

PHYC90045 Introduction to Quantum Computing

Week 1



Lecture 1

- 1.1 Non-technical overview of the quantum world (& QC)
- 1.2 Qubits: mathematical preliminaries I

Lecture 2

- 2.1 Qubits: mathematical preliminaries II
- 2.2 Single qubit logic gates

Lab 1

QUI, Single qubits and logic gates

PHYC90045 Introduction to Quantum Computing

Lecture 1 overview



In this lecture:

1.1 Non-technical overview of the quantum world (and QC)

- Easy intro to qubits, superposition, entanglement, QC

1.2 Qubits: mathematical preliminaries I

- Superposition and measurements
- Linear algebra and Dirac notation
- Measurement, quantum amplitudes, complex numbers and phase
- Projective operators
- Linear dependence and basis states
 - Rieffel, Chapter 3
 - Kaye, 2.6
 - Nielsen & Chuang, 1.3.2-1.3.4

PHYC90045 Introduction to Quantum Computing



1.1 Non-technical overview of the quantum world (and QC)

PHYC90045
Lecture 1

PHYC90045 Introduction to Quantum Computing

The Quantum World

Some meta-physics...

The diagram shows an iceberg floating in water. The visible part above the surface is labeled "nature of the physical world". Below the surface, the submerged part is labeled "nature of ‘reality’". A Greek letter Ψ (Psi) is shown at the base of the iceberg. To the right, it says "Advances in experimental and theoretical quantum science:" followed by a list: "→ quantum sensing", "quantum communication", "quantum computing", and "...". The University of Melbourne logo is in the top right corner.

PHYC90045 Introduction to Quantum Computing

Key concepts for quantum computing

Quantum superposition:
Systems can be in indeterminate (multiple) states prior to measurement

Quantum measurement:
Result of any given measurement a-priori unknown, system “collapses” to an outcome

Quantum entanglement:
Systems can be linked such that measurement of one part correlates to that of another part

PHYC90045 Introduction to Quantum Computing

Start with a familiar concept – the atom

Planck (1900): postulated that energy is quantised

Bohr (1913): constructed a simple “quantised” model of the atom

The diagram shows a central nucleus consisting of a proton and an electron. The electron orbits the nucleus in circular paths. On the left, there is a mathematical formula for the energy levels of the Bohr model: $E_n = -\frac{m}{2\hbar^2} \left(\frac{e^2}{4\pi \epsilon_0} \right)^2 \frac{1}{n^2}$. It also shows energy levels labeled n=1, n=2, n=3, ..., n=∞. Transitions between levels are indicated by arrows: a vertical arrow from the ground state (n=1) to an excited state (n=2), and a horizontal arrow from the excited state (n=2) to a higher level (n=3). On the right, it says "Consider lowest two levels" and shows two concentric circles representing the first two energy levels. The outer level is labeled |1⟩ and the inner level is labeled |0⟩. Below this, it says "Simply re-label states to a bit notation" and "→ quantum bit, or ‘qubit’" followed by "(Not to be confused with cubit!)".

The diagram illustrates the principles of quantum mechanics. On the left, a yellow circle represents an orbital path with a green electron at one point and a red electron at another. Below it, two horizontal lines represent the state $|1\rangle$ (yellow) and $|0\rangle$ (blue). The text "One electron in quantum superposition." is written below the diagram. In the center, the text "quantum superposition" is above a sphere containing a multi-colored electron cloud with labels $|1\rangle$, Ψ , and $|0\rangle$. Below this, the text " $|0\rangle$ ‘and’ $|1\rangle$ " is shown. To the right, the text "measurement/observation" is above a black-and-white photograph of a beach ball with a red arrow pointing to it, labeled $|0\rangle$. Below this, the text "state collapse → random outcome" is shown.

The diagram illustrates the principles of quantum mechanics. On the left, two overlapping circles represent energy levels; a green dot is in the inner circle and a red dot is in the outer circle. Below these are two horizontal lines: the top one has a blue dot and is labeled $|1\rangle$; the bottom one has a blue dot and is labeled $|0\rangle$. The text "One electron in quantum superposition." is written below the diagram.

In the center, a sphere is divided into two hemispheres: the left half is labeled $|0\rangle$ and the right half is labeled $|1\rangle$. A central symbol Ψ represents the wavefunction. The text "quantum superposition" is written above the sphere.

To the right, a hand holds a small glowing sphere. The text "measurement/observation" is written above it, and "state collapse → random outcome" is written below it. The text "We write $|\psi\rangle \sim |0\rangle + |1\rangle$ " is centered at the bottom.

PHYC90045 Introduction to Quantum Computing



MONASH UNIVERSITY
MELBOURNE

Quantum entanglement

Imagine two qubits. We can prepare two distinct classes of overall state:

a) Separable state: independent quantum superpositions

$$\text{qubit-1} \sim |0\rangle + |1\rangle$$

$$\text{qubit-2} \sim |0\rangle + |1\rangle$$

(NB. ignore normalisation for now)

Or we write $|0\rangle + |1\rangle \times |0\rangle + |1\rangle \rightarrow |0\rangle|0\rangle + |0\rangle|1\rangle + |1\rangle|0\rangle + |1\rangle|1\rangle$

Measurement outcomes for each qubit: random and independent

b) An entangled state: e.g. in above notation $|0\rangle|1\rangle + |1\rangle|0\rangle$

Measurement on qubit-1:

$$\text{qubit-1} = |0\rangle \text{ (random)} \rightarrow \text{qubit-2} = |1\rangle$$

$$\text{qubit-1} = |1\rangle \text{ (random)} \rightarrow \text{qubit-2} = |0\rangle$$

(and vice-versa if we first measured qubit-2)

Measurement outcomes for e.g. qubit-1 are random, but result of qubit-2 depends on qubit-1 outcome → the qubits are somehow connected...

PHYC90045 Introduction to Quantum Computing

Multiple qubits and binary representation

Basic representation of binaries as quantum information:

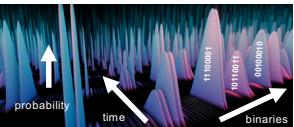


Independent quantum superpositions → superposition over N -bit binaries $|000\dots0\rangle, \dots, |111\dots1\rangle$



Not very useful... measurement of qubits collapses to one random N -bit string

Quantum computation: qubits interact to create complex superpositions and entangled states

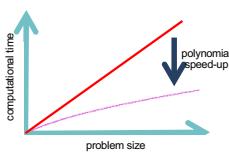
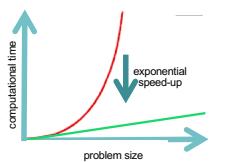



PHYC90045 Introduction to Quantum Computing

Types of quantum computers

Broadly, there are two classes of quantum computers:

- Intermediate quantum computers (IQC)**
 - medium-scale system ($10 \sim 1000$ qubits)
 - simplified control and error correction
 - potentially polynomial speed-up (e.g. \sqrt{CPU})
 - optimisation, machine learning, chemistry,...
 - pathway to full-scale universal QC...
- Full universal quantum computers (UQC)**
 - large-scale system (>1000 qubits)
 - high redundancy for quantum error correction
 - potentially exponential speed-up for some problems (e.g. Shor's factoring algorithm)
 - large class of problems: financial, data-base analysis, security, bio-molecular simulation
 - polynomial to exponential speed-up

PHYC90045 Introduction to Quantum Computing

Timeline and state of the art

Random QC → simulate with all 2^N binaries: max $N = 50$ qubits (peta bytes)

Quantum advantage 100-1000 qubits?

Largest simulation of a QC: quantum factoring for 60 qubits (exa to tera bytes)*

Hardware race: IBM, Google, Intel, D-Wave, Rigetti, Microsoft, SQC,...

Universal QC: (millions of qubits) poly to exponential speed-up on various problems

*A. Dang et al., arXiv:1712.07311





In the meantime, the era of quantum software and app development has begun...

PHYC90045 Introduction to Quantum Computing



1.2 Qubits: mathematical preliminaries I

PHYC90045
Lecture 1

PHYC90045 Introduction to Quantum Computing



Superposition, stochastic measurements

Recap: unlike bits which are in a definite state (0 or 1), a qubit can be a *superposition*



e.g. an atom in both "0" and "1" even though still only one electron (maths: more to it)

If we measure which state qubit is in, we will get a *probabilistic* outcome of "0" or "1".

e.g. If we prepare a qubit in an **equal** (50:50) superposition and measure:

Prepare/repeat same qubit many times → { 50% of the time, "1" will be measured
50% of the time, "0" will be measured

Superpositions are not always equal. Another state could lead to different probabilities, eg.

{ 80% of the time, "1" will be measured (prob = 0.8)
20% of the time, "0" will be measured (prob = 0.2)

Here we will briefly develop the **mathematical framework** to describe qubits.

PHYC90045 Introduction to Quantum Computing



Linear Algebra and Dirac notation

- A lot of quantum mechanics comes down to linear algebra (matrices and vectors), but uses a slightly different notation introduced by Dirac:

$|\psi\rangle$ ← A "ket" is a member of a **linear vector space** which represents *the state* of a qubit.

We write the general state of a qubit in ket notation as:

$$|\psi\rangle = a_0 |0\rangle + a_1 |1\rangle$$

where a_0 and a_1 are in general complex amplitudes.

PHYC90045 Introduction to Quantum Computing

Complex numbers: basics

$|\psi\rangle = a_0 |0\rangle + a_1 |1\rangle$ where a_0 and a_1 are in general complex amplitudes.

Complex numbers recap: $i = \sqrt{-1}$, and so $i^2 = -1$

$z = x + iy$ "Real" part is denoted by $\text{Re}[z] = x$
 $z = x + iy$ "Imaginary" part is denoted by $\text{Im}[z] = y$

Addition: $z_1 + z_2 = x_1 + iy_1 + x_2 + iy_2 = (x_1 + x_2) + i(y_1 + y_2)$

Multiplication: $z_1 z_2 = (x_1 + iy_1)(x_2 + iy_2) = x_1 x_2 + ix_1 y_2 + iy_1 x_2 - y_1 y_2$

Conjugate: $z = x + iy \rightarrow z^* = x - iy$

$z z^* = (x + iy)(x - iy) = x^2 - ixy + ixy + y^2 = x^2 + y^2$

i.e. $z z^* = x^2 + y^2 = |z|^2$ i.e. $|z| = \sqrt{x^2 + y^2}$

PHYC90045 Introduction to Quantum Computing

Complex numbers: polar notation

$z = x + iy$ in terms of $x = \text{Re}[z]$ and $y = \text{Im}[z]$

Polar: $z = |z| \cos \theta + i|z| \sin \theta = |z| e^{i\theta}$

NB identity: $e^{i\theta} = \cos \theta + i \sin \theta$

In the QUI, we use "polar notation". e.g. for amplitudes in the state:

$|\psi\rangle = a_0 |0\rangle + a_1 |1\rangle$ the amplitudes a_0 and a_1 are complex numbers.

Ignoring subscripts, consider a complex amplitude a .

$a = \text{Re}[a] + i \text{Im}[a] = |a| e^{i\theta} \rightarrow |a| = \sqrt{\text{Re}[a]^2 + \text{Im}[a]^2}$
 $\theta = \tan^{-1}(\text{Im}[a]/\text{Re}[a])$

$|\psi\rangle = a_0 |0\rangle + a_1 |1\rangle \rightarrow |\psi\rangle = |a_0| e^{i\theta_0} |0\rangle + |a_1| e^{i\theta_1} |1\rangle$

PHYC90045 Introduction to Quantum Computing

Linear Algebra and Dirac notation

$|\psi\rangle = a_0 |0\rangle + a_1 |1\rangle$

For qubits we can use column vectors to represent a convenient basis for kets:

$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
 $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
 Computational basis states

$a_0 |0\rangle + a_1 |1\rangle = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix}$
 $a_0, a_1 \in \mathbb{C}$
 General qubit state
 a_0 and a_1 are "amplitudes"

PHYC90045 Introduction to Quantum Computing

Dual vectors

$\langle \psi |$ A “bra” is a **row vector**.

