

We now calculate

$$\begin{aligned}
\mathbb{E}\left(|V \cdot d|^2\right) &= \int_{T_n} |v_0 d_0 + \cdots v_{n-1} d_{n-1}|^2 \, dV \\
&= \sum_{i\in[n]} d_i^2 \int_{T_n} v_i^2 dV + \sum_{i,j\in[n];i\neq j} d_i d_j \int_{T_n} v_i v_j \, dV \\
&= \sum_{i\in[n]} d_i^2 \cdot \left(\int_{T_n} v_0^2 \, dV\right) + \sum_{i,j\in[n];i\neq j} d_i d_j \cdot \left(\int_{T_n} v_0 v_1 \, dV\right),
\end{aligned}$$

where the final line follows because the integral is symmetric in the v_i.

Using Proposition A.2.2, we see that the integral in the first term evaluates to $\frac{2}{n(n+1)}$, and the integral in the second term evaluates to $\frac{1}{n(n+1)}$. Thus

$$\begin{aligned}\mathbb{E}\left(|V\cdot d|^2\right) &= \frac{2}{n(n+1)}\sum_{i\in[n]} d_i^2 + \frac{1}{n(n+1)}\sum_{i,j\in[n];i\neq j} d_i d_j \\ &= \frac{1}{n(n+1)}\sum_{i\in[n]} d_i^2 + \frac{1}{n(n+1)}\sum_{i,j\in[n]} d_i d_j \\ &= \frac{1}{n(n+1)}\sum_{i\in[n]} d_i^2 + \frac{1}{n(n+1)}\left(\sum_{i\in[n]} d_i\right)^2.\end{aligned}$$

The sum in the second term is 0 by definition (noted in Section 6.1), and hence

$$\mathbb{E}\left(|V\cdot d|^2\right) = \frac{1}{n(n+1)}\sum_{i\in[n]} d_i^2. \tag{A.6}$$

As we chose k to maximise the bound in eq. (A.4), we can immediately deduce the inequality (recalling that $|d_0| \geq |d_1| \geq \cdots$)

$$\frac{S}{4\sqrt{k}} \geq \frac{|d_0|+|d_1|}{4\sqrt{2}} \implies |d_0|+|d_1| \leq S\sqrt{\frac{2}{k}}.$$

We thus see that $|d_0| \leq S\sqrt{2/k}$, and hence that $|d_i| \leq S\sqrt{2/k}$ for all i.

The maximum value of $\sum_{i=0}^{k-1} d_i^2$ is therefore obtained by setting $|d_0| = \cdots = |d_{\lceil\sqrt{k/2}\rceil}| = S\sqrt{k/2}$ and $|d_{\lceil\sqrt{k/2}\rceil+1}| = \cdots = |d_{k-1}| = 0$, and so

$$\sum_{i=0}^{k-1} d_i^2 \leq S^2\cdot\frac{2}{k}\cdot\left\lceil\sqrt{\frac{k}{2}}\right\rceil \leq s^2\cdot\frac{2}{k}\left(\sqrt{\frac{k}{2}}+1\right) = S^2\frac{2}{k} + S^2\sqrt{\frac{2}{k}}. \tag{A.7}$$

Since $\sum_{i=0}^{k-1}|d_i| = S$, we see that $|d_{k-1}| \leq \frac{S}{k}$, and hence $|d_k|,\ldots,|d_{n-1}| \leq \frac{S}{k}$. Also, $\sum_{i=k}^{n-1}|d_i| = \eta - S$, and thus we can similarly maximise this sum to deduce that

$$\sum_{i=k}^{n-1} d_i^2 \leq \frac{S^2}{k^2}\cdot\left\lceil\frac{\eta-S}{S/k}\right\rceil \leq \frac{S(\eta-S)}{k} + \frac{S^2}{k^2}. \tag{A.8}$$

