

Quantum Computing, Complexity, and Noise

With Prerequisite Mathematics and Quantum Mechanics

Andrew Jackson





PREFACE

This document was designed - for a summer school in Summer 2024 - to cover - in depth - all mathematics required for basic quantum mechanics, before presenting quantum mechanics along the same path as British undergraduate degrees. There then follows a brief introduction to quantum computing and an exposition of noise in quantum systems.

It is traditional for an author to begin any textbook with the statement that all mistakes contained herein are solely my fault and are the responsibility of nobody else. But if we're allowing authors to designate who is to blame for mistakes in their work, I would like to lay all mistakes - and the associated blame/reprisals - at the feet of my brother, Ben. If you find any errors within this textbook, please let me know by emailing me at: Andrew.Jackson.1@warwick.ac.uk, and I will pass your scorn on to him.

Thanks to the following for pointing out errors in this textbook, which I have now corrected:

CHAPTER 1

BACKGROUND MATHEMATICS

1	Derivatives: Partial and Otherwise	6
2	Differential and Partial Differential Equations	9
3	Matrices	21
4	Traces and Norms	31
5	Taylor Series	32
6	Functions and Taylor Series of Matrices	35
7	Special Functions	36

1 Derivatives: Partial and Otherwise

1.1 A Necessary Formality: Limits

Before I can define and examine any element of calculus. I must first define limits. These are defined formally as:

Definition 1.1: Limits

For an function of x , $f : \mathbb{R} \rightarrow \mathbb{R}$, the limit of f as x approaches $c \in \mathbb{R}$ is $L \in \mathbb{R}$, which is denoted as:

$$\lim_{x \rightarrow c} (f(x)) = L, \quad (1.1)$$

if and only if:

$$\forall \epsilon \in \mathbb{R}, \exists \delta \in \mathbb{R} \text{ such that } \forall x \in (c - \delta, c + \delta), |f(x) - L| \leq \epsilon. \quad (1.2)$$

1.2 Initial Definition of Derivatives

We start with a formal definition of derivatives.

Definition 1.2: Derivative

Define the derivative of a function, f , at x is defined by:

$$\lim_{h \rightarrow 0} \left(\frac{f(x+h) - f(x)}{h} \right). \quad (1.3)$$

The derivative is not guaranteed to exist for all functions.

1.3 Some Example Derivatives

We can then consider some examples of using the definition of derivatives.

The Exponential Function

If $f(x) = e^x$,

$$(e^x)' = \lim_{h \rightarrow 0} \left(\frac{f(x+h) - f(x)}{h} \right) = \lim_{h \rightarrow 0} \left(\frac{e^{x+h} - e^x}{h} \right) = e^x \lim_{h \rightarrow 0} \left(\frac{e^h - 1}{h} \right) = e^x \lim_{h \rightarrow 0} \left(\frac{(e^h)'}{1} \right). \quad (1.4)$$

We can see this is satisfied if $(e^x)' = e^x$.

The Logarithm

If $f(x) = \ln(x)$, then:

$$(\ln(x))' = \lim_{h \rightarrow 0} \left(\frac{\ln(x+h) - \ln(x)}{h} \right) = \lim_{h \rightarrow 0} \left(\frac{\ln\left(\frac{x+h}{x}\right)}{h} \right) = \lim_{h \rightarrow 0} \left(\frac{1}{h} \ln\left(1 + \frac{h}{x}\right) \right) \quad (1.5)$$

$$= \lim_{h \rightarrow 0} \left(\ln\left(\left[1 + \frac{h}{x}\right]^{1/h}\right) \right). \quad (1.6)$$

Let $H = h/x$. Then,

$$(\ln(x))' = \lim_{H \rightarrow 0} \left(\ln\left(\left[1 + H\right]^{(1/H) \cdot (1/x)}\right) \right) = \frac{1}{x} \lim_{H \rightarrow 0} \left(\ln\left(\left[1 + H\right]^{1/H}\right) \right) = \frac{1}{x}. \quad (1.7)$$

1.4 Basic Theorems of Differential Calculus

Theorem 1.3: Product Rule

If u and v are once-differentiable functions of x , then

$$(u(x)v(x))' = u'(x)v(x) + u(x)v'(x) \quad (1.8)$$

Proof for Theorem.

Using the definition of differentiation:

$$(u(x)v(x))' = \lim_{h \rightarrow 0} \left(\frac{u(x+h)v(x+h) - u(x)v(x)}{h} \right) \quad (1.9)$$

$$= \lim_{h \rightarrow 0} \left(\frac{u(x+h)v(x+h) - u(x)v(x) + u(x+h)v(x) - u(x+h)v(x)}{h} \right) \quad (1.10)$$

$$= \lim_{h \rightarrow 0} \left(\frac{u(x+h)[v(x+h) - v(x)]}{h} \right) + \lim_{h \rightarrow 0} \left(\frac{[u(x+h) - u(x)]v(x)}{h} \right) \quad (1.11)$$

$$= \lim_{h \rightarrow 0} \left(u(x+h) \right) \lim_{h \rightarrow 0} \left(\frac{[v(x+h) - v(x)]}{h} \right) + \lim_{h \rightarrow 0} \left(\frac{[u(x+h) - u(x)]}{h} \right) v(x) \quad (1.12)$$

$$= u'(x)v(x) + u(x)v'(x) \quad (1.13)$$

Theorem 1.4: Generalized Product Rule

Presupposing the product rule for any finite number of functions of x , f_1, f_2, \dots, f_k ,

$$\left(f_1(x)f_2(x)\dots f_k(x)\right)' = \sum_{j=1}^k \left(f_j'(x) \prod_{\substack{q=1 \\ q \neq j}}^k [f_q(x)]\right). \quad (1.14)$$

Proof for Theorem.

I proceed by induction: the base case is given by the product rule. Then let $\Gamma(n)$ denote the statement that Eqn. 1.14 holds true when $k = n$. Assume $\psi(n)$ is true and consider $\left(f_1(x)f_2(x)\dots f_n(x)f_{n+1}(x)\right)'$. By the product rule:

$$\left(f_1(x)f_2(x)\dots f_n(x)f_{n+1}(x)\right)' = \left(f_1(x)f_2(x)\dots f_n(x)\right)' f_{n+1}(x) + f_1(x)f_2(x)\dots f_n(x)f_{n+1}'(x). \quad (1.15)$$

Then as $\Gamma(n)$ was assumed true:

$$\left(f_1(x)f_2(x)\dots f_n(x)f_{n+1}(x)\right)' = \left([f_1(x)f_2(x)\dots f_n(x)]f_{n+1}(x)\right)' \quad (1.16)$$

$$= [f_1(x)f_2(x)\dots f_n(x)]' f_{n+1}(x) + [f_1(x)f_2(x)\dots f_n(x)](f_{n+1}(x))' \quad (1.17)$$

$$= \sum_{j=1}^n \left(f_j'(x) \prod_{\substack{q=1 \\ q \neq j}}^n [f_q(x)]\right) f_{n+1}(x) + f_1(x)f_2(x)\dots f_n(x)f_{n+1}'(x) \quad (1.18)$$

$$= \sum_{j=1}^{n+1} \left(f_j'(x) \prod_{\substack{q=1 \\ q \neq j}}^{n+1} [f_q(x)]\right). \quad (1.19)$$

Therefore, $\Gamma(n+1)$ is true if $\Gamma(n)$ is, i.e. $\Gamma(n) \Rightarrow \Gamma(n+1)$, and the product rule shows that $\Gamma(2)$ is true. This implies $\Gamma(n)$ holds for all values of $n \in \mathbb{Z}$ greater than one. ■

This then gives a much nicer way of deriving the derivative of x^n than the usual way.

Theorem 1.5: Derivative of x^n

$$(x^n)' = nx^{n-1}$$

Proof for Theorem.

By re-expressing x^n , and using the generalized product rule, we can derive:

$$(x^n)' = \left(\prod_{j=1}^n (x) \right)' = \sum_{k=1}^n \left[\prod_{j=1}^{k-1} (x) \cdot 1 \cdot \prod_{j=k+1}^n (x) \right] = \sum_{k=1}^n [x^{n-1}] = nx^{n-1}. \quad (1.20)$$

Theorem 1.6: Quotient Rule

$$\left(\frac{u(x)}{v(x)} \right)' = \frac{u'(x)v(x) - u(x)v'(x)}{[v(x)]^2} \quad (1.21)$$

Proof for Theorem.

$$\left(\ln \left(\frac{u(x)}{v(x)} \right) \right)' = \left(\ln(u(x)) \right)' - \left(\ln(v(x)) \right)' = \frac{u'(x)}{u(x)} - \frac{v'(x)}{v(x)} \quad (1.22)$$

But similarly,

$$\left(\ln \left(\frac{u(x)}{v(x)} \right) \right)' = \left(\frac{u(x)}{v(x)} \right)' \frac{v(x)}{u(x)}. \quad (1.23)$$

Equating these last two derivations:

$$\left(\frac{u(x)}{v(x)} \right)' \frac{v(x)}{u(x)} = \frac{u'(x)}{u(x)} - \frac{v'(x)}{v(x)} \quad (1.24)$$

$$\Rightarrow \left(\frac{u(x)}{v(x)} \right)' = \frac{u(x)}{v(x)} \left(\frac{u'(x)}{u(x)} - \frac{v'(x)}{v(x)} \right) = \frac{u(x)}{v(x)} \frac{u'(x)v(x) - u(x)v'(x)}{u(x)v(x)} \quad (1.25)$$

$$= \frac{u'(x)v(x) - u(x)v'(x)}{[v(x)]^2} \quad (1.26)$$

2 Differential and Partial Differential Equations

Within all of physics, the ability to solve differential equations is key. Most - in fact almost all - of the main equations describing physical systems are differential equations. Examples include:

- Newton's second law of motion: a central tool of classical mechanics
- Schrodinger Equation: the central equation of quantum mechanics
- Einstein Equation: a central equation of general relativity

- Navier-Stokes Equation: the equivalent of Newton's second law of motion in fluid dynamics
- Maxwell's Equations: a complete description of electromagnetism

They are also of great use further afield, such as the BlackScholes equation, in finance; Lanchester's laws, in military strategy; and the SIR model, in epidemiology.

2.1 What is a Differential Equation?

A differential equation defines an equality between a function and its various derivatives. For example

$$\frac{df(t)}{dt} = f(t) \quad (1.27)$$

is a differential equation.

By "solving" a differential equation, we mean determining the function that has the prescribed relationship with its derivatives.

2.2 Key Pre-requisites

A few things are important to remember for later and form the foundation:

These are what the rest is built on so check them, understand them, and make sure you agree.

Point 1

In the standard algebra you learn from secondary school, if the same thing is done to both sides of an equality, they remain equal i.e. if f and g are functions of t :

$$f(t) = g(t) \Rightarrow f(t) + 7 = g(t) + 7 \quad (1.28)$$

Similarly, the same applies for integration:

$$f(t) = g(t) \Rightarrow \int f(t)dt = \int g(t)dt \quad (1.29)$$

Note this is only true in cases where the functions are equivalent (i.e. the same for all values) not if they only equate at a certain value you are seeking.

Point 2: The Fundamental Theorem of Calculus

thmp

$$\int \left(\frac{df}{dt} \right) dt = f(t) + c$$

Where c can be any constant.

Point 3

$$\int \left(f(y) \frac{dy}{dt} \right) dt = \int \left(f(y) \right) dy \quad (1.30)$$

Point 4

If you can somehow eliminate all derivatives from the equation, whilst maintaining the equality, you can rearrange the equation to find an expression for the function.

Many ways of solving differential equations revolve around this.

2.3 First Order Ordinary Differential Equations

The most simple form of differential equations are those that only feature first order derivatives. In Sec. 2.3, I will proceed through a series of methods of solving such equations.

Separation of Variables

Definition 2.1: Separable Differential Equations

A first order differential equation is separable if and only if it can be re-written as:

$$\frac{dy}{dx} = f(x)g(y), \quad (1.31)$$

up to a different choice of variables.

Theorem 2.2: Solution of Separable Differential Equations

Any separable differential equation, as in Eqn. 1.31, has the solution:

$$\int \left(\frac{1}{g(y)} \right) dy = \int \left(f(x) \right) dx, \quad (1.32)$$

this can likely be simplified further.

Proof for Theorem.

Starting from Eqn. 1.31:

$$\frac{dy}{dx} = f(x)g(y) \iff \frac{1}{g(y)} \frac{dy}{dx} = f(x) \iff \int \left(\frac{1}{g(y)} \frac{dy}{dx} \right) dx = \int \left(f(x) \right) dx \quad (1.33)$$

$$\iff \int \left(\frac{1}{g(y)} \right) dy = \int \left(f(x) \right) dx \quad (1.34)$$

This method of solving differential equations may best be explained by an example.

Example: Problem

Solve,

$$\frac{dy}{dx} = \frac{y}{x} \quad (1.35)$$

Example: Solution

To utilize point 3 we need to get all of the y s and it's derivatives on one side, and all the x s on the other:

$$\frac{dy}{dx} = \frac{y}{x} \Rightarrow \frac{1}{y} \frac{dy}{dx} = \frac{1}{x} \quad (1.36)$$

Then we integrate with respect to x , using point 3:

$$\int \left(\frac{1}{y} \frac{dy}{dx} \right) dx = \int \left(\frac{1}{x} \right) dx \Rightarrow \int \left(\frac{1}{y} \right) dy = \ln |x| + c \Rightarrow \ln |y| = \ln |x| + c \Rightarrow y = \mathcal{C}x, \quad (1.37)$$

where $\mathcal{C} = e^c$. Note the $+c$ is only required on one side as it can absorb the one generated by the other integral. The value of \mathcal{C} can be determined by initial conditions that may be provided in the question (none are provided in this example), but these will be discussed later.

Homogeneous and Inhomogeneous Differential Equations

Definition 2.3: Homogeneous and Inhomogeneous Differential Equations

A first order homogeneous linear differential equation is a first order differential equation which may be expressed as:

$$\frac{dy}{dx} + f(x)y = 0, \quad (1.38)$$

where f is some function purely of x . A first order inhomogeneous linear differential equation is a first order differential equation which may be expressed as:

$$\frac{dy}{dx} + f(x)y = g(y), \quad (1.39)$$

where f is some function purely of x and g is some function purely of y . If given an inhomogeneous differential equation as in Eqn. 1.39, call the instance of Eqn. 1.38 - with the same $f(x)$ - the corresponding homogeneous differential equation.

I am very lucky that we can find a general solution for homogeneous linear differential equations, which we now do.

Theorem 2.4: General Solution of Homogeneous Linear Differential Equations

The solution to any differential equation of the form:

$$\frac{dy}{dx} + f(x)y = 0, \quad (1.40)$$

is:

$$y = e^{-\int f(x) dx} \quad (1.41)$$

Proof for Theorem.

To confirm the claimed solution, I simply substitute it into the left hand side of Eqn. 1.40 and using the chain rule:

$$\frac{d}{dx} \left(e^{-\int f(x) dx} \right) + f(x) e^{-\int f(x) dx} = \frac{d}{dx} \left(- \int f(x) dx \right) e^{-\int f(x) dx} + f(x) e^{-\int f(x) dx}. \quad (1.42)$$

By the fundamental theorem of calculus, this equates to:

$$\left[-f(x) + f(x) \right] e^{-\int f(x) dx} = 0. \quad (1.43)$$

This is exactly as required to prove that the claimed solution is a correct solution. ■

Note that, due to the linearity of the differential equation, multiplying the solution proven above by any complex number also provides a solution.

Before continuing on to look at inhomogeneous linear first order differential equations, I show that the solutions for homogeneous linear differential remain useful when examining first order inhomogeneous linear differential equations.

Theorem 2.5: General Solution of First Order Inhomogeneous Linear Differential Equations

For any first order inhomogeneous linear differential equation, with solution $y_I(x)$; if $y_H(x)$ is a solution of the corresponding homogeneous differential equation, then, for any $\lambda \in \mathbb{C}$, $y_I(x) + \lambda y_H(x)$ is also a solution for the aforementioned inhomogeneous differential equation.

Proof for Theorem.

By assumption,

$$\frac{dy_H}{dx} + f(x)y_H = 0 \text{ and } \frac{dy_I}{dx} + f(x)y_I = g(y_I). \quad (1.44)$$

Then substituting $y_I(x) + \lambda y_H(x)$ into the left hand side of Eqn. 1.39 gives:

$$\frac{d}{dx} \left(y_I(x) + \lambda y_H(x) \right) + f(x) \left[y_I(x) + \lambda y_H(x) \right] \quad (1.45)$$

$$= \frac{d}{dx} \left(y_I(x) \right) + \lambda \frac{d}{dx} \left(y_H(x) \right) + f(x) \left[y_I(x) \right] + \lambda f(x) \left[y_H(x) \right] \quad (1.46)$$

$$= \left\{ \frac{dy_I}{dx} + f(x)y_I(x) \right\} + \lambda \left\{ \frac{dy_H}{dx} + f(x)y_H(x) \right\} \quad (1.47)$$

$$= \left\{ g(y_I) \right\} + \lambda \left\{ 0 \right\} = g(y_I), \quad (1.48)$$

which is exactly as required to be a solution of the inhomogeneous differential equation we were examining. ■

Integrating Factor

Definition 2.6: Integrating Factors

If given a differential equation of the form:

$$\frac{dy}{dx} + f(x)y = g(x), \quad (1.49)$$

then the corresponding integrating factor, $I(x)$, is a function of x , defined as:

$$I(x) = e^{\int f(x) dx} \quad (1.50)$$

Theorem 2.7

The solution to any differential equation of the form:

$$\frac{dy}{dx} + f(x)y = g(x), \quad (1.51)$$

is:

$$y = \frac{1}{I(x)} \int \left(I(x)g(x) \right) dx. \quad (1.52)$$

Proof for Theorem.

Starting from Eqn. 1.51 and multiplying through by the relevant integrating factor:

$$I(x) \frac{dy}{dx} + I(x)f(x)y = I(x)g(x). \quad (1.53)$$

Then integrating both sides with respect to x :

$$\int \left(I(x) \frac{dy}{dx} + I(x) f(x) y \right) dx = \int \left(I(x) g(x) \right) dx. \quad (1.54)$$

Focusing - for just one second - exclusively on the left hand side of Eqn. 1.54:

$$\int \left(I(x) \frac{dy}{dx} + I(x) f(x) y \right) dx = \int \left(\frac{dy}{dx} e^{\int f(x) dx} + f(x) y e^{\int f(x) dx} \right) dx \quad (1.55)$$

$$= \int \left(\frac{d}{dx} \left[y e^{\int f(x) dx} \right] \right) dx = y e^{\int f(x) dx} = y I(x). \quad (1.56)$$

Substituting this back into Eqn. 1.54 gives:

$$y I(x) = \int \left(I(x) g(x) \right) dx \Rightarrow y = \frac{1}{I(x)} \int \left(I(x) g(x) \right) dx. \quad (1.57)$$

The use of integrating factors may also be demonstrated by an example. Consider,

$$\frac{dy}{dx} + \frac{y}{x} = 0 \quad (1.58)$$

From here we might simply see that the integrating factor would be x , but for clarity we use the formal methods. Let \mathcal{I} denote the integrating factor:

$$\mathcal{I} = e^{\int \left(\frac{1}{x} \right) dx} = e^{\ln |x|} = x \quad (1.59)$$

We have dropped the $+c$ as it could simply cancel out when we multiply through by it and is of no use.

Then we multiply the differential equation through by the integrating factor:

$$x \frac{dy}{dx} + x \frac{y}{x} = 0 \Rightarrow x \frac{dy}{dx} + y = 0 \Rightarrow \frac{d}{dx} (xy) = 0 \Rightarrow y = \frac{c}{x} \quad (1.60)$$

2.4 Second Order Ordinary Differential Equations

Homogeneous Second Order Ordinary Differential Equations with Constant Coefficients

Definition 2.8: Homogeneous Second Order Ordinary Differential Equations With Constant Coefficients

A homogeneous second order ordinary differential equation, with constant coefficients is a second order differential equation of the form:

$$a \frac{d^2 y}{dx^2} + b \frac{dy}{dx} + cy = 0. \quad (1.61)$$

Theorem 2.9: General Solution of Homogeneous Second Order Ordinary Differential Equations With Constant Coefficients

Any differential equation of the form:

$$a \frac{d^2 y}{dx^2} + b \frac{dy}{dx} + cy = 0. \quad (1.62)$$

has solutions of the form:

$$Ae^{\lambda_1 x} + Be^{\lambda_2 x}, \quad (1.63)$$

where A and B can be any complex numbers, and λ_1, λ_2 are the distinct^a solutions of the quadratic equations:

$$a\lambda^2 + b\lambda + c = 0. \quad (1.64)$$

^aIf they are not distinct, then this solution does not apply

Proof for Theorem.

I use the ansatz $e^{\lambda x}$ and substitute it into Eqn. 1.62:

$$a \frac{d^2}{dx^2} \left(e^{\lambda x} \right) + b \frac{d}{dx} \left(e^{\lambda x} \right) + ce^{\lambda x} = 0 \iff a\lambda^2 e^{\lambda x} + b\lambda e^{\lambda x} + ce^{\lambda x} = 0 \quad (1.65)$$

$$\iff \left(a\lambda^2 + b\lambda + c \right) e^{\lambda x} = 0 \quad (1.66)$$

As the solution must obey this equation for any value of x , this can only be satisfied if:

$$a\lambda^2 + b\lambda + c = 0. \quad (1.67)$$

The solutions of this equations were already defined as λ_1 and λ_2 . Hence both,

$$e^{\lambda_1 x} \text{ and } e^{\lambda_2 x} \quad (1.68)$$

are solutions to the differential equation in Eqn. 1.62. As Eqn. 1.62 is clearly linear, this implies that the general solution is:

$$Ae^{\lambda_1 x} + Be^{\lambda_2 x}, \quad (1.69)$$

where A and B can be any complex numbers. ■

I then just have to handle the case that was excluded from the above theorem: if $a\lambda^2 + b\lambda + c = 0$ has repeated roots.

Substitution

The essence of using a substitution is that by making a substitution we change a differential equation into one we can more easily solve. Once we have solved the differential equation, we can then undo the substitution to get the solution to the original differential equation.

This is akin to how you solve integrals by substitution.

I think the first thing to learn is how to use a substitution before concerning ourselves with how to choose a substitution.

For example, consider:

$$x \frac{dy}{dx} + y = e^{x(1+y)} \quad (1.70)$$

This isn't exactly the nicest looking differential equation. If we are given the substitution $u = xy$, then:

$$\frac{du}{dx} = \frac{d}{dx} \left(xy \right) = y + x \frac{dy}{dx} \quad (1.71)$$

Using the product rule.

This is conveniently the LHS of the differential equation so we simply substitute it in.

$$\frac{du}{dx} = e^{x(1+y)}$$

We then finish off our substitution by subbing in u to the right hand side:

$$\frac{du}{dx} = e^{x+u}$$

This is a much nicer differential equation and can be solved simply by separating variables:

$$\frac{du}{dx} = e^{x+u} \Rightarrow e^{-u} \frac{du}{dx} = e^x \Rightarrow \int \left(e^{-u} \frac{du}{dx} \right) dx = \int \left(e^x \right) dx \Rightarrow -e^{-u} + c = e^x \Rightarrow e^{-u} + e^x = c$$

We then need to undo the effects of our substitution:

$$e^{-u} + e^x = c \Rightarrow e^{-xy} + e^x = c \quad (1.72)$$

This implicitly defines y but if you want to clearly derive a formula for y :

$$e^{-xy} + e^x = c \Rightarrow y = -\frac{1}{x} \ln(c - e^x) \quad (1.73)$$

This solves the differential equation, which you can check by substituting the equation into the original differential equation.

Choosing a Substitution

The difficult part is deciding the substitution to make. The answer is whichever substitution gives a differential equation you can more easily solve is the correct substitution. There is no definitive rule, the best way is to see as many substitutions as you can to gain the intuition as to which may work.

In the exams I looked at, the substitution to make was strongly implied (January 2019, Q2a). We can see from the example above that a substitution where a complicated sum of terms, with terms we maybe don't want to deal with, can be replaced with a simple derivative are useful and can guide our choice of substitution. One final tip is to look at how the variables present already group e.g. in the example above xy was already present in the exponent on the RHS which can tip us off that the substitution $u = xy$ could simplify things.

Note that it is entirely valid to use multiple substitutions.

Theorem 2.10: Homogeneous Second Order Linear Differential Equations (With Repeated Roots)

If $a\lambda^2 + b\lambda + c = 0$ has only a single solution, λ_1 , then the differential equation:

$$a\frac{d^2y}{dx^2} + b\frac{dy}{dx} + cy = 0 \quad (1.74)$$

has the solution:

$$\left[Ax + B \right] e^{\lambda_1 x}, \quad (1.75)$$

, where A and B can be any complex numbers.