For a qubit state,
 $|\psi\rangle = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \quad a_0, a_1 \in \mathbb{C}$

we define the corresponding *dual vector* to be:
 $\langle \psi | = [a_0^* \quad a_1^*]$

PHYC90045 Introduction to Quantum Computing

Measurement and quantum amplitudes

- In quantum mechanics the outcomes of measurements are probabilistic
- Qubits can be in **superpositions**
 $a_0|0\rangle + a_1|1\rangle = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix}$
- If we were to measure (in the computational basis) we would randomly measure “0” with probability:
 $|a_0|^2$
 or “1” with probability,
 $|a_1|^2$
- Since probabilities must sum to 1, all qubit states are normalised:
 $|a_0|^2 + |a_1|^2 = 1$

PHYC90045 Introduction to Quantum Computing

Amplitudes in the QUI

Quantum mechanics represents the *wave function*. Complex numbers represent the amplitude *and phase* of this wave.

$|\psi\rangle = a_0|0\rangle + a_1|1\rangle \rightarrow |\psi\rangle = |a_0|e^{i\theta_0}|0\rangle + |a_1|e^{i\theta_1}|1\rangle$

Recall: $a = \text{Re}[a] + i\text{Im}[a] = |a|e^{i\theta} \rightarrow |a| = \sqrt{\text{Re}[a]^2 + \text{Im}[a]^2}$
 $\theta = \tan^{-1}(\text{Im}[a]/\text{Re}[a])$
 $e^{i\theta} = \cos\theta + i\sin\theta$

In the QUI, phase is represented using the phase wheel colour map, and probability by histogram, e.g. two different single-qubit states:

The figure shows four phase wheels, each with a color gradient from red to purple. The first wheel has a central dot at 0° and is labeled $\theta_0 = 0$. The second wheel has a central dot at 45° and is labeled $\theta_1 = \pi/4$. The third wheel has a central dot at 90° and is labeled $\theta_0 = \pi/2$. The fourth wheel has a central dot at 135° and is labeled $\theta_1 = 3\pi/4$. To the left of the wheels is a vertical blue arrow pointing upwards, labeled "Probability". Above the wheels are labels: $|a_0|^2$, $|a_0|e^{i\theta_0}|0\rangle$, $|a_1|^2$, and $|a_1|e^{i\theta_1}|1\rangle$.

PHYC90045 Introduction to Quantum Computing

Inner Product

$\langle \psi | \phi \rangle$ A “braket” is an **inner product** (analogous to dot product for vectors in 3D)

For two quantum states $|\psi\rangle = \begin{bmatrix} a \\ b \end{bmatrix}$, $|\phi\rangle = \begin{bmatrix} c \\ d \end{bmatrix}$

We can define an inner product between them

$$\langle \psi | \phi \rangle \equiv \langle \psi | \phi \rangle$$

$$= [a^* \ b^*] \begin{bmatrix} c \\ d \end{bmatrix}$$

$$= a^*c + b^*d$$

PHYC90045 Introduction to Quantum Computing

Outer Product

$|\psi\rangle \langle \phi|$ is an **outer product**

For two quantum states $|\psi\rangle = \begin{bmatrix} a \\ b \end{bmatrix}$, $|\phi\rangle = \begin{bmatrix} c \\ d \end{bmatrix}$

We can define an inner product between them:

$$|\psi\rangle \langle \phi| = \begin{bmatrix} a \\ b \end{bmatrix} \otimes [c^* \ d^*]$$

$$= \begin{bmatrix} ac^* & ad^* \\ bc^* & bd^* \end{bmatrix}$$

PHYC90045 Introduction to Quantum Computing

Orthogonality

Two states are **orthogonal** if their inner product is zero

$$\langle \psi | \phi \rangle = 0$$

“Z-basis” (computational basis)

For $|0\rangle$ and $|1\rangle$

$$\langle 0 | 1 \rangle = [1 \ 0] \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0$$

Computational basis states are orthogonal

“X-basis” (+/- states)

For

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

$$|-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$\langle + | - \rangle = \frac{1}{2} [1 \ 1] \begin{bmatrix} 1 \\ -1 \end{bmatrix} = 0$$

These states are also orthogonal

PHYC90045 Introduction to Quantum Computing

Qubit state after measurement



Measurements in quantum mechanics necessarily *disturb the measured system*.

If we make a measurement in quantum mechanics, we disturb the system, and have to update the wave function:

$$|\psi\rangle \rightarrow |\psi'\rangle$$

If measure the state "0", the wave function becomes

$$|\psi'\rangle = |0\rangle$$

If measure the state "1", the wave function becomes

$$|\psi'\rangle = |1\rangle$$

This dramatic change of the state is known as wave function "collapse". In the first Lab we will see how we can use this fact to detect an eavesdropper.

PHYC90045 Introduction to Quantum Computing

Projective Operators



Consider a qubit in the state: $|\psi\rangle = a_0|0\rangle + a_1|1\rangle$

In the computational basis, we can define two projectors:

$$P_0 = |0\rangle\langle 0| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

$$P_1 = |1\rangle\langle 1| = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

The projection operators separate out the basis state in question, e.g.:

$$P_0|\psi\rangle = |0\rangle\langle 0| (a_0|0\rangle + a_1|1\rangle) = a_0|0\rangle\langle 0|0\rangle + a_1|0\rangle\langle 0|1\rangle = a_0|0\rangle$$

Or in matrix representation: $P_0|\psi\rangle = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} a_0 \\ 0 \end{pmatrix}$

PHYC90045 Introduction to Quantum Computing

Projective Measurement



The probability of measurement outcome in terms of projection operators is:

$$p(0) = \langle\psi|P_0|\psi\rangle = a_0^*a_0 = |a_0|^2 \quad p(1) = \langle\psi|P_1|\psi\rangle = a_1^*a_1 = |a_1|^2$$

The state after measurement 'collapses' to: $|\psi'\rangle = \frac{P_m|\psi\rangle}{\sqrt{p(m)}}$

Consider measurement on a state $|\psi\rangle = a_0|0\rangle + a_1|1\rangle \rightarrow |\psi'\rangle$

If "0" is measured, the resulting state is: $|\psi'\rangle = \frac{P_0|\psi\rangle}{|a_0|} = \frac{a_0|0\rangle}{|a_0|} = |0\rangle$

If "1" is measured, the resulting state is: $|\psi'\rangle = \frac{P_1|\psi\rangle}{|a_1|} = \frac{a_1|0\rangle}{|a_1|} = |1\rangle$

(up to global phase)

PHYC90045 Introduction to Quantum Computing

QUI: measurement operation

In the QUI the measurement operation looks like this:

By default, measurements are made in the computational basis (ie. 0 or 1).

When you run the circuit, the QUI will randomly select a measurement outcome based on the amplitudes of the state at that point:

$|\psi\rangle = a_0|0\rangle + a_1|1\rangle$

In some circuit diagrams notation is:

PHYC90045 Introduction to Quantum Computing

Linear Dependence

A set of vectors is *linearly dependent* if you can write

$$a_1|\psi_1\rangle + a_2|\psi_2\rangle + \dots = 0$$

with $a_1 \neq 0, a_2 \neq 0, \dots$

A set of vectors is *linearly independent* if they are not linearly dependent.

$\left\{ |0\rangle, |1\rangle, \frac{|0\rangle + |1\rangle}{\sqrt{2}} \right\}$ are linearly dependent

$\{|0\rangle, |1\rangle\}$ are linearly independent

$\left\{ \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right\}$ are also linearly independent

PHYC90045 Introduction to Quantum Computing

Basis states

Every possible qubit state can be expressed as a linear combination of two linearly independent vectors.

Every vector space is spanned by d linearly independent vectors. This set of vectors is known as a **basis**. d is the **dimension** of the vector space.

$\{|0\rangle, |1\rangle\}$ The computational, or "Z basis"

$\left\{ \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right\}$ The Hadamard (+/-) or "X basis"

Every qubit state can be expressed in Z-basis as: $a_0|0\rangle + a_1|1\rangle$

Or in the X-basis as as: $a_+ \frac{|0\rangle + |1\rangle}{\sqrt{2}} + a_- \frac{|0\rangle - |1\rangle}{\sqrt{2}}$

(more on this later)

PHYC90045 Introduction to Quantum Computing

Week 1



Lecture 1

- 1.1 Non-technical overview of the quantum world (& QC)
- 1.2 Qubits: mathematical preliminaries I

Lecture 2

- 2.1 Qubits: mathematical preliminaries II
- 2.2 Single qubit logic gates

Lab 1

QUI, Single qubits and logic gates

PHYC90045 Introduction to Quantum Computing

Week 1



Lecture 1

- 1.1 Non-technical overview of the quantum world (& QC)
- 1.2 Qubits: mathematical preliminaries I

Lecture 2

- 2.1 Qubits: mathematical preliminaries II
- 2.2 Single qubit logic gates

Lab 1

QUI, Single qubit gates, BB84 protocol

PHYC90045 Introduction to Quantum Computing



2.1 Qubits: mathematical preliminaries II

PHYC90045
Lecture 2

PHYC90045 Introduction to Quantum Computing

Lecture 2 overview



In this lecture:

2.1 Qubits: mathematical preliminaries II

- Operations
- Expectations values
- The Bloch sphere
- Pauli matrices and the axes of the Bloch sphere

2.2 Single qubit operations

- Qubit operations as rotations on the Bloch sphere
- X, Y, Z, H, S, T gate operations, QUI representation
- Arbitrary rotation gate
- (Euler angle rotations)
- Quantum circuit diagrams
- Using the QUI

NB. A lot of maths detail here, but you will gain familiarity/practice in the Labs using QUI

PHYC90045 Introduction to Quantum Computing

Operations

In quantum mechanics, *unitary* operators acting on quantum states produce new quantum states. These operators can be described by *unitary* matrices.

The new state is given by: $|\psi'\rangle = U |\psi\rangle$

Unitary operations are ones for which: $U^\dagger U = I$

Where the dagger represents taking the transpose (t) and complex conjugate ($*$).

$$U^\dagger = U^{t*}$$

In quantum mechanics, all unitary operations are **reversible**.

It's possible to efficiently express every classical computation using equivalent reversible logic gates, but there can be a cost in terms of additional bits and operations.

PHYC90045 Introduction to Quantum Computing

Operations Don't Commute!

For operators (e.g. matrices), remember
 $AB \neq BA$
Order matters!

PHYC90045 Introduction to Quantum Computing

Expectation Values

The *expectation value* of an operator is given by:

$$\langle A \rangle = \langle \psi | A | \psi \rangle$$

Some operators correspond to physical observables – the expectation value then gives the average value of that quantity when measured in a given state.

For example, consider measuring the total energy of the system represented by the energy operator, i.e. the "Hamiltonian" \mathcal{H} which in matrix representation is

$$\mathcal{H} = \begin{pmatrix} E_0 & 0 \\ 0 & E_1 \end{pmatrix}$$

Expectation values:

$$\langle 0 | \mathcal{H} | 0 \rangle = E_0 \quad \langle 1 | \mathcal{H} | 1 \rangle = E_1$$

PHYC90045 Introduction to Quantum Computing

Expectation values in a superposition state

The University of Melbourne

For example, consider measuring the total energy of the system

$$\mathcal{H} = \begin{pmatrix} E_0 & 0 \\ 0 & E_1 \end{pmatrix}$$

For the equal superposition state:

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

The expectation value is:

$$\langle + | H | + \rangle = \frac{E_0 + E_1}{2}$$

i.e. in this superposition (X basis) the energy is an average of the two values, and it generalizes for other superpositions (ex).

PHYC90045 Introduction to Quantum Computing

The Bloch Sphere

The University of Melbourne

A convenient geometric representation of single qubit states is the Bloch sphere:

PHYC90045 Introduction to Quantum Computing

Polar co-ordinates and global phase

The University of Melbourne

Recall, arbitrary qubit state: $|\psi\rangle = a_0|0\rangle + a_1|1\rangle = |a_0|e^{i\theta_0}|0\rangle + |a_1|e^{i\theta_1}|1\rangle$

$$|a_0|^2 + |a_1|^2 = 1$$

Which we can rearrange as:

$$|\psi\rangle = e^{i\theta_0}(|a_0| |0\rangle + |a_1| e^{i(\theta_1 - \theta_0)} |1\rangle) = e^{i\theta_{\text{global}}} \left(\cos \frac{\theta_B}{2} |0\rangle + \sin \frac{\theta_B}{2} e^{i\phi_B} |1\rangle \right)$$

The global phase is unimportant, only the relative phase matters (for now). The real variables θ_B and ϕ_B dictate the position of this state on the Bloch sphere.