By combining eq. (A.6), eq. (A.7) and eq. (A.8), we see that

$$\begin{aligned}
\mathbb{E}\left(|V\cdot d|^2\right) &\leq \frac{1}{n(n+1)}\left(S^2\frac{2}{k}+S^2\sqrt{\frac{2}{k}}+\frac{S(\eta-S)}{k}+\frac{S^2}{k^2}\right),\\
&= \frac{1}{n(n+1)}\left(S^2\sqrt{\frac{2}{k}}+\frac{S(\eta+S)}{k}+\frac{S^2}{k^2}\right)\\
&\leq \frac{1}{n(n+1)}\cdot\frac{5S\eta}{\sqrt{k}}\\
&= \frac{1}{n(n+1)}\cdot 20\eta\cdot\frac{S}{4\sqrt{k}}\\
&\leq \frac{20\eta}{n(n+1)}\cdot\mathbb{E}\left(\delta^{(\mathcal{B})}\right)
\end{aligned}$$

Substituting this back into eq. (A.5), we get

$$\mathrm{Var}\left(\delta^{(\mathcal{B})}\right)\leq\frac{20\eta}{n+1}\cdot\mathbb{E}\left(\delta^{(\mathcal{B})}\right)+n(n-1)\,\mathrm{Cov}\left(|V^{(1)}\cdot d|,|V^{(2)}\cdot d|\right)$$

Here we include a conjecture about $\mathrm{Cov}\left(|V^{(1)}\cdot d|,|V^{(2)}\cdot d|\right)$.

Conjecture A.2.3. *In this scenario,*

$$\left|\mathrm{Cov}\left(|V^{(1)}\cdot d|,|V^{(2)}\cdot d|\right)\right|\leq\frac{\nu}{n}\,\mathrm{Var}\left(|V\cdot d|\right)\left(\leq\frac{\nu}{n}\mathbb{E}\left(|V\cdot d|^2\right)\right)$$

for some constant ν.

Reasoning. $V^{(1)}$ and $V^{(2)}$ are derived from columns of a unitary matrix, $W^{(1)}$ and $W^{(2)}$, for which the only dependence is that $W^{(1)}\cdot W^{(2)}=0$. This property is one restriction on two n-dimensional vectors, suggesting that $\|\,\mathrm{Cov}(W^{(1)},W^{(2)})\|$ will be roughly the same as $\|\,\mathrm{Var}(W)\|/n$, where W is a generic $W^{(i)}$. One would expect that by applying the same function to $W^{(1)}$ and $W^{(2)}$ (and W), neither the covariance nor the variance could increase by more than a constant factor. This leads to the statement of the conjecture.

If we take Conjecture A.2.3 to be true, then we see that

$$\mathrm{Var}\left(\delta^{(\mathcal{B})}\right)\leq\frac{20\eta(1+\nu)}{n+1}\cdot\mathbb{E}\left(\delta^{(\mathcal{B})}\right)$$

Applying the Chebyshev inequality

The Chebyshev inequality (see eq. (1.5)) implies that

$$\mathbb{P}\left[\left|\delta^{(\mathcal{B})}-\mathbb{E}\left(\delta^{(\mathcal{B})}\right)\right|\leq\frac{1}{2}\mathbb{E}\left(\delta^{(\mathcal{B})}\right)\right]\geq 1-\frac{20\eta(1+\nu)}{(n+1)\mathbb{E}\left(\delta^{(\mathcal{B})}\right)}.$$

Now, $\mathbb{E}\left(\delta^{(\mathcal{B})}\right) \geq \frac{\eta}{4\sqrt{n}}$ (from eq. (6.8)), which implies that

$$\begin{aligned}\mathbb{P}\left[\delta^{(\mathcal{B})} \geq \frac{\eta}{8\sqrt{n}}\right] &\geq \Pr\left[\left|\delta^{(\mathcal{B})} - \mathbb{E}\left(\delta^{(\mathcal{B})}\right)\right| \leq \frac{1}{2}\mathbb{E}\left(\delta^{(\mathcal{B})}\right)\right] \\ &\geq 1 - \frac{80(1+\nu)}{\sqrt{n}}.\end{aligned}$$