Proof for Theorem.

Again, this can be demonstrated by substituting the candidate solution into the right hand

side of Eqn. 1.74:

$$a \frac{d^2}{dx^2} \left([Ax + B] e^{\lambda_1 x} \right) + b \frac{d}{dx} \left([Ax + B] e^{\lambda_1 x} \right) + c [Ax + B] e^{\lambda_1 x} \quad (1.76)$$

$$= a \frac{d}{dx} \left(A e^{\lambda_1 x} + \lambda_1 [Ax + B] e^{\lambda_1 x} \right) + b \left(A e^{\lambda_1 x} + \lambda_1 [Ax + B] e^{\lambda_1 x} \right) + c [Ax + B] e^{\lambda_1 x} \quad (1.77)$$

$$= aA\lambda_1 e^{\lambda_1 x} + a\lambda_1 A e^{\lambda_1 x} + a\lambda_1^2 [Ax + B] e^{\lambda_1 x} + bA e^{\lambda_1 x} + b\lambda_1 [Ax + B] e^{\lambda_1 x} + c [Ax + B] e^{\lambda_1 x} \quad (1.78)$$

$$= \left(2aA\lambda_1 + a\lambda_1^2 B + bA + b\lambda_1 B + cB \right) e^{\lambda_1 x} + \left(a\lambda_1^2 A + b\lambda_1 A + cA \right) x e^{\lambda_1 x} \quad (1.79)$$

As λ_1 is defined as the solution of $a\lambda^2 + b\lambda + c = 0$, $a\lambda_1^2 A + b\lambda_1 A + cA = 0$. Therefore the above is equivalent to:

$$\left(2aA\lambda_1 + a\lambda_1^2 B + bA + b\lambda_1 B + cB \right) e^{\lambda_1 x} = \left(2aA\lambda_1 + bA + [a\lambda_1^2 + b\lambda_1 + c] B \right) e^{\lambda_1 x} \quad (1.80)$$

$$= \left(2a\lambda_1 + b \right) A e^{\lambda_1 x}. \quad (1.81)$$

Considering the quadratic formula and that λ_1 is the only solution to $a\lambda^2 + b\lambda + c = 0$, $\lambda_1 = \frac{-b}{2a}$,

$$a \frac{d^2}{dx^2} \left([Ax + B] e^{\lambda_1 x} \right) + b \frac{d}{dx} \left([Ax + B] e^{\lambda_1 x} \right) + c [Ax + B] e^{\lambda_1 x} = \left(2a \left[\frac{-b}{2a} \right] + b \right) A e^{\lambda_1 x} \quad (1.82)$$

$$= 0. \quad (1.83)$$

2.5 Initial Value Problems and Other Boundary Conditions

The discerning reader may have noticed that in all the above differential equations no single function is derived as a solution. There were always A s or B s that may be arbitrary real numbers within the solution. This was not wrong, but nature is not like this: a ball thrown through the air does not have a range of solutions describing its motion. It has but one.

So how does the multitude of solutions we obtain from the differential equations describing the physics of the situation reduce to exactly one?

The answer lies in the initial conditions (or other boundary conditions).

Definition 2.11: Initial Conditions

An initial condition is a restriction on a solution to a differential equation that demands that the solution to the problem - a function - takes a particular value when the argument is set to a particular value (typically, that - if the solution is a function of time - the solution takes a particular value, perhaps being in a particular position, when $t = 0$).

These initial conditions are satisfied by taking our general solution to the differential equation, without the initial conditions, and using the required conditions derive the values of the unknown/arbitrary values - in the above cases these were A s and B s - and solve for the particular values of them that make the required conditions true.

This point may be best illustrated with an example.

Example Initial Value Problem

Solve the differential equation:

$$\frac{d^2y}{dx^2} - 2\frac{dy}{dx} + y = 0, \quad (1.84)$$

subject to the initial condition:

$$y(0) = 0 \text{ and } y'(1) = 4e. \quad (1.85)$$

Solution

We already know that this differential equation, without the initial condition has solution:

$$y(x) = \left[Ax + B \right] e^x. \quad (1.86)$$

So to impose the conditions, I must solve - for A and B - the simultaneous equations:

$$y(0) = \left[A0 + B \right] e^0 = 0, \quad (1.87)$$

$$y'(1) = Ae^1 + \left[A + B \right] e^1 = 4e. \quad (1.88)$$

These simultaneous equations can be seen to have solution:

$$A = 2 \quad (1.89)$$

$$B = 0. \quad (1.90)$$

Hence, the solution to the initial value problem is:

$$y(x) = \left[2x + 0 \right] e^x = 2xe^x. \quad (1.91)$$

2.6 Choice of Method

The techniques here are the standard techniques but these will not always work. Choosing exactly which technique to try boils down to recognising a situation where a particular one may help, either through trial and error or intuition. but there are guidelines: some equations obviously can't have their variables separated, using the auxiliary equation will fail when an equation is not linear.

NOT ALL METHODS WORK FOR ALL EQUATIONS, YOU HAVE TO ADAPT. TRY THINGS, SEE WHAT WORKS, LEARN FROM WHAT FAILS.

Very Rough Algorithm

Here is a *very basic* rough guide on what steps to go through when choosing a method: Consider the general second order ordinary differential equation:

$$A(f(t), t) \frac{d^2 f}{dt^2} + B(f(t), t) \frac{df}{dt} + C(t)F(t) = D(t) \quad (1.92)$$

In order:

- 1) Can you separate variables? (this should be relatively clear to see). If yes, use separation of variables.
- 2) Are A, B, C constant? If yes, use the auxiliary equation.
- 3) Try using an integrating factor.
- 4) Look for a substitution.

Of course if you are told (or hinted) to use a certain method, use that. If your intuition tells you something will work, that's what we're aiming for, try that.

First order differential equations are more likely to be separable or have an integrating factor.

Second order equations are more likely to need substitutions if the auxiliary equation method isn't available.

3 Matrices

3.1 Basic Definitions

For the purposes of quantum mechanics, we only need to consider square matrices.

Definition 3.1: Square Matrices

A $n \times n$ square matrix over a set \mathbb{F} , that is typically \mathbb{R} or \mathbb{C} , is a set of elements of \mathbb{F} arranged into a two-dimensional grid. The element of \mathbb{F} in the i th row and j th column of the matrix A is labelled A_{ij} . The set of all $n \times n$ square matrices over \mathbb{F} is often denoted as $\mathbb{F}^{n \times n}$.

The important features of matrices are how they interact with each other under specific operations. The two main operations are addition and multiplication, defined as:

Definition 3.2: Matrix Addition

We define matrix addition of two matrices, A and B , in terms of the elements of the resulting matrix and the input matrices:

$$(A + B)_{ij} = A_{ij} + B_{ij}. \quad (1.93)$$

Definition 3.3: Matrix Multiplication

We define matrix multiplication of two matrices, A and B , in terms of the elements of the resulting matrix and the input matrices:

$$(AB)_{ij} = \sum_{k=1}^n (A_{ik} B_{kj}). \quad (1.94)$$

3.2 Commutators

For the purposes of quantum mechanics, perhaps the most important aspect of matrices is that they do not commute under matrix multiplication.

Definition 3.4: Commutation

Two objects, A and B , are said to commute under binary operation, $*$, if and only if:

$$A * B = B * A. \quad (1.95)$$

Therefore, given two matrices, we say that they do not commute under matrix multiplication. A key example of this is:

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \text{ and } \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}. \quad (1.96)$$

$$\text{Therefore, } \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \neq \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (1.97)$$

The main way we capture the commutativity / non-commutativity of two matrices is commutators. We define these as:

Definition 3.5: Commutators

Given two matrices, A and B , define the commutator, $[A, B]$, of the two matrices as:

$$[A, B] = AB - BA. \quad (1.98)$$

The main way we use this concept is to rearrange the order of matrix multiplication.

Lemma 1. *For any four matrices, A, B, C, D :*

$$ABCD = ACBD + A[B, C]D. \quad (1.99)$$

Proof. Start from the definition of a commutator between two matrices, B and C :

$$[B, C] = BC - CB \Rightarrow BC = [B, C] + CB. \quad (1.100)$$

Therefore,

$$ABCD = A(BC)D = A([B, C] + CB)D = ACBD + A[B, C]D. \quad (1.101)$$

□

Note that we have achieved the required rearrangement at the cost of the additional term. The single most important value a commutator can take is zero. This is the case if and only if the two argument matrices commute.

Theorem 3.6: Commutator of Commuting Matrices

The commutator of any two matrices is zero if and only if they commute.

Proof for Theorem.

I prove this theorem in two parts:

The commutator of any two matrices is zero if they commute

Let A and B be any two matrices that commute. Then - by definition of commutation - $AB = BA$. Therefore,

$$[A, B] = AB - BA = AB - AB = 0. \quad (1.102)$$

The commutator of any two matrices is zero only if they commute

For the other half of the theorem note that this is equivalent to if two matrices have a commutator equal to zero, then they must commute. Assume

$$[A, B] = 0 \Rightarrow AB - BA = 0 \Rightarrow AB = BA. \quad (1.103)$$

Therefore, A and B commute. ■

3.3 Matrices as Transforms on a Vector Space

The first thing to define is a vector space.

Definition 3.7: Vector Spaces (informally)

We - for our purposes herein - define a vector space over some set of numbers (such as \mathbb{R} or \mathbb{C})^a as the set of all vectors of a particular size where the entries are all in the specified set. The set of vectors of size $n \in \mathbb{N}$ with all elements being in \mathbb{R} or \mathbb{C} is denoted as \mathbb{R}^n or \mathbb{C}^n , respectively.

^aFormally this can be any field, but do not worry about that for now

I now take a slightly different perspective on matrices but one motivated by something you should be familiar with.

You are likely aware that vectors can be represented as columns of numbers e.g.

$$\begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} \text{ or } \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \text{ or } \begin{pmatrix} 1+i \\ 0 \\ i \\ 0 \\ 7 \end{pmatrix} \text{ or } \begin{pmatrix} 8+7i \\ \frac{2}{3} \\ 2i \\ 0 \\ \pi \end{pmatrix} \text{ or } 5 \begin{pmatrix} 1 \\ 1 \\ 2i \\ 0 \\ 6 \end{pmatrix} \quad (1.104)$$

But I then note that I can consider the column vectors - as above - as a matrix and apply the previously established rules of matrix multiplication to allow a matrix (of the right size) to act on the vector:

Definition 3.8: Multiplying a Vector by a Matrix

For any $n \times n$ matrix, A , and column vector of size n , \vec{v} , define the multiplication of \vec{v} by A (also known as the application of A to \vec{v}) as:

$$(A\vec{v})_i = \sum_{k=1}^n \left(A_{ik} \vec{v}_k \right). \quad (1.105)$$

The key thing to observe is that - generally - $A\vec{v} \neq \vec{v}$. Although there are - very! - important cases where $A\vec{v} = \vec{v}$, as we shall see later. Hence we can see that multiplication by a matrix, in this case A , has changed/transformed \vec{v} . As A can be applied to *any* vector of the correct size (i.e. of the same size as \vec{v}), A can be seen as transforming the set of all vectors of that size. Therefore, A , or any $n \times n$ matrix, can - and often is, like we will do later - be alternatively viewed as a transformation on the vector space of all vectors of size n . I now give an example that hopefully will prove illustrative.

Consider a classical¹ particle in three-dimensional space. Its position can be described entirely by a vector from \mathbb{R}^3 . Any 3×3 matrix over the real numbers can be seen as transforming the vector representing the particles position: we can therefore interpret it as

¹If this does not mean anything to you yet, do not worry: just pretend that the word classical is not here.

representing moving the particle. Consider the particular example:

$$\vec{v} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad (1.106)$$

which described the particle as being in a particular position. If I then pick a 3×3 matrix over the real numbers (and label it A):

$$A = \begin{pmatrix} 1, 2, 3 \\ 4, 5, 6 \\ 7, 8, 9 \end{pmatrix}, \quad (1.107)$$

we can view this matrix, A , as moving the particle from the location represented by \vec{v} to the location represented by $A\vec{v}$, i.e. to:

$$A\vec{v} = \begin{pmatrix} 1, 2, 3 \\ 4, 5, 6 \\ 7, 8, 9 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 4 \\ 7 \end{pmatrix}. \quad (1.108)$$

3.4 Determinants

I define the determinant of a matrix as:

Definition 3.9: Determinant of a Matrix

For a $n \times n$ square matrix, M , the determinant, denoted \det , is defined recursively:
 $\forall i \in \mathbb{N}^{\leq n}$,

$$\det(M) = \sum_{j=1}^n \left[(-1)^{i+j} M_{i,j} \det(m_{i,j}) \right], \quad (1.109)$$

where $m_{i,j}^M$ is a matrix obtained from M by removing the i th row and the j th column.

I then prove several important properties of the determinant:

Theorem 3.10

Exchanging any two neighbouring rows in a matrix multiplies its determinant by -1 .

Proof for Theorem.

■ Consider two matrices: M and M' , where M' is M but with neighbouring rows r_1 and r_2

exchanged (assume WLOG that $r_2 = r_1 + 1$). Then consider:

$$\det(M') = \sum_{j=1}^n \left[(-1)^{r_1+j} M'_{r_1,j} \det(m_{r_1,j}^{M'}) \right] \quad (1.110)$$

$$\det(M) = \sum_{j=1}^n \left[(-1)^{r_2+j} M_{r_2,j} \det(m_{r_2,j}^M) \right] = \sum_{j=1}^n \left[(-1)^{r_1+1+j} M_{r_2,j} \det(m_{r_2,j}^M) \right]. \quad (1.111)$$

Due to the exchanging of the rows: $m_{r_2,j}^M = m_{r_1,j}^{M'}$ and $M_{r_2,j} = M'_{r_1,j}$. Therefore,

$$\det(M) = - \sum_{j=1}^n \left[(-1)^{r_1+j} M_{r_1,j} \det(m_{r_1,j}^M) \right] = -\det(M'). \quad (1.112)$$

Theorem 3.11: Determinant of a Matrix with Repeated Rows

Any matrix where two rows are identical has a determinant of zero.

Proof for Theorem.

Let M be a matrix where rows r_1 and r_2 are identical. Then let M' be M but with a series of exchanges performed such that the two identical rows are neighbouring. Then $\det(M')$ is $\det(M)$ up to a possible factor of -1 . Exchanging the two neighbouring and identical rows in M' must multiply its determinant by a factor of -1 but as those rows are identical it must also leave the determinant unchanged. Therefore the determinant must be zero, so that both statements can be satisfied. ■

3.5 Inverse of a Matrix

An important class of matrices to examine are the identities:

Definition 3.12: The identity of size $n \in \mathbb{N}$

For each $n \in \mathbb{N}$, there is a unique $n \times n$ matrix such that for any $n \times n$ matrix, A :

$$AI = IA = A. \quad (1.113)$$

I then define the inverse of a matrix.

Definition 3.13: Inverse of a Matrix

The inverse, M^{-1} , of a matrix, M , is any matrix such that:

$$MM^{-1} = M^{-1}M = I. \quad (1.114)$$

I then turn to how to calculate the inverse.

Theorem 3.14: Calculating a Matrix Inverse

For any matrix, M , for which an inverse exists^a, that inverse is:

$$M^{-1} = \frac{C^T}{\det(M)}, \quad (1.115)$$

where $C_{i,j} = (-1)^{i+j} \det(m_{i,j}^M)$.

^aThis is the case whenever it does not have 0 for an eigenvalue.

Proof for Theorem.

Consider the definition of matrix multiplication, then for any $n \times n$ matrix,

$$(MM^{-1})_{ij} = \sum_{k=1}^n \left(A_{ik} (-1)^{j+k} \det(m_{j,k}^A) \right) \frac{1}{\det(A)} = \begin{cases} \det(A) \frac{1}{\det(A)} = 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}, \quad (1.116)$$

with the second case following from its summation's equivalence to the determinant of a matrix with repeated rows. Therefore,

$$MM^{-1} = I. \quad (1.117)$$

It is also required that $M^{-1}M = I$ but this is similar and I leave it as an exercise. ■

3.6 Eigenvalues and Eigenvectors

When an individual matrix is being considered, perhaps its most important properties are its eigenvalues and eigenvectors. These are defined by:

Definition 3.15: Eigenvalues and Eigenvectors

For any matrix, $M \in \mathbb{C}^{n \times n}$, an eigenvector, $|\lambda\rangle \in \mathbb{C}^n$, and its corresponding eigenvalue, $\lambda \in \mathbb{C}$, of M satisfy the equation:

$$M|\lambda\rangle = \lambda|\lambda\rangle. \quad (1.118)$$

Eigenvalues and eigenvectors are important, but perhaps the most important thing to know about them is how to find them. To consider this, examine the equation defining eigenvector and eigenvalues:

$$M|\lambda\rangle = \lambda|\lambda\rangle \iff (M - \lambda\hat{I})|\lambda\rangle = 0. \quad (1.119)$$

Assuming $|\lambda\rangle$ is not zero, this only has a solution if:

$$\left| M - \lambda \hat{I} \right| = 0 \quad (1.120)$$

Eqn. 1.120 is called the characteristic equation of M ; if M is an $n \times n$ matrix, it is an n th order polynomial in λ and hence has n roots which are the eigenvalues of M . Hence the eigenvalues can be obtained by solving the characteristic polynomial.

Using each of the now known eigenvalues, denoted λ_0 , in:

$$M|\lambda_0\rangle = \lambda_0|\lambda_0\rangle, \quad (1.121)$$

provides a series of n (assuming again that M is a $n \times n$ matrix) simultaneous equations, each with n variables. Hence this system of equations can be solved to find the corresponding eigenvector $|\lambda_0\rangle$.

3.7 Diagonalizations of Matrices

As previously established, a $n \times n$ matrix has n eigenvectors. For ease, index them so that for each positive integer, $j, \leq n$ there is a corresponding distinct eigenvector, $|\lambda_j\rangle$. Similarly, let $\mathbf{1}_j$ denote the n -element vector where all elements but the j th are zero and the j th is one. E.g. if $n = 3$,

$$\mathbf{1}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}. \quad (1.122)$$

It is reasonably easy to see that there exists a matrix V such that, $\forall j \in \mathbb{N}^{\leq n}$:

$$|\lambda_j\rangle = V\mathbf{1}_j. \quad (1.123)$$

If I then define D_M to be the diagonal matrix where $(D_M)_{j,j}$ is the eigenvalue of M corresponding the the eigenvector $|\lambda_j\rangle$.

Theorem 3.16: Diagonalization of a Matrix

For any $n \times n$ matrix, M , with n linearly-independent eigenvectors, may be expressed as:

$$M = VD_MV^{-1}, \quad (1.124)$$

where V is defined by:

$$|\lambda_j\rangle = V\mathbf{1}_j. \quad (1.125)$$

Proof for Theorem.

As M is a $n \times n$ matrix with n linearly-independent eigenvectors, any n -element vector may be expressed as a linear combination of the eigenvectors of M . Therefore, if any other matrix has the exact same eigenvectors and eigenvalues as M it must affect every vector the same as M and so must be equal to M . In light of this, consider:

$$VD_MV^{-1}|\lambda_j\rangle = VD_MV^{-1}V\mathbf{1}_j = VD_M\mathbf{1}_j = V(D_M)_{j,j}\mathbf{1}_j = (D_M)_{j,j}V\mathbf{1}_j = (D_M)_{j,j}|\lambda_j\rangle. \quad (1.126)$$

As $(D_M)_{j,j}$ was defined as the eigenvalue of M corresponding to $|\lambda_j\rangle$, this shows:

$$M = VD_MV^{-1}. \quad (1.127)$$

3.8 Conjugate Transpose

One operation on matrices that does not immediately seem to be important but ends up being immensely important to quantum mechanics is the conjugate transpose.

Definition 3.17: Conjugate Transpose of a Matrix

For any matrix, M , the conjugate transpose of M , denoted M^\dagger , is defined:

$$(M^\dagger)_{i,j} = [(M)_{j,i}]^*, \quad (1.128)$$

where $[\cdot]^*$ denotes the complex conjugate of its argument.

3.9 Commonly Used Subsets of Matrices

Throughout physics, we repeatedly find that certain subsets of matrices are especially important. Here I present two that are particularly important to quantum mechanics.

Definition 3.18: Hermitian Matrices

A matrix, M , is Hermitian if and only if it is its own conjugate transpose.

Definition 3.19: Unitary Matrices

A matrix, M , is unitary if and only if it has an inverse and that inverse is its conjugate transpose.

I then prove the most important features of these two families of matrices.

Theorem 3.20: Product of a unitary matrix and its Conjugate Transpose

For any unitary matrix, U ,

$$UU^\dagger = U^\dagger U = I. \quad (1.129)$$

Proof for Theorem.

By definition of a unitary matrix, $U^\dagger = U^{-1}$, therefore:

$$UU^\dagger = UU^{-1} = I \quad (1.130)$$

$$U^\dagger U = U^{-1}U = I. \quad (1.131)$$

Theorem 3.21: Eigenvalue of Hermitian Matrices

For any Hermitian matrix, all of its eigenvalues are real.

Proof for Theorem.

Let v_λ be an eigenvector of matrix, M , with eigenvalue, $\lambda \in \mathbb{C}$. Then,

$$Mv_\lambda = \lambda v_\lambda \quad (1.132)$$

Taking the complex transpose of each side:

$$v_\lambda^\dagger M^\dagger = \lambda^* v_\lambda^\dagger. \quad (1.133)$$

Multiplying, from the right, both sides by v_λ and noting that - as it is Hermitian - $M^\dagger = M$ gives:

$$v_\lambda^\dagger M v_\lambda = \lambda^* v_\lambda^\dagger v_\lambda \Rightarrow \lambda v_\lambda^\dagger v_\lambda = \lambda^* v_\lambda^\dagger v_\lambda \Rightarrow \lambda = \lambda^*. \quad (1.134)$$

Hence, v_λ must be real. ■

I leave as an exercise, the proof that the V in the diagonalization of any matrix is unitary. Finally, I state an important result without proof.

Theorem 3.22: Relation Between Hermitian and Unitary Matrices

Any unitary matrix may be expressed as e^{itH} , where H is some Hermitian matrix.

Proof for Theorem.

■ Proof Omitted. ■

4 Traces and Norms

Definition 4.1: Traces

For any matrix, M , its trace, denoted as:

$$\text{Tr}\left(M\right), \quad (1.135)$$

is the sum of the eigenvalues of M .

Definition 4.2: Schatten Norms

$\forall p \in \mathbb{N}$, define the Schatten p -norm as a function from matrices to the reals:

$$\|M\|_p = \left[\text{Tr}\left((M^\dagger M)^{p/2}\right) \right]^{1/p}. \quad (1.136)$$

As Schatten p -norms are defined on matrices, via representation theory, they are also defined on linear operators.

Definition 4.3: Trace Norm

The trace norm (also known as the nuclear norm) is the Schatten p -norm with $p = 1$.

Definition 4.4: Trace Distance

For any two matrices, ρ, σ , define the trace distance, $\mathcal{D}(\rho, \sigma)$, as:

$$\mathcal{D}(\rho, \sigma) = \frac{1}{2} \|\rho - \sigma\|_1. \quad (1.137)$$

Definition 4.5: Variation Distance

Given two probability distributions over the set of possible outcomes, Ω , (i.e. mappings that take a possible outcome, $s \in \Omega$, as input and return a number from $[0, 1]$), P and Q , define the variational distance between them, $VD(P, Q)$, by:

$$VD(P, Q) = \frac{1}{2} \sum_{s \in \Omega} \left| P(s) - Q(s) \right|. \quad (1.138)$$

5 Taylor Series

Theorem 5.1: Integration by Parts

For any differentiable functions u and v ,

$$\int [uv'] dx = uv - \int [u'v] dx. \quad (1.139)$$

Proof for Theorem.

Let u, v be differentiable functions of x and start from the product rule (as in Def. ??)

$$(uv)' = u'v + uv'. \quad (1.140)$$

I then take the integral of both sides of Eqn. 1.140:

$$\int [(uv)'] dx = \int [u'v + uv'] dx = \int [u'v] dx + \int [uv'] dx. \quad (1.141)$$

Hence, applying the fundamental theorem of calculus (axiom 14) to the left-most part of Eqn. 1.141 gives:

$$uv = \int [u'v] dx + \int [uv'] dx. \quad (1.142)$$

Eqn. 1.142 can then be re-arranged to give the formula for integration by parts (as in Eqn. ??):

$$\int [uv'] dx = uv - \int [u'v] dx. \quad (1.143)$$

■

Theorem 5.2: Taylor's Theorem

For any function, f , such that for any interval on the real line, I , there exists some constant, $C \in \mathbb{R}$, such that:

$$\forall x \in I, \left| f^{(k)}(x) \right| < C^{k+1} k!, \quad (1.144)$$

where $f^{(k)}(x)$ denotes the k th derivative of f with respect to x^a . $\forall k \in \mathbb{N}, \forall p \in \mathbb{R}$:

$$f(x) = \sum_{j=0}^k \left(\frac{f^{(j)}(p)}{j!} (x-p)^j \right) + \int_p^x \left(\frac{f^{(k+1)}(y)}{k!} (x-y)^k \right) dy. \quad (1.145)$$

^aThis condition is equivalent to assuming the function f is analytic.