$|\psi\rangle = a_0|0\rangle + a_1|1\rangle$

$$\rightarrow \cos \frac{\theta_B}{2} |0\rangle + \sin \frac{\theta_B}{2} e^{i\phi_B} |1\rangle$$

$\sqrt{|a_0|^2 + |a_1|^2} = 1$

PHYC90045 Introduction to Quantum Computing

States on the Bloch sphere

$|\psi\rangle = a_0 |0\rangle + a_1 |1\rangle \rightarrow \cos \frac{\theta_B}{2} |0\rangle + \sin \frac{\theta_B}{2} e^{i\phi_B} |1\rangle$

PHYC90045 Introduction to Quantum Computing

The Pauli Matrices

Operations transform states and are equivalent to moving around on the Bloch sphere.

Qubits: the most important operators are the generators of rotations under the SU(2) group. In matrix representation these are the Pauli matrices:

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

Property – all square to identity (check):

$$X^2 = X X = I \quad Y^2 = Y Y = I \quad Z^2 = Z Z = I$$

Odd and even powers:

$$X^3 = X X X = IX = X \quad X^4 = X X X X = II = I \quad \text{etc}$$

Why is this important? – quantum logic gates are ultimately written as exponentials of Pauli operators...let's see how this works...

PHYC90045 Introduction to Quantum Computing

Exponential of a Matrix

We can define the exponential of a matrix using the power series for the exponential:

$$\exp(A) = I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \frac{A^4}{4!} + \dots$$

The exponential of a Pauli Matrix (eg. X) including an angle parameter:

$$\begin{aligned} \exp(i\theta X) &= I + i\theta X - \frac{\theta^2}{2!} X^2 - \frac{i\theta^3}{3!} X^3 + \frac{\theta^4}{4!} X^4 + \dots \\ &= I + i\theta X - \frac{\theta^2}{2!} I - \frac{i\theta^3}{3!} X + \frac{\theta^4}{4!} I + \dots \\ &= \left(1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} + \dots\right) I + i \left(\theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} + \dots\right) X \\ &= \cos \theta I + i \sin \theta X \end{aligned}$$

(used power series for cos and sin)

PHYC90045 Introduction to Quantum Computing

General exponentiation of Paulis



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

We just proved: $\exp(i\theta X) = I \cos \theta + i X \sin \theta$

To generalise we form a 2×2 matrix Σ from the Pauli matrices as: $\Sigma = \hat{\mathbf{n}} \cdot \mathbf{X}$

where $\hat{\mathbf{n}}$ is a unit spatial 3-vector and \mathbf{X} is a vector of Paulis: $\mathbf{X} = (X, Y, Z)$

$\exp(i\theta \hat{\mathbf{n}} \cdot \mathbf{X}) = I \cos \theta + i \hat{\mathbf{n}} \cdot \mathbf{X} \sin \theta$

$\hat{\mathbf{n}} = \frac{\mathbf{n}}{|\mathbf{n}|}$

e.g. $\hat{\mathbf{n}} = (1, 0, 0) \rightarrow \hat{\mathbf{n}} \cdot \mathbf{X} = (1, 0, 0) \cdot \mathbf{X} = X$

Recover previous result: $\exp(i\theta X) = I \cos \theta + i X \sin \theta$

NB. Lots of practice in labs!

PHYC90045 Introduction to Quantum Computing

Expectation of Pauli Matrices



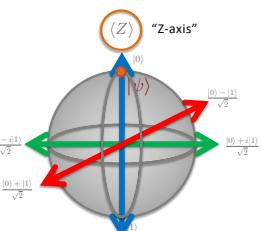
Recall: expectation value for an operator: $\langle A \rangle = \langle \psi | A | \psi \rangle$

For the state: $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$\langle X \rangle = [1 \ 0] \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 0$

$\langle Y \rangle = [1 \ 0] \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 0$

$\langle Z \rangle = [1 \ 0] \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 1$



"Z-axis"

PHYC90045 Introduction to Quantum Computing

Expectation of Pauli Matrices



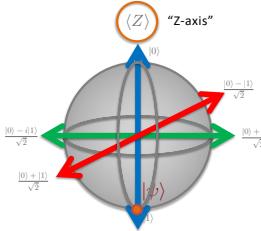
Recall: expectation value for an operator: $\langle A \rangle = \langle \psi | A | \psi \rangle$

For the state: $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

$\langle X \rangle = [0 \ 1] \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0$

$\langle Y \rangle = [0 \ 1] \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0$

$\langle Z \rangle = [0 \ 1] \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = -1$



"Z-axis"

PHYC90045 Introduction to Quantum Computing

Expectation values of Pauli matrices

Recall: expectation value for an operator: $\langle A \rangle = \langle \psi | A | \psi \rangle$

For the state: $|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$

$\langle X \rangle = \frac{1}{2} [1 \ 1] \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 1$

$\langle Y \rangle = \frac{1}{2} [1 \ 1] \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 0$

$\langle Z \rangle = \frac{1}{2} [1 \ 1] \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 0$

...and so on.

PHYC90045 Introduction to Quantum Computing

Pauli matrices & Bloch sphere axes

The expectation values of the Pauli matrices (operators) define the axes on the Bloch sphere.

PHYC90045 Introduction to Quantum Computing

2.2 Single qubit operations

PHYC90045
Lecture 2

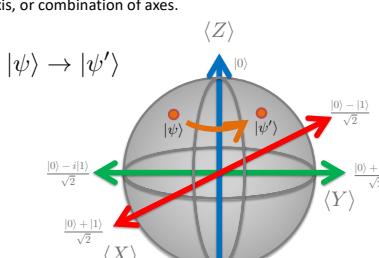
PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

Operations as rotations on Bloch sphere

Now we have the full Bloch sphere laid out, we will look at how single qubit operations have a convenient geometric interpretation as rotations about a specific axis, or combination of axes.



Recap: amplitudes in the QUI

Quantum mechanics represents the *wave function*. Complex numbers represent the amplitude *and phase* of this wave.

$$|\psi\rangle = a_0|0\rangle + a_1|1\rangle \rightarrow |\psi\rangle = |a_0|e^{i\theta_0}|0\rangle + |a_1|e^{i\theta_1}|1\rangle$$

Recall: $a = \text{Re}[a] + i\text{Im}[a] = |a|e^{i\theta} \rightarrow |a| = \sqrt{\text{Re}[a]^2 + \text{Im}[a]^2}$

$$\theta = \tan^{-1}(\text{Im}[a]/\text{Re}[a])$$

$$e^{i\theta} = \cos\theta + i\sin\theta$$

In the QUI, phase is represented using the phase wheel colour map, and probability by histogram, e.g. two different states:

PHYC90045 Introduction to Quantum Computing

X Gate (the X operator): π around X-axis



Circuit symbol: 

Matrix representation: $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

$X (a_0 |0\rangle + a_1 |1\rangle) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_0 \end{bmatrix}$

i.e. $X (a_0 |0\rangle + a_1 |1\rangle) = a_1 |0\rangle + a_0 |1\rangle$

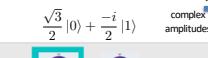
Action on ket states: $a_0 |0\rangle + a_1 |1\rangle \rightarrow a_1 |0\rangle + a_0 |1\rangle$

QUI example:

complex amplitudes

$\frac{\sqrt{3}}{2}|0\rangle + \frac{-i}{2}|1\rangle$

$\frac{-i}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle$





PHY90045 Introduction to Quantum Computing

Y Gate (the Y operator): π around Y-axis

Circuit symbol: 

Matrix representation: $Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$

$$Y(a_0|0\rangle + a_1|1\rangle) = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} -ia_1 \\ ia_0 \end{bmatrix}$$

i.e. $Y(a_0|0\rangle + a_1|1\rangle) = -ia_1|0\rangle + ia_0|1\rangle$

Action on ket states: $a_0|0\rangle + a_1|1\rangle \rightarrow -ia_1|0\rangle + ia_0|1\rangle$

QUI example:

$\frac{\sqrt{3}}{2}|0\rangle + \frac{-i}{2}|1\rangle$

complex amplitudes



$\frac{-1}{2}|0\rangle + \frac{i\sqrt{3}}{2}|1\rangle$



PHYC90045 Introduction to Quantum Computing

Z Gate (the Z operator): π around Z-axis

Circuit symbol:

Matrix representation: $Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

$$Z(a_0|0\rangle + a_1|1\rangle) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} a_0 \\ -a_1 \end{bmatrix}$$

i.e. $Z(a_0|0\rangle + a_1|1\rangle) = a_0|0\rangle - a_1|1\rangle$

Action on ket states: $a_0|0\rangle + a_1|1\rangle \rightarrow a_0|0\rangle - a_1|1\rangle$

QUI example:

$\frac{\sqrt{3}}{2}|0\rangle + \frac{-i}{2}|1\rangle$ $\frac{-1}{2}|0\rangle + \frac{i}{2}|1\rangle$

complex amplitudes

$\frac{\sqrt{3}}{2}|0\rangle + \frac{i}{2}|1\rangle$

Z GATE

Rotate around the Z axis by π radians.

Handwriting practice lines consisting of five horizontal lines for letter formation.

PHYC90045 Introduction to Quantum Computing

H Gate (the H operator): π around X+Z-axis



Circuit symbol: 

Matrix representation:
$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

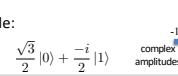
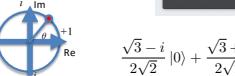
Action on ket states: $|0\rangle \rightarrow \frac{|0\rangle + |1\rangle}{\sqrt{2}}$ $|1\rangle \rightarrow \frac{|0\rangle - |1\rangle}{\sqrt{2}}$
 $a_0|0\rangle + a_1|1\rangle \rightarrow \frac{a_0 + a_1}{\sqrt{2}}|0\rangle + \frac{a_0 - a_1}{\sqrt{2}}|1\rangle$

QUI example:

complex amplitudes

$\frac{\sqrt{3}}{2}|0\rangle + \frac{-i}{2}|1\rangle$

$\frac{\sqrt{3}-i}{2\sqrt{2}}|0\rangle + \frac{\sqrt{3}+i}{2\sqrt{2}}|1\rangle$

PHYC90045 Introduction to Quantum Computing

Hadamard gate as change of basis

Normally, we talk about making measurements in the computational Z-basis: "0", "1"
NB. "Z-basis" because these are eigenstate of Z: $Z|0\rangle = +|0\rangle$, $Z|1\rangle = -|1\rangle$

But we could equally think of making measurements in the X-basis: "+" state or "-" state. The Hadamard gate takes us from the Z-basis to the X-basis (and vice-versa):

$$H|+\rangle = |0\rangle \quad H|-\rangle = |1\rangle \quad H(a_+|+\rangle + a_-|-\rangle) = a_+|0\rangle + a_-|1\rangle$$

A Hadamard gate directly before a measurement "changes the basis" of the measurement from 0/1 to +/-.

NB. The +/- states are eigenstates of the X operator with eigenvalues +/- 1:

$$X|\pm\rangle = \frac{X(|0\rangle \pm |1\rangle)}{\sqrt{2}} = \frac{(|1\rangle \pm |0\rangle)}{\sqrt{2}} = \frac{\pm(|0\rangle \pm |1\rangle)}{\sqrt{2}} \rightarrow X|\pm\rangle = \pm|\pm\rangle$$

Similarly, the Hadamard gate can be considered to change basis back again:

$$H|0\rangle = |+\rangle \quad H|1\rangle = |-\rangle$$

PHYC90045 Introduction to Quantum Computing

S Gate (the S operator): $\pi/2$ rotation

Circuit symbol: 

Matrix representation: $S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$

$S(a_0|0\rangle + a_1|1\rangle) = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} a_0 \\ ia_1 \end{bmatrix}$
i.e. $S(a_0|0\rangle + a_1|1\rangle) = a_0|0\rangle + ia_1|1\rangle$

Action on ket states: $a_0|0\rangle + a_1|1\rangle \rightarrow a_0|0\rangle + ia_1|1\rangle$

QUI example:



The diagram shows a complex plane with axes labeled "Re" and "Im". A vector representing the state $a_0|0\rangle + a_1|1\rangle$ is rotated by $\pi/2$ radians around the origin, moving from the first quadrant to the second quadrant. Below the plane, two qubits are shown: the first qubit's state vector is rotated by $\pi/2$ in the complex plane, while the second qubit's state vector remains unchanged.