As $n \to \infty$, we see that the probability tends to 1, and hence it is almost certain that we will get a 'good' basis. The analysis presented in Section 6.2 then follows to yield an algorithm that requires only $\tilde{O}(\sqrt{n}/\epsilon)$ queries. □

Appendix B

Classical Simulation of Quantum Circuits

B.1 Higher-dimensional analogues of matchgates

The Jordan-Wigner transform (see Section 1.1.3) maps fermions of one flavour to (2-dimensional) qubits. This transform has a particularly advantageous property, described below, that is the crux of the proof given in [JM08, JMS15] that circuits built from matchgates are classically-simulatable.

Is it possible to extend the Jordan-Wigner transform to map multiple flavours of fermions to d-dimensional qudits, while still retaining this property? This would lead directly to higher-dimensional analogues of matchgates.

We provide evidence that such a mapping is not possible for odd d, and that for even d, the circuits that are achieved can be trivially reduced to those built from matchgates.

Classically-simulatable quantum circuits on qubits

To ease notation, we define the $2N$ Majorana fermions to be

$$\begin{aligned} b_{2i} = c_i &= (a_i + a_i^\dagger) = Z_1 \dots Z_{i-1} X_i \\ b_{2i+1} = d_i &= i(a_i - a_i^\dagger) = Z_1 \dots Z_{i-1} Y_i, \end{aligned}$$

where we have used the Jordan-Wigner transform expressed in eq. (1.10).

A surprising property that these operators obey is that the span of their quadratic products form a Lie algebra, that is:

$$[b_i\, b_j\, , b_k b_l\,] \in \mathrm{span}(\{b_p b_q\}_{p,q}) \tag{B.1}$$

This property leads directly to the classical simulability of circuits comprising elements of the corresponding Lie group i.e. matchgates) as gates. For more details, see [JMS15, JM08] and Section 7.2.

Classically-simulatable quantum circuits on qudits?

Here, we (non-rigorously) illustrate some of the difficulties in generalising the Jordan-Wigner operators to d dimensions, while still retaining a closed algebra like that in eq. (B.1).

For this we consider odd d and begin with the assumption that the b operators take the form

$$b_{i,\alpha} = T_1 \dots T_{i-1} (A_\alpha)_i S_{i+1} \dots S_N,$$

where S, T and A_α are invertible Hermitian $d \times d$ matrices. We also impose the conditions

$$A_\alpha T = t_\alpha T A_\alpha, \quad A_\alpha S = s_\alpha S A_\alpha, \quad ST = \sigma TS, \quad A_\alpha A_b = \rho_{\alpha\beta} A_\beta A_\alpha, \tag{B.2}$$

where $t_\alpha, s_\alpha, \alpha, \rho_{\alpha\beta} \neq 0$. (Note that in the 2-dimensional case, $T = Z, S = \mathbb{1}, A_1 = X, A_2 = Y, t_\alpha = -1, s_\alpha = 1, \sigma = 1, \rho_{\alpha\beta} = -1$ for $a \neq b$.)

For $p < q$, we have that

$$\begin{aligned} b_{p,g} b_{q,h} &= T_1^2 \dots T_{p-1}^2 (A_g T)_p (ST)_{p+1} \dots (ST)_{q-1} (SA_h)_q S_{q+1}^2 \dots S_N^2 \\ b_{q,h} b_{p,g} &= T_1^2 \dots T_{p-1}^2 (TA_g)_p (TS)_{p+1} \dots (TS)_{q-1} (A_h S)_q S_{q+1}^2 \dots S_N^2, \end{aligned} \tag{B.3}$$

and we consider the following commutator for $i < j$

$$\begin{aligned} &[b_{i,a} b_{j,b}, b_{i,a} b_{j+1,c}] \\ &\propto T_1^4 \dots T_{i-1}^4 (A_\alpha T)_i^2 (ST)_{i+1}^2 \dots (ST)_{j-1}^2 (S^2 T A_\beta)_j (S^3 A_c)_{j+1} S_{j+2}^4 \dots S_N^4, \end{aligned}$$

which has been simplified using the commutation relations in eq. (B.2). In the 2-dimensional case, this is one of the few commutator expressions that is non-zero, and so we aim for a non-zero value here also.