Proof for Theorem.

I start with a definition of an integral, $J[x, n]$. This is not new maths or a new axiom, it is just notation.

$$J[x, n] = \int_0^x \left(\frac{f^{(n+1)}(y)}{n!} (x-y)^n \right) dy. \quad (1.146)$$

Then, integrating Eqn. 1.146 by parts:

$$\left[\frac{f^{(n+1)}(y)}{n!} \int_0^x \left((x-y)^n \right) dy \right]_0^x - \int_0^x \left(\frac{f^{(n+2)}(y)}{n!} \int_0^x \left((x-y)^n \right) dy \right) dy. \quad (1.147)$$

The integral $\int_0^x \left((x-y)^n \right) dy$ can be evaluated, hence:

$$J[x, n] = \left[- \frac{f^{(n+1)}(y)}{n!} \frac{(x-y)^{n+1}}{n+1} \right]_0^x + \int_0^x \left(\frac{f^{(n+2)}(y)}{n!} \frac{(x-y)^{n+1}}{n+1} \right) dy \quad (1.148)$$

$$= \left[\frac{f^{(n+1)}(y)}{(n+1)!} (x-y)^{n+1} \right]_x^0 + \int_0^x \left(\frac{f^{(n+2)}(y)}{(n+1)!} (x-y)^{n+1} \right) dy \quad (1.149)$$

$$= \frac{f^{(n+1)}(0)}{(n+1)!} x^{n+1} + J[x, n+1]. \quad (1.150)$$

To prove the existence of a Taylor series for any value of n , I then start from a base case using the fundamental theorem of calculus:

$$f(x) - f(0) = \int_0^x \left(f'(y) \right) dy \Rightarrow f(x) = f(0) + \int_0^x \left(f'(y) \right) dy, \quad (1.151)$$

which is the the required Taylor series when $n = 0$. Then, assuming the required Taylor series exists for a given value of $n \in \mathbb{Z}$:

$$f(x) = \sum_{j=0}^n \left(\frac{f^{(j)}(0)}{j!} (x)^j \right) + \int_0^x \left(\frac{f^{(k+1)}(y)}{k!} (x-y)^n \right) dy = \sum_{j=0}^n \left(\frac{f^{(j)}(0)}{j!} (x)^j \right) + J[x, n]. \quad (1.152)$$

Then, using Eqn. 1.150 to re-write $J[x, n]$:

$$f(x) = \sum_{j=0}^n \left(\frac{f^{(j)}(0)}{j!} (x)^j \right) + \frac{f^{(n+1)}(0)}{(n+1)!} x^{n+1} + J[x, n+1] = \sum_{j=0}^{n+1} \left(\frac{f^{(j)}(0)}{j!} (x)^j \right) + J[x, n+1]. \quad (1.153)$$

Therefore, the required Taylor series of $n+1$ exists. Therefore, axiom 18 allows me to - by induction - assert that the required Taylor series exists for any positive integer. As all the Taylor series produced above were always about zero, they are technically Maclaurin series. I now show that a Taylor series can always be obtained from a suitable Maclaurin series.

Define a new function in terms of the previously established general function: $g(x) = f(x-z)$, where $z \in \mathbb{R}$ and can be chosen as needed. For any chosen g a corresponding f exists, and for any chosen, f a corresponding g exists. Then,

$$g(z) = f(0), \quad (1.154)$$

$$g'(z) = f'(0), \quad (1.155)$$

$$g''(z) = f''(0), \quad (1.156)$$

$$g'''(z) = f'''(0), \quad (1.157)$$

$$\text{and so on.} \quad (1.158)$$

This provides for a Taylor series to be taken about any point. To calculate this, an appropriate f can always be found. ■

Theorem 5.3: Infinite Taylor's Theorem

For any function, f , meeting all conditions of Taylor's theorem, there exists some neighbourhood^a of any point, $p \in \mathbb{R}$, such that:

$$f(x) = \sum_{j=0}^{\infty} \left(\frac{f^{(j)}(p)}{j!} (x-p)^j \right). \quad (1.159)$$

^aA neighbourhood of a specified point just means some area around that point of unspecified size.

Proof for Theorem.

By the inherited conditions for Taylor's theorem, for any interval centred on $p \in \mathbb{R}$, there exists some constant, $C \in \mathbb{R}$, such that:

$$\left| f^{(k)}(x) \right| < C^{k+1} k!. \quad (1.160)$$

Hence,

$$\frac{|f^{(k)}(x)|}{k!} |x-y|^k < \frac{C^{k+1} k!}{k!} |x-y|^k = C^{k+1} |x-y|^k. \quad (1.161)$$

This is then used to show that, $\forall k \in \mathbb{N}$,

$$\int_p^x \left(\frac{f^{(k+1)}(x)}{k!} (x-y)^k \right) dy \leq \left| \int_p^x \left(\frac{f^{(k+1)}(y)}{k!} (x-y)^k \right) dy \right| \quad (1.162)$$

$$\leq \int_p^x \left(\left| \frac{f^{(k+1)}(y)}{k!} (x-y)^k \right| \right) dy \leq \int_p^x \left(\frac{|f^{(k+1)}(y)|}{k!} |x-y|^k \right) dy \leq C^{k+1} \int_p^x (|x-y|^k) dy. \quad (1.163)$$

If I restrict x to be within ξ^{-1} - where $\xi = 1.1C$ - of p , then if y is restricted to within the limits of the integral in Eqn. 1.163, $|x-y| \leq \xi^{-1}$ and $x-p \leq \xi^{-1}$. Therefore, using this restriction:

$$\int_p^x \left(\frac{f^{(k+1)}(x)}{k!} (x-y)^k \right) dy \leq C^{k+1} \int_p^x (|x-y|^k) dy \leq C^{k+1} \xi^{-k} \int_p^x (1) dy \quad (1.164)$$

$$\leq C^{k+1} \xi^{-k} (x-p) \leq C^{k+1} \xi^{-(k+1)} = \frac{C^{k+1}}{C^{k+1}(1.1)^{k+1}} \leq (1.1)^{-k}. \quad (1.165)$$

It can be shown, that $\forall k \in \mathbb{N}$, $(1.1)^{-k} > (1.1)^{-(k+1)}$, so I can extend the series to be infinite with zero error, i.e.

$$f(x) = \sum_{j=0}^{\infty} \left(\frac{f^{(j)}(p)}{j!} (x-p)^j \right). \quad (1.166)$$

6 Functions and Taylor Series of Matrices

For many functions that we can apply to the real or complex numbers, we can also apply them to matrices. We define these functions via their Taylor series. The most important one of these is exponentials. The Taylor expansion of the exponential function is: $\forall x \in \mathbb{C}$,

$$e^x = \sum_{j=1}^{\infty} \left(\frac{x^j}{j!} \right). \quad (1.167)$$

Therefore, for any matrix M , define e^M by:

$$e^M = \sum_{j=1}^{\infty} \left(\frac{M^j}{j!} \right). \quad (1.168)$$

I note that, in practice, this is not how anybody finds the exponential of a matrix. Instead, start from the diagonalization of the matrix to be exponentiated:

$$M = V D_M V^{-1}. \quad (1.169)$$

Then take the above Taylor series of this new form of M :

$$e^M = \sum_{j=1}^{\infty} \left(\frac{[VD_M V^{-1}]^j}{j!} \right) = \sum_{j=1}^{\infty} \left(\frac{V[D_M V^{-1} V]^j V^{-1}}{j!} \right) = V \sum_{j=1}^{\infty} \left(\frac{[D_M]^j}{j!} \right) V^{-1} = V e^{D_M} V^{-1}. \quad (1.170)$$

This final expression of e^M is much easier to compute: the exponential of a diagonal matrix - which D_M is - is easily done by taking the exponential of each diagonal element.

7 Special Functions

7.1 Hermite Polynomials

The Hermite polynomials are an infinite set of polynomials indexed by a parameter $n \in \mathbb{N}$. They are defined recursively as in Def. 7.1.

Definition 7.1: Hermite Polynomials

Define the n th Hermite polynomial, $H_n(x)$, by:

$$H_n(x) = \left(2x - \frac{d}{dx} \right)^n 1. \quad (1.171)$$

Therefore, the first few Hermite polynomials can be calculated as:

- $H_0 = 1$,
- $H_1 = 2x$,
- $H_2 = 4x^2 - 2$,
- $H_3 = 8x^3 - 12x$,
- $H_4 = 16x^4 - 48x^2 + 12$,
- $H_5 = 32x^5 - 160x^3 + 120x$.

I then derive our first simple result about Hermite polynomials.

Lemma 2. $H_{n+1}(x) = 2xH_n(x) - H'_n(x)$

Proof. Using the definition of the n th Hermite polynomial - in Def. 7.1:

$$H_{n+1}(x) = \left(2x - \frac{d}{dx} \right)^{n+1} 1 = \left(2x - \frac{d}{dx} \right) \left(2x - \frac{d}{dx} \right)^n 1 = \left(2x - \frac{d}{dx} \right) H_n(x) \quad (1.172)$$

$$= 2xH_n(x) - H'_n(x) \quad (1.173)$$

□

Lemma 2 is used immediately in Theorem 7.1.

Theorem 7.2: Derivative of Hermite Polynomials

For any $n \in \mathbb{N}$,

$$H'_{n+1}(x) = 2(n+1)H_n(x) \quad (1.174)$$

Proof for Theorem.

I proceed by induction. For the base case, it is easy to see that this formula holds when $n = 0$. To begin the inductive step, assume that the formula holds for all $k \leq n$. Then consider,

$$H'_{k+1}(x) = \frac{d}{dx} \left[\left(2x - \frac{d}{dx} \right) \left(2x - \frac{d}{dx} \right)^n 1 \right] = \frac{d}{dx} \left[\left(2x - \frac{d}{dx} \right) H_k(x) \right] \quad (1.175)$$

$$= \frac{d}{dx} [2xH_k(x)] - \frac{d^2}{dx^2} [H_k(x)] = 2H_k(x) + 2xH'_k(x) - H''_k(x). \quad (1.176)$$

Using the assumption that the formula holds for H_k :

$$H'_{k+1}(x) = 2H_k(x) + 4xkH'_k(x) - 4k(k-1)H_{k-1}(x) \quad (1.177)$$

$$= 2H_k(x) + 2k(2xH'_k(x) - 2(k-1)H_{k-1}(x)). \quad (1.178)$$

The application of Lemma 2 then allows this to be rewritten as:

$$H'_{k+1}(x) = 2(k+1)H_k(x). \quad (1.179)$$

Therefore, by induction, the formula holds for all $n \in \mathbb{N}$. ■

I then conclude with a final theorem that will be useful later.

Theorem 7.3

$\forall n \in \mathbb{N}^{\geq 2}$,

$$H_n(x) = 2xH_{n-1}(x) - 2(n-1)H'_{n-2}(x) \quad (1.180)$$

Proof for Theorem.

Using Lemma 2, $H_{n+1}(x) = 2xH_n(x) - H'_n(x)$. Then using Theorem 7.1:

$$H_{n+1}(x) = 2xH_n(x) - H'_n(x) = 2xH_n(x) - 2nH'_{n-1}(x) \quad (1.181)$$

$$\iff H_n(x) = 2xH_{n-1}(x) - H'_{n-1}(x) = 2xH_{n-1}(x) - 2(n-1)H'_{n-2}(x) \quad (1.182)$$
■

CHAPTER 2

A FIRST LOOK AT QUANTUM MECHANICS AND WHY WE CAN TREAT WAVE FUNCTIONS AS VECTORS

1	Schrodinger's Equation	40
2	Examples Showcasing Quantum Effects	41
3	Practical Usage: Using Wells as Qubits	48
4	Setting Up Treating Wavefunctions as Vectors	48

Our initial look at quantum mechanics takes as very limited postulates:

- For any system at any time, there exists a function - known as a wavefunction - completely describing the system.
- The state of the system evolves according to Schrodinger's Equation (given later).
- The probability of measuring a particle as being within a given region is the integral - over that region - of the square of the absolute value of the wavefunction.

1 Schrodinger's Equation

Definition 1.1: A Hamiltonian

A Hamiltonian is an operator corresponding to the total energy of the system. When applied to any state with a definite energy, it maps that state to itself but multiplied by the corresponding energy.

Definition 1.2: The Hamiltonian of a Particle

The Hamiltonian considering to a particle in a potential, $V(x)$, is:

$$\frac{\hbar^2}{2m} \frac{d^2}{dx^2} + \hat{V}(x) \quad (2.1)$$

Definition 1.3: Schrödinger equation

The Schrödinger equation is:

$$i\hbar \frac{d\psi}{dt} = \hat{H}\psi, \quad (2.2)$$

where $\hbar = \frac{6.62607015 \times 10^{-34}}{2\pi} \text{ kg } m^2 s^{-1}$ (the exactness follows from kilograms being defined in terms of this constant) and is called the reduced Planck constant.

What I have given as the Schrödinger equation is often known as the time dependent Schrödinger equation. Another useful equation - known as the time independent Schrödinger equation - may be derived from it.

Theorem 1.4: the Time Independent Schrödinger Equation

For any quantum system with Hamiltonian, \hat{H} , for any stationary state of the system, $\psi_n(x)$, with energy, $E_n \in \mathbb{R}$, satisfies:

$$-\frac{\hbar^2}{2m} \frac{d^2 \psi_{n,x}(x)}{dx^2} + \hat{V}(x) \psi_{n,x}(x) = E_n \psi_{n,x}(x) \quad (2.3)$$

Proof for Theorem.

Start from the time dependent Schrödinger Equation and look for solutions of the form

$$\psi_n(x, t) = \psi_{n,x}(x) \cdot \psi_{n,t}(t):$$

$$i\hbar \frac{d\psi_n}{dt} = \hat{H}\psi_n \quad (2.4)$$

$$\iff i\hbar \frac{d}{dt} \left(\psi_{n,t} \psi_{n,x}(x) \right) = -\frac{\hbar^2}{2m} \frac{d^2}{dx^2} \left(\psi_{n,x}(x) \cdot \psi_{n,t}(t) \right) + \hat{V}(x) \psi_{n,x}(x) \cdot \psi_{n,t}(t) \quad (2.5)$$

$$\iff i\hbar \psi_{n,x}(x) \frac{d\psi_{n,t}}{dt} = -\frac{\hbar^2}{2m} \psi_{n,t}(t) \frac{d^2 \psi_{n,x}(x)}{dx^2} + \hat{V}(x) \psi_{n,x}(x) \psi_{n,t}(t) \quad (2.6)$$

$$\iff \frac{i\hbar}{\psi_{n,t}(t)} \frac{d\psi_{n,t}}{dt} = -\frac{\hbar^2}{2m} \frac{1}{\psi_{n,x}(x)} \frac{d^2 \psi_{n,x}(x)}{dx^2} + \hat{V}(x). \quad (2.7)$$

I see that I have achieved a separation of variables. To have this equation hold in all places at all times, I requires that both sides of the equation equate to a constant. I - suspiciously - call this constant E_n . Therefore, I now have two differential equations to solve:

$$\frac{i\hbar}{\psi_{n,t}(t)} \frac{d\psi_{n,t}}{dt} = E_n \quad (2.8)$$

$$-\frac{\hbar^2}{2m} \frac{1}{\psi_{n,x}(x)} \frac{d^2 \psi_{n,x}(x)}{dx^2} + \hat{V}(x) = E_n. \quad (2.9)$$

Fortunately, the first of these is in a form I can easily solve. A fact more easily seen if the two differential equations are expressed as:

$$\frac{d\psi_{n,t}}{dt} = \frac{-iE_n}{\hbar} \psi_{n,t}(t) \quad (2.10)$$

$$-\frac{\hbar^2}{2m} \frac{d^2 \psi_{n,x}(x)}{dx^2} + \hat{V}(x) \psi_{n,x}(x) = E_n \psi_{n,x}(x). \quad (2.11)$$

The second differential equation is the Time-Independent Schrodinger Equation (TISE) I've been aiming for. It provides a way to derive the stationary states (the states which the system will stay in under the time evolution governed by \hat{H}) of system being considered. With the stationary state labelled $\psi_{n,x}(x)$ having energy E_n . ■

2 Examples Showcasing Quantum Effects

Our first particular examples to examine quantum mechanics consist of a single free particle in various potentials, in a single dimension. Each of the following examples are determined by and differ only in the potentials a particle is placed in.

2.1 Superposition: Infinite Square Well

Our first - slightly unphysical - example is defined by a potential that is zero within a specified one-dimensional region and infinite everywhere else. This results in the particle remaining trapped within this specified region. We formally define the problem to be solved

as examining the behaviour of a particle within the potential:

$$V(x) = \begin{cases} 0, & \text{if } 0 \leq x \leq a, \\ \infty, & \text{otherwise} \end{cases}. \quad (2.12)$$

Examining the TISE within the square well (where $V(x) = 0$):

$$\hat{H}\psi = -\frac{\hbar^2}{2m} \frac{d^2\psi}{dx^2} \Rightarrow -\frac{\hbar^2}{2m} \frac{d^2\psi}{dx^2} = E\psi \Rightarrow \frac{d^2\psi}{dx^2} = -\frac{2mE}{\hbar^2}\psi. \quad (2.13)$$

For convenience, let $k^2 = \frac{2mE}{\hbar^2}$. So the above differential equation has the solution:

$$\psi(x) = A \cos(kx) + B \sin(kx), \quad (2.14)$$

where A and B are arbitrary constants. They are decided by the boundary conditions that are imposed. In this case, the boundary conditions come from the wavefunction disappearing (equalling zero) in the regions where $v(x) = \infty$. This is as there is no chance of measuring the particle being there. This translates to the conditions:

- $\psi(0) = 0 \iff A \cos(0) + B \sin(0) = A = 0$.
- $\psi(a) = 0 \iff A \cos(ka) + B \sin(ka) = B \sin(ka) = 0$

Hence, k must take some value such that ka is a multiple of π . However, there are many such values - of k - that satisfy this constraint: *all* are valid. In equations, we need to find the values of k_n such that, for some integer value of n :

$$k_n a = n\pi \iff k_n = \frac{n\pi}{a}. \quad (2.15)$$

Therefore,

$$\psi_n(x) = B_n \sin\left(\frac{n\pi}{a}x\right). \quad (2.16)$$

Note that there are many different solutions, indexed by the integer value n . The values of B are decided by the *normalization condition*. The normalization condition is derived from the fact that the probability of measuring the particle anywhere must be one i.e. $\forall n \in \mathbb{N}$

$$\int_{-\infty}^{\infty} \left(|\psi_n(x)|^2 \right) dx = 1. \quad (2.17)$$

If we use the above derived formula for $\psi_n(x)$ in this condition:

$$\int_0^a \left(\left| B_n \sin\left(\frac{n\pi}{a}x\right) \right|^2 \right) dx = |B_n|^2 \int_0^a \left(\left| \sin\left(\frac{n\pi}{a}x\right) \right|^2 \right) dx = 1 \quad (2.18)$$

$$\iff B_n = + \sqrt{\frac{1}{\int_0^a \left(\left| \sin\left(\frac{n\pi}{a}x\right) \right|^2 \right) dx}}. \quad (2.19)$$

We don't actually need to bother evaluating this right now. What matters is that B_n takes a value that can be evaluated.

Definition 2.1: Superposition of States

As we noted in the chapter on the pre-requisite mathematics, when we considered linear differential equations if two functions are a solution to a homogeneous linear differential equation, then any linear combination (basically any sum of the two of them, each multiplied by some coefficient) is also a solution to that same differential equation. Such solutions are known as superpositions and correspond to a genuine quantum phenomena: that a system can exist in a combination of different classical states, when not being measured.

I note that, within the well, the Schrodinger equation of the particle is a homogeneous linear differential equation, so any combination of found solutions to the Schrodinger equation would also satisfy the Schrodinger equation. This new solution would have to be normalized - so the probabilities still sum to one - but this does not prevent these superposition solutions from existing.

2.2 Entering Classically Forbidden Regions: Finite Square Well

Making our first example a little more physical, we now define a new potential that is zero within the zero-potential region and takes a specific real - generally unspecified, as its exact value is immaterial - value, V_0 , everywhere else.

$$V(x) = \begin{cases} V_0 \in \mathbb{R}, & \text{if } -a \leq x \leq a, \\ \infty, & \text{otherwise} \end{cases}. \quad (2.20)$$

Is the particle still trapped within the zero-potential region? Examining the TISE within the square well (where $V(x) = 0$), we derive:

$$\psi(x) = A \cos(kx) + B \sin(kx), \quad (2.21)$$

but by the symmetry of the situation, about $x = 0$, we reason that $B = 0$. We then examine the schrodinger equation within the regions of potential V_0 , considering the cases where the energy, E , is less than V_0 (i.e. $E - V_0 < 0$):

$$\frac{\hbar^2}{2m} \frac{d^2\psi}{dx^2} = (V_0 - E)\psi. \quad (2.22)$$

Letting $k' = \frac{\sqrt{2m(V_0 - E)}}{\hbar}$, this has the solution:

$$\psi(x) = Ce^{k'x} + De^{-k'x}, \quad (2.23)$$

where C and D are - as yet - unknown values. If we impose the condition that infinitely far from the well the probability of finding the particle is zero (which is reasonable), then the overall solution of the wave function is:

$$\psi(x) = \begin{cases} B \cos(kx) & \text{if } -a \leq x \leq a \\ C e^{k'x} & \text{if } -a > x \\ D e^{-k'x} & \text{if } a < x \end{cases}. \quad (2.24)$$

Again we evaluate the values of the coefficients using the boundary conditions, which are now different. What we do is known as wavefunction matching; basically joining up all of the functions within the different regions at the known points they meet. The conditions for doing this are:

- The wavefunction must be continuous everywhere.
- The derivative of the wavefunction must be continuous everywhere.

Therefore, considering matching the wavefunctions at $x = a$, the conditions are:

$$A \cos(ka) = C e^{-k'a}, \quad (2.25)$$

$$-Ak \sin(ka) = -Ck' e^{-k'a}. \quad (2.26)$$

The interesting thing about satisfying these two equations simultaneously is that it is not possible for all values of k and k' . This is the root of quantisation! Dividing the latter of the above by the former:

$$\frac{A \cos(ka)}{-Ak \sin(ka)} = \frac{C e^{-k'a}}{-Ck' e^{-k'a}} \iff \frac{\cos(ka)}{k \sin(ka)} = \frac{1}{k'} \iff \tan(ka) = \frac{k'}{k}. \quad (2.27)$$

I do not actually much care about matching the wavefunctions. The interesting thing here is the quantisation. Recall that the values of k and k' are defined in terms of energies. Hence, we can rewrite the above as:

$$\tan\left(\sqrt{2mE}a/\hbar\right) = \frac{\sqrt{2m(V_0 - E)}/\hbar}{\sqrt{2mE}/\hbar} \iff \tan\left(\sqrt{2mE}a/\hbar\right) = \sqrt{\frac{V_0 - E}{E}}. \quad (2.28)$$

This is actually a pain to solve analytically so we are far better off attempting to solve this numerically (I'm not actually going to as I can use this as an excuse to use approximations). We can rearrange to make the equation slightly nicer, by letting $x = \sqrt{E}$ and choosing units such that $\hbar = 1$,

$$\tan\left(\sqrt{2m}ax\right) = \sqrt{\frac{V_0}{x^2} - 1}, \quad (2.29)$$

but to solve this numerically we have to choose a particular value of m , the particle's mass, a , the radius of the well, and V_0 , the value of the potential outside the region where it is zero. We don't actually care what values they take; this example is merely illustrative, so just set

- $a = 1$,
- $m = \frac{1}{2}$,
- $V_0 = 1$.

This gives reduces the equation to solve to:

$$\tan(x) = \sqrt{\frac{1}{x^2} - 1}. \quad (2.30)$$

Therefore, $x \approx 0.739085133215161\dots$ is the first solution but there are infinitely many.

Definition 2.2: Tunnelling

Tunnelling is the phenomena in quantum mechanics where a particle appears where it theoretically can be but should not have been able to get to due to energy barriers. We say the particle has tunnelled through the barrier.