PHYC90045 Introduction to Quantum Computing

T Gate (the T operator): $\pi/4$ rotation

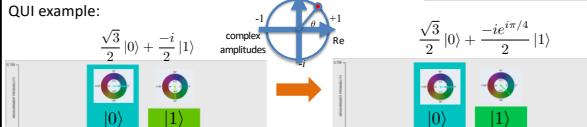
Circuit symbol: 

Matrix representation: $T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$

$T(a_0|0\rangle + a_1|1\rangle) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} a_0 \\ e^{i\pi/4}a_1 \end{bmatrix}$
i.e. $T(a_0|0\rangle + a_1|1\rangle) = a_0|0\rangle + e^{i\pi/4}a_1|1\rangle$

Action on ket states: $a_0|0\rangle + a_1|1\rangle \rightarrow a_0|0\rangle + e^{i\pi/4}a_1|1\rangle$

QUI example:



The diagram shows a complex plane with axes labeled "Re" and "Im". A vector representing the state $a_0|0\rangle + a_1|1\rangle$ is rotated by $\pi/4$ radians around the origin, moving from the first quadrant to the second quadrant. Below the plane, two qubits are shown: the first qubit's state vector is rotated by $\pi/4$ in the complex plane, while the second qubit's state vector remains unchanged.

The QUI provides the ability to code an arbitrary rotation gate:

Arbitrary axis rotation

The “parameters” menu allows you to specify any axis and angle:

Cartesian cords for axis of rotation \mathbf{n}

Angle of rotation θ_R about \mathbf{n}

Global phase θ_B : details in next slide

PHYC90045 Introduction to Quantum Computing


 THE UNIVERSITY OF
MELBOURNE

Arbitrary axis rotation: maths

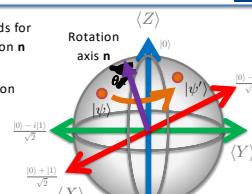
ARBITRARY ROTATION GATE - PARAMETERS

Rotation axis	$x = 1$	$y = 1$	$z = 1$
$\theta =$ rotation angle (radians)	$1 - \pi$		
Global phase (radians)	$0 - \pi$		

Cartesian cords for axis of rotation \hat{n}

Angle of rotation θ_R about \hat{n}

Global phase θ_g



Mathematically, the operation is: $|\psi'\rangle = R_{\hat{n}}(\theta_R) |\psi\rangle$

Where the rotation operator is:

$$R_{\hat{n}}(\theta_R) = e^{i\theta_R} e^{-i\frac{\theta_R}{2} \hat{n} \cdot \mathbf{X}} = e^{i\theta_R} \left(I \cos \frac{\theta_R}{2} - i \hat{n} \cdot \mathbf{X} \sin \frac{\theta_R}{2} \right)$$

$$\langle X \rangle - \text{axis: } \hat{n} = (1, 0, 0), \quad \langle Y \rangle - \text{axis: } \hat{n} = (0, 1, 0), \quad \langle Z \rangle - \text{axis: } \hat{n} = (0, 0, 1)$$

$$\langle X + Z \rangle - \text{axis: } \hat{n} = (1, 0, 1)/\sqrt{2}$$

PHY90045 Introduction to Quantum Computing



THE UNIVERSITY OF
MELBOURNE

Arbitrary axis rotation – example

Let's see how it works by comparing with a Z gate.

Rotation about Z-axis by π radians – but a global phase choice is made.

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad a_0 |0\rangle + a_1 |1\rangle \rightarrow a_0 |0\rangle - a_1 |1\rangle$$

Rotation about Z-axis by π radians, using the R-gate.

$$R_{\hat{\mathbf{n}}}(\theta_R) = e^{i\theta_R} \left(I \cos \frac{\theta_R}{2} - i \hat{\mathbf{n}} \cdot \mathbf{X} \sin \frac{\theta_R}{2} \right)$$

$$\hat{\mathbf{n}} = (0, 0, 1) \quad \theta_R = \pi \quad \hat{\mathbf{n}} \cdot \mathbf{X} = (0, 0, 1) \cdot (X, Y, Z) = Z$$

$$I \cos \frac{\theta_R}{2} - i \hat{\mathbf{n}} \cdot \mathbf{X} \sin \frac{\theta_R}{2} \rightarrow \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \text{cancel } \frac{\pi}{2} - i \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \sin \frac{\pi}{2} = -i \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = -iZ$$

Set global phase to $\theta_g = \pi/2$ radians: $e^{i\pi/2} = i$

$$R_Z(\pi)|_{\theta_g=\pi/2} = Z$$

PHYS90045 Introduction to Quantum Computing

Note angles in context – abundant use of θ

Phase angle of complex amplitudes in polar coordinates:

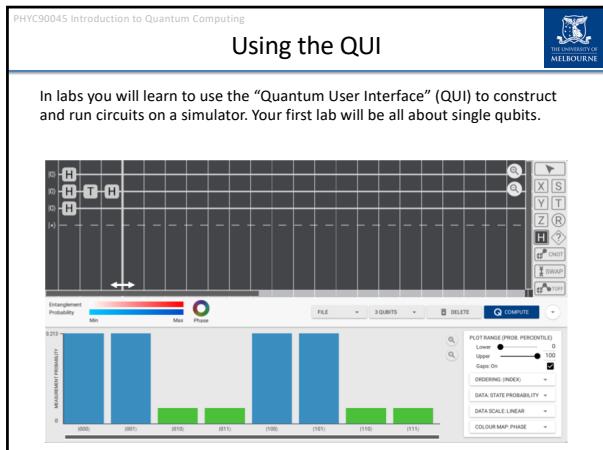
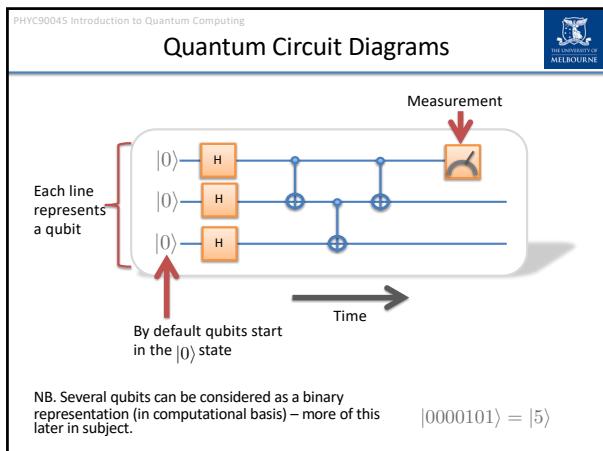
$$|\psi\rangle = a_0 |0\rangle + a_1 |1\rangle \rightarrow |\psi\rangle = |a_0|e^{i\theta_0} |0\rangle + |a_1|e^{i\theta_1} |1\rangle$$

Angle specifying position on the Bloch sphere:

$$|\psi\rangle = a_0 |0\rangle + a_1 |1\rangle \rightarrow \cos \frac{\theta_B}{2} |0\rangle + \sin \frac{\theta_B}{2} e^{i\phi_B} |1\rangle$$

Angle of rotation of a qubit state on the Bloch sphere about a specified axis, \mathbf{n} :

$$|\psi'\rangle = R_{\mathbf{n}}(\theta_R) |\psi\rangle \quad R_{\mathbf{n}}(\theta_R) = e^{i\theta_R} \left(I \cos \frac{\theta_R}{2} - i \mathbf{n} \cdot \mathbf{X} \sin \frac{\theta_R}{2} \right)$$



© L. Hollenberg, C. Hill 2019

PHYC90045 Introduction to Quantum Computing

Week 1 so far



Lecture 1

- 1.1 Non-technical overview of the quantum world (& QC)
- 1.2 Qubits: mathematical preliminaries I

Lecture 2

- 2.1 Qubits: mathematical preliminaries II
- 2.2 Single qubit logic gates

Lab 1

QUI, Single qubits and logic gates

PHYC90045 Introduction to Quantum Computing

Week 2



Lecture 3

- 3.1 Two qubit systems and operations
- 3.2 Entanglement

Lecture 4

- 4.1 Dense coding
- 4.2 Teleportation

Lab 2

Two qubit operations, entanglement, dense coding, teleportation

PHYC90045 Introduction to Quantum Computing



3.1 Two qubit systems and operations

PHYC90045
Lecture 3

PHYC90045 Introduction to Quantum Computing

Lecture overview



In this lecture:

3.1 Two qubit systems and operations

- Multiple qubits and binary numbers
- Linear algebra of two-qubit systems
- Two-qubit logic gates
- Universality

3.2 Entanglement

- Separable states
- Entangled states
- Entropy of entanglement
- Entanglement in the QM

- Reiffel, Chapter 3
- Kaye, 2.6, 4.1-4.2
- Mike and Ike, 1.2.2, 1.3.2-1.3.4

PHYC90045 Introduction to Quantum Computing

Recap: qubits and binary numbers

The University of Melbourne

Computers represent digits as binary numbers. Similarly, we can think as the state of several qubits as a binary digits.

For example, the number 5 can be represented in binary as 101, and this can be encoded directly in the state of three individual atoms.

This lecture we will talk about multi-qubit systems (i.e. 2-qubit systems).

PHYC90045 Introduction to Quantum Computing

Two qubits: tensor product

The University of Melbourne

Two atoms, each with one electron in a superposition of the bit states:

$$|\psi\rangle = a|0\rangle + b|1\rangle \quad |\phi\rangle = c|0\rangle + d|1\rangle$$

Then the joint state of both atoms is:

$|\psi\rangle \otimes |\phi\rangle$

Tensor product!

PHYC90045 Introduction to Quantum Computing

Tensor product

The University of Melbourne

Two atoms, each with one electron in a superposition of the bit states:

$$|\psi\rangle = a|0\rangle + b|1\rangle \quad |\phi\rangle = c|0\rangle + d|1\rangle$$

For these two atoms in the states indicated:

$$\begin{aligned} |\psi\rangle \otimes |\phi\rangle &= (a|0\rangle + b|1\rangle) \otimes (c|0\rangle + d|1\rangle) \\ &= ac|0\rangle \otimes |0\rangle + ad|0\rangle \otimes |1\rangle + bc|1\rangle \otimes |0\rangle + bd|1\rangle \otimes |1\rangle \\ &= ac|00\rangle + ad|01\rangle + bc|10\rangle + bd|11\rangle \end{aligned}$$

PHYS90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

Tensor product of vectors

$$\begin{bmatrix} a \\ b \end{bmatrix} \otimes \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} ac \\ ad \\ bc \\ bd \end{bmatrix}$$

00 amplitude
 01 amplitude
 10 amplitude
 11 amplitude



$$|\psi\rangle = a|0\rangle + b|1\rangle$$

$$|\phi\rangle = c|0\rangle + d|1\rangle$$

PHYC90045 Introduction to Quantum Computing

 THE UNIVERSITY OF
MELBOURNE

Tensor product of operators

Similarly, we can define a Kronecker tensor product of qubit operators:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \otimes \begin{bmatrix} m & n \\ p & q \end{bmatrix} = \begin{bmatrix} am & an & bm & bn \\ ap & aq & bp & bq \\ cm & cn & dm & dn \\ cp & cq & dp & dq \end{bmatrix}$$

PHYC90045 Introduction to Quantum Computing

THE UNIVERSITY OF MELBOURNE

Single qubit gates on multi-qubit systems

We can apply single-qubit operators to multi-qubit systems:

$$(X \otimes I) |0\rangle \otimes |0\rangle \rightarrow X_1 |00\rangle = |10\rangle$$

$$(I \otimes X) |0\rangle \otimes |0\rangle \rightarrow X_2 |00\rangle = |01\rangle$$

Simplest way to think of it: the subscript represents which qubit the operation is applied to.