We attempt to match this expression with the quadratic terms in eq. (B.3), and notice that the sites at the ends of the chain suggest[1] that $T^4 = T^2 \Rightarrow T^2 = \mathbb{1}$ and, similarly, that $S^4 = S^2 \Rightarrow S^2 = \mathbb{1}$.

This simplifies the quadratic terms and commutator significantly, and we see that

$$\begin{aligned} b_{p,g} b_{q,h} &= (A_g T)_p (ST)_{p+1} \dots (ST)_{q-1} (SA_h)_q \\ b_{q,h} b_{p,g} &= (TA_g)_p (TS)_{p+1} \dots (TS)_{q-1} (A_h S)_q \\ [b_{i,a} b_{j,b}, b_{i,a} b_{j+1,c}] &\propto (A_\alpha)_i^2 (TA_b)_j (SA_c)_{j+1}. \end{aligned}$$

Comparing the commutator to the quadratic terms, the possibilities are[2]

[1] Actually, this suggests that $T^4 \propto T^2$ and $S^4 \propto S^2$, but for our purposes it is more illustrative to treat these as equalities.

[2] Once again we have replaced the proportionalities with equalities for illustration.

1. $p = i, q = j$, and hence $SA_c = \mathbb{1} \Rightarrow A_c = S \ \forall c$, which gives only a trivial result (as we would expect our A_c's to be different)
2. $p = i, q = j + 1$, and hence that $TA_b = \mathbb{1} \Rightarrow A_b = T \ \forall b$, which also gives a trivial result (as we would expect our A_b's to be different)
3. $p = j, q = j + 1$, and hence that $(A_\alpha)^2 = \mathbb{1} \ \forall a$.

Moving forward with Point 3, we see that $T^2 = S^2 = A_\alpha^2 = \mathbb{1}$. Now, from eq. (B.2), we see that $STS = \sigma T$. Taking the trace of both sides, we see that $\operatorname{Tr} T = \sigma \operatorname{Tr} T$, and so either $\operatorname{Tr} T = 0$ (which is a contradiction, since T has an odd number of ± 1 eigenvalues), or $\sigma = 1$. Hence we conclude that T and S commute. Using the other relations in eq. (B.2), we conclude that T, S and all of the A_α's mutually commute, and WLOG can be considered to be diagonal. Circuits built from diagonal operators are trivially classically simulatable, and hence this approach does not yield a useful result.

Instead of quadratic terms in the commutator, we may consider cubic terms. However, we find that $[b_{i,a} b_{j,b} b_{k,c}\, , b_{i,a} b_{j,b} b_{k+1,d}]$ exhibits similar problems to $[b_{i,a} b_{j,b}\, , b_{i,a} b_{j+1,c}]$: we find that $T^6 = T^3 \Rightarrow T^3 = \mathbb{1}$. Since T is Hermitian, it has real eigenvalues, and hence these eigenvalues can only be ± 1, which leads us to the conclusion that $T^2 = \mathbb{1}$, as in the case discussed above. This also holds true for S and the A_α's, and we arrive at another trivial classical simulation result.

It is clear from this that increasing the degree of terms in the commutator will not produce a more useful result, and so it appears that for odd d this approach will not yield a generalisation of the Jordan-Wigner operators that leads to non-trivial classical simulation results.

For even d, it is possible to have $\operatorname{Tr} T = 0$ even when $T^2 = \mathbb{1}$. Continuing the analysis quickly leads to the conclusion that $\sigma, t_\alpha, s_\alpha = \pm 1$ and that $A_\alpha^2 = \mathbb{1}$. Hence we conclude that all of the operators either commute or anti-commute. By considering the commutator $[b_{i,a} b_{j,b}\, , b_{i,a} b_{j,c}]$, it becomes clear that there must be a set of A_α's that mutually anti-commute in order to ensure that not all of the commutators are equal to zero. This leads directly to the b_i's forming a Clifford algebra, as in the 2-dimensional case. While this produces a generalisation of the Jordan-Wigner operators in even dimensions, the resultant quantum circuits can easily be decomposed into matchgates and so do not yield any new classes of classically-simulatable quantum circuits.