2.3 A Textbook Confinement: Quantum Harmonic Oscillator

The quantum harmonic oscillator is defined as a system of a single particle in a potential:

$$V(x) = \frac{1}{2}qx^2, \quad (2.31)$$

where q is some real, positive value characterizing the exact potential. Therefore, q can be expressed as $m\omega^2$, where m is the particle's mass and ω is an arbitrary value - for now - that makes the equivalence to q true. Then consider the Shrodinger equation of this system:

$$-\frac{\hbar^2}{2m} \frac{d^2\psi}{dx^2} + \frac{1}{2}m\omega^2 x^2 \psi = E\psi. \quad (2.32)$$

To make life easier, I define a new function¹, ϕ , by:

$$\phi(x) = \psi(x\sqrt{m\omega/\hbar}). \quad (2.33)$$

A further simplification comes from defining a change of variables:

$$y = x\sqrt{m\omega/\hbar}. \quad (2.34)$$

The key consequence of both these transforms is that:

$$\frac{d^2\psi}{dx^2} = \frac{m\omega}{\hbar} \frac{d^2\phi}{dy^2}. \quad (2.35)$$

¹which is just a rescaling of the wavefunction

Using these results in the Schrodinger equation - and re-expressing E as $\frac{\hbar\omega}{2}\epsilon$ gives:

$$-\frac{\hbar^2}{2m} \frac{m\omega}{\hbar} \frac{d^2}{dy^2} \left(\phi(y) \right) + \frac{1}{2} m \omega^2 \frac{y^2}{m\omega/\hbar} \phi(y) = \frac{\hbar\omega}{2} \epsilon \phi(y) \quad (2.36)$$

$$\Rightarrow -\frac{\hbar\omega}{2} \frac{d^2}{dy^2} \left(\phi(y) \right) + \frac{\hbar\omega}{2} y^2 \phi(y) = \frac{\hbar\omega}{2} \epsilon \phi(y) \quad (2.37)$$

$$\Rightarrow -\frac{d^2}{dy^2} \left(\phi(y) \right) + y^2 \phi(y) = \epsilon \phi(y) \quad (2.38)$$

I then seek to solve this differential equation with the ansatz: $f(y)e^{-y^2/2}$. Substituting this into the Schrodinger equation:

$$-\frac{d^2}{dy^2} \left(f(y)e^{-y^2/2} \right) + y^2 f(y)e^{-y^2/2} = \epsilon f(y)e^{-y^2/2} \quad (2.39)$$

$$\iff -\left(f''(y)e^{-y^2/2} - yf'(y)e^{-y^2/2} - f(y)e^{-y^2/2} - yf'(y)e^{-y^2/2} + y^2 f(y)e^{-y^2/2} \right) \quad (2.40)$$

$$+ y^2 f(y)e^{-y^2/2} = \epsilon f(y)e^{-y^2/2} \quad (2.41)$$

$$\iff \left(f''(y) - 2yf'(y) + (\epsilon - 1)f(y) \right) e^{-y^2/2} = 0 \quad (2.42)$$

$$\iff f''(y) - 2yf'(y) + (\epsilon - 1)f(y) = 0. \quad (2.43)$$

I then turn to a purely mathematical result, using earlier derived properties of the Hermite polynomials.

Theorem 2.3: Solution to a Differential Equation

The differential equation:

$$f''(y) - 2yf'(y) + (\epsilon - 1)f(y) = 0 \quad (2.44)$$

is solved by $H_n(y)$ if $\epsilon = 2n + 1$.

Proof for Theorem.

Substituting the claimed solution into the differential equation gives:

$$H_n''(y) - 2yH_n'(y) + (\epsilon - 1)H_n(y) = 0. \quad (2.45)$$

Using Theorem 7.1, this can be re-written as:

$$4n(n-1)H_{n-2}(y) - 4ynH_{n-1}(y) + (\epsilon - 1)H_n(y) = 0 \quad (2.46)$$

$$\iff 2n \left(2(n-1)H_{n-2}(y) - 2yH_{n-1}(y) \right) + (\epsilon - 1)H_n(y) = 0. \quad (2.47)$$

Using Theorem 7.1, this reduces to:

$$-2nH_n(y) + (\epsilon - 1)H_n(y) = 0 \iff (\epsilon - (2n + 1))H_n(y) = 0. \quad (2.48)$$

Then the assumption that $\epsilon = 2n + 1$ implies that this final equation is true. ■

I note that if $\epsilon \neq 2n + 1$ then the differential equation has no solution that tends to zero at $\pm\infty$ as we require². So then the solution to the problem of finding the wavefunctions is:

$$\phi(y) = H_n(y)e^{-y^2/2}. \quad (2.49)$$

All that remains to do is undo the substitutions that were made to simplify the problem:

$$\psi(y\sqrt{m\omega/\hbar})_n = C'_n H_n(y)e^{-y^2/2} \Rightarrow \psi_n(x) = C_n H_n\left(\frac{x}{\sqrt{m\omega/\hbar}}\right)e^{-x^2 m\omega/2\hbar}, \quad (2.50)$$

where C_n and C'_n are normalization coefficients. I have not explicitly calculated a full wavefunction yet so I fell like I owe you one. I'll calculate the two lowest energy states.

Lowest Energy State

The lowest energy state of a system is often called its ground state. The ground state is:

$$\psi_0(x) = C_0 H_0\left(\frac{x}{\sqrt{m\omega/\hbar}}\right)e^{-x^2 m\omega/2\hbar} = C_0 e^{-x^2 m\omega/2\hbar}. \quad (2.51)$$

I find C_0 by imposing the normalization condition (using the same substitution as before):

$$1 = \int_{-\infty}^{\infty} \left(|\psi(x)|^2\right) dx = C_0^2 \int_{-\infty}^{\infty} \left(e^{-x^2 m\omega/2\hbar}\right) dx = C_0^2 \sqrt{\hbar/m\omega} \int_{-\infty}^{\infty} \left(e^{-y^2}\right) dy \quad (2.52)$$

Fortunately, the integral at the end of the above equation is a well known integral, known as the Gaussian integral. It is known to equate to $\sqrt{\pi}$. Therefore,

$$1 = C_0^2 \sqrt{\hbar/m\omega} \sqrt{\pi} \Rightarrow C_0 = \left(\frac{m\omega}{\hbar\pi}\right)^{1/4}. \quad (2.53)$$

Therefore, the wavefunction of the ground state of the quantum harmonic oscillator is:

$$\psi_0(x) = \left(\frac{m\omega}{\hbar\pi}\right)^{1/4} e^{-x^2 m\omega/2\hbar}. \quad (2.54)$$

The First Excited State

The first excited state has the wavefunction:

$$\psi_1(x) = C_1 H_1\left(\frac{x}{\sqrt{m\omega/\hbar}}\right)e^{-x^2 m\omega/2\hbar} = C_1 \frac{2x}{\sqrt{m\omega/\hbar}} e^{-x^2 m\omega/2\hbar}. \quad (2.55)$$

Then applying the normalization condition and absorbing a factor of $\frac{2}{m\omega/\hbar}$ into C_1^2 , for convenience:

$$1 = \int_{-\infty}^{\infty} \left(C_1^2 x^2 e^{-x^2 m\omega/\hbar}\right) dx = C_1^2 \int_{-\infty}^{\infty} \left(x^2 e^{-x^2 m\omega/\hbar}\right) dx. \quad (2.56)$$

²Which allows for the total probability to be finite.

Using another known solution of an integral:

$$1 = C_1^2 \int_{-\infty}^{\infty} \left(x^2 e^{-x^2 m\omega/\hbar} \right) dx = C_1^2 \frac{\sqrt{\pi}}{2(m\omega/\hbar)^{3/2}} \quad (2.57)$$

$$\Rightarrow C_1^2 = \frac{2(m\omega/2\hbar)^{3/2}}{\sqrt{\pi}} = \left(4 \frac{m^3 \omega^3}{\hbar^3 \pi} \right)^{1/2} \Rightarrow C_1 = \left(4 \frac{m^3 \omega^3}{\hbar^3 \pi} \right)^{1/4}. \quad (2.58)$$

Therefore, the wavefunction of the first excited state is:

$$\psi_1(x) = \left(4 \frac{m^3 \omega^3}{\hbar^3 \pi} \right)^{1/4} x e^{-x^2 m\omega/2\hbar} \quad (2.59)$$

3 Practical Usage: Using Wells as Qubits

I now present the first instance of practically using quantum phenomena, by examining what is a quantum computer at its most basic level. The core component is a quantum version of a bit. We will go into more detail about what a bit is later, but for now just consider a qubit to be a quantum object with exactly two states. We can build a qubit - effectively - by considering only the two lowest energy states of a physical system. Then, consider two of the harmonic potential wells side by side. Assume that they are far enough apart and the potentials are steep enough that the probability of the particle in one oscillator tunnelling to the other is negligible. So how to encode two states - importantly with the same energy - in this system?

First, isolate the system so the overall energy it contains is completely fixed. Then define one state, that we denote $|0\rangle$ as one of the oscillators being in the first excited state and the other being in the ground state. Define the other state, which we denote as $|1\rangle$ as the other oscillator being the one in the excited state.

Due to the isolation, there are exactly two states that the system can be in.

Note the notation we have use to denote the two states. Each has full wavefunctions that we could easily write down - and in fact, already know due to the above - but this notation - known as Dirac notation - is more convenient. And all the properties of the state can be derived from the known wavefunction as required.

4 Setting Up Treating Wavefunctions as Vectors

In the immediately preceding section, we saw that the state of some systems, such as the qubit we constructed, can be represented by objects such as $|0\rangle$ and $|1\rangle$. In fact these two objects are vectors and so any state of a two-dimensional system can be represented as a vector of the appropriate size. We can then use matrices to apply operations, time evolutions, and measurements to them.

CHAPTER 3

INTRODUCTION TO QUANTUM MECHANICS AS LINEAR ALGEBRA

1	Classical Systems in a Convenient Formalism	50
2	Quantum Systems in the Same Formalism	51
3	Operators and Time Evolutions in Quantum Mechanics .	53
4	Measurement	54
5	From Dirac Notation to Density Matrices	55
1	Components of a Quantum Computation in the Circuit Model	59
2	Commonly Used Gates	62
1	Trace Estimation Algorithm .	65
2	Simulation of Quantum Systems	67
3	Simulating Fermion Systems: Transforms to spin Systems .	71
1	Basics of Classical and Quantum Complexity Theory . . .	84
1	Introduction To Noise and Error	91
2	A Background on CPTP Maps	96
3	A Characterisation of Error .	99

1 Classical Systems in a Convenient Formalism

First, consider a classical system that can be in any of n_l states. I index each possible state from 1 to n_l and label each of those states as $|b_j\rangle$, where j is the index of the states. I can then define a set, \mathcal{H}_{class} , containing all possible states of the classical system i.e. $\mathcal{H}_{class} = \{|b_j\rangle | 1 \leq j \leq n_l\}$.¹

A good example is a set of n_s switches (e.g. light switches), then each of the possible states, i.e. $|b_j\rangle$, is defined by a particular combination of each of the n_s switches being on or off (e.g. if $n_s = 2$, $|b_1\rangle$ is the state where the first switch is off and the second switch is on). Therefore, the state of the system can be completely described by giving the index of the state, for example, $|b_k\rangle \in \mathcal{H}_{class}$, that the set of switches are in. Setting this example aside for a second, an important subset of classical systems is when there are only two possible states of the system, i.e. $|\mathcal{H}_{class}| = 2$, then I term such a system a bit.

Definition 1.1

A bit is a system with exactly two possible states.

For bits, it is conventional to label the two possible states $|0\rangle$ and $|1\rangle$.

In the earlier example about switches, each individual switch, in isolation, can be considered as a bit, as it can be either on or off. Therefore having two states.

This gives a convenient way to look at $\mathcal{H}_{class}^{switches}$ (the set of all states of n_s switches): each state in $\mathcal{H}_{class}^{switches}$ consists of a single choice from each of the possible states of each switch. More mathematically and concretely, indexing the switches and letting \mathcal{H}_j^{single} be the set of states for a single switch (with index j). Then, using the labelling convention, mentioned above, for the states of a bit (which each individual switch can be considered as):

$$\mathcal{H}_j^{single} = \{|0_j\rangle, |1_j\rangle\}, \quad (3.1)$$

where the j inside the bracket denotes which qubit the state refers to. This allows $\mathcal{H}_{class}^{switches}$ to be written as the tensor product of these single switch sets (\mathcal{H}_j^{single}):

$$\mathcal{H}_{class}^{switches} = \bigotimes_{j=1}^{n_s} \left(\mathcal{H}_j^{single} \right). \quad (3.2)$$

This then implies that the size of $\mathcal{H}_{class}^{switches}$ is $|\mathcal{H}_{class}^{switches}| = 2^{n_s}$. The final component of the classical systems to consider is that there exists a function of the system's state that returns the energy of the system when in that state, referred to as the Hamiltonian.

Definition 1.2

A classical Hamiltonian is a function from the set of possible states of the corresponding system to \mathbb{R} , that returns the energy of the input state of the system.

¹This presentation of states is known as bracket or Dirac notation: any state, in this notation, is denoted as $|label\rangle$, where *label* is the unique identifier / “name” of the state.

2 Quantum Systems in the Same Formalism

I can then expand the set of possible states of the system, to form the state space of a quantum system, by expanding the set of possible states into a vector space over the field of complex numbers. This vector space is determined by being the span of the set of classical states considered before, subject to a normalisation condition ².

Definition 2.1

For a system that can be in a superposition of the basis states^a $\{|b_j\rangle | 1 \leq j \leq n_l\}$, i.e. for any state of the system, $|\psi\rangle$, such that:

$$|\psi\rangle = \sum_{j=1}^{n_l} \left(\alpha_j |b_j\rangle \right), \quad (3.3)$$

where $\forall |b_j\rangle \in \{|b_j\rangle | 1 \leq j \leq n_l\}$, $\alpha_j \in \mathbb{C}$.

The normalisation condition, followed by all quantum systems, is:

$$\sum_{j=1}^{n_l} \left(|\alpha_j|^2 \right) = 1. \quad (3.4)$$

^aA basis state is the element of a set such that any state of the system can be expressed as a linear combination - with complex coefficients - of elements in the basis set. The set is known as the basis.

So the set of possible states of a single quantum bit (a qubit) is the set:

$$\left\{ \alpha_0 |0\rangle + \alpha_1 |1\rangle \mid \alpha_0, \alpha_1 \in \mathbb{C} \text{ such that } |\alpha_0|^2 + |\alpha_1|^2 = 1 \right\}, \quad (3.5)$$

where a qubit is defined as in Definition 2.

Definition 2.2

A qubit is a quantum system with two basis states^a.

^aA set of basis states of a quantum is a set such that every state the system could be in is expressible as a linear combination (with complex coefficients) of basis states i.e. the set of possible states of the system is the span of the basis states.

States that are not a basis state of this vector space but still in the vector space, are said to be in a superposition. Though, in fact, the term superposition is always defined relative to a basis and no basis has a particular claim to being *the* basis.

As this vector space also admits an inner product that defines a metric on the vector space and is complete, it can also be referred to as a Hilbert space and is often referred to as *the* Hilbert space of the system.

²This normalization condition is a consequence of the Born rule (which governs measurement in quantum systems and is defined in Definition 4) and the requirement for the probabilities each measurement outcome to always sum to one.

Definition 2.3

A Hilbert space is a real or complex inner product space that is a complete metric space with respect to the metric induced by the inner product. For the purposes of this thesis, the Hilbert space of a quantum system can just be considered as the set of all possible states (including superpositions) that the quantum system being considered can be in.

For a single qubit - as the set of possible states of a quantum system can be considered as the span of the possible states of the corresponding classical system - its set of possible states, i.e. its Hilbert space, follows from Eqn. (3.1) and can be expressed as:

$$\mathcal{H}_j^{single, quantum} = Span\{|0\rangle_j, |1\rangle_j\}. \quad (3.6)$$

Returning to the example of n_s switches, each switch's two states in the classical system become two basis states - as shown in Eqn. (3.6). Each switch's Hilbert space is then defined as the Hilbert space spanned by the two basis states, subject to the normalisation condition (i.e. Definition 2).

For the whole system of n_s switches, considered as a quantum system, the Hilbert space, $\mathcal{H}_{quantum}^{switches}$, is the tensor product of the Hilbert space of each switch (as the tensor product of two Hilbert spaces can be considered as the Hilbert space of their respective systems combined into a single system), analogously to Eqn. (3.2) i.e.

$$\mathcal{H}_{quantum}^{switches} = \otimes_{j=1}^{n_s} \left(\mathcal{H}_j^{single, quantum} \right). \quad (3.7)$$

For example, if there are two switches (i.e. $n_s = 2$), the Hilbert space of the two switches - considered as a single system - is:

$$Span(|0\rangle_1 \otimes |0\rangle_2, |0\rangle_1 \otimes |1\rangle_2, |1\rangle_1 \otimes |0\rangle_2, |1\rangle_1 \otimes |1\rangle_2), \quad (3.8)$$

i.e any state of the two qubits can be expressed as:

$$\alpha_{00}|0\rangle_1 \otimes |0\rangle_2 + \alpha_{01}|0\rangle_1 \otimes |1\rangle_2 + \alpha_{10}|1\rangle_1 \otimes |0\rangle_2 + \alpha_{11}|1\rangle_1 \otimes |1\rangle_2, \quad (3.9)$$

where each $\alpha_{j,k}$ is a complex number such that $|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$. It is worth noting that expressing states as formally and fully as $|0\rangle_1 \otimes |0\rangle_2$ is quite a laborious notation and so, for example, it is quite common - both in this thesis and more widely - to express this state as $|0\rangle_1|0\rangle_2$ - omitting the tensor product symbol - or, going even further to be concise, as $|00\rangle$. I note that any choice of four orthogonal states in the Hilbert space of two qubits can be used to express any state as in Eqn. 3.9. This particular choice is known as the computational basis and may be extended to any number - $N \in \mathbb{N}$ - qubits as the N -fold tensor product of elements in $\{|0\rangle, |1\rangle\}$

In the quantum setting, the Hamiltonian, which was a function for classical systems, is upgraded to an operator (defined in next section) and for that reason I suspend a full description of quantum Hamiltonians until the next section.

3 Operators and Time Evolutions in Quantum Mechanics

Most simply, operators are things that do something to states of the system. Mathematically, I define them, factoring in the requirements of quantum mechanics, as linear maps from the system under consideration's Hilbert space onto itself.

An operator, \hat{A} , (that acts on a Hilbert space, \mathcal{H}) is said to be linear if, for any states, $|\psi\rangle, |\phi\rangle \in \mathcal{H}$; and $\alpha_\phi, \alpha_\psi \in \mathbb{C}$:

$$\hat{A}(\alpha_\psi|\psi\rangle + \alpha_\phi|\phi\rangle) = \alpha_\psi\hat{A}|\psi\rangle + \alpha_\phi\hat{A}|\phi\rangle. \quad (3.10)$$

For ease of "bookkeeping," I denote the set of all operators on a given Hilbert space, \mathcal{H} , by $\mathbb{B}(\mathcal{H})$.

The most important operator for any quantum system is its Hamiltonian: in moving from a classical to a quantum system, the Hamiltonian is "upgraded" to a Hermitian operator so it can now act on states. I define the Hamiltonian to map an energy basis state (eigenstate of the Hamiltonian) to itself multiplied by its energy, a real value (so then the energy is an eigenvalue of the Hamiltonian).

Defining the eigenstates (ensuring they form an orthogonal basis of the Hilbert space) and eigenvalues uniquely defines the Hamiltonian (up to physically irrelevant variations).

Definition 3.1

The Hamiltonian of a quantum system is a Hermitian operator such that the eigenstates of the Hamiltonian are the stationary states (i.e. the states such that a system in that state does not change - except for an irrelevant overall phase - with time) of the system, and the corresponding eigenvalues are the energy of the respective stationary states.

A large part of the centrality of a system's Hamiltonian is its role in the system's equation of motion. However, there is no short, simple derivation of the governing equation for the dynamics of quantum systems. The dynamics were observed experimentally and the justification for the Schrödinger equation, presented below, is its good agreement with experiments. For any quantum system with Hamiltonian, \hat{H} , if $|\psi\rangle$ represents the state of the system, the Schrödinger equation is:

$$i\hbar \frac{d|\psi\rangle}{dt} = \hat{H}|\psi\rangle, \quad (3.11)$$

where $\hbar = \frac{6.62607015 \times 10^{-34}}{2\pi} \text{ kg m}^2 \text{ s}^{-1}$ and is called the reduced Planck constant. But, because theorists are too cool to write out long numbers, I use "God-given" units for the rest of this thesis, that allows me to set $\hbar = 1$.

For my purposes, it is more convenient to derive an operator that applies the time evolution for a specified amount of time, rather than solve the Schrödinger equation every time. To be clear, I seek an operator, \hat{U}_t , such that, for any $t \in \mathbb{R}$, if $|\psi(0)\rangle$ is an arbitrary state at

an arbitrary time and $|\psi(t)\rangle$ is the result of evolving that state for time t under the relevant Hamiltonian:

$$|\psi(t)\rangle = \hat{U}_t |\psi(0)\rangle. \quad (3.12)$$

To find such an operator, I start from the Schrödinger equation (Eqn. (3.11)) and derive:

$$\left(i \frac{d}{dt} - \hat{H}\right) |\psi(t)\rangle = 0 \iff \left(i \frac{d}{dt} - \hat{H}\right) \hat{U}_t |\psi(0)\rangle = 0. \quad (3.13)$$

Then, as this must hold for any choice of initial state, $|\psi(0)\rangle$, I require that:

$$\left(i \frac{d}{dt} - \hat{H}\right) \hat{U}_t = 0. \quad (3.14)$$

This is a differential equation of a known, standard form and so the correct solution can easily be seen but, for completeness, I check that the solution $\hat{U}_t = e^{-i\hat{H}t}$ is correct:

$$\left(i \frac{d}{dt} - \hat{H}\right) e^{-i\hat{H}t} = \hat{H} e^{-i\hat{H}t} - \hat{H} e^{-i\hat{H}t} = 0. \quad (3.15)$$

Likewise, any multiple of \hat{U}_t is a solution by the same argument as the above. As the Hamiltonian is defined to be Hermitian, this form of the time evolution operator, \hat{U}_t , means that time evolutions, for any time period, t , are applied by unitary operators. I.e. allowing the time evolution for any length of time, according to any Hermitian Hamiltonian, will apply a unitary operator. This will be important later.

A final important operator is the identity. The identity applied to a state or system represents doing nothing to it. It is denoted \hat{I} .

Sometimes when the aim is to represent doing nothing to a sub-part, S_{sub} of a system, the identity (on that sub-part only) is represented by $I_{S_{\text{sub}}}$. Though occasionally the subscript is also omitted.

4 Measurement

It is necessary to start with a definition of what can be observed, before the process of observing it can be considered.

Definition 4.1

In quantum mechanics, an observable is a quantity that can be measured. It is represented by a Hermitian matrix.

Like was the case for the Schrödinger equation, the rules for measurement cannot be derived even if given the Schrödinger equation. So, based on experimental observation, I assert the Born rule Born, “Zur Quantenmechanik der Stoßvorgänge” (as in Definition 4) to govern measurement outcomes in quantum mechanics:

Definition 4.2

For any observable with matrix representation, A , if that observable is measured, the outcome will be an eigenvalue of A . Additionally, for a system in the state $|\psi\rangle$, the probability of measuring the eigenvalue, λ , is $\langle\psi|\hat{\Pi}_\lambda|\psi\rangle$, where $\hat{\Pi}_\lambda$ is the projector into the eigenspace of λ . This is known as the Born rule.

5 From Dirac Notation to Density Matrices

While Dirac notation (the style of quantum mechanics constructed so far) functions perfectly well, it does have the unfortunate habit of treating quantum and classical probability separately. Meaning if I want to consider both simultaneously (and mixing them can be advantageous), I would have to manually handle the classical probabilities. This could quickly become onerous. So re-consider the previously established mechanics to handle quantum and classical probability on the same footing.

The fundamental hurdle to this is the Born rule. So I need some way to re-express quantum states (and how operators act on them) such that quantum (via the Born rule) and classical probabilities are managed similarly.

In Dirac notation, for any state, $|\psi\rangle = \sum_{j=1}^{n_l} (\alpha_j |b_j\rangle)$ (where $\{|b_j\rangle\}_{j=1}^{n_l}$ is an orthonormal basis), the probability of measuring a given basis state³, $|b_k\rangle$, is:

$$|\langle b_k|\psi\rangle|^2 = \left| \sum_{j=1}^{n_l} \left(\alpha_j \langle b_k|b_j\rangle \right) \right|^2 = |\alpha_k|^2. \quad (3.16)$$

Inspired by this, let $|\psi\rangle$ be any state from the previous formalism. In the new formalism, the same state would be represented by:

$$|\psi\rangle\langle\psi|, \quad (3.17)$$

which, considering the state $|\psi\rangle$ as a vector⁴, allows us to consider $|\psi\rangle\langle\psi|$ as a matrix. Call this a density matrix, which is generally denoted as ρ . This means that a state in an equal superposition of the measurement basis states $|b_1\rangle\langle b_1|$ and $|b_2\rangle\langle b_2|$, would be represented as:

$$\frac{1}{2}|\psi_1\rangle\langle\psi_1| + \frac{1}{2}|\psi_2\rangle\langle\psi_2| + \frac{1}{2}|\psi_1\rangle\langle\psi_2| + \frac{1}{2}|\psi_2\rangle\langle\psi_1|. \quad (3.18)$$

In this case, the probability of measuring each basis states is $\frac{1}{2}$.