To work out the operator we are applying in matrix representation, we use the Kronecker (tensor) product with the identity:

$$X \otimes I = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Identity is applied to second qubit where there's no operation

PHYC90045 Introduction to Quantum Computing

Two-qubit projective measurement

THE UNIVERSITY OF MELBOURNE

Examples of two-qubit projectors. Eg. For measuring the first qubit in the computational basis:

$$P_0 = |0\rangle\langle 0| \otimes I$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$P_1 = |1\rangle\langle 1| \otimes I$$

$$= \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

PHYC90045 Introduction to Quantum Computing

Two-qubit measurement & collapse

THE UNIVERSITY OF MELBOURNE

Measurement on a two-qubit state:

(1) Apply projector into the measured state $|\psi'\rangle = \frac{P_m |\psi\rangle}{\sqrt{p(m)}}$

(2) Renormalize the state

For example, consider the general two-qubit state: $|\psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$

If the first qubit were measured to be "0", apply P_0 and renormalize to get the collapsed state: $|\psi'\rangle = \frac{a|00\rangle + b|01\rangle}{\sqrt{|a|^2 + |b|^2}}$

If the first qubit were measured to be "1", apply P_1 and renormalize to get the collapsed state: $|\psi'\rangle = \frac{c|10\rangle + d|11\rangle}{\sqrt{|c|^2 + |d|^2}}$

normalization

Later (and Lab-2): this generalizes to measurements on multi-qubit states.

PHYC90045 Introduction to Quantum Computing

Multi-qubit states: binary and decimal

THE UNIVERSITY OF MELBOURNE

n qubits

$|0\rangle$ shorthand notation $|0\rangle \xrightarrow{\text{H}^{\otimes n}} |\psi\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} \dots \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}}$

$|0\rangle$ $|\psi\rangle = \left[\frac{1}{\sqrt{2}} \right]^n (|00\dots0\rangle + \dots + |11\dots1\rangle)$

i.e. even superposition over binary rep of integers: $i = 0$ to $2^n - 1$

In general we use two representations in the QM ($N = 2^n$):

"binary"

$|\psi\rangle = a_{0\dots00}|0\dots00\rangle + a_{0\dots01}|0\dots01\rangle + a_{0\dots10}|0\dots10\rangle + \dots + a_{1\dots1}|1\dots1\rangle$

"decimal"

$|\psi\rangle = a_0|0\rangle + a_1|1\rangle + a_2|2\rangle + \dots + a_{N-1}|N-1\rangle$

e.g. $a_{101}|101\rangle$

$|\psi\rangle = \sum_i a_i|i\rangle$

$a_i = |a_i|e^{i\theta_i}$

PHYC90045 Introduction to Quantum Computing

Two qubit logic gates: CNOT

The University of Melbourne

Two qubit gates can be constructed using an interaction between the two systems. Most important is the Controlled-NOT (CNOT) gate.

Symbol for "control"
Control qubit
Target qubit
Symbol for binary addition (flip)

How states transform: CNOT truth table

00>	→ 00>
01>	→ 01>
10>	→ 11>
11>	→ 10>

Rule: The target is flipped iff the control qubit is "1".

As a matrix:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$a|00> + b|01> + c|10> + d|11>$
 $\rightarrow a|00> + b|01> + d|10> + c|11>$

PHYC90045 Introduction to Quantum Computing

Example: CNOT on superposition

The University of Melbourne

$\alpha|0> + \beta|1>$
 $|0>$
 $|\psi>$ $|\psi'>$

Before the CNOT, the state is:
 $|\psi> = (\alpha|0> + \beta|1>) \otimes |0> = \alpha|00> + \beta|10>$

After the CNOT, the state is:
 $|\psi'> = \alpha|00> + \beta|11>$

PHYC90045 Introduction to Quantum Computing

Control Phase Gate

The University of Melbourne

Two qubit gates can be constructed using an interaction between the two systems.

Control qubit
Target qubit

How states transform:

00>	→ 00>
01>	→ 01>
10>	→ 10>
11>	→ - 11>

Rule: the phase of the target flipped iff the control qubit is "1".

As a matrix:

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

$a|00> + b|01> + c|10> + d|11>$
 $\rightarrow a|00> + b|01> + c|10> - d|11>$

Fun fact: CZ is diagonal so it doesn't matter which one you think of as control/target.

PHYC90045 Introduction to Quantum Computing

SWAP gate

A SWAP operation can be implemented using an interaction between the two qubits – the states of the two qubits are swapped (not the physical qubits).

Qubit 1

Qubit 2

How states transform:

$$\begin{aligned} |00\rangle &\rightarrow |00\rangle \\ |01\rangle &\rightarrow |10\rangle \\ |10\rangle &\rightarrow |01\rangle \\ |11\rangle &\rightarrow |11\rangle \end{aligned}$$

$$a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle \rightarrow a|00\rangle + c|01\rangle + b|10\rangle + d|11\rangle$$

As a matrix:

$$SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rule: the two qubits are swapped.

NB. Unlike CNOT, SWAP gates do not generate entanglement (but sqrt SWAP does!).

PHYC90045 Introduction to Quantum Computing

Toffoli gate

Double control-NOT:

Control qubit

Control qubit

Target qubit

How states transform:

$$\begin{aligned} |000\rangle &\rightarrow |000\rangle \\ |001\rangle &\rightarrow |001\rangle \\ |010\rangle &\rightarrow |010\rangle \\ |011\rangle &\rightarrow |011\rangle \\ |100\rangle &\rightarrow |100\rangle \\ |101\rangle &\rightarrow |101\rangle \\ |110\rangle &\rightarrow |111\rangle \\ |111\rangle &\rightarrow |110\rangle \end{aligned}$$

$$a|000\rangle + b|001\rangle + c|010\rangle + d|011\rangle \\ e|100\rangle + f|101\rangle + g|110\rangle + h|111\rangle \\ \rightarrow a|000\rangle + b|001\rangle + c|010\rangle + d|011\rangle \\ e|100\rangle + f|101\rangle + h|110\rangle + g|111\rangle$$

Rule: the target is flipped iff **both** the control qubits are in "1" state.

Toffoli gate plus NOT is universal for classical computation. It was used in the proof that classical computation can be made reversible!

PHYC90045 Introduction to Quantum Computing

Universality

In classical computing the NAND gate is universal, i.e. every Boolean function can be implemented as a sequence of NAND (NOT AND) gates:

In quantum computing every quantum circuit can be expressed as a sequence of:

CNOT

Single Qubit Rotations

PHYC90045 Introduction to Quantum Computing

Universality

In classical computing the NAND gate is universal, i.e. every Boolean function can be implemented as a sequence of NAND (NOT AND) gates:

In quantum computing every quantum circuit can be expressed as a sequence of:

CNOT Hadamard T Gate (Z-axis, $\pi/4$)

PHYC90045 Introduction to Quantum Computing

Construction for *any* two qubit unitary

Any two qubit gate can be decomposed into just 3 CNOTs and single qubit rotations:

PHYC90045 Introduction to Quantum Computing

Example

How can you decompose Toffoli as CNOTs and single qubit rotations?

$= \dots -|H\rangle \oplus |T\rangle \oplus |T\rangle \oplus |T\rangle \oplus |T\rangle \oplus |T\rangle -|H\rangle \dots$

PHYC90045 Introduction to Quantum Computing

 THE UNIVERSITY OF MELBOURNE

3.2 Entanglement

PHYC90045
Lecture 3

PHYC90045 Introduction to Quantum Computing

 THE UNIVERSITY OF MELBOURNE

Separable states



$$|\psi\rangle = a|0\rangle + b|1\rangle$$

$$|\phi\rangle = c|0\rangle + d|1\rangle$$

A separable state is one which can be written as

$$|\Phi\rangle = |\psi\rangle \otimes |\phi\rangle$$

All separable states (of two qubits) can be written as:

$$|\psi\rangle = ac|00\rangle + ad|01\rangle + bc|10\rangle + bd|11\rangle$$

PHYC90045 Introduction to Quantum Computing

 THE UNIVERSITY OF MELBOURNE

Examples of separable states

Consider the state:

$$|\psi\rangle = \frac{|00\rangle + |01\rangle}{\sqrt{2}}$$

It is *separable* because:

$$|\psi\rangle = |0\rangle \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

Consider the state:

$$|\psi\rangle = \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2}$$

It is also *separable* because:

$$|\psi\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

PHYC90045 Introduction to Quantum Computing

Constructing a Bell state

This is one of four states named after the physicist John Bell (who figured out how to experimentally explore reality of entanglement).

Consider the following circuit in the QUI:

Execution: $|00\rangle \xrightarrow{\text{H}} \frac{|00\rangle + |10\rangle}{\sqrt{2}} \xrightarrow{\text{CNOT}} \frac{|00\rangle + |11\rangle}{\sqrt{2}}$

Question: Is $\frac{|00\rangle + |11\rangle}{\sqrt{2}}$ separable?

PHYC90045 Introduction to Quantum Computing

Entanglement

Answer: No! We can never find a, b, c, d, i.e.

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}} \neq (a|0\rangle + b|1\rangle) \otimes (c|0\rangle + d|1\rangle)$$

A state which is not separable is called an **entangled** state.

Entanglement is a uniquely quantum mechanical property, with no direct classical analogue.

PHYC90045 Introduction to Quantum Computing

Entanglement Measure

We would like to have a measure of *how much* entanglement a state has. Some states are more entangled than others:

$ 00\rangle$	Not entangled, separable
$\sqrt{0.99} 00\rangle + \sqrt{0.01} 11\rangle$	Entangled, but close to a separable state
$\frac{1}{\sqrt{2}} 00\rangle + \frac{1}{\sqrt{2}} 11\rangle$	Maximally entangled

Entanglement is a type of correlation between two systems, say A and B.
To see how much correlation there is between A and B: We will measure B and ask how many bits of information (as measured by entropy) this can tell us about the state of A?

In the QUI we measure the degree of entanglement using an informatic “entropy” measure: *Entanglement Entropy (EE)*

PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

Entanglement in the QUI – time slider

The *time slider* is the vertical bar which moves left and right to show the quantum state at each time step. When there is entanglement it will show it.

The entanglement entropy (EE) is shown in a red colour scale between min and max values possible. Each segment corresponds to the entropy between the system of qubits above and below for that particular bi-partition.



A quantum circuit diagram with four horizontal lines representing qubits. The qubits are labeled qubit-1, qubit-2, qubit-3, and qubit-4 from top to bottom. The circuit consists of several operations: a Hadamard (H) on qubit-2, a Z gate on qubit-3, a T gate on qubit-3, and a Vx gate on qubit-4. A vertical red line, representing the 'time slider', is positioned between qubit-3 and qubit-4. Three orange arrows point from the text labels to this red line, indicating the segments of entanglement entropy between different partitions of the qubits.

qubit-1 |(0)

qubit-2 |(0) H

qubit-3 |(0) H Z

qubit-4 |(0) T Vx

Entanglement entropy between qubit 1 and qubits {2 & 3 & 4} partition

Entanglement entropy between qubits {1 & 2} and qubits {3 & 4} partitions

Entanglement entropy between qubit 4 and qubits {1 & 2 & 3} partition



 THE UNIVERSITY OF
 MELBOURNE

PHYS90045 Introduction to Quantum Computing

Aside: how we determine entanglement entropy

How much entanglement is present in a general state?

$$|\psi\rangle = a_{00} |00\rangle + a_{01} |01\rangle + a_{10} |10\rangle + a_{11} |11\rangle$$

Can be hard to tell. It's not in anything like product form. For that we will use SVD.
 Arrange as a matrix:

$$A = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix}$$

Taking Singular Value Decomposition (SVD):

$$A = \sum \lambda_i |\psi_i\rangle \langle \psi_i|$$

Allows us to express the state in this convenient form:

$$|\psi\rangle = \sum \lambda_i |\psi_i\rangle \langle \psi_i|$$

This form is known as the "Schmidt Decomposition"

PHYC90045 Introduction to Quantum Computing

Aside: how we determine entanglement entropy

Schmidt Decomposition:

$$|\psi\rangle = \sum_i \lambda_i |u_i\rangle |v_i\rangle$$

Several terms might have a singular value of 0. The number of non-zero terms is called the **Schmidt rank**.