Bibliography

[ABJ+03] Joseph B Altepeter, David Branning, Evan Jeffrey, T C Wei, Paul G Kwiat, Robert T Thew, Jeremy L O'Brien, Michael A Nielsen, and Andrew G White. Ancilla-assisted quantum process tomography. *Physical Review Letters*, 90(19):193601, 2003.

[ABRdW15] Andris Ambainis, Aleksandrs Belovs, Oded Regev, and Ronald de Wolf. Efficient Quantum Algorithms for (Gapped) Group Testing and Junta Testing. *arXiv:1507.03126 [quant-ph]*, July 2015. arXiv: 1507.03126.

[ACK14] Jayadev Acharya, Clément L Canonne, and Gautam Kamath. A chasm between identity and equivalence testing with conditional queries. *arXiv preprint arXiv:1411.7346*, 2014.

[ÁSS12] Gonzalo A Álvarez, Alexandre M Souza, and Dieter Suter. Iterative rotation scheme for robust dynamical decoupling. *Physical Review A*, 85(5):052324, 2012.

[Atk08] Kendall E Atkinson. *An introduction to numerical analysis*. John Wiley & Sons, 2008.

[Bal05] R. C. Ball. Fermions without Fermion Fields. *Physical Review Letters*, 95(17):176407, October 2005.

[BBM12] Eric Blais, Joshua Brody, and Kevin Matulef. Property testing lower bounds via communication complexity. *Computational Complexity*, 21(2):311–358, 2012.

[BC14] Daniel J Brod and Andrew M Childs. The computational power of matchgates and the XY interaction on arbitrary graphs. *Quantum Information & Computation*, 14(11-12):901–916, 2014.

[BFF+01] Tugkan Batu, Eldar Fischer, Lance Fortnow, Ravi Kumar, Ronitt Rubinfeld, and Patrick White. Testing random variables for independence and identity. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 442–451. IEEE, 2001.

[BFR+10] Tugkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D. Smith, and Patrick White. Testing Closeness of Discrete Distributions. *arXiv:1009.5397 [cs, math, stat]*, September 2010. arXiv: 1009.5397.

[BG07] Joakim Bergli and Leonid Glazman. Spin echo without an external permanent magnetic field. *Physical Review B*, 76(6):064301, 2007.

[BHH11] S. Bravyi, A. W. Harrow, and A. Hassidim. Quantum Algorithms for Testing Properties of Distributions. *IEEE Transactions on Information Theory*, 57(6):3971–3981, June 2011.

[BK02] Sergey B Bravyi and Alexei Yu Kitaev. Fermionic quantum computation. *Annals of Physics*, 298(1):210–226, 2002.

[BO01] C. D. Batista and G. Ortiz. Generalized Jordan-Wigner Transformations. *Physical Review Letters*, 86(6):1082–1085, February 2001.

[Bon15] Annalisa De Bonis. Constraining the number of positive responses in adaptive, non-adaptive, and two-stage group testing. *Journal of Combinatorial Optimization*, pages 1–34, September 2015.

[CDVV14] Siu-On Chan, Ilias Diakonikolas, Gregory Valiant, and Paul Valiant. Optimal Algorithms for Testing Closeness of Discrete Distributions. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 1193–1203, Philadelphia, PA, USA, 2014. Society for Industrial and Applied Mathematics.

[CEMM98] Richard Cleve, Artur Ekert, Chiara Macchiavello, and Michele Mosca. Quantum algorithms revisited. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 454, pages 339–354. The Royal Society, 1998.

[CFGM16] S. Chakraborty, E. Fischer, Y. Goldhirsh, and A. Matsliah. On the Power of Conditional Samples in Distribution Testing. *SIAM Journal on Computing*, pages 1261–1296, January 2016.