Similarly, if there is (classical) uncertainty about which state a system is in I can just sum the probabilities multiplied by the relevant density matrix. e.g. if there is an equal, classical probability of the state $|\psi_1\rangle\langle\psi_1|$ or $|\psi_2\rangle\langle\psi_2|$ being prepared, the system can be considered to be in the state:

$$\frac{1}{2}|\psi_1\rangle\langle\psi_1| + \frac{1}{2}|\psi_2\rangle\langle\psi_2|. \quad (3.19)$$

³when performing a measurement that projects into one of the basis states.

⁴and $\langle\psi|$ is the Hermitian conjugate of $|\psi\rangle$.

This has the nice benefit that the probability of measuring each basis state is the coefficient of that state's density matrix (i.e. $|\psi_1\rangle\langle\psi_1|$ or $|\psi_2\rangle\langle\psi_2|$) in the state's representation in Eqn. (3.19). Therefore it is easy to see (in Eqn. (3.18) and Eqn. (3.19)) that classical uncertainties and superpositions are represented on an equal footing. Each is represented in the coefficients of the density matrices (e.g. $|\psi_1\rangle\langle\psi_1|$ or $|\psi_1\rangle\langle\psi_2|$) that sum to give the density matrix representing the state.

As a technicality, this new formalism does not represent the set of all states of a system as a Hilbert space. Instead note that density matrices can be viewed as operators (as they are matrices and hence must represent linear operators on a corresponding vector space, decided by the matrix size and the field the matrices are over) that act on the Hilbert space of the system (the same system but in the old formalism). This set of density matrices is exactly the set of valid linear operators, $\mathbb{B}(\mathcal{H})$, on the Hilbert space of the old formalism.

I now present the application of operators to density matrices. For this, assume the aim is to perform the equivalent of (in the old formalism) applying the operator \hat{U} to the state $|\psi\rangle$.

It is easy to see how this is done by defining the state $|\phi\rangle = \hat{U}|\psi\rangle$ and equating the density matrix representations of the two states:

$$|\phi\rangle\langle\phi| = \hat{U}|\psi\rangle\langle\psi|\hat{U}^\dagger. \quad (3.20)$$

It is therefore simple to conclude that for any operator, \hat{U} , and any density matrix, ρ , I apply the operator as:

$$\hat{U}\rho\hat{U}^\dagger. \quad (3.21)$$

The one outstanding element of density matrices left to cover is measurement. In the density matrix formalism, a measurement is defined via a set of projection operators (with the condition they project into orthogonal subspaces). Denote the projector corresponding to each outcome measurement, α , as $\hat{\Pi}_\alpha$.

$\hat{\Pi}_\alpha$ projects any state into the subspace defined by any state in that subspace having the property such that the quantity measured by the measurement takes the value α i.e. any state in the subspace would give measurement outcome α if measured.

Then the probability of getting outcome α from the measurement (on ρ) is:

$$Prob(\text{measuring } \alpha) = \text{Tr}\left(\hat{\Pi}_\alpha\rho\right). \quad (3.22)$$

The state of the system after the measurement is the state the projectors project it into, but normalised:

$$\rho' = \frac{\hat{\Pi}_\alpha\rho\hat{\Pi}_\alpha^\dagger}{\text{Tr}(\hat{\Pi}_\alpha\rho\hat{\Pi}_\alpha^\dagger)} = \frac{\hat{\Pi}_\alpha\rho\hat{\Pi}_\alpha^\dagger}{\text{Tr}(\hat{\Pi}_\alpha^\dagger\hat{\Pi}_\alpha\rho)}. \quad (3.23)$$

Definition 5.1

While density matrices are more general than Dirac notation, a fair fraction of the time they just mean all gates / time evolutions / have to be written out twice. I therefore reserve the right to use either Dirac notation or density matrices depending on which is convenient.

A more general notion of measurement that that presented above comes from Positive Operator Valued Measures (POVMs) defined in Definition 5. This notion of measurement will not be used in considerations of quantum systems or when designing protocols/algorithms but can be used for considering more general measurements in proofs.

Definition 5.1

A Positive Operator Valued Measure (POVM) is a set, $\{\hat{E}_i\}_i$, of positive semi-definite Hermitian operators such that:

$$\sum_i \left(\hat{E}_i \right) = \hat{I}. \quad (3.24)$$

Each element of the set, \hat{E}_i , corresponds to a measurement outcome, indexed by i , and the probability of that outcome is:

$$\text{Tr} \left(\hat{E}_i \rho \right), \quad (3.25)$$

where ρ is a density matrix the measurement is being performed on.

CHAPTER 4

BASIC QUANTUM COMPUTING

As the first line of every quantum computing paper tells us, quantum computers promise to radically transform what is computationally possible. Quantum computing was originally conceived of via the quantum Turing machine, but this is unwieldy for practical or even theoretical usage and overly abstract for algorithm design.

An alternative, and yet still universal D. Deutsch, “[Quantum Theory, the ChurchTuring principle and the Universal Quantum Computer](#)”, approach is the circuit model.

1 Components of a Quantum Computation in the Circuit Model

In brief, a quantum computation is executed (in the circuit model), by preparing a number of qubits in specific states, applying operations to those qubits, and then measuring the qubits. During this computation, the qubits may experience quantum effects e.g. being entangled with each other and being in superpositions; giving rise to the expected greater computational power of quantum computers. It is worth pausing to consider each component of the circuit model.

1.1 Wires

The basis of the circuit model is that qubits are represented by wires (i.e. horizontal lines), with each qubit represented by a single wire. The flow of time is then represented by moving rightward along each wire: the same distance along different wires represents the same point in time for different qubits. A trivial example of a wire is given in Fig. 4.1. Fig. 4.1 is not



Figure 4.1: Example of a single wire, representing a single qubit. This represents nothing happening to the qubit.

the most exciting figure, but I'll add each component of an interesting circuit to Fig. 4.1 as it is introduced.

1.2 State Preparation

This is the initial step in the circuit model and consists of taking the required number of qubits from unknown, arbitrary states (as, in general, you're unaware of what state the qubits were used for before you started using the computer) and preparing them in a known, specified (by the algorithm being executed) state. Typically, this is a tensor product of single-qubit gates. In the circuit model, the state each qubit is prepared in is shown to the left of the left-most end of the wire (representing the earliest point in time of the qubit represented by the wire), as in Fig. 4.2.



Figure 4.2: Example circuit of a single wire, representing a single qubit prepared in the state $|0\rangle$ and then left.

1.3 Gates / Operations

The representation of the operations applied to qubits in the course of the circuit is via gates. These are defined in Definition 1.3.

Definition 1.1

A gate depicts an operation on a set of qubits, with each qubit depicted by both an input wire (representing the qubit before the operation) and an output wire (representing each qubit after the operation). For an example of a gate see Fig. 4.3, which has input wires on the left and output wires on the right.

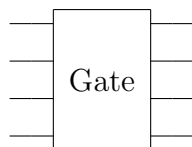


Figure 4.3: Example gate

These gates are physically applied via the time evolution of a specific Hamiltonian, for a specific duration, to produce the intended gate. For this reason, due to reasons discussed in Sec. 3, all gates appearing in a circuit must be unitary.

An important aspect of the application of operations to a qubit, and hence the application of gates to wires, is denoting which qubits an operator is applied on and in what order the operations are performed. The first point is conveyed by which wires a gate acts on; the second is conveyed by the left-most gates being applied first, followed by the next left-most. So the order gates are applied follows the representation of the flow of time in the wires.

In light of this, for a circuit to make sense, gates are required to be time-wise connected, as defined in Definition 1.3.

Definition 1.2

Two gates are said to be time-wise connected if no output wire of one connects to an output wire of the other or itself, and no input wire of one connects to an input wire of the other or itself. For example see Fig. 4.4, which features two copies of the gate in Fig. 4.3 connected time-wise.

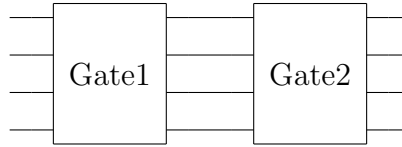


Figure 4.4: Two time-wise connected gates, representing the operations represented by the two gates applied sequentially, progressing from left to right.

Example gates can be added to the example circuit developed in Fig. 4.1 and Fig. 4.2, and this is shown in Fig. 4.5.

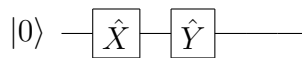


Figure 4.5: Example circuit of a single wire, representing a single qubit prepared in the state $|0\rangle$, followed by the application of a Pauli \hat{X} gate then a Pauli \hat{Y} gate.

Typically, a physical quantum device is restricted to applying a finite number of gates, by the physics of how it applies gates usually. The set of all possible gates on a given device is known as the gateset. Some gatesets are known to be ‘universal,’ meaning all efficiently implementable circuits can be expressed using only gates from the gateset.

1.4 Measurement

The final element of a circuit is the measurements. These provide the results of a circuit, and do exactly as the name suggests. The measurements of a circuit are the last thing to happen to their respective qubits and, as such, are represented by ‘D’s at the end of a wire, as in Fig. 4.6. A measurement is applied, typically, to a single qubit, though not all qubits are

required to be measured. Any wire in a circuit that just ends abruptly can be disregarded past that point. It is not measured or considered any further.

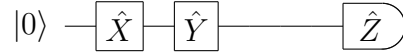


Figure 4.6: Example circuit of a single wire, representing a single qubit prepared in the state $|0\rangle$, followed by the application of a Pauli \hat{X} gate then a Pauli \hat{Y} gate, and then measurement in the Pauli \hat{Z} basis.

2 Commonly Used Gates

There are some gates that must be presented for use in this thesis. I stick to a few, well-worn, and commonly used gates. The full characterization of these gates follows from their matrices, provided in the tables below: the single-qubit gates are shown in Table. 4.7 and the two qubit gates shown in Table. 4.8.

Any other gates used are denoted by their operator, as defined, being contained in a gate.

Some gates form groups, with specific properties, two such sets that will continually reappear are the Pauli gates and the Clifford gates.

Definition 2.1

Define \mathbb{P}_1 as the set $\{\hat{I}, \hat{X}, \hat{Y}, \hat{Z}\}$. Where \hat{I} is the identity on a single qubit and $\hat{X}, \hat{Y}, \hat{Z}$ are the single-qubit Pauli operators.

Similarly, $\mathbb{P}_1^{\otimes N}$ denotes the set of operators defined by a single element of \mathbb{P}_1 acting on each of N qubits (the elements of \mathbb{P}_1 acting on different qubits can be different).

Clifford gates were originally defined in Ref. Gottesman, “[Theory of Fault-Tolerant Quantum Computation](#)” with a different definition but, for my purposes, the following suffices:

Definition 2.2

A two-qubit gate, $\hat{\mathcal{M}}$, is defined to be Clifford if $\forall \hat{x} \in \mathbb{P}_1^{\otimes 2}$,

$$\hat{\mathcal{M}}^\dagger \hat{x} \hat{\mathcal{M}} \in \mathbb{P}_1^{\otimes 2}. \quad (4.1)$$

I will sometimes refer to this as two-qubit Clifford gates normalising $\mathbb{P}_1^{\otimes 2}$.

Gate Name	Circuit Depiction	Matrix Representation
Identity	$\text{---} \boxed{\hat{I}} \text{---}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
Pauli X	$\text{---} \boxed{\hat{X}} \text{---}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
Pauli Y	$\text{---} \boxed{\hat{Y}} \text{---}$	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$
Pauli Z	$\text{---} \boxed{\hat{Z}} \text{---}$	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
Hadamard	$\text{---} \boxed{H} \text{---}$	$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
Phase	$\text{---} \boxed{\hat{S}} \text{---}$	$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$
Inverse Phase	$\text{---} \boxed{\hat{S}^\dagger} \text{---}$	$\begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix}$
T	$\text{---} \boxed{\hat{T}} \text{---}$	$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$
$R_x(\theta)$	$\text{---} \boxed{\hat{R}_x(\theta)} \text{---} \text{ or } \text{---} \boxed{e^{-i\hat{X}\theta/2}} \text{---}$	$\begin{pmatrix} \cos(\theta/2) & -i \sin(\theta/2) \\ -i \sin(\theta/2) & \cos(\theta/2) \end{pmatrix}$
$R_z(\theta)$	$\text{---} \boxed{\hat{R}_z(\theta)} \text{---} \text{ or } \text{---} \boxed{e^{-i\hat{Z}\theta/2}} \text{---}$	$\begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{-i\theta/2} \end{pmatrix}$
Arbitrary Gate	$\text{---} \boxed{\hat{U}} \text{---}$	$\begin{pmatrix} U_{1,1} & U_{1,2} \\ U_{2,1} & U_{2,2} \end{pmatrix}$

Table 1.1: A table of the names, circuit depictions, and matrix representations (in the basis $\{|0\rangle, |1\rangle\}$) of commonly used single-qubit gates. Note that the Hadamard gate is denoted as H , *without a hat*, as opposed to Hamiltonians which will always be written with a hat. $\theta \in [0, 2\pi]$ is a parameter that specifies a particular gate in a family of gates.

Figure 4.7: Table of Common Single-Qubit Gates

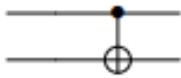

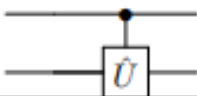
Gate Name	Circuit Depiction	Matrix Representation
controlled-X / CNOT / $c\hat{X}$		$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
controlled-Z / $c\hat{Z}$		$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$
controlled- \hat{U} / $c\hat{U}$		$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & U_{1,1} & U_{1,2} \\ 0 & 0 & U_{2,1} & U_{2,2} \end{pmatrix}$

Table 1.2: A table of the names, circuit depictions, and matrix representations (in the basis $\{|0\rangle|0\rangle, |0\rangle|1\rangle, |1\rangle|0\rangle, |1\rangle|1\rangle\}$) of commonly used two-qubit gates. Note that \hat{U} can be any single-qubit gate.

Figure 4.8: Table of Common Two-Qubit Gates

CHAPTER 5

BASIC EXAMPLE QUANTUM ALGORITHMS

1 Trace Estimation Algorithm

The key part of the DQC1 trace estimation algorithm is the circuit presented in Figure 5.1. Consider the algorithm in the density matrix formalism. Then the initial state is:

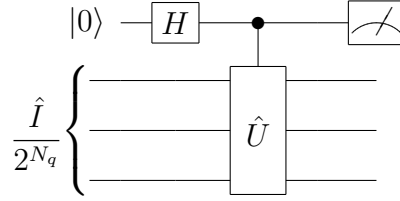


Figure 5.1: DQC1 circuit for estimating $\text{Tr}(U)/2^{N_q}$ for a N_q -qubit unitary \hat{U} . Its real and imaginary parts are obtained by measuring in the X and Z Pauli basis' respectively on the first qubit Knill and Laflamme, “[Power of One Bit of Quantum Information](#)”.

$$\rho_{\text{init}} = |0\rangle\langle 0| \otimes \frac{\hat{I}}{2^{N_q}}. \quad (5.1)$$

Therefore the state of the qubits immediately before the measurement is:

$$\rho_{\text{preM}} = (\hat{cU}) H \rho_{\text{init}} H^\dagger (\hat{cU})^\dagger = \frac{1}{2^{N_q+1}} (\hat{cU}) \left[(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| + |1\rangle\langle 1|) \otimes \hat{I} \right] (\hat{cU})^\dagger \quad (5.2)$$

$$= \frac{1}{2^{N_q+1}} \left(|0\rangle\langle 0| \otimes \hat{I} + |0\rangle\langle 1| \otimes U^\dagger + |1\rangle\langle 0| \otimes U + |1\rangle\langle 1| \otimes \hat{I} \right). \quad (5.3)$$

The probability of measuring $|+\rangle\langle+|$ is then:

$$\frac{1}{2^{N_q+1}} \text{Tr} \left[(|+\rangle\langle+| \otimes \hat{I}) \left(|0\rangle\langle 0| \otimes \hat{I} + |0\rangle\langle 1| \otimes U^\dagger + |1\rangle\langle 0| \otimes U + |1\rangle\langle 1| \otimes \hat{I} \right) \right] \quad (5.4)$$

$$= \frac{1}{2^{N_q+1}} \text{Tr}(\hat{I}) + \frac{1}{2^{N_q+2}} \text{Tr}(\hat{U}^\dagger) + \frac{1}{2^{N_q+2}} \text{Tr}(\hat{U}) = \frac{1}{2} + \frac{\text{Re}[\text{Tr}(\hat{U})]}{2^{N_q+1}}. \quad (5.5)$$

The imaginary component of $\text{Tr}(\hat{U})$ can be obtained using the same methods and a similar circuit. This implies that by repeated measurement of the output of the circuit in Fig. 5.1 (or its counterpart for the imaginary component), an estimate of the probability of measuring $|1\rangle\langle 1|$ can be obtained and hence an estimate of $\frac{\text{Tr}(\hat{U})}{2^{N_q}}$ can be obtained. Managing the error due to estimating the probability of measuring $|1\rangle\langle 1|$ imperfectly is handled in Lemma 3.

Lemma 3. *For any desired absolute error, ϵ_A , and confidence, c , the empirical mean of i.i.d.¹ random variables bounded between 0 and 1 can be estimated to within the desired error - with the required confidence - with $\mathcal{O}\left(\frac{1}{\epsilon_A^2}\right)$ samples.*

Proof. Let,

- $\tilde{S} = \frac{1}{n}(s_1 + s_2 + \dots + s_n)$,

- $\langle \tilde{S} \rangle$ be the expected value of \tilde{S}

where each s_j is an independent random variable and each s_j is bounded within $[a_j, b_j]$ then Hoeffding's inequality states:

$$\mathbb{P}(|\tilde{S} - \langle \tilde{S} \rangle| \geq \epsilon_A) \leq 2 \exp \left\{ - \frac{2n^2 \epsilon_A^2}{\sum_{j=1}^n [(b_j - a_j)^2]} \right\}. \quad (5.6)$$

So in this case, where each s_j is a measurement outcome, $\forall 1 \leq j \leq n$, $b_j = 1$, $a_j = 0$,

$$\mathbb{P}(|\tilde{S} - \langle \tilde{S} \rangle| \geq \epsilon_A) \leq 2 \exp \left(- \frac{2n^2 \epsilon_A^2}{n} \right) = 2 \exp \left(- 2n \epsilon_A^2 \right). \quad (5.7)$$

If I then set $\mathbb{P}(|\tilde{S} - \langle \tilde{S} \rangle| \leq \epsilon_A)$ to the desired confidence, c , then I can calculate the required number of samples: set $\mathbb{P}(|\tilde{S} - \langle \tilde{S} \rangle| \geq \epsilon_A) = 1 - c$, then:

$$1 - c \geq 2 \exp \left(- 2n \epsilon_A^2 \right) \Rightarrow - \frac{\ln \left(\frac{1 - c}{2} \right)}{2 \epsilon_A^2} \leq n. \quad (5.8)$$

Therefore,

$$n = \mathcal{O} \left(\frac{1}{\epsilon_A^2} \right). \quad (5.9)$$

□

¹i.i.d. stands for 'identical, independently distributed.'

2 Simulation of Quantum Systems

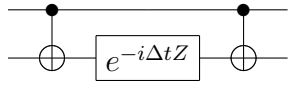
The basis of the simulation method employed in this analysis is based on the following gadget. Consider the Hamiltonian:

$$\mathcal{H} = Z_1 \otimes Z_2 \otimes Z_3 \otimes \dots \otimes Z_N$$

Hence the time evolution operator of this Hamiltonian is:

$$\mathcal{U} = e^{-i\mathcal{H}t} = e^{-i(Z_1 \otimes Z_2 \otimes Z_3 \otimes \dots \otimes Z_N)t}$$

The following gadget implements the time evolution according to the Hamiltonian, $\mathcal{H} = Z_1 \otimes Z_2$.



Proof. Considering the unitary corresponding to the circuit:

$$\begin{aligned} \mathcal{U} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \left(\hat{\mathcal{I}} \otimes \begin{bmatrix} e^{-i\Delta t} & 0 \\ 0 & e^{i\Delta t} \end{bmatrix} \right) \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} e^{-i\Delta t} & 0 & 0 & 0 \\ 0 & e^{i\Delta t} & 0 & 0 \\ 0 & 0 & e^{-i\Delta t} & 0 \\ 0 & 0 & 0 & e^{i\Delta t} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} e^{-i\Delta t} & 0 & 0 & 0 \\ 0 & e^{i\Delta t} & 0 & 0 \\ 0 & 0 & e^{i\Delta t} & 0 \\ 0 & 0 & 0 & e^{-i\Delta t} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{aligned}$$

Then, for the purposes of comparison, consider the Hamiltonian we wish to apply: $e^{-i\mathcal{H}t}$. Expressing the Hamiltonian in question as a matrix:

$$\mathcal{H} = Z_1 \otimes Z_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Therefore,

$$-i\mathcal{H}\Delta t = \begin{bmatrix} -i\Delta t & 0 & 0 & 0 \\ 0 & i\Delta t & 0 & 0 \\ 0 & 0 & i\Delta t & 0 \\ 0 & 0 & 0 & -\Delta t \end{bmatrix}$$

Exponentiating this: DONE WRONG

$$e^{-i\mathcal{H}\Delta t} = \begin{bmatrix} e^{-i\Delta t} & 0 & 0 & 0 \\ 0 & e^{i\Delta t} & 0 & 0 \\ 0 & 0 & e^{i\Delta t} & 0 \\ 0 & 0 & 0 & e^{-i\Delta t} \end{bmatrix}$$

□

This can be done with as many qubits as you like inductively. SHOW THIS INDUCTIVELY, I THINK I'VE SHOWN THIS BEFORE.

Using the gadget for the time evolution of:

$$\mathcal{H} = Z_1 \otimes Z_2 \otimes Z_3 \otimes \dots \otimes Z_N$$

The time evolution of any Hamiltonian consisting of Pauli gates (as below) may be implemented through the addition of single qubit gates.

$$\mathcal{H} = \mathbb{P}_1 \otimes \mathbb{P}_2 \otimes \dots \otimes \mathbb{P}_n$$

Where \mathbb{P}_i are Pauli operators.

Proof. Consider the time evolution operator:

$$e^{-i(\mathbb{P}_1 \otimes \mathbb{P}_2 \otimes \dots \otimes \mathbb{P}_n)\Delta t} = \sum_{k=0}^{\infty} \left(\frac{(-i(\mathbb{P}_1 \otimes \mathbb{P}_2 \otimes \dots \otimes \mathbb{P}_n)\Delta t)^k}{k!} \right) = \sum_{k=0}^{\infty} \left(\frac{(-i\mathbb{P}_1\mathbb{P}_2 \dots \mathbb{P}_n\Delta t)^k}{k!} \right)$$

As each Pauli operator acts on a different qubit, they all commute, so we can express this as:

$$e^{-i\mathbb{P}_1\mathbb{P}_2 \dots \mathbb{P}_n\Delta t} = \sum_{k=0}^{\infty} \left(\frac{(-i\Delta t)^k (\mathbb{P}_1)^k (\mathbb{P}_2)^k \dots (\mathbb{P}_n)^k}{k!} \right)$$

To go further, we require the following lemma:

Lemma 4.

$$X = HZH \text{ and } Y = R_x(\frac{\pi}{2})ZR_x(\frac{-\pi}{2})$$

Proof.

$$HZH = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = X$$

and

$$R_x(\frac{\pi}{2})ZR_x(\frac{-\pi}{2}) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -i \\ -i & 1 \end{bmatrix} = Y$$

□

Using these derived formulae to express each \mathbb{P}_i in terms of Z gates sandwiched between two gates: Γ such that:

$$\mathbb{P}_i = \Gamma_i Z_i \Gamma_i^{-1}$$

The particular formulae for $\mathbb{P}_i = X, Y$ have been verified to be of this form. So we can rewrite the time evolution operator above as:

$$\begin{aligned} e^{-i\mathbb{P}_1\mathbb{P}_2\cdots\mathbb{P}_n\Delta t} &= \sum_{k=0}^{\infty} \left(\frac{(-i\Delta t)^k (\Gamma_1 Z_1 \Gamma_1^{-1})^k (\Gamma_2 Z_2 \Gamma_2^{-1})^k \cdots (\Gamma_n Z_n \Gamma_n^{-1})^k}{k!} \right) \\ &= \sum_{k=0}^{\infty} \left(\frac{(-i\Delta t)^k \Gamma_1 (Z_1)^k \Gamma_1^{-1} \Gamma_2 (Z_2)^k \Gamma_2^{-1} \cdots \Gamma_n (Z_n)^k \Gamma_n^{-1}}{k!} \right) \\ &= \sum_{k=0}^{\infty} \left(\Gamma_1 \Gamma_2 \cdots \Gamma_n \frac{(-i\Delta t)^k (Z_1)^k (Z_2)^k \cdots (Z_n)^k}{k!} \Gamma_1^{-1} \Gamma_2^{-1} \cdots \Gamma_n^{-1} \right) \\ &= \Gamma_1 \Gamma_2 \cdots \Gamma_n \sum_{k=0}^{\infty} \left(\frac{(-i\Delta t)^k (Z_1)^k (Z_2)^k \cdots (Z_n)^k}{k!} \right) \Gamma_1^{-1} \Gamma_2^{-1} \cdots \Gamma_n^{-1} \\ &= \Gamma_1 \Gamma_2 \cdots \Gamma_n \sum_{k=0}^{\infty} \left(\frac{(-iZ_1 Z_2 \cdots Z_n \Delta t)^k}{k!} \right) \Gamma_1^{-1} \Gamma_2^{-1} \cdots \Gamma_n^{-1} \\ &= \Gamma_1 \Gamma_2 \cdots \Gamma_n e^{-iZ_1 Z_2 \cdots Z_n \Delta t} \Gamma_1^{-1} \Gamma_2^{-1} \cdots \Gamma_n^{-1} \end{aligned}$$

Each Γ can be easily (and classically) identified and may be efficiently implemented by a single gate. \square

The question is when is a trotter decomposition viable?