If a state has a Schmidt rank of 1:

$$|\psi\rangle = |u_0\rangle \otimes |v_0\rangle$$

Then the state is separable, and not entangled.

If a state has a Schmidt rank greater than 1, then the state is entangled. Schmidt rank is a very coarse measure of entanglement. We would like a finer measure.



PHYC90045 Introduction to Quantum Computing

 THE UNIVERSITY OF MELBOURNE

Aside: how we determine entanglement entropy

$$|\psi\rangle = \sum \lambda_i |u_i\rangle |v_i\rangle$$

A more fine-grained measure of entanglement is the **entanglement entropy**. Form a probability distribution:

$$p_i = \lambda_i^2$$

From which you can calculate the entanglement entropy:

$$S = - \sum_i p_i \log p_i$$

This is a measure of entanglement. The higher the entanglement entropy, the more entanglement.

PHYC90045 Introduction to Quantum Computing



THE UNIVERSITY OF
MELBOURNE

Aside: Entropy of entanglement

Entanglement is a type of correlation between two systems, say A and B.

To see how much correlation there is between A and B: We will measure B and ask how many bits of information (as measured by entropy) this can tell us about the state of A?

For example, taking the Bell state (first qubit is A (Alice's), second qubit is B (Bob's)):

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

Recall: Bob knows from his measurement what Alice's outcome will be.

Entropy of entanglement: state is already in Schmidt Decomposition form, so we read off the probabilities:

$$|\psi\rangle = \sum \lambda_i |u_i\rangle |v_i\rangle \quad p_i = \lambda_i^2$$

PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

Aside: Entropy of Entanglement

Bob knows some bits of information about Alice's state. How much? That's measured by the *entropy*.

Entanglement entropy is given by:

$$S = - \sum_i p_i \log p_i$$

where p_i is the probability of measuring i th state of Alice's qubit.

For this case of a Bell state, from Schmidt form we have

$$p_0=50\%, p_1=50\%,$$

$$S = - \frac{1}{2} \log \frac{1}{2} - \frac{1}{2} \log \frac{1}{2} = \frac{1}{2} + \frac{1}{2} = 1$$

Here, and throughout this subject, logarithms are taken base 2 (unless otherwise stated).

Therefore, a Bell State has 1 bit of entanglement (max possible).

PHYC90045 Introduction to Quantum Computing

Aside: Entropy measure of a separable state

The University of Melbourne

For the separable state: $|00\rangle$ We measure the state of the second (Bob's) qubit. 100% of the time, the first qubit (Alice's) collapses to the state $|0\rangle$

Schmidt form -> the entropy of entanglement is therefore:
$$S = -1 \times \log 1 = 0$$

All separable states have an entropy of entanglement of 0.

For the state: $\sqrt{0.99}|00\rangle + \sqrt{0.01}|11\rangle$

Schmidt form -> the entropy of entanglement is therefore: $\rightarrow S = 0.0808$

Entanglement Entropy generalizes between two subsystems of qubits A and B, and is how the measure of entanglement is calculated in the QUI.

Week 2

Lecture 3

- 3.1 Two qubit systems and operations
- 3.2 Entanglement

Lecture 4

- 4.1 Dense coding
- 4.2 Teleportation

Lab 2

Two qubit operations, entanglement, dense coding, teleportation

Lecture overview



In this lecture:

4.1 Dense coding

4.2 Teleportation

- Reiffel: 5.3
- Kaye: Ch 5
- Mike and Ike: 1.3.5, 1.3.7, 2.3

4.1 Dense coding

PHYC90045
Lecture 4

Recap: constructing a Bell state

This is one of four states named after the physicist John Bell (who figured out how to experimentally explore reality of entanglement).

Consider the following circuit in the QUI:



Execution: $|00\rangle \xrightarrow{\text{H}} \frac{|00\rangle + |10\rangle}{\sqrt{2}} \xrightarrow{\text{CNOT}} \frac{|00\rangle + |11\rangle}{\sqrt{2}}$

Question: Is $\frac{|00\rangle + |11\rangle}{\sqrt{2}}$ separable?

Recap: entanglement

Answer: No! We can never find a, b, c, d, i.e.

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}} \neq (a|0\rangle + b|1\rangle) \otimes (c|0\rangle + d|1\rangle)$$

A state which is not separable is called an **entangled** state.

Entanglement is a uniquely quantum mechanical property, with no direct classical analogue.

Recap: Entanglement Entropy

We would like to have a measure of *how much* entanglement a state has. Some states are more entangled than others:

$$|00\rangle$$

Not entangled, separable

$$\sqrt{0.99} |00\rangle + \sqrt{0.01} |11\rangle$$

Entangled, but close to a separable state

$$\frac{1}{\sqrt{2}} |00\rangle + \frac{1}{\sqrt{2}} |11\rangle$$

Maximally entangled

Entanglement is a type of correlation between two systems, say A and B.

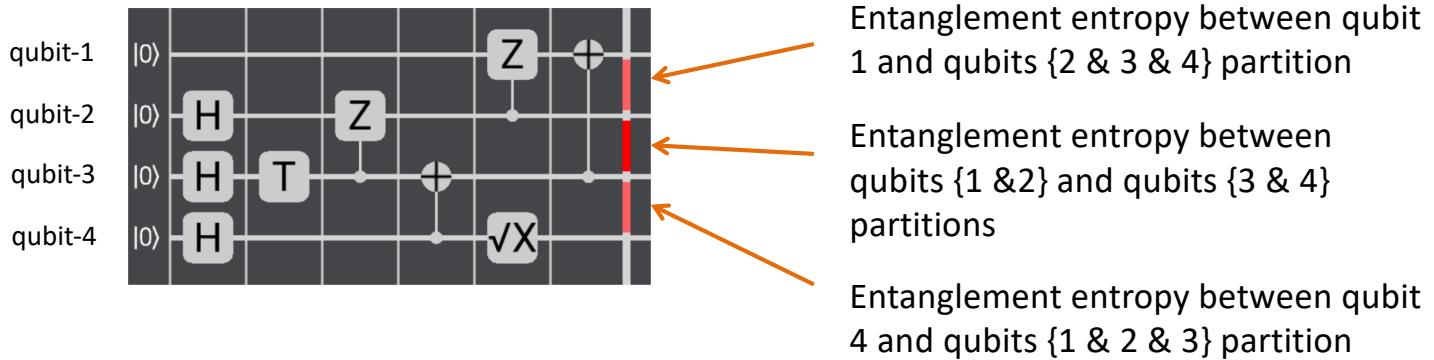
To see how much correlation there is between A and B: We will measure B and ask how many bits of information (as measured by entropy) this can tell us about the state of A?

In the QUI we measure the degree of entanglement using an informatic “entropy” measure: *Entanglement Entropy (EE)*

Recap: entanglement in the QUI – time slider

The *time slider* is the vertical bar which moves left and right to show the quantum state at each time step. When there is entanglement it will show it.

The entanglement entropy (EE) is shown in a red colour scale between min and max values possible. Each segment corresponds to the entropy between the system of qubits above and below for that particular bi-partition.



Entanglement and quantum computing

A state which is *not separable* is **entangled**. For example:

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

In this lecture we will see how entangled states can be critical in various quantum computing tasks and apply these in the Lab to gain experience in how entangled states work.

In particular we will discuss

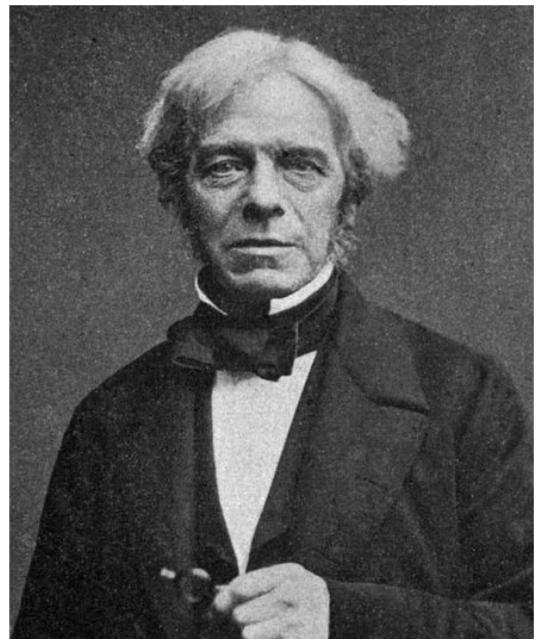
1. Dense Coding
2. No-cloning theorem
3. Quantum teleportation

Entanglement as a resource

When asked what practical use electricity was, Faraday reportedly replied:

“Why sir, there is every probability that you will be able to tax it”

Entanglement is similar, a **resource** useful for many quantum information tasks.



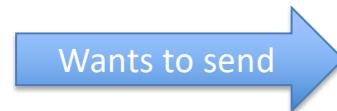
Faraday

Dense Coding

Alice would like to send **two classical bits** to Bob.



01



01



Alice

Bob

Alice and Bob can use a **quantum NBN**, and share some initial entanglement – can they get any advantage?

Dense Coding

Entanglement makes it possible.



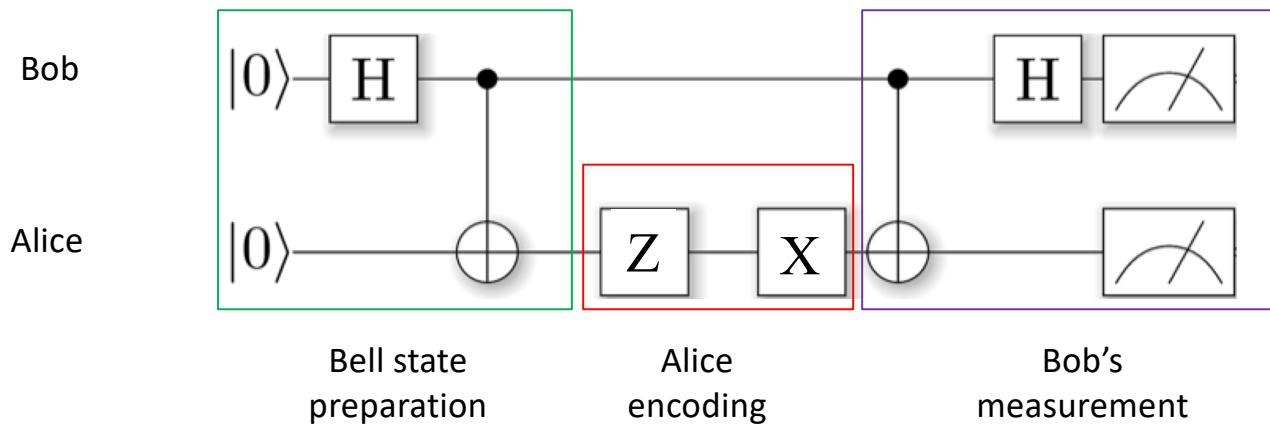
Alice



Bob

- (1) Alice and Bob share an entangled state
- (2) Alice flips her qubit one of four ways,
based on the state she wants to send
- (3) Alice sends her qubit to Bob
- (4) Bob measures correlations between the
qubits, to reveal which of the four (ie. two
bits) operations Alice applied