[CFMdW09] Sourav Chakraborty, Eldar Fischer, Arie Matsliah, and Ronald de Wolf. Quantum queries for testing distributions, 2009.

[CFMdW10] Sourav Chakraborty, Eldar Fischer, Arie Matsliah, and Ronald de Wolf. New Results on Quantum Property Testing. *arXiv:1005.0523 [quant-ph]*, May 2010. arXiv: 1005.0523.

[CG95] Siddhartha Chib and Edward Greenberg. Understanding the Metropolis-Hastings algorithm. *The American Statistician*, 49(4):327–335, 1995.

[CL00] P. M. Chaikin and T. C. Lubensky. *Principles of Condensed Matter Physics*. Cambridge University Press, September 2000.

[CR14] Clément Canonne and Ronitt Rubinfeld. Testing probability distributions underlying aggregated data. In *International Colloquium on Automata, Languages, and Programming*, pages 283–295. Springer, 2014.

[CRS15] Clément L Canonne, Dana Ron, and Rocco A Servedio. Testing probability distributions using conditional samples. *SIAM Journal on Computing*, 44(3):540–616, 2015.

[D+00] David P DiVincenzo et al. The physical implementation of quantum computation. *arXiv preprint quant-ph/0002077*, 2000.

[DBGV05] A. De Bonis, L. Gasieniec, and U. Vaccaro. Optimal Two-Stage Algorithms for Group Testing Problems. *SIAM Journal on Computing*, 34(5):1253–1270, January 2005.

[DDJB14] Nilanjana Datta, Tony Dorlas, Richard Jozsa, and Fabio Benatti. Properties of subentropy. *Journal of Mathematical Physics*, 55(6):062203, 2014.

[DJ92] David Deutsch and Richard Jozsa. Rapid Solution of Problems by Quantum Computation. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 439(1907):553–558, December 1992.

[DKN15] Ilias Diakonikolas, Daniel M Kane, and Vladimir Nikishkin. Testing identity of structured distributions. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1841–1854. Society for Industrial and Applied Mathematics, 2015.

[DN05] Christopher M Dawson and Michael A Nielsen. The Solovay-Kitaev algorithm. *arXiv preprint quant-ph/0505030*, 2005.

[FJO+15] Moein Falahatgar, Ashkan Jafarpour, Alon Orlitsky, Venkatadheeraj Pichapathi, and Ananda Theertha Suresh. Faster algorithms for testing under conditional sampling. *CoRR, vol. abs/1504.04103*, 2015.

[FK90] Holger Frahm and VE Korepin. Critical exponents for the one-dimensional Hubbard model. *Physical Review B*, 42(16):10553, 1990.

[FM98] Ray Freeman and Michael J Minch. *Spin choreography: basic steps in high resolution NMR*. Oxford University Press New York, 1998.

[GG81] Israel Gohberg and Seymour Goldberg. *Basic operator theory*. Birkhäuser, 1981.

[GGR98] Oded Goldreich, Shari Goldwasser, and Dana Ron. Property Testing and Its Connection to Learning and Approximation. *J. ACM*, 45(4):653–750, July 1998.

[GMV06] Sudipto Guha, Andrew McGregor, and Suresh Venkatasubramanian. Streaming and sublinear approximation of entropy and information distances. In *Proceedings of the seventeenth annual ACM-SIAM symposium*

on Discrete algorithm, pages 733–742. Society for Industrial and Applied Mathematics, 2006.

[Gol10] Oded Goldreich. *Property Testing: Current Research and Surveys*. Springer, October 2010. Google-Books-ID: HIdqCQAAQBAJ.

[Got97] Daniel Gottesman. Stabilizer codes and quantum error correction. *arXiv preprint quant-ph/9705052*, 1997.

[Got98] Daniel Gottesman. The Heisenberg representation of quantum computers. *arXiv preprint quant-ph/9807006*, 1998.

[GW14] Geoffrey Grimmett and Dominic Welsh. *Probability: an introduction*. Oxford University Press, 2014.