Lemma 5.

A Trotter decomposition may be performed for any Hamiltonian where every term consists of no more than 1 Pauli acting on each qubit and some real coefficient.

Proof. The limiting factor on which Hamiltonians the Trotter decomposition can be used on and what parts it can be broken up into is that for every sub-Hamiltonian, \mathcal{S} , the operator $e^{-i\mathcal{S}t}$ must be implementable (at least approximately) on our quantum computer and therefore must be unitary.

This requires that \mathcal{S} be Hermitian.

Claim

If A and B are two commuting Hermitian matrices then AB is hermitian.

Proof.

$$AB = ([AB]^*)^* = (B^* A^*)^* = (BA)^* = (AB)^*$$

\square

All the Pauli matrices are Hermitian, as is the identity, and any two Pauli's that act on different qubits commute so the product of any two such Paulis is Hermitian. Then consider the product of k such Paulis:

$$x_1 \cdot x_2 \cdots x_k$$

Assume this is Hermitian and let x_A be a Pauli that acts on a different qubit to all the paulis so far. Then,

$$x_1 \cdot x_2 \cdots x_k \cdot x_A = x_1 \cdot x_2 \cdots x_A \cdot x_k = x_1 \cdot x_A \cdot x_2 \cdots x_k = x_A \cdot x_1 \cdot x_2 \cdots x_k$$

Therefore, x_A commutes with $x_1 \cdot x_2 \cdots x_k$ and both are Hermitian.

So $x_1 \cdot x_2 \cdots x_k \cdot x_A$ is Hermitian.

As this has already been shown to be true for $k = 2$ then by induction it is true for any number of Paulis. \square

First order trick

Lemma 6.

For a first order Trotter decomposition, the gadgets corresponding to each term may be performed in any order.

Proof. Consider a Hamiltonian, \mathcal{H} , that may be split into \mathcal{N} Hermitian sub-Hamiltonians:

$$\mathcal{H} = \sum_{i=1}^{\mathcal{N}} \left(\mathcal{S}_i \right)$$

Consider applying the first order Trotter decomposition to this Hamiltonian:

$$\left[\prod_{j=1}^{\mathcal{N}} \left(e^{\frac{-i\mathcal{S}_j t}{n}} \right) \right]^n$$

Then consider a permutation, $\mathcal{P} : \{1, 2, \dots, \mathcal{N}\} \Rightarrow \{1, 2, \dots, \mathcal{N}\}$, on the indices. Due to the commutative nature of addition,

$$\mathcal{H} = \sum_{i=1}^{\mathcal{N}} \left(\mathcal{S}_i \right) = \sum_{i=1}^{\mathcal{N}} \left(\mathcal{S}_{\mathcal{P}(i)} \right) \quad \forall \mathcal{P}$$

Therefore, for all possible \mathcal{P} ,

$$\left[\prod_{j=1}^{\mathcal{N}} \left(e^{\frac{-i\mathcal{S}_{\mathcal{P}(j)} t}{n}} \right) \right]^n$$

is an equally valid Trotter decomposition. It does not necessarily equate to the other Trotter decomposition but the same limits on its error apply. \square

3 Simulating Fermion Systems: Transforms to spin Systems

To consider the simulation of fermion systems we first look at a simple dimer.

DIMER IMAGE: I have it somewhere

The Hubbard dimer consists of two sites. Each site has two orbitals, one for each spin. Therefore the basis for the entire system is given in the table below

	0	\uparrow	\downarrow	$\uparrow\downarrow$
0	$ \vec{0}\rangle$	$C_{1,\uparrow}^\dagger \vec{0}\rangle$	$C_{1,\downarrow}^\dagger \vec{0}\rangle$	$C_{1,\uparrow}^\dagger C_{1,\downarrow}^\dagger \vec{0}\rangle$
\uparrow	$C_{2,\uparrow}^\dagger \vec{0}\rangle$	$C_{1,\uparrow}^\dagger C_{2,\uparrow}^\dagger \vec{0}\rangle$	$C_{1,\downarrow}^\dagger C_{2,\uparrow}^\dagger \vec{0}\rangle$	$C_{1,\uparrow}^\dagger C_{1,\downarrow}^\dagger C_{2,\uparrow}^\dagger \vec{0}\rangle$
\downarrow	$C_{2,\downarrow}^\dagger \vec{0}\rangle$	$C_{1,\uparrow}^\dagger C_{2,\downarrow}^\dagger \vec{0}\rangle$	$C_{1,\downarrow}^\dagger C_{2,\downarrow}^\dagger \vec{0}\rangle$	$C_{1,\uparrow}^\dagger C_{1,\downarrow}^\dagger C_{2,\downarrow}^\dagger \vec{0}\rangle$
$\uparrow\downarrow$	$C_{2,\downarrow}^\dagger C_{2,\uparrow}^\dagger \vec{0}\rangle$	$C_{1,\uparrow}^\dagger C_{2,\downarrow}^\dagger C_{2,\uparrow}^\dagger \vec{0}\rangle$	$C_{1,\downarrow}^\dagger C_{2,\downarrow}^\dagger C_{2,\uparrow}^\dagger \vec{0}\rangle$	$C_{1,\uparrow}^\dagger C_{1,\downarrow}^\dagger C_{2,\downarrow}^\dagger C_{2,\uparrow}^\dagger \vec{0}\rangle$

If we consider a system containing exactly one up spin electron and exactly one down spin electron. This then restricts the system to a subspace of Hilbert space. This subspace is closed under time evolution according to the Hubbard Hamiltonian (shown in appendix B) and this subspace has the basis:

	0	\uparrow	\downarrow	$\uparrow\downarrow$
0				$C_{1,\uparrow}^\dagger C_{1,\downarrow}^\dagger \vec{0}\rangle$
\uparrow			$C_{1,\downarrow}^\dagger C_{2,\uparrow}^\dagger \vec{0}\rangle$	
\downarrow		$C_{1,\uparrow}^\dagger C_{2,\downarrow}^\dagger \vec{0}\rangle$		
$\uparrow\downarrow$	$C_{2,\downarrow}^\dagger C_{2,\uparrow}^\dagger \vec{0}\rangle$			

Any two of the above states are orthonormal to each other

Proof. Consider the product of any two of these states:

$$\begin{aligned} \langle \vec{0} | C_{i,\sigma} C_{j,-\sigma} C_{k,\sigma}^\dagger C_{l,-\sigma}^\dagger | \vec{0} \rangle &= \langle \vec{0} | C_{i,\sigma} C_{k,\sigma}^\dagger C_{j,-\sigma} C_{l,-\sigma}^\dagger | \vec{0} \rangle = \langle \vec{0} | \left(\delta_{i,k} - C_{k,\sigma}^\dagger C_{i,\sigma} \right) \left(\delta_{j,l} - C_{l,-\sigma}^\dagger C_{j,-\sigma} \right) | \vec{0} \rangle \\ &= \delta_{i,k} \delta_{j,l} \langle \vec{0} | \vec{0} \rangle = \delta_{i,k} \delta_{j,l} \end{aligned}$$

Therefore, the inner two product of any two of the above states is one if the states are the same and zero otherwise. \square

Any state in the $\uparrow\downarrow$ subspace can clearly be decomposed into a linear combination of these states. In conjunction with the above theorem, this proves the given states form a basis.

This leaves just 4 basis states left that form a sub-basis for the sub-space of the half-filled system. It's worth noting that each configuration of n_\uparrow and n_\downarrow has it's own sub-basis formed from a diagonal of the basis states of the entire Hilbert space. So then any state in the half-filling sub-space may be expressed in the form:

$$|\psi\rangle = \alpha_{1,1} C_{1,\uparrow}^\dagger C_{1,\downarrow}^\dagger |\vec{0}\rangle + \alpha_{2,1} C_{2,\uparrow}^\dagger C_{1,\downarrow}^\dagger |\vec{0}\rangle + \alpha_{1,2} C_{1,\uparrow}^\dagger C_{2,\downarrow}^\dagger |\vec{0}\rangle + \alpha_{2,2} C_{2,\uparrow}^\dagger C_{2,\downarrow}^\dagger |\vec{0}\rangle$$

Basis State	Index
$C_{1,\uparrow}^\dagger C_{1,\downarrow}^\dagger \vec{0}\rangle$	1
$C_{1,\uparrow}^\dagger C_{2,\downarrow}^\dagger \vec{0}\rangle$	2
$C_{1,\downarrow}^\dagger C_{2,\uparrow}^\dagger \vec{0}\rangle$	3
$C_{2,\uparrow}^\dagger C_{2,\downarrow}^\dagger \vec{0}\rangle$	4

Where $\alpha \in \mathbb{C}$

For the purposes of forming the Hamiltonian matrix later it is useful to apply an indexing to the basis states and this is done in the table below:

The Hamiltonian

The dimer only has two nodes, therefore, the Hubbard Hamiltonian takes the form:

$$\mathcal{H} = C_{1,\uparrow}^\dagger C_{2,\uparrow} + C_{2,\uparrow}^\dagger C_{1,\uparrow} + C_{1,\downarrow}^\dagger C_{2,\downarrow} + C_{2,\downarrow}^\dagger C_{1,\downarrow} + u\hat{n}_{1,\uparrow}\hat{n}_{1,\downarrow} + u\hat{n}_{2,\uparrow}\hat{n}_{2,\downarrow}$$

In order to apply the transforms (more on them later), an ordering is required to be applied to the orbits of the fermionic system. To the orbital of location i with spin σ assign the index: $2i - \delta_{\sigma,\uparrow}$.

Resultantly, apply the following map to the C/A operators:

$$\begin{aligned} C_{i,\sigma}^\dagger &\longrightarrow a_{2i-\delta_{\sigma,\uparrow}}^\dagger \\ C_{i,\sigma} &\longrightarrow a_{2i-\delta_{\sigma,\uparrow}} \end{aligned}$$

This gives the modified Hamiltonian:

$$\mathcal{H} = a_1^\dagger a_3 + a_{2,\uparrow}^\dagger a_1 + a_2^\dagger a_4 + a_4^\dagger a_2 + u\hat{n}_1\hat{n}_2 + u\hat{n}_3\hat{n}_4$$

Under this indexing, the i th orbital can be directly related to the i th qubit. The only remaining question is how to relate the two system's lie algebra of operators such that they are isomorphic and hence can bisimulate. This is handled by encodings described below.

3.1 Jordan Wigner Encoding

Decide on the encoding to use going forward

Using the standard Jordan-Wigner transform formulae

$$a_j = \frac{1}{2} \left(X_j + iY_j \right) \otimes_{k=1}^{j-1} \left(Z_k \right)$$

Which additionally implies:

$$a_j^\dagger = \frac{1}{2} \left(X_j - iY_j \right) \otimes_{k=1}^{j-1} \left(Z_k \right)$$

As the Jordan-Wigner transform acts on each creation and annihilation operator individually, each term in a Hamiltonian can be considered on it's own. It is then useful to consider some common terms:

$$\begin{aligned} a_i^\dagger a_j &= \frac{1}{2} \left(X_i - iY_i \right) \otimes_{k=1}^{i-1} \left(Z_k \right) \frac{1}{2} \left(X_j + iY_j \right) \otimes_{k=1}^{j-1} \left(Z_k \right) \\ &= \frac{1}{4} \otimes_{k=i+1}^{j-1} \left(Z_k \right) \left(X_i - iY_i \right) \left(X_j + iY_j \right) \end{aligned}$$

Alternatively if $i > j$

$$a_i^\dagger a_j = \frac{1}{4} \otimes_{k=j+1}^{i-1} \left(Z_k \right) \left(X_j - iY_j \right) \left(X_i + iY_i \right)$$

Finally, if $i = j$:

$$\begin{aligned} n_j &= a_j^\dagger a_j = \frac{1}{4} \otimes_{k=j+1}^{j-1} \left(Z_k \right) \left(X_j - iY_j \right) \left(X_j + iY_j \right) = \frac{1}{4} \left(Z_k \right) \left(X_j - iY_j \right) \left(X_j + iY_j \right) \\ &= \frac{1}{4} \left(X_j^2 - iY_j X_j + iX_j Y_j + Y_j^2 \right) = \frac{1}{4} \left(2I_j + i \left[X_j, Y_j \right] \right) = \frac{1}{4} \left(2 + i(2iZ_j) \right) \\ &= \frac{1}{4} \left(2 - 2Z_j \right) = \frac{1}{2} \left(1 - Z_j \right) \end{aligned}$$

CHAPTER 6

FORMAL NOTIONS OF COMPUTING AND COMPLEXITY THEORY

In this section I present six modes of computation: deterministic classical computing, probabilistic classical computing, non-deterministic classical computing, universal quantum computing, one clean qubit quantum computing, and analogue quantum computing.

0.1 Deterministic Classical Computing

Basic Overview

The first step in this overview consists in defining exactly what “deterministic classical computing” means. By “deterministic” I mean that the computer will compute in an entirely predetermined way: it will not make any random choices, it will not - as is the case with non-deterministic computing - split into many hypothetical computations and return (overall) accept/yes if any of them return accept/yes. At any time, there is a single state of the computation that is entirely and non-randomly determined by the input to the computation and the algorithm to be executed. For any step in the computation, there is a single definite step that is performed. This term is perhaps best defined by what it is not: non-deterministic or random¹.

In the context of this thesis, “classical” basically means not quantum. A computation is classical if it is performed entirely according to the rules of classical, Newtonian mechanics. There is no superposition or interference present.

This mode of computation is perhaps the simplest, and is in fact the first that was defined or considered, and all subsequent modes of computing are defined - in some way or another - by how they are different from this mode. In particular, the formal models of later modes of computation are defined by how various aspects of their models differ from the formal model of deterministic classical computing: the deterministic Turing machine. Defined in the next section.

¹As defined later.

Formal Model Presentation

Models of various modes of computation are formally represented by Turing machines, but the origins of the Turing machine lie in defining deterministic classical computation, and in its most simple form that is the mode of computation it captures. Defined in 1936 by Alan Turing, “[On Computable Numbers, with an Application to the Entscheidungsproblem](#)”, the Turing machine is an abstract mathematical model of computation that describes a hypothetical device that solves decision problems (problems where the answer is exclusively either yes or no). A basic overview of it can be seen in Fig 6.1². The deterministic Turing

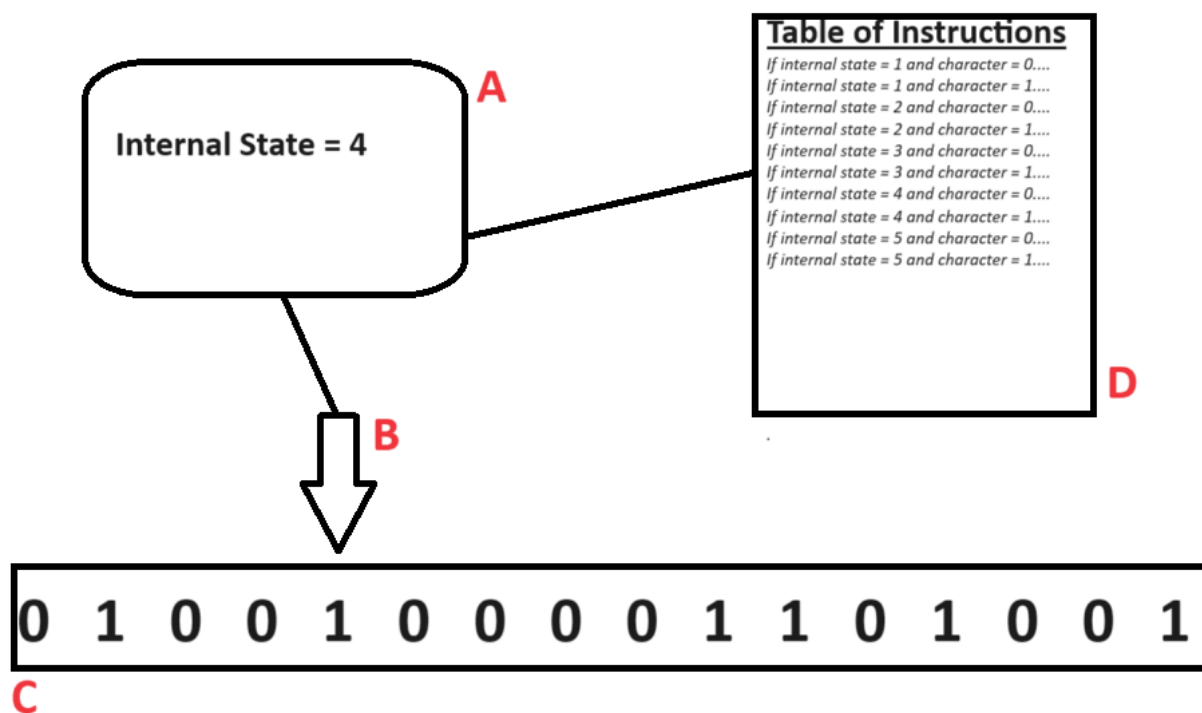


Figure 6.1: A labelled diagram of a Turing machine where: A is the control, keeping track of the internal state of the Turing machine; B is the head, it is over a single character on the tape at a time - but can move one character at a time - and can only read and rewrite the single character it is above; C is the tape, initially holding the input string but this is continually changed as the computation progresses; D is the table of instructions, representing the transition function of the Turing machine and dictating how the Turing machine performs each step.

machine, as depicted in Fig. 6.1, is formally defined in Def. 0.1.

²The author would like to dedicate this image to Hannah Leitch, for her - totally not sarcastic - support in his artistic endeavours.

Definition 0.1

A deterministic Turing machine can be defined by a 7-tuple: $(Q, A, \delta, \Gamma, F, \Sigma, q_0, b)$, where:

- Q is the set of all possible states that the Turing machine may be in
- A is the tape alphabet, the set of all symbols that can appear on the tape, its elements are called letters
- $F \subset Q$ is the set of accepting states. States not in F are referred to as rejecting states
- $\Sigma \in A \setminus \{b\}$ is the set of letters allowed in the input string
- $q_0 \in Q$ is the state that the Turing machine starts in
- $b \in A$ is the blank state, every cell holds this letter initially - unless it is one of the cells initials holding the input string - and until otherwise specified
- $\delta: (Q \setminus F) \times A \rightarrow Q \times A \times \{-1, 0, +1\}$ is the transition function, a function equivalent to the table of instructions. Given the current state of the Turing machine and the letter beneath the head, it outputs: the state the Turing machine changes to, the letter to be written in the cell beneath the head, and the change in the index of the head's position. It may also be represented by a transition table, which lists what to do in each of the possible situations that may occur. In fact this is a more tangible representation of the transition rules than an abstract function.

It's worth noting here that a Turing machines can only solve decision problems (problems where the answer is either yes or no) and it indicates it's decision based on whether it halts in an accepting state or not.

What This Model Represents

This model of computation models a "standard" classical computer: exactly what most people think of when you say "computer." Throughout this thesis, when I say classical computer, this model of computation is exactly what I am referring to. This model of computation is - at present - the dominant paradigm.

0.2 Probabilistic Classical Computing**Basic Overview**

The notion of probabilistic classical computing is very similar, and yet formally different, from deterministic computation. The idea is to slightly upgrade deterministic classical computers with the ability to make - genuinely - random decisions.

Formal Model Presentation

The formal model of probabilistic computing is the probabilistic Turing machine. Its definition is nearly identical to that of the deterministic classical Turing machine - so I will not repeat it here. Therefore, the probabilistic Turing machine can be depicted very similarly to

how the deterministic Turing machine is in Fig. 6.1, and I do so in Fig. 6.2³.

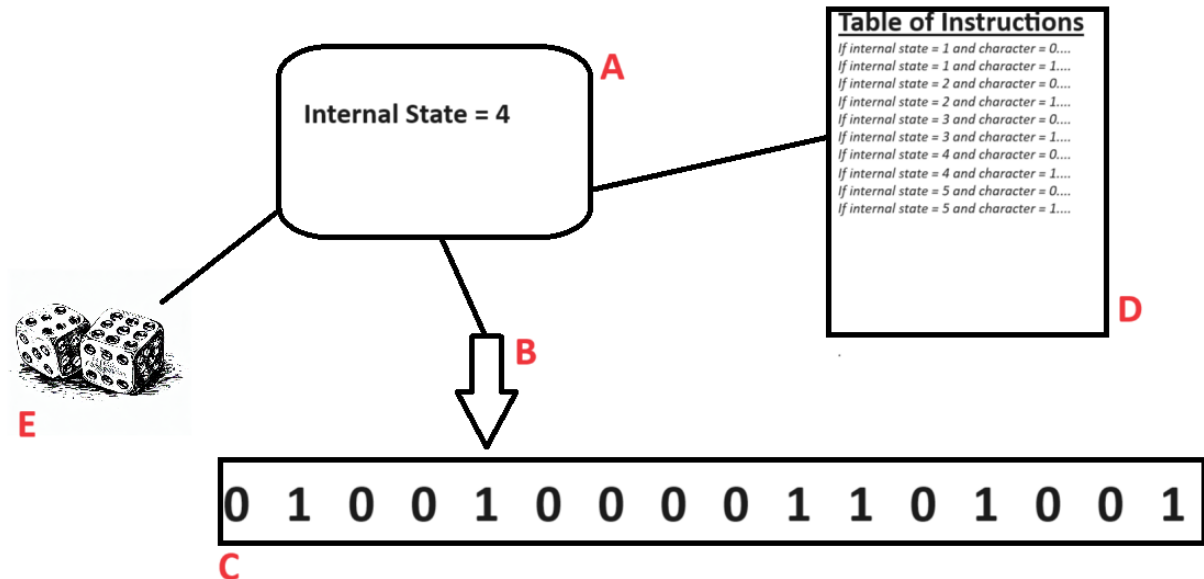


Figure 6.2: A labelled diagram of a Turing machine where: A is the control, keeping track of the internal state of the Turing machine; B is the head, it is over a single character on the tape at a time - but can move one character at a time - and can only read and rewrite the single character it is above; C is the tape, initially holding the input string but this is continually changed as the computation progresses; D is the table of instructions, representing the transition function of the Turing machine and dictating how the Turing machine performs each step; E is some method - such as dice, as depicted - , of making random choices.

The key difference between deterministic and probabilistic Turing machines lies in their respective transition functions (represented - in the deterministic Turing machine definition - as $\delta: (Q \setminus F) \times A \rightarrow X$, where $X \subset Q \times A \times \{-1, 0, +1\}$). For a deterministic Turing machine this is a deterministic map, telling the Turing machine what to do with certainty and if given the same input will return exactly the same output. In a probabilistic Turing machine, the transition function can output a random - according to some probability distribution, with a dependence on any part of the input to the transition function - instruction. This allows the probabilistic Turing machine to act randomly when required. A important thing to note is that the probabilistic Turing machine is still classical and no quantum affects are present in the computation.