Dense Coding Circuit

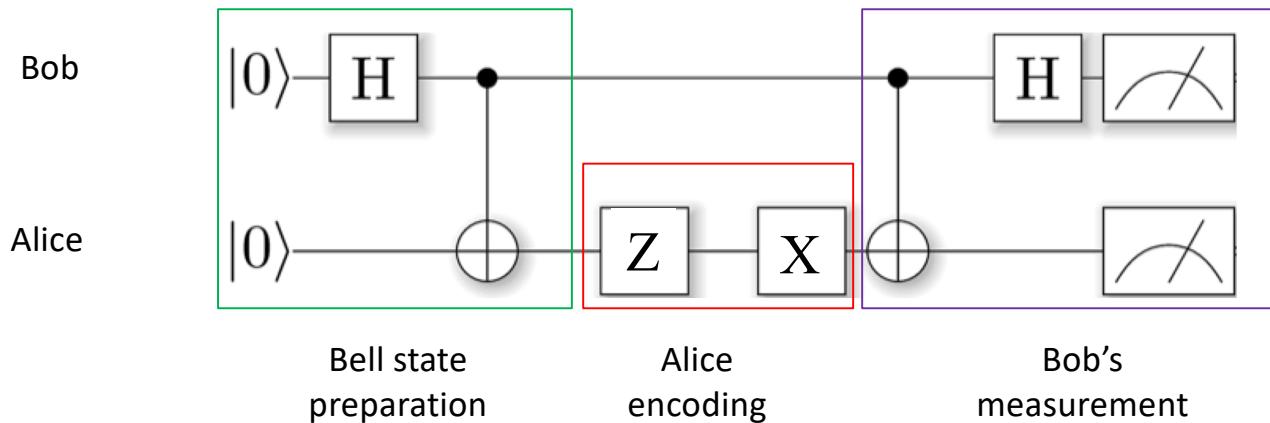


Bell state preparation:

$$|00\rangle \rightarrow \frac{|0\rangle + |1\rangle}{\sqrt{2}} |0\rangle$$

$$\text{CNOT} \rightarrow \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

Dense Coding Circuit

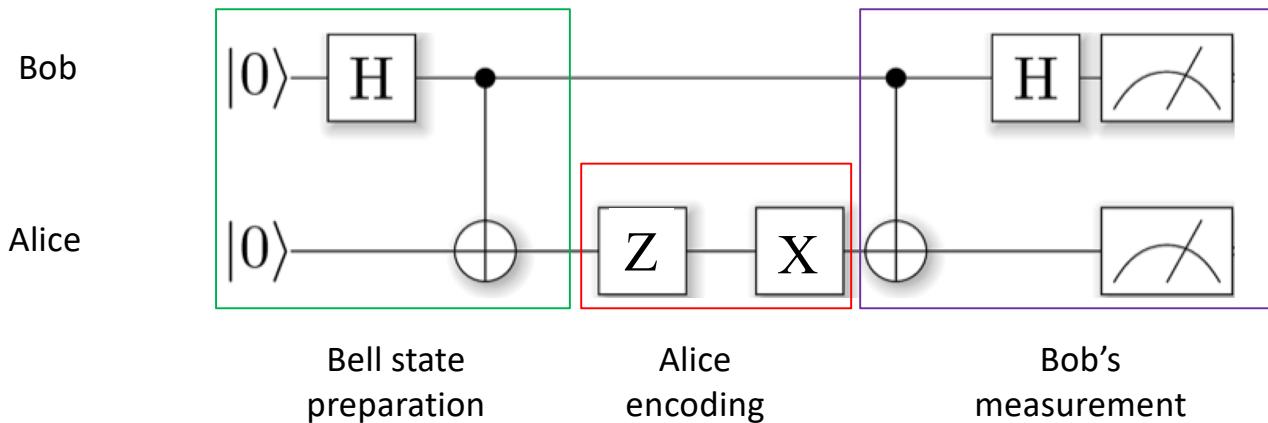


$$\begin{aligned} |00\rangle &\rightarrow \frac{|0\rangle + |1\rangle}{\sqrt{2}} |0\rangle \\ &\rightarrow \frac{|00\rangle + |11\rangle}{\sqrt{2}} \end{aligned}$$

0, 0	$\frac{ 00\rangle + 11\rangle}{\sqrt{2}}$	$\rightarrow \frac{ 00\rangle + 11\rangle}{\sqrt{2}}$
0, 1	$X_2 \frac{ 00\rangle + 11\rangle}{\sqrt{2}}$	$\rightarrow \frac{ 01\rangle + 10\rangle}{\sqrt{2}}$
1, 0	$Z_2 \frac{ 00\rangle + 11\rangle}{\sqrt{2}}$	$\rightarrow \frac{ 00\rangle - 11\rangle}{\sqrt{2}}$
1, 1	$X_2 Z_2 \frac{ 00\rangle + 11\rangle}{\sqrt{2}}$	$\rightarrow \frac{ 01\rangle - 10\rangle}{\sqrt{2}}$

Alice applies one of four different operations to her qubit, based on the **classical information** she would like to send.

Dense Coding Circuit



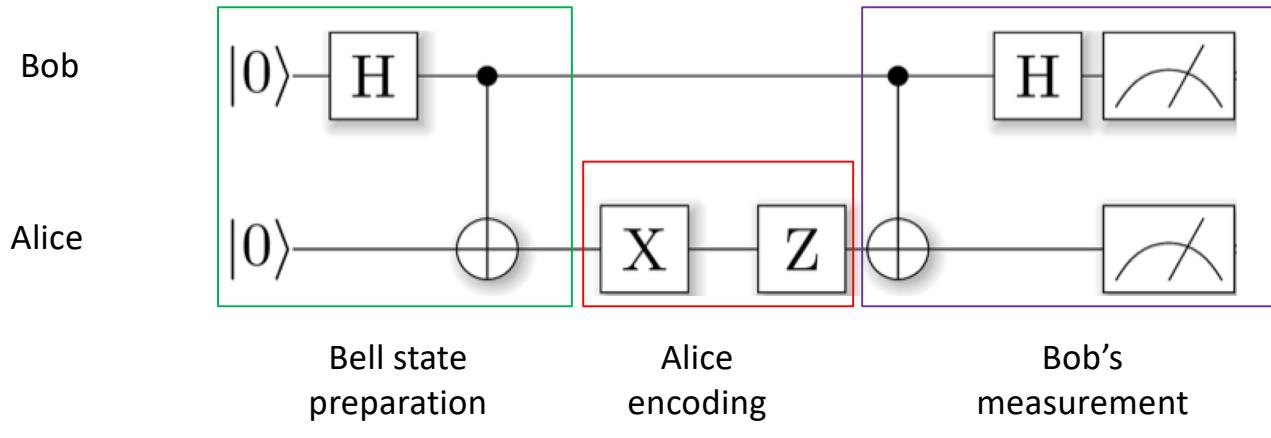
$0, 0$	$\frac{ 00\rangle + 11\rangle}{\sqrt{2}} \rightarrow \frac{ 00\rangle + 11\rangle}{\sqrt{2}}$
$0, 1$	$X_2 \frac{ 00\rangle + 11\rangle}{\sqrt{2}} \rightarrow \frac{ 01\rangle + 10\rangle}{\sqrt{2}}$
$1, 0$	$Z_2 \frac{ 00\rangle + 11\rangle}{\sqrt{2}} \rightarrow \frac{ 00\rangle - 11\rangle}{\sqrt{2}}$
$1, 0$	$X_2 Z_2 \frac{ 00\rangle + 11\rangle}{\sqrt{2}} \rightarrow \frac{ 01\rangle - 10\rangle}{\sqrt{2}}$

CNOT $\xrightarrow{\hspace{1cm}}$

$\frac{ 00\rangle + 10\rangle}{\sqrt{2}}$	$ 00\rangle$
$\frac{ 01\rangle + 11\rangle}{\sqrt{2}}$	$ 01\rangle$
$\frac{ 00\rangle - 10\rangle}{\sqrt{2}}$	$ 10\rangle$
$\frac{ 01\rangle - 11\rangle}{\sqrt{2}}$	$ 11\rangle$

H $\xrightarrow{\hspace{1cm}}$

Dense Coding Circuit



$$H|+\rangle = |0\rangle, H|-\rangle = |1\rangle$$

$0, 0$	$\frac{ 00\rangle + 11\rangle}{\sqrt{2}} \rightarrow \frac{ 00\rangle + 11\rangle}{\sqrt{2}}$
$0, 1$	$X_2 \frac{ 00\rangle + 11\rangle}{\sqrt{2}} \rightarrow \frac{ 01\rangle + 10\rangle}{\sqrt{2}}$
$1, 0$	$Z_2 \frac{ 00\rangle + 11\rangle}{\sqrt{2}} \rightarrow \frac{ 00\rangle - 11\rangle}{\sqrt{2}}$
$1, 0$	$X_2 Z_2 \frac{ 00\rangle + 11\rangle}{\sqrt{2}} \rightarrow \frac{ 01\rangle - 10\rangle}{\sqrt{2}}$

CNOT

$\frac{ 00\rangle + 10\rangle}{\sqrt{2}} = \frac{ 0\rangle + 1\rangle}{\sqrt{2}} 0\rangle = +\rangle 0\rangle$
$\frac{ 01\rangle + 11\rangle}{\sqrt{2}} = \frac{ 0\rangle + 1\rangle}{\sqrt{2}} 1\rangle = +\rangle 1\rangle$
$\frac{ 00\rangle - 10\rangle}{\sqrt{2}} = \frac{ 0\rangle - 1\rangle}{\sqrt{2}} 0\rangle = -\rangle 0\rangle$
$\frac{ 01\rangle - 11\rangle}{\sqrt{2}} = \frac{ 0\rangle - 1\rangle}{\sqrt{2}} 1\rangle = -\rangle 1\rangle$

$|00\rangle$
 $|01\rangle$
 $|10\rangle$
 $|11\rangle$

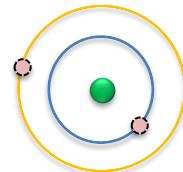
Two bits communicated but only one qubit “sent”.
 Makes use of pre-existing entanglement.

4.2 Teleportation

PHYC90045
Lecture 4

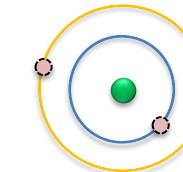
A Quantum Computing Bus?

To understand the role entanglement can play in quantum information processing, we will consider how it can be used to transmit quantum information around our quantum computer (and potentially between quantum computers)



$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$$

Alice



$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$$



Bob

Communication around the quantum computer is an important primitive. We could physically move quantum systems, but there is a (potentially) better way: **teleportation**

Sending classical information

How would we do this classically? Measure everything about the state, then send that information down (classical) bus and recreate a perfect copy elsewhere.

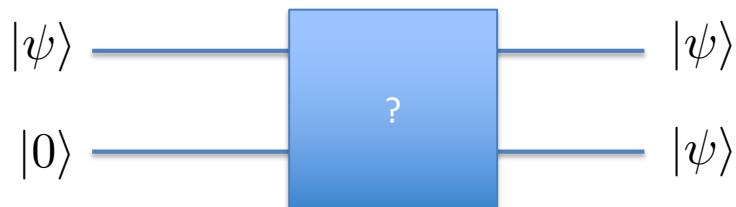


Image by ChtiTux, Used here under CC-by-SA 1.0 license.

Problem: we can't do this in quantum mechanics because classical measurement (1) *collapses the system*, and (2) *this clones the system* which we can't do in quantum mechanics.

No-cloning theorem

Can we make a circuit which clones the input state?



That is, we ask if it is possible to make a unitary transformation s.t.

$$\begin{aligned}(\alpha|0\rangle + \beta|1\rangle) \otimes |0\rangle &\rightarrow (\alpha|0\rangle + \beta|1\rangle) \otimes (\alpha|0\rangle + \beta|1\rangle) \\&= \alpha^2|00\rangle + \alpha\beta|01\rangle + \beta\alpha|10\rangle + \beta^2|11\rangle\end{aligned}$$

No-cloning theorem: the answer is **no**.

Proof of no-cloning theorem

If we had a cloning circuit, we could use it on two arbitrary states, $|\psi\rangle$ and $|\phi\rangle$

$$U|\phi\rangle|0\rangle = |\phi\rangle|\phi\rangle \quad U|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle$$

Inner product on LHS: $\langle 0| \langle \phi| U^\dagger U |\psi\rangle |0\rangle = \langle \phi|\psi\rangle$

Inner product on RHS: $\langle \psi| \langle \psi|\phi\rangle |\phi\rangle = \langle \psi|\phi\rangle^2$

But the only solutions to $x^2=x$ are $x=0$ or $x=1$. We can only have a circuit clone states *which are orthogonal (x=0 case), not arbitrary states.*

There can be no unitary transformation which clones two arbitrary states.

Teleportation

Entanglement makes it possible.