[Haa33] Alfred Haar. Der Massbegriff in der Theorie der kontinuierlichen Gruppen. *Annals of mathematics*, pages 147–169, 1933.

[Hah50] Erwin L Hahn. Spin echoes. *Physical Review*, 80(4):580, 1950.

[Has70] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, January 1970.

[HKM15] J. Hutchinson, J. P. Keating, and F. Mezzadri. On relations between one-dimensional quantum and two-dimensional classical spin systems. *arXiv:1503.07712 [cond-mat, physics:math-ph]*, March 2015. arXiv: 1503.07712.

[ID91] Claude Itzykson and Jean-Michel Drouffe. *Statistical Field Theory: Volume 2, Strong Coupling, Monte Carlo Methods, Conformal Field Theory and Random Systems*. Cambridge University Press, March 1991.

[Jar92] M Jarrell. Hubbard model in infinite dimensions: A quantum Monte Carlo study. *Physical Review Letters*, 69(1):168, 1992.

[JM08] Richard Jozsa and Akimasa Miyake. Matchgates and classical simulation of quantum circuits. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 464(2100):3089–3106, December 2008.

[JMS15] Richard Jozsa, Akimasa Miyake, and Sergii Strelchuk. Jordan-Wigner formalism for arbitrary 2-input 2-output matchgates and their classical simulation. *Quantum Information & Computation*, 15(7&8):541–556, 2015.

[Joz08] Richard Jozsa. Embedding classical into quantum computation. *arXiv:0812.4511 [quant-ph]*, December 2008. arXiv: 0812.4511.

[JW28] P. Jordan and E. Wigner. Über das Paulische äquivalenzverbot. *Zeitschrift fur Physik*, 47:631–651, September 1928.

[JWB03] Dominik Janzing, Pawel Wocjan, and Thomas Beth. Identity check is QMA-complete. *arXiv preprint quant-ph/0305050*, 2003.

[KL98] E. Knill and R. Laflamme. Power of one bit of quantum information. *Physical Review Letters*, 81:5672, December 1998.

[KL05] K Khodjasteh and DA Lidar. Fault-tolerant quantum dynamical decoupling. *Physical Review Letters*, 95(18):180501, 2005.

[KL11] Wan-Jung Kuo and Daniel A Lidar. Quadratic dynamical decoupling: Universality proof and error analysis. *Physical Review A*, 84(4):042329, 2011.

[KSV02] Alexei Yu Kitaev, Alexander Shen, and Mikhail N Vyalyi. *Classical and quantum computation*, volume 47 of *Graduate Studies in Mathematics*. American Mathematical Society Providence, 2002.

[Lie04] Elliott H Lieb. Two theorems on the Hubbard model. In *Condensed Matter Physics and Exactly Soluble Models*, pages 55–58. Springer, 2004.

[LM13] Elliott H Lieb and Daniel C Mattis. *Mathematical physics in one dimension: exactly soluble models of interacting particles*. Academic Press, 2013.

[MdW13] Ashley Montanaro and Ronald de Wolf. A Survey of Quantum Property Testing. *arXiv:1310.2035 [quant-ph]*, October 2013. arXiv: 1310.2035.

[MM06] Ettore Majorana and Luciano Maiani. A symmetric theory of electrons and positrons. In *Ettore Majorana Scientific Papers*, pages 201–233. Springer, 2006.

[Mon15] Ashley Montanaro. Quantum speedup of monte carlo methods. In *Proc. R. Soc. A*, volume 471, page 20150301. The Royal Society, 2015.

[MRR+53] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, June 1953.

[NC10] Michael A Nielsen and Isaac L Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[OW15] Ryan O'Donnell and John Wright. Quantum Spectrum Testing. *arXiv:1501.05028 [quant-ph]*, January 2015. arXiv: 1501.05028.

[Pfe03] Walter Pfeifer. *The Lie algebras su(N)*. Springer, 2003.