What This Model Represents

While the ability to make random choices is very useful in modern computing e.g. for cryptography, in practice genuine randomness is not used in actual computers, as this is very

³Due to my lack of art skills, please note that the dice appearing in this image were generated by DALL-E via Microsoft paint

hard to obtain, and a pseudorandom generator⁴ is typically used in the hope that “good enough” numbers can be obtained for whatever purpose - however this is not known to be true or false. Therefore the notion of a probabilistic Turing machine is used as it has not been proven that these pseudorandom generators can adequately simulate probabilistic Turing machines and, even if it could, it saves a lot of time and effort to just assume the random choices can be made rather than explicitly build the pseudorandom generators into a deterministic Turing machine.

0.3 Non-Deterministic Classical Computing

Basic Overview

Non-deterministic classical computing is the most un-physical model of computing presented and considered in this thesis. It is not supposed to model any *real* method of computation: its purpose is mostly for comparison with other models of computation or considering aspects of computer science other than actual computations to solve problems. The key aspect it is used to investigate is verifiability, that is if there exists a certificate - i.e. a string of characters from a valid alphabet - that enables a deterministic classical Turing machine to verify that any instance of a decision problem has been correctly decided, then there is a non-deterministic classical Turing machine that can decide the same decision problem. The idea of non-determinism is that a non-deterministic Turing machine takes every choice at what could otherwise have been a random decision, and then accepts if *any* of the choices ends with the Turing machine in an accepting state. In this sense, it can be seen as a Turing machine accepts any input where there is any chance of it accepting if it were to be probabilistic. The series of choices that the probabilistic Turing machine would have to have made can then be seen as the certificate to enable the deterministic classical Turing machine verify the answer.

Formal Model Presentation

In line with the above description of non-determinism, I depict the non-deterministic as in Fig. 6.3. I then present a more formal definition of non-deterministic classical Turing machines, in Def. 0.3.

⁴Meaning a deterministic attempt to emulate genuine randomness while not actually being random itself, a key example is the Mersenne Twister Matsumoto and T. Nishimura, “[Mersenne twister: a 623-Dimensionally Equidistributed Uniform Pseudo-random Number Generator](#)” which is the default way of generating “random” numbers in many programming languages including Python, Julia (until Julia 1.6 LTS), Ruby, and R.

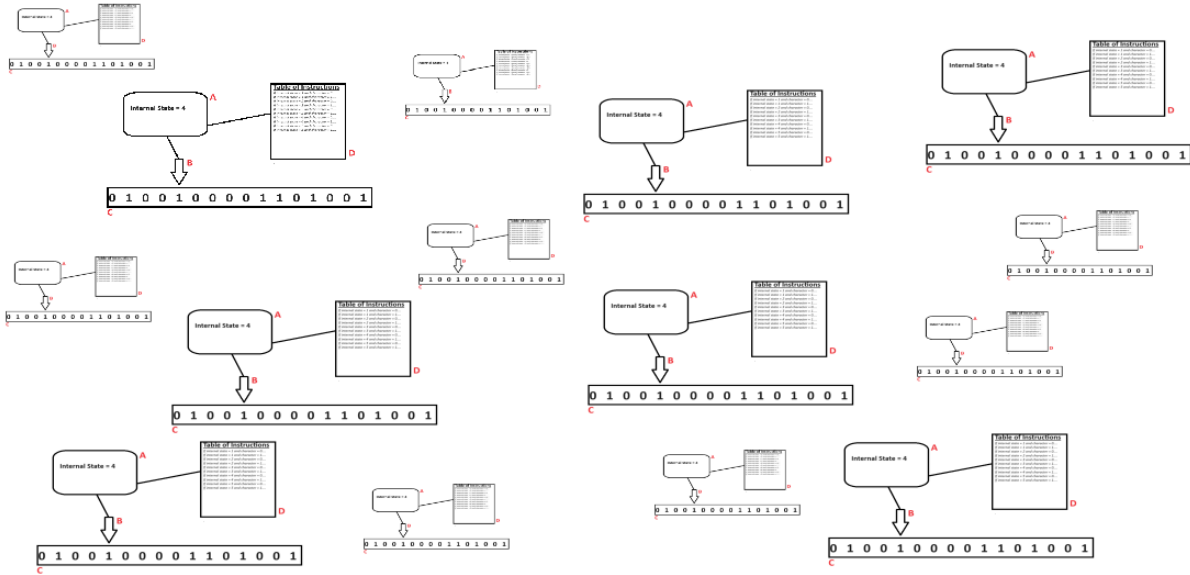


Figure 6.3: A labelled diagram of a non-deterministic Turing machine where the computation can be seen as splitting into many computations of a deterministic classical Turing machine - as depicted in Fig. 6.1 - where the non-deterministic Turing machine accepts if any of the collection of deterministic classical Turing machines do.

Definition 0.2

A non-deterministic Turing machine, like the deterministic Turing machine, can be defined by a 7-tuple: $(Q, A, \delta, \Gamma, F, \Sigma, q_0, b)$, where:

- Q is the set of all possible states that the Turing machine may be in
- A is the "tape alphabet," the set of all symbols that can appear on the tape, its elements are called letters
- $F \subset Q$ is the set of accepting states. States not in F are referred to as rejecting states
- $\Sigma \in A \setminus \{b\}$ is the set of letters allowed in the input string
- $q_0 \in Q$ is the state that the Turing machine starts in
- $b \in A$ is the blank state, every cell holds this letter unless/until otherwise specified
- $\delta: (Q \setminus F) \times A \rightarrow X$, where $X \subset Q \times A \times \{-1, 0, +1\}$, is a many-to-one map referred to as the transition function. Given the current state of the Turing machine and the letter beneath the head, it outputs many possible courses of action for the Turing machine.

This definition is identical to that of deterministic Turing machines until the definition of δ where many different possible actions are given. The non-deterministic Turing machine can be viewed as producing many different copies of itself at each step and carrying out all the possible actions. It is defined to accept an input string if one or more of the copies created halts in an accepting state.

What This Model Represents

Non-deterministic classical computation - as I mentioned before - is not representing a physical model of computation. What this model of computation aims to capture is the verification of computations. That is, checking that a provided answer is correct. I.e., if a problem can be checked with a specific amount of time using a deterministic classical Turing machine, then it can be solved - in the same amount of time - by a non-deterministic classical Turing machine.

0.4 Universal Quantum Computing

Basic Overview

Quantum computing is introduced in far more technical detail in Sec. 4. Here I will simply mention that quantum computing is comparable to the deterministic classical Turing machine, but while the deterministic classical computer operates according to classical mechanics a quantum computer operates according to quantum mechanics. The particular mode of quantum computing currently being considered is universal quantum computing. It is called universal because it can perform any task that any quantum computer is capable of - though not necessarily as quickly, or with less error. This term will become clearer after the later comparisons with other models of quantum computing.

Formal Model Presentation

The initial model of quantum computing was the quantum Turing machine David Deutsch and Penrose, “Quantum theory, the ChurchTuring principle and the universal quantum computer”. However, while useful for early results, this proved to be too unwieldy for practical usage. Fortunately, Lemma 7 allows us to avoid having to use quantum Turing machines and instead use the circuit model of quantum computing, which is introduced in far more detail in Sec. 4. A very simple example of a quantum computing circuit is given in Fig. 6.4⁵, to provide a basic idea of what’s being discussed in this section.

Lemma 7 (in Ref. David Deutsch and Penrose, “Quantum theory, the ChurchTuring principle and the universal quantum computer”). *Any decision problem decidable by a quantum Turing machine in polynomial (in the input size) time - with at least a $2/3$ probability of success - is also decidable by a uniform family of circuits, each using at most a polynomial (in the input size) number of gates - also with at least a $2/3$ probability of success.*

What This Model Represents

The quantum computing model of computing represents a range of hardware where the bits/qubits performing a computation are able to operate according to quantum mechanics. These devices are still in their infancy but are developing quickly. This model represents what many people would think of when they think of quantum computing. However it is

⁵This is the exact example, in Fig. 4.6, that will be constructed piece by piece in the full presentation of the circuit model.

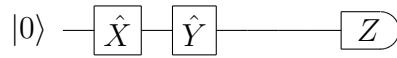


Figure 6.4: Example circuit of a single wire, representing a single qubit prepared in the state $|0\rangle$, followed by the application of a Pauli \hat{X} gate then a Pauli \hat{Y} gate, and then measurement in the Pauli \hat{Z} basis.

not the only model of quantum computing, it is the model sometimes referred to as digital or gate-based quantum computing and hence excludes models such as the one clean qubit model or analogue quantum computing.

0.5 One Clean Qubit Quantum Computing

Basic Overview

The one clean qubit model of quantum computing is a model of quantum computing inspired by NMR quantum computing J. A. Jones, “[Quantum Computing with NMR](#)”; Morimae, Fujii, and H. Nishimura, “[Power of One Non-Clean Qubit](#)”. The key difference between one clean qubit quantum computing and universal quantum computing is that in the one clean qubit model only a single qubit can be prepared in a definite pre-decided state: all other qubits are prepared in either $|0\rangle$ or $|1\rangle$ with equal probability. I.e., the difference between universal and one clean qubit quantum computing is that the one clean qubit model is much more restricted in the initial states it can prepare.

Formal Model Presentation

The formal model of the one clean qubit quantum computing is completely analogous to that of universal quantum computing except that all but one of the qubits are initially prepared in a uniformly random product of $|0\rangle$ and $|1\rangle$ states. The textbook algorithm that uses this model of computation is the trace estimation algorithm Knill and Laflamme, “[Power of One Bit of Quantum Information](#)”, which uses a circuit as depicted in Fig. 6.5⁶:

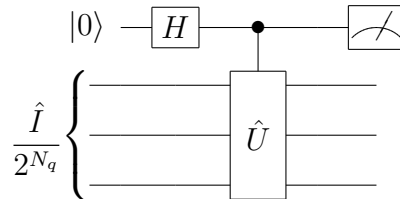


Figure 6.5: Example computation in the one clean qubit model. This circuit is used in the trace estimation algorithm to estimate the trace of an operator, \hat{U}

⁶This figure is also Fig. 5.1 and will be examined in more detail in Sec. ??.

What This Model Represents

As mentioned before, the one clean qubit model of quantum computing was initially inspired by NMR quantum computing and what was easily implementable on that kind of hardware. In this thesis, it is used as a model of quantum computing that is distinct from universal quantum computing. Universal quantum computing can emulate the one clean qubit model so the universal model is more powerful than the one clean qubit model but it is not known if the one clean qubit model can emulate the universal model. Although it is expected that it cannot. Therefore, if an algorithm can be demonstrated to function in the one clean qubit model, there are complexity theoretic implications but also it shows that the algorithm does not require the full power of quantum computing.

0.6 Analogue and Hybrid Quantum Computing

Basic Overview

Analogue quantum computing is a model of quantum computing that does not even try to be as widely applicable as universal quantum computing. It is strictly limited to modelling the behaviour of quantum systems. In general, it can be used to emulate a whole host of quantum systems and can investigate many properties. However, in this thesis, I will exclusively consider using analogue quantum computers to simulate the time evolutions of quantum systems. Despite its limits on what it can do, an analogue quantum simulator is redeemed by how well it does it. Typically an analogue quantum computer/simulator can simulate systems with much less noise - i.e. much more accurately - than can be done with a digital device, and can simulate much larger systems.

Formal Model Presentation

In this thesis, I model analogue quantum computers using something akin to the circuit model. However, this model differs from the circuit model used for the universal quantum computer as different operators can be applied to the system. In analogue quantum computing the only operators that can be applied are time evolutions according to specific Hamiltonians. Within this thesis I also consider - and define - hybrid analogue simulators, where an analogue quantum simulator is augmented with the ability to perform digital gates (as are found in universal quantum computing). This model again uses circuits to present computations - and is very similar to the model used for analogue quantum computers but with the addition of digital gates to the circuit.

What This Model Represents

The model of analogue quantum computing represents a range of hardware where certain Hamiltonians can be physically imitated. Most notably cold atoms in an optical lattice are used. The hybrid model represents a similar setup but in situations where companies/ research groups have additionally added the ability to apply digital gates to their system⁷. This

⁷I don't believe this is currently, publicly accessible but I know a few companies and groups are working towards this.

gives the best of both worlds: the simulation abilities and error-rates of analogue simulation, and the universality of universal quantum simulation.

1 Basics of Classical and Quantum Complexity Theory

Complexity theory is to computer science what the theory of evolution is to biology: it is always lurking in the background, determining what is and isn't possible. The central idea is to collect the problems that can always be solved with a given amount of resources (on a given model of computation e.g. classical, non-deterministic, quantum) into sets, referred to as complexity classes.

The most basic, and standard, notions of complexity are all formulated in terms of decision problems, which are defined in Definition 1.

Definition 1.1

A decision problem is a problem where the input is a string of characters (from a well defined alphabet, often denoted as Σ) and the output must be a single bit.

1.1 Defining Complexity Classes

Arguably the most important classical complexity class is P. This is due to Cobham's thesis Edmonds, "[Paths, Trees, and Flowers](#)"; Cobham, "[The Intrinsic Computational Difficulty of Functions](#)". While not a rigorous theorem, Cobham's thesis was formulated based on experience and states that for any computational device, only problems with time requirements scaling, at most, polynomially in the size of the problem can be solved in a practically achievable amount of time. Such problems are referred to as being tractable.

Definition 1.2

A decision problem is in the complexity class P if and only if any instance can be decided on a classical computer, as modelled in Sec. 0.1, in a time bounded by a polynomial function of the size of the problem, n , such as $n^2 + n + 1$.

An example in this class is checking if a number is a multiple of 2. This can be accomplished by looking at a single bit of a binary input string.

Another important question to ask about problems, other than how long they take to solve, is how long it takes to check that a given answer to a problem is correct. This process is known as verification and, as previously established, whether a computation can be done in polynomial time is important for if it is practical, so another important class is the problems that can be verified in polynomial time: NP.

Definition 1.3

A decision problem is in the complexity class NP if and only if a non-deterministic classical computer, as modelled in Sec. 0.3, can decide any instance of the problem in polynomial time.

Another important complexity class is BPP. This is the most important class, due to Cobham's thesis, for computers with the ability to make random decisions (i.e. with the ability for a program the computer runs to have stochastic steps).

Definition 1.4

A decision problem is in the complexity class BPP if and only if any instance can be decided on a classical computer, with the ability to make random choices, as modelled in Sec. 0.2, in a time bounded by a polynomial function of the size of the problem, with probability of correctness at least $\frac{2}{3}$ for any instance.

BPP finds itself in a strange position: its relationship to almost any significant complexity class is unknown. It is expected to be equivalent to P but this cannot be proven and the closest anyone has ever gotten is the SipserLautemann theorem Lautemann, “BPP and the Polynomial Hierarchy” where it was shown $\text{BPP} \subseteq \Sigma_2 \cap \Pi_2$ (the definition of $\Sigma_2 \cap \Pi_2$ is given in Appendix ?? alongside the background of the polynomial hierarchy - which it forms a part of. For now it suffices to state that $\Sigma_2 \cap \Pi_2$ is expected to be a lot larger than P, or even NP).

Another expected relation nobody can prove is that BPP is strictly contained in BQP. BQP, again due to Cobham's thesis, is the central complexity class for quantum computers.

Definition 1.5

BQP is the set of all decision problems where the runtime to correctly decide any input instance with probability greater than $\frac{2}{3}$, on a quantum computer, as modelled in Sec. 0.4, is bounded by a polynomial of the size of the input.

I conclude this subsection on background definitions by defining a notation that proves useful in discussing complexity classes, what they can do, and how they related to each other.

Definition 1.6

For a complexity class, C , a C -device is anything that can decide all problems in C . A C -algorithm is an algorithm that can be run on a C -device, and saying a problem can be decided with C -resources or an algorithm solving the problem can run with C -resources is equivalent to saying the problem can be decided with a C -device and/or using a C -algorithm.

1.2 The relationship between P and NP

Clearly $P \subseteq NP$, as for any language in P, a solution can be verified in polynomial time by disregarding any certificate⁸ and recalculating the solution.

So is $NP \subseteq P$ (implying $P = NP$)? This remains an open question but is widely suspected not to be true.

1.3 Completeness and Hardness

To define the notion of completeness, it is necessary to introduce the concept of reductions:

Definition 1.7

A reduction of problem A to problem B is the process of reformulating problem A as problem B, so that solving problem B provides a solution to problem A. If this can be done in polynomial time, it is known as a Cook reduction.

An amazing consequence of reductions between decision problems was found in 1971 by Stephen Cook, “[The Complexity of Theorem-Proving Procedures](#)”, and independently in 1973 by Leonid Levin Trakhtenbrot, “[A Survey of Russian Approaches to Perebor \(Brute-Force Searches\) Algorithms](#)”, first in the context of NP-Completeness (defined below) but a generalised notion of completeness, that can be found in many complexity classes, was soon realised.

Definition 1.8

A problem, X, is NP-Complete if and only if:

- 1) $X \in NP$
- 2) $\forall Y \in NP$, Y is Cook reducible to X

And the more general notion of completeness can be defined as:

Definition 1.9

A decision problem, X, is complete for a complexity class, C_{la} if and only if:

- 1) $X \in C_{la}$
- 2) $\forall Y \in C_{la}$, Y is Cook reducible to X

The complexity theory shown so far can be summarised in the diagram in Fig. 6.6, depicting the relationships between the complexity classes P, NP, and NP-Complete, assuming $P \neq NP$. The final classical complexity class to be defined is NP-Hard:

⁸which just means an input to the machine to do the verifying - corresponding to a specific instance of the problem

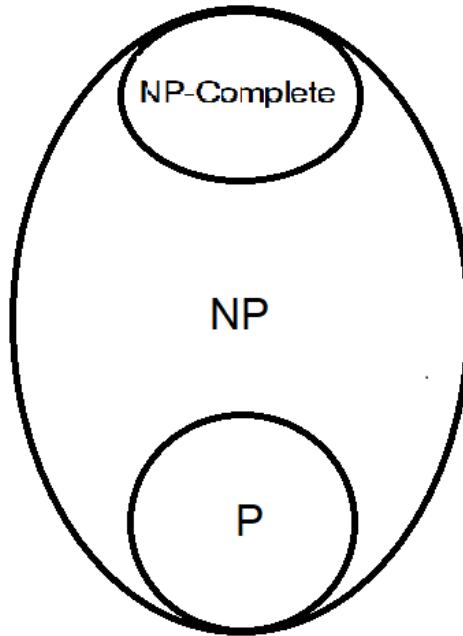


Figure 6.6: Diagram of expected relation between P and NP

Definition 1.10

A problem is said to be NP-Hard if it is at least as hard as the hardest problems in NP

The concept of hardness generalises in the same way completeness did.

1.4 P and BQP

A key question in quantum computing, and given the amount of money being thrown at quantum computing it would be nice to know, is whether quantum computers have any benefit over the pre-existing, well understood, classical computers? This may be formalised in comparing two of the complexity classes defined previously: P and BQP.

It is known that $P \subseteq BQP$ (as a BQP-device can just act exclusively classically to emulate a P-device) but quantum computers would only be useful if $P \neq BQP$. Unfortunately, this is not known. Like many problems in complexity theory, there is a suspicion that it is true but it has not been proven. The expected relation between P, NP, and BQP is shown in Fig. 6.7.

1.5 One Clean Qubit and DQC1

Similarly to how Cobham's thesis provides the tractable quantum complexity class for the circuit model, BQP, the tractable class for the one clean qubit model is DQC1.

DQC1 (Deterministic quantum computation with one clean qubit) Knill and Laflamme, "Power of One Bit of Quantum Information"; Shepherd, *Computation with Unitaries and One Pure Qubit* is a complexity class believed to properly contain BPP but not be equiv-

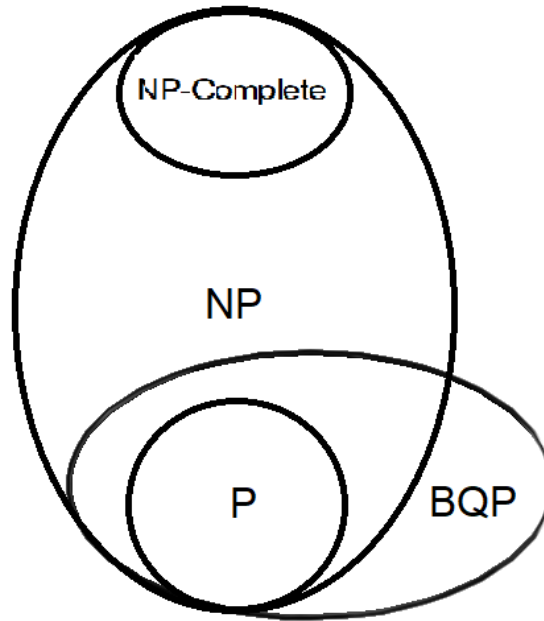


Figure 6.7: Diagram of expected relation between P, NP, and BQP

alent to BQP Shor and Jordan, “[Estimating Jones Polynomials is a Complete Problem for One Clean Qubit](#)”; Aharonov, V. Jones, and Landau, “[A Polynomial Quantum Algorithm for Approximating the Jones Polynomial](#)”; Datta and Shaji, “[Quantum Discord and Quantum Computing - An Appraisal](#)”. Computations in the one clean qubit model are - in general - therefore not expected to be efficiently simulable classically. DQC1 can be more formally defined as:

Definition 1.11

DQC1 is defined as the complexity class of all decision problems that can be decided with probability at least $\frac{2}{3}$, using the one clean qubit model, as modelled in Sec. 0.5, in polynomial (in the input length) time.

The expected relation between DQC1 and other classes is shown in Fig. 6.8.

1.6 Functional Classes

For all complexity classes defined above, we can also define a functional class of problems where the output is not necessarily a yes or no. The functional class, fX , corresponding to complexity class, X , is defined as the set of functions computable with the same resources as are specified in the definition of X .

Another class consisting of problems where the correct answer is an integer, that is important to consider but contains problems much too hard to actually solve, is $\#P$.

This class is most important, for the purposes of this thesis, for denoting just how hard some problems are to solve and how far beyond efficient classical or quantum computing they

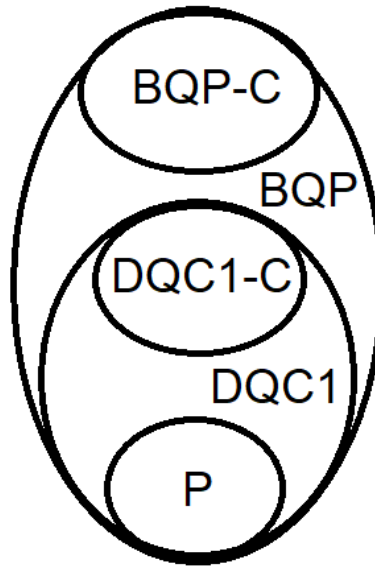


Figure 6.8: Diagram of expected relations between P , $DQC1$, and BQP (-C is used as a shorthand for -Complete)

(probably) are.

Definition 1.12

$\#P$ is defined as the class of all functions: $\{0, 1\}^* \rightarrow \mathbb{N}$ such that there exists a non-deterministic Turing machine, $\mathcal{T}_\#$, (taking inputs from alphabet Σ^*) such that $\forall x \in \Sigma^*$, the number of accepting paths/branches (i.e. the sets of possible non-deterministic ‘choices’ the Turing machine can make that ends in accepting) $\mathcal{T}_\#$ has on input x is $f(x)$.

Clearly, this problem is harder than the hardest problems in NP but its exact relation to other complexity classes remains largely unknown.

CHAPTER 7

NOISE IN QUANTUM SYSTEMS

Start with idea system is actually a lot larger than we would like. and is an open system and then look at how we represent this with

1 Introduction To Noise and Error

Regardless of what you aim to do with quantum computing, whatever clever algorithms you have planned; eventually have to start dealing with reality. Namely, that not everything goes perfectly all the time (e.g. you start your PhD and a worldwide pandemic begins six months in): gates are not always correctly applied, they sometimes perform an incorrect mapping between states of the qubits; the correct initial state for a computation is not always prepared; measurements are sometimes performed incorrectly; etc.

I refer to this as error occurring in the gates, state preparation, or measurements, respectively. To have any characterisation of - and hence any measure of - error, I first need to define various metrics on $\mathbb{B}(\mathcal{H})$ (for arbitrary Hilbert spaces \mathcal{H}).

In this context, $\mathbb{B}(\mathcal{H})$ represents the set of all mixed states of a system with the Hilbert space, \mathcal{H} , hence these measures allow me to quantise the difference between quantum states. This lays the groundwork for quantifying the error that occurs in a quantum computation.

1.1 Quantifying Error

Quantifying how badly the implementation of an operator goes wrong within some quantum technology necessitates a measure of how different the ideal and erroneous operators' matrix representations are. For this purpose, I present the below concepts.

Definition 1.1: Test

test $\forall p \in \mathbb{N}$, define the Schatten p -norm as a function from matrices to the reals:

$$\|M\|_p = \left[\text{Tr} \left((M^\dagger M)^{p/2} \right) \right]^{1/p}. \quad (7.1)$$

As Schatten p -norms are defined on matrices, via representation theory, they are also defined on linear operators.