[Por09] Ely Porat. Problem 33: Group Testing—Open Problems in Sublinear Algorithms. *Sublinear.info*, 2009.

[Rob55] Herbert Robbins. A remark on Stirling's formula. *The American Mathematical Monthly*, 62(1):26–29, 1955.

[Sch02] Michelle Schatzman. *Numerical Analysis: a mathematical introduction*. Oxford University Press, 2002.

[Suz76] Masuo Suzuki. Relationship between d-Dimensional Quantal Spin Systems and (d+1)-Dimensional Ising Systems Equivalence, Critical Exponents and Systematic Approximants of the Partition Function and Spin Correlations. *Progress of Theoretical Physics*, 56(5):1454–1469, January 1976.

[Suz93] Masuo Suzuki. *Quantum Monte Carlo Methods in Condensed Matter Physics*. World Scientific, 1993.

[Sỳk74] Stanislav Sỳkora. Quantum theory and the Bayesian inference problems. *Journal of Statistical Physics*, 11(1):17–27, 1974.

[Sza97] Stanislaw J Szarek. Metric entropy of homogeneous spaces. *arXiv preprint math/9701213*, 1997.

[Tas98a] Hal Tasaki. From Nagaoka's Ferromagnetism to Flat-Band Ferromagnetism and Beyond An Introduction to Ferromagnetism in the Hubbard Model. *Progress of Theoretical Physics*, 99(4):489–548, 1998.

[Tas98b] Hal Tasaki. The Hubbard model: an introduction and selected rigorous results. *Journal of Physics: Condensed Matter*, 10(20):4353, 1998.

[Tch67] P Tchébychef. Des valeurs moyennes (traduction du russe, n. de khanikof. *Journal de mathématiques pures et appliquées*, 12:177–184, 1867.

[TD02] Barbara M. Terhal and David P. DiVincenzo. Classical simulation of noninteracting-fermion quantum circuits. *Physical Review A*, 65(3), March 2002. arXiv: quant-ph/0108010.

[Tsv07] Alexei M Tsvelik. Quantum Field Theory in Condensed Matter Physics. *Cambridge University Press*, 2007.

[Val02] Leslie G Valiant. Quantum circuits that can be simulated classically in polynomial time. *SIAM Journal on Computing*, 31(4):1229–1254, 2002.

[Vaz98] Umesh Vazirani. On the power of quantum computation. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, pages 1759–1767, 1998.

[VC05] F. Verstraete and J. I. Cirac. Mapping local Hamiltonians of fermions to local Hamiltonians of spins. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09012, September 2005.

[VdNDRB09] M. Van den Nest, W. Dür, R. Raussendorf, and H. J. Briegel. Quantum algorithms for spin models and simulable gate sets for quantum computation. *Physical Review A*, 80(5):052334, November 2009.

[VFGE12] Victor Veitch, Christopher Ferrie, David Gross, and Joseph Emerson. Negative quasi-probability as a resource for quantum computation. *New Journal of Physics*, 14(11):113011, 2012.

[VKL99] Lorenza Viola, Emanuel Knill, and Seth Lloyd. Dynamical decoupling of open quantum systems. *Physical Review Letters*, 82(12):2417, 1999.

[VV11] G. Valiant and P. Valiant. The Power of Linear Estimators. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 403–412, October 2011.

[Wey27] Hermann Weyl. Quantenmechanik und Gruppentheorie. *Zeitschrift für Physik*, 46(1-2):1–46, 1927.

[WS07] WM Witzel and S Das Sarma. Concatenated dynamical decoupling in a solid-state spin bath. *Physical Review B*, 76(24):241303, 2007.

[YWL11] Wen Yang, Zhen-Yu Wang, and Ren-Bao Liu. Preserving qubit coherence by dynamical decoupling. *Frontiers of Physics in China*, 6(1):2–14, 2011.

[ZWL14] Nan Zhao, Jörg Wrachtrup, and Ren-Bao Liu. Dynamical decoupling design for identifying weakly coupled nuclear spins in a bath. *Physical Review A*, 90(3):032319, 2014.