Definition 1.2

The trace norm (also known as the nuclear norm) is the Schatten p -norm with $p = 1$.

Definition 1.3

For any two matrices, ρ, σ , define the trace distance, $\mathcal{D}(\rho, \sigma)$, as:

$$\mathcal{D}(\rho, \sigma) = \frac{1}{2} \|\rho - \sigma\|_1. \quad (7.2)$$

Definition 1.4

Given two probability distributions over the set of possible outcomes, Ω , (i.e. mappings that take a possible outcome, $s \in \Omega$, as input and return a number from $[0, 1]$), P and Q , define the variational distance between them, $VD(P, Q)$, by:

$$VD(P, Q) = \frac{1}{2} \sum_{s \in \Omega} |P(s) - Q(s)|. \quad (7.3)$$

1.2 A First Bound on Error

When initially considering error in quantum circuits, the first question to ask is does error ruin everything i.e. does error cascade through the circuit growing exponentially, destroying any chance of a correct computation.

To counteract this fear, see the below Lemma 9. But before this lemma can be presented, some pre-requisites must be first presented. They may seem like a lot for this brief aside, but they are re-used later so their full statement and proof can be justified.

Lemma 8. *Let,*

- ρ and σ be density matrices on a Hilbert space, \mathcal{H} (i.e. $\rho, \sigma \in \mathbb{B}(\mathcal{H})$).
- $\{\hat{E}_m\}$ be a POVM.

Then, for any \hat{E}_m in the POVM,

$$\left| \left(\hat{E}_m(\rho - \sigma) \right) \right| \leq \left(\hat{E}_m |\rho - \sigma| \right) \quad (7.4)$$

Where for any operator (or, equivalently, density matrix), \hat{A} , define $|\hat{A}| = \sqrt{\hat{A}^\dagger \hat{A}}$.

Proof.

As both ρ and σ are Hermitian, $\rho - \sigma$ is also Hermitian.

I can split the spectrum of $\rho - \sigma$ into the set of positive, or zero, eigenvalues and define $\{+\}$ be the set of indices, j , such that λ_j is in this set. Similarly, I can define $\{-\}$ as the set of indices not in $\{+\}$. Then, defining $|\lambda_j\rangle$ as the eigenstate of $\rho - \sigma$ with eigenvalue λ_j :

$$\rho - \sigma = \sum_j \left(\lambda_j |\lambda_j\rangle \langle \lambda_j| \right) = \sum_{j \in \{+\}} \left(\lambda_j |\lambda_j\rangle \langle \lambda_j| \right) - \sum_{j \in \{-\}} \left(|\lambda_j| |\lambda_j\rangle \langle \lambda_j| \right) \quad (7.5)$$

Label the two summations in Eqn. (7.5) as \hat{F} and \hat{S} , respectively in order of appearance. As each of \hat{F} and \hat{S} is a positive operator, and each acts non-trivially only on a sub-space of the Hilbert space completely disjoint from the sub-space the other acts non-trivially on, $\rho - \sigma$ can be re-expressed as:

$$\rho - \sigma = \hat{F} - \hat{S} \Rightarrow |\rho - \sigma| = |\hat{F} - \hat{S}| = \hat{F} + \hat{S} \quad (7.6)$$

This implies:

$$\left| \left(\hat{E}_m(\rho - \sigma) \right) \right| \leq \left[\hat{E}_m(\hat{F} + \hat{S}) \right] = \left(\hat{E}_m |\rho - \sigma| \right) \quad (7.7)$$

□

Let,

- $\rho, \sigma \in \mathbb{B}(\mathcal{H})$ be density matrices on a Hilbert space, \mathcal{H} .
- $\{\hat{E}_m\}$ be a POVM.
- q_m, p_m be probabilities of the outcome indexed as m , from the POVM on states ρ and σ , respectively.
- P, Q be the probability distributions resulting from the POVM on each of the states, ρ and σ , respectively.

Then,

$$VD(P, Q) \leq \mathcal{D}(\rho, \sigma) \quad (7.8)$$

Proof. From how p_m, q_m are defined

$$p_m = \left(\hat{E}_m \rho \right) \quad (7.9)$$

$$q_m = \left(\hat{E}_m \sigma \right) \quad (7.10)$$

Then the definition of total variational distance implies:

$$VD(P, Q) = \frac{1}{2} \sum_m \left(|p_m - q_m| \right) = \frac{1}{2} \sum_m \left\{ |[\hat{E}_m(\rho - \sigma)]| \right\} \quad (7.11)$$

Then, via Lemma 8, this implies:

$$VD(P, Q) \leq \frac{1}{2} \left[\sum_m \left(\hat{E}_m \right) |\rho - \sigma| \right] \quad (7.12)$$

Then, as $\{\hat{E}_m\}$ is a POVM,

$$VD(P, Q) \leq \frac{1}{2} (|\rho - \sigma|) = \mathcal{D}(\rho, \sigma) \quad (7.13)$$

□

With the pre-requisite lemmas presented, a first bound on the error is possible.

Lemma 9. *If \hat{C} is an ideal circuit consisting of m gates, contained in the set $\{\hat{C}_j\}_{j=1}^m$, and \hat{C}' is an imperfect implementation of \hat{C} (that is still unitary), with imperfect gates $\{\hat{C}'_j\}_{j=1}^m$, such that $\forall j \in \mathbb{Z}^{\leq m}$, $\|\hat{C}_j - \hat{C}'_j\|_1 \leq \epsilon$, then the variational distance between probability distributions generated by the outcomes of ideal (\hat{C}) and erroneous (\hat{C}') implementations of the circuit is bounded by $m\epsilon$. I.e. if P and P' are the probability distributions generated by any measurements on states resulting from applying \hat{C} and \hat{C}' , respectively, to an arbitrary input state, then:*

$$VD(P, P') \leq m\epsilon. \quad (7.14)$$

1

Proof. Let N be the number of qubits in the circuits. Then, in an abuse of notation, each circuit, \hat{C} and \hat{C}' , is treated as the unitary (in $\mathbb{U}(2^N)$) its gates implement. The same is true for each gate in each circuit i.e. each element of $\{\hat{C}_j\}_{j=1}^m$ is considered as the corresponding element of $\mathbb{U}(2^N)$.

Consider $\|\hat{C} - \hat{C}'\|_1$ and aim to show $\|\hat{C} - \hat{C}'\|_1 \leq m\epsilon$. To this end I proceed by induction.

Base case: $m = 1$

If $m = 1$, $\|\hat{C} - \hat{C}'\|_1 = \|\hat{C}_1 - \hat{C}'_1\|_1 \leq \epsilon = m\epsilon$. So the base case holds.

Inductive step

Define:

- $\hat{V}_m = \hat{C}_{m-1} \hat{C}_{m-2} \cdots \hat{C}_1 \in \mathbb{U}(2^N)$
- $\hat{V}'_m = \hat{C}'_{m-1} \hat{C}'_{m-2} \cdots \hat{C}'_1 \in \mathbb{U}(2^N)$

¹The first approximately half of the proof (Up to Eqn. (7.20)) of Lemma 9 is from Ref. Gharibian, [Quantum Complexity Theory \(Lecture Notes\)](#) (This has been taken offline since I used it.), from there on it is original.

Assume the claim holds for $m = m_0 - 1$, then consider:

$$\|\hat{C} - \hat{C}'\|_1 = \|\hat{C}_{m_0} \hat{V}_{m_0} - \hat{C}'_{m_0} \hat{V}'_{m_0}\|_1 \quad (7.15)$$

$$= \|\hat{C}_{m_0} \hat{V}_{m_0} - \hat{C}'_{m_0} \hat{V}'_{m_0} + \hat{C}_{m_0} \hat{V}'_{m_0} - \hat{C}'_{m_0} \hat{V}'_{m_0}\|_1 \quad (7.16)$$

$$\leq \|\hat{C}_{m_0} (\hat{V}_{m_0} - \hat{V}'_{m_0})\|_1 + \|(\hat{C}_{m_0} - \hat{C}'_{m_0}) \hat{V}'_{m_0}\|_1 \quad (7.17)$$

$$\leq \|\hat{V}_{m_0} - \hat{V}'_{m_0}\|_1 + \|\hat{C}_{m_0} - \hat{C}'_{m_0}\|_1 \quad (7.18)$$

$$\leq (m_0 - 1)\epsilon + \epsilon = m_0\epsilon. \quad (7.19)$$

The inductive step shows if the bound holds for a $m_0 - 1$ operator circuit, it holds for a m_0 operator circuit. Then, as the base case ($m_0 - 1 = 0$) holds, the bound holds for all positive integers. Then consider:

$$\mathcal{D}[\hat{C}\rho\hat{C}^\dagger, \hat{C}'\rho(\hat{C}')^\dagger] = \frac{1}{2} \|\hat{C}\rho\hat{C}^\dagger - \hat{C}'\rho(\hat{C}')^\dagger\|_1 \quad (7.20)$$

$$= \frac{1}{2} \|\hat{C}\rho\hat{C}^\dagger - \hat{C}'\rho(\hat{C}')^\dagger + \hat{C}\rho(\hat{C}')^\dagger - \hat{C}\rho(\hat{C}')^\dagger\|_1 \quad (7.21)$$

$$\leq \frac{1}{2} \|\hat{C}\rho[\hat{C}^\dagger - (\hat{C}')^\dagger]\|_1 + \frac{1}{2} \|(\hat{C} - \hat{C}')\rho(\hat{C}')^\dagger\|_1 \quad (7.22)$$

$$\leq \frac{1}{2} \|\rho[\hat{C}^\dagger - (\hat{C}')^\dagger]\|_1 + \frac{1}{2} \|(\hat{C} - \hat{C}')\rho\|_1 \quad (7.23)$$

$$\leq \frac{1}{2} \|\rho\|_1 \cdot \|\hat{C}^\dagger - (\hat{C}')^\dagger\|_1 + \frac{1}{2} \|\hat{C} - \hat{C}'\|_1 \cdot \|\rho\|_1 \quad (7.24)$$

$$= \|\hat{C} - \hat{C}'\|_1, \quad (7.25)$$

where \cdot denotes scalar multiplication and is used here to clarify and keep the equations clear. Using Eqn. (7.19) in Eqn. (7.25):

$$\mathcal{D}[\hat{C}\rho\hat{C}^\dagger, \hat{C}'\rho(\hat{C}')^\dagger] \leq m\epsilon. \quad (7.26)$$

Finally, let P, P' be the probability distributions generated by the measurements on the states $\hat{C}\rho\hat{C}^\dagger$ and $\hat{C}'\rho(\hat{C}')^\dagger$, respectively. Then the variational distance between probability distributions generated by the outcomes of ideal (\hat{C}) and erroneous (\hat{C}') implementations, $\text{VD}(P, P')$, is bounded, via Theorem 1.2, as in Eqn. (7.27).

$$\text{VD}(P, P') \leq \mathcal{D}[\hat{C}\rho\hat{C}^\dagger, \hat{C}'\rho(\hat{C}')^\dagger] \leq m\epsilon. \quad (7.27)$$

□

Lemma 9 is notably simplistic: the error in each gate is considered constant i.e. the difference between different gates or the effect they may have on each other is completely neglected; a bound, such as ϵ , even if it existed for a fixed size system would be exceptionally hard to obtain if the surrounding circuit is allowed to have an effect on the error in gates; I assume the erroneous gates are unitary; and finally, Lemma 9 neglects error in measurement and state preparation. However, it does give some indication that error in quantum computers is manageable.

2 A Background on CPTP Maps

The “gold standard” way to represent error in quantum operations is that an erroneous implementation of an operator/measurement/state preparation is equivalent to the ideal/errorless operation followed by a completely positive trace-preserving map. As such, Completely Positive trace-preserving (CPTP) maps are a key notion in any consideration of error. These maps, as the name suggests, are any map from the system under consideration to itself that is completely positive and trace-preserving. I pause here to introduce these maps and their important properties that will be used later in this chapter.

Definition 2.1

Any linear operator, A , acting on any Hilbert space is positive if, for any element, $|x\rangle$, in the Hilbert space:

$$\langle x|A|x\rangle \geq 0. \quad (7.28)$$

Definition 2.2

For any Hilbert space, \mathcal{H} , any map, $\Phi : \mathbb{B}(\mathcal{H}) \rightarrow \mathbb{B}(\mathcal{H})$, is positive if positive operators are mapped exclusively to positive operators. If, $\forall N \in \mathbb{N}$, $\Phi \otimes I_N$ (where I_N is the identity on N qubits) is positive, then Φ is completely positive.

Definition 2.3

For any Hilbert space, \mathcal{H} , a map, Φ , is trace-preserving if $\forall \rho \in \mathbb{B}(\mathcal{H})$,

$$\text{Tr} \left[\Phi(\rho) \right] = \text{Tr} \left(\rho \right). \quad (7.29)$$

For completeness, once completely-positive and trace-preserving maps have been defined, I formally define CPTP maps.

Definition 2.4

A completely positive trace-preserving map between matrices (or, via their representations, states) is a map that is :

- Completely Positive.
- trace-preserving.

A central theorem of use in presenting the following properties of CPTP maps is the Stinespring Dilation Theorem. The most quantum-information-friendly background and proof of the Stinespring Dilation Theorem (to my knowledge) can be found in Ref. Watrous, “[Basic Notions of Quantum Information](#)”, following from the proof of Theorem 2.22.

Theorem 2.5: Stinespring Dilation Theorem (Special Case)

Let \mathcal{H}_A be a Hilbert space, then for any completely-positive trace-preserving map, $\Phi : \mathbb{B}(\mathcal{H}_A) \rightarrow \mathbb{B}(\mathcal{H}_A)$, there exists: another Hilbert space, \mathcal{H}_C ; an element of that space, $|\phi\rangle \in \mathcal{H}_C$; and a unitary, $\hat{U}_{AC} : \mathcal{H}_A \otimes \mathcal{H}_C \rightarrow \mathcal{H}_A \otimes \mathcal{H}_C$, such that, $\forall \rho_A \in \mathbb{B}(\mathcal{H}_A)$:

$$\Phi(\rho_A) = \text{Tr}_c \left[\hat{U}_{AC} \circ \left(\rho_A \otimes |\phi\rangle\langle\phi| \right) \circ \hat{U}_{AC}^\dagger \right], \quad (7.30)$$

Where \circ denotes operator composition but is additionally overloaded to be matrix multiplication between matrices.

Proof for Theorem.

■ Proof omitted ■

A first application of Theorem 2 is in Lemma 10, which details and proves the existence of a useful form for considering CPTP maps.

Lemma 10. *For any Hilbert space, \mathcal{H} , any CPTP map, Φ , acting on $\mathbb{B}(\mathcal{H})$, has Kraus decomposition i.e. can be expressed as:*

$$\Phi(\rho) = \sum_{j=1} \left(\hat{\mathcal{K}}_j \circ \rho \circ \hat{\mathcal{K}}_j^\dagger \right), \quad (7.31)$$

Where each $\hat{\mathcal{K}}_j$ is a linear operator on \mathcal{H} (called a Kraus operator), and,

$$\sum_{j=1} \left(\hat{\mathcal{K}}_j^\dagger \circ \hat{\mathcal{K}}_j \right) = \hat{I}. \quad (7.32)$$

Note that the number of Kraus operators is always finite.

Proof. Via the Stinespring Dilation Theorem, any CPTP map, $\Phi : \mathbb{B}(\mathcal{H}) \rightarrow \mathbb{B}(\mathcal{H})$, can be expressed as: $\forall \rho \in \mathbb{B}(\mathcal{H})$

$$\Phi(\rho) = \text{Tr}_c \left[\hat{U}_{AC} \circ \left(\rho \otimes |\phi\rangle\langle\phi| \right) \circ \hat{U}_{AC}^\dagger \right]. \quad (7.33)$$

Let $\{|e_j\rangle\}_{j=1}^{N_E}$ be a basis of the extra Hilbert space added (denoted \mathcal{H}_c in the Stinespring Dilation Theorem). Then $\Phi(\rho)$ can be re-expressed as:

$$\Phi(\rho) = \sum_{j=1}^{N_E} \left[\left(\hat{I}_A \otimes \langle e_j| \right) \circ \hat{U}_{AC} \circ \left(\rho \otimes |\phi\rangle\langle\phi| \right) \circ \hat{U}_{AC}^\dagger \circ \left(\hat{I}_A \otimes |e_j\rangle \right) \right]. \quad (7.34)$$

Defining $\hat{\mathcal{K}}_j$ as $\left(\hat{I}_A \otimes \langle e_j| \right) \circ \hat{U}_{AC} \circ \left(I_A \otimes |\phi\rangle \right)$. Therefore,

$$\Phi(\rho) = \sum_{j=1}^{N_E} \left(\hat{\mathcal{K}}_j \circ \rho \circ \hat{\mathcal{K}}_j^\dagger \right). \quad (7.35)$$

All that then remains is to show that the condition in Eqn. (7.32) is satisfied:

$$\sum_{j=1} \left(\hat{\mathcal{K}}_j^\dagger \circ \hat{\mathcal{K}}_j \right) = \sum_{j=1} \left[(I_A \otimes \langle \phi |) \circ \hat{\mathcal{U}}_{AC}^\dagger \circ (I_A \otimes |e_j\rangle) \circ (I_A \otimes \langle e_j |) \circ \hat{\mathcal{U}}_{AC} \circ (I_A \otimes |\phi\rangle) \right] \quad (7.36)$$

$$= (\hat{I}_A \otimes \langle \phi |) \circ \hat{\mathcal{U}}_{AC}^\dagger \circ \sum_{j=1} \left(\hat{I}_A \otimes |e_j\rangle \langle e_j| \right) \circ \hat{\mathcal{U}}_{AC} \circ (\hat{I}_A \otimes |\phi\rangle) = \hat{I}. \quad (7.37)$$

□

The converse of Lemma 10 also holds, as shown in Lemma 11.

Lemma 11. *For any Hilbert space, \mathcal{H} , and any map, Φ acting on $\mathbb{B}(\mathcal{H})$, of the form,*

$$\Phi(\rho) = \sum_{j=1} \left(\hat{\mathcal{K}}_j \circ \rho \circ \hat{\mathcal{K}}_j^\dagger \right), \quad (7.38)$$

where each $\hat{\mathcal{K}}_j$ is a linear operator on \mathcal{H} , and,

$$\sum_{j=1} \left(\hat{\mathcal{K}}_j^\dagger \circ \hat{\mathcal{K}}_j \right) = \hat{I}. \quad (7.39)$$

(i.e. ϕ has a Kraus decomposition) is a completely positive trace-preserving map.

Proof. I need to show that the map in Eqn. (7.38) is both trace-preserving and completely positive. Denote this map by Φ .

Trace-preserving

$$\text{Tr}[\Phi(\rho)] = \text{Tr} \left[\sum_{j=1} \left(\hat{\mathcal{K}}_j \circ \rho \circ \hat{\mathcal{K}}_j^\dagger \right) \right] = \sum_{j=1} \left[\text{Tr} \left(\hat{\mathcal{K}}_j^\dagger \circ \hat{\mathcal{K}}_j \circ \rho \right) \right] = \text{Tr} \left[\sum_{j=1} \left(\hat{\mathcal{K}}_j^\dagger \circ \hat{\mathcal{K}}_j \right) \circ \rho \right] \quad (7.40)$$

$$= \text{Tr}(\rho). \quad (7.41)$$

Hence the trace remains unchanged after the application of Φ .

Completely-Positive

Let \mathcal{H}' be an arbitrary Hilbert space of dimension $n \in \mathbb{N}$ and $|x\rangle \in \mathcal{H} \otimes \mathcal{H}'$, then:

$$\langle x | [\Phi(\rho) \otimes \hat{I}_n] | x \rangle = \langle x | \sum_{j=1} \left[(\hat{\mathcal{K}}_j \otimes \hat{I}_n) \circ (\rho \otimes \hat{I}_n) \circ (\hat{\mathcal{K}}_j^\dagger \otimes \hat{I}_n) \right] | x \rangle \quad (7.42)$$

$$= \sum_{j=1} \left\{ \langle x | [(\hat{\mathcal{K}}_j \otimes \hat{I}_n) \circ (\rho \otimes \hat{I}_n) \circ (\hat{\mathcal{K}}_j^\dagger \otimes \hat{I}_n)] | x \rangle \right\} = \sum_{j=1} \left[\langle x_j | (\rho \otimes \hat{I}_n) | x_j \rangle \right], \quad (7.43)$$

where $|x_j\rangle$ is defined as $(\hat{\mathcal{K}}_j^\dagger \otimes \hat{I}_n)|x\rangle$ and is still in $\mathcal{H} \otimes \mathcal{H}'$. I then note that ρ being positive implies that $\rho \otimes \hat{I}_n$ is positive, hence:

$$\langle x | [\Phi(\rho) \otimes \hat{I}_n] | x \rangle = \sum_{j=1} \left[\langle x_j | (\rho \otimes \hat{I}_n) | x_j \rangle \right] \geq 0. \quad (7.44)$$

So Φ is completely positive. Therefore Φ is a completely positive, trace-preserving map, a CPTP map. \square

3 A Characterisation of Error

In this thesis, error in an operation will always be modelled as CPTP maps following or preceding the correct implementation of the intended (ideal) operation. There may be some other conditions on the error, but it will always be modelled as CPTP maps.

The characterisation of all error as being CPTP error is a strong claim. This can be argued for by considering all error that could affect the system as being the result of a unitary (as all changes in a known closed system are representable by a unitary operator) acting on the entire universe, including the system. However the state of the universe is not known and so, from our perspective, is stochastic. Thus, the actual error can be considered as consisting of different unitaries applied with varying probabilities and therefore can be characterised by the map:

$$\mathcal{E}(\rho) = \sum_{k=1} \left(p_k \hat{\mathcal{U}}_k \rho \hat{\mathcal{U}}_k^\dagger \right), \quad (7.45)$$

where,

- $\mathcal{E}(\cdot)$ is the CPTP map representing the error acting on a density matrix (given as the argument).
- Each $\hat{\mathcal{U}}_k$ is a unitary that acts on both the system and environment, there is a unknown number of them so the summation above has no upper limit.
- p_k is a probability such that $\sum_{k=1} (p_k) = 1$.

Therefore, by Lemma 11, the error applied can be modelled as a CPTP map acting on both the system and its environment. Alternatively, I can require certain things of post-error states and use this to imply the error map is a CPTP map. These requirements are:

Let ρ be any density matrix of the system the error acts on, and let \mathcal{E} be the error map,

1. $\mathcal{E}(\rho)$ can be treated identically to a density matrix for measurement purposes.
2. The outcome probabilities for any measurement on $\mathcal{E}(\rho)$ are positive.
3. The outcome probabilities for any measurement on $\mathcal{E}(\rho)$ sum to one.

Theorem 3.1

The above requirements imply that \mathcal{E} is a CPTP map.

Proof for Theorem.

For a measurement of the state of ρ (in some orthogonal basis) e.g. measuring a single qubit in the Z-basis; let $|\alpha\rangle$ be any output state, Π_α be a projector onto that state, $[\alpha]$ be the set of all possible measurement outcomes and \mathbb{P}_α be the probability of measuring that state.

\mathcal{E} is completely positive

Let ρ' be the density matrix of any state of any system in which the system \mathcal{E} acts on is a subsystem, similarly let α' be any possible outcome of any measurement on the entire system.

Requirement 2 means $\mathbb{P}_{\alpha'} \geq 0$, therefore, for any measurement outcome, α' :

$$0 \leq \text{Tr} \left[\Pi_{\alpha'} (\hat{I} \otimes \mathcal{E})(\rho') \right] = \sum_{b' \in [\alpha']} \left[\langle b' | \alpha' \rangle \langle \alpha' | (\hat{I} \otimes \mathcal{E})(\rho') | b' \rangle \right] = \langle \alpha' | (\hat{I} \otimes \mathcal{E})(\rho') | \alpha' \rangle. \quad (7.46)$$

As any density matrix is a positive operator, this implies \mathcal{E} is a completely positive map.

\mathcal{E} is trace-preserving

Requirement 3 means $\sum_\alpha (\mathbb{P}_\alpha) = 1$, therefore:

$$1 = \sum_\alpha \left[\text{Tr} \left(\Pi_\alpha \mathcal{E}(\rho) \right) \right] = \text{Tr} \left[\sum_\alpha \left(\Pi_\alpha \right) \mathcal{E}(\rho) \right] = \text{Tr} \left[\mathcal{E}(\rho) \right], \quad (7.47)$$

where the last equality follows from the sum of projectors onto disjoint subspaces, the union of which is the entire space, is equivalent to the identity.

As any density matrix also has trace one, \mathcal{E} is trace-preserving. ■