Alice

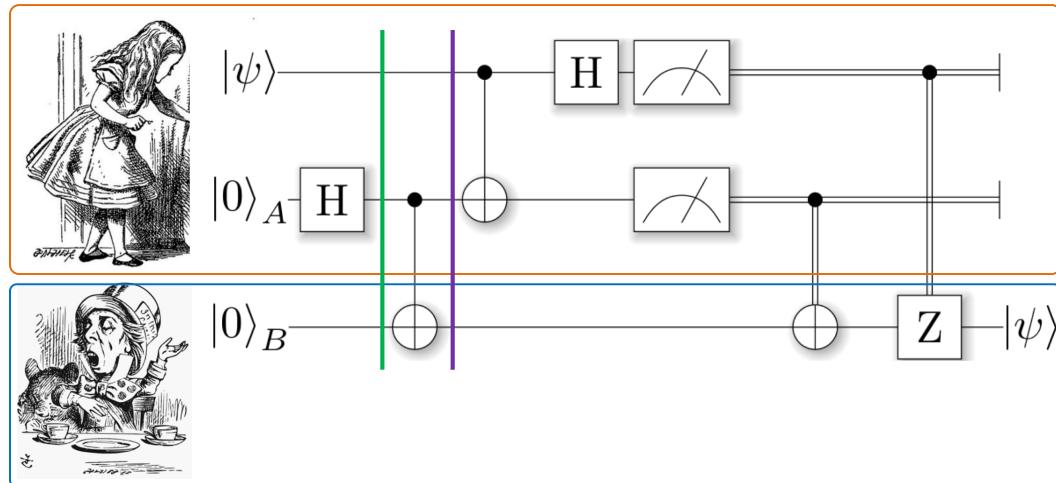
- (1) Alice has a qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$
- (2) Alice and Bob share an entangled state
- (3) Alice measures correlations between her qubit and half of the entangled state
- (4) Alice sends the results of the measurements to Bob
- (5) Bob uses them to reconstruct the original state in his qubit

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$



Bob

Teleportation



$$(\alpha |0\rangle + \beta |1\rangle) \otimes |00\rangle$$

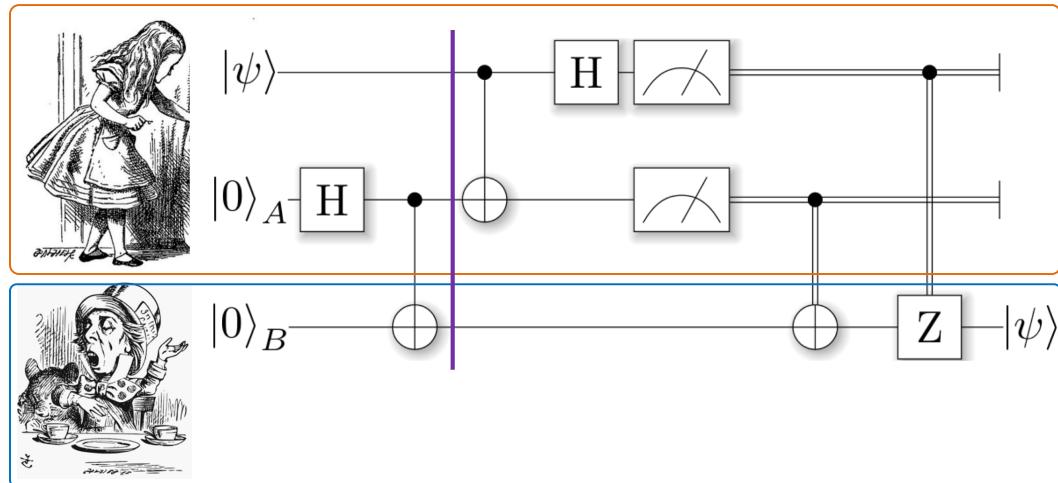
Hadamard (A) $\xrightarrow{\text{green}} (\alpha |0\rangle + \beta |1\rangle) \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |0\rangle$

CNOT(A-B) $\xrightarrow{\text{purple}} (\alpha |0\rangle + \beta |1\rangle) \otimes \frac{|00\rangle + |11\rangle}{\sqrt{2}}$

Alice's state

Bell state preparation (shared)

Teleportation

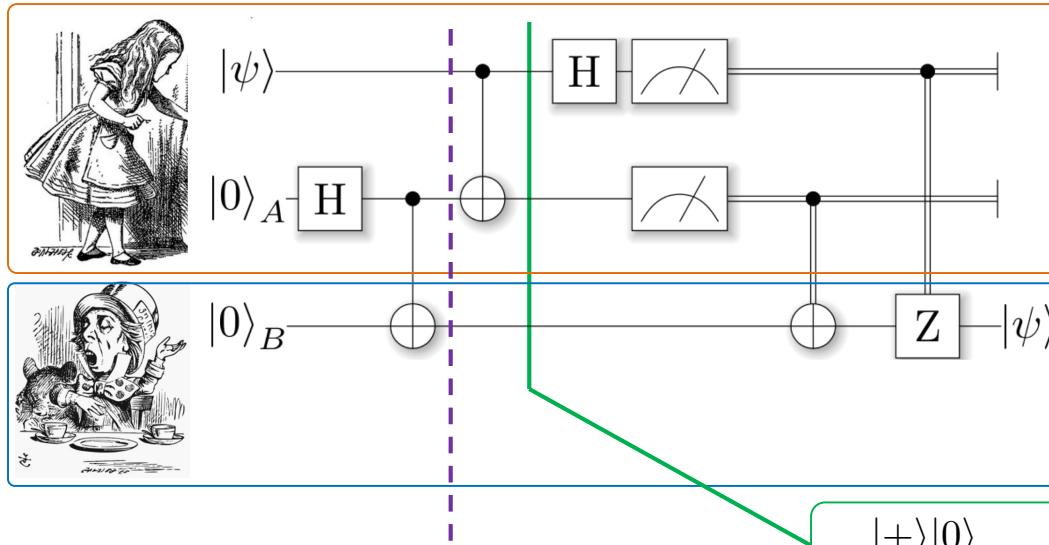


Total system state: $(\alpha|0\rangle + \beta|1\rangle) \frac{|00\rangle + |11\rangle}{\sqrt{2}}$

Alice's state $|\psi\rangle$ Shared entangled state A & B

Expand: $\longrightarrow \frac{1}{\sqrt{2}} (\alpha|000\rangle + \alpha|011\rangle + \beta|100\rangle + \beta|111\rangle)$

Teleportation



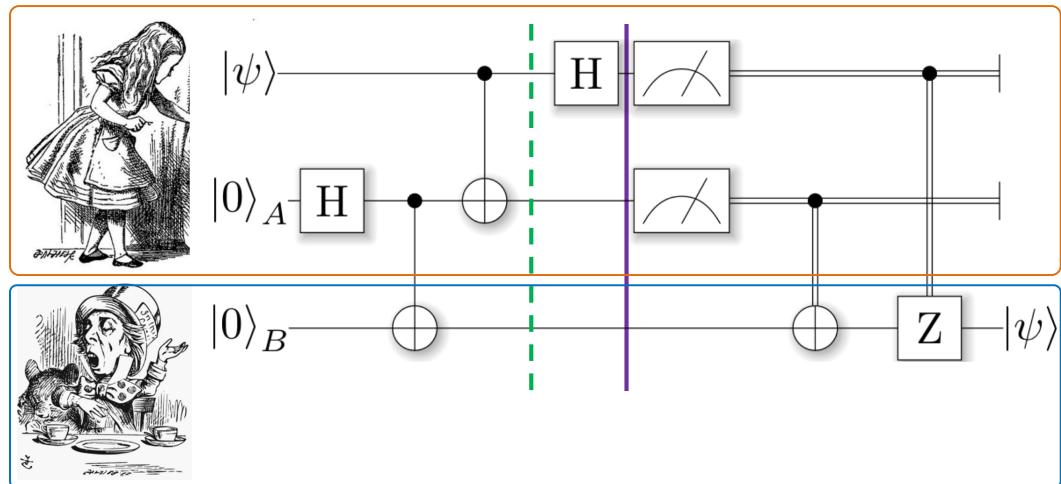
$$|\psi\rangle = \frac{1}{\sqrt{2}} (\alpha |000\rangle + \alpha |011\rangle + \beta |100\rangle + \beta |111\rangle)$$

$$\xrightarrow{\text{CNOT}[1,2]} \alpha \frac{|000\rangle + |011\rangle}{\sqrt{2}} + \beta \frac{|110\rangle + |101\rangle}{\sqrt{2}}$$

Rewrite
(ex):

$$\begin{aligned}
 & \frac{|+\rangle|0\rangle}{2}(\alpha|0\rangle + \beta|1\rangle) \\
 & + \frac{|+\rangle|1\rangle}{2}(\alpha|1\rangle + \beta|0\rangle) \\
 & + \frac{|-\rangle|0\rangle}{2}(\alpha|0\rangle - \beta|1\rangle) \\
 & + \frac{|-\rangle|1\rangle}{2}(\alpha|1\rangle - \beta|0\rangle)
 \end{aligned}$$

Teleportation

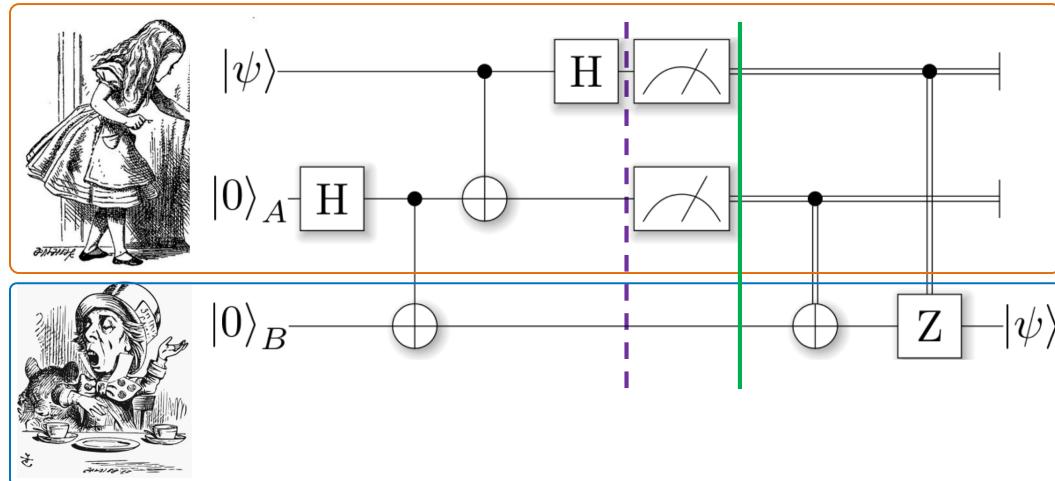


$$\begin{aligned} & \frac{|+\rangle|0\rangle}{2}(\alpha|0\rangle + \beta|1\rangle) \\ & + \frac{|+\rangle|1\rangle}{2}(\alpha|1\rangle + \beta|0\rangle) \\ & + \frac{|-\rangle|0\rangle}{2}(\alpha|0\rangle - \beta|1\rangle) \\ & + \frac{|-\rangle|1\rangle}{2}(\alpha|1\rangle - \beta|0\rangle) \end{aligned}$$

Hadamard
→

$$\begin{aligned} & \frac{|0\rangle|0\rangle}{2}(\alpha|0\rangle + \beta|1\rangle) \\ & + \frac{|0\rangle|1\rangle}{2}(\alpha|1\rangle + \beta|0\rangle) \\ & + \frac{|1\rangle|0\rangle}{2}(\alpha|0\rangle - \beta|1\rangle) \\ & + \frac{|1\rangle|1\rangle}{2}(\alpha|1\rangle - \beta|0\rangle) \end{aligned}$$

Teleportation



$$\begin{aligned}
 & \frac{|0\rangle|0\rangle}{2} (\alpha|0\rangle + \beta|1\rangle) \\
 + & \frac{|0\rangle|1\rangle}{2} (\alpha|1\rangle + \beta|0\rangle) \\
 + & \frac{|1\rangle|0\rangle}{2} (\alpha|0\rangle - \beta|1\rangle) \\
 + & \frac{|1\rangle|1\rangle}{2} (\alpha|1\rangle - \beta|0\rangle)
 \end{aligned}$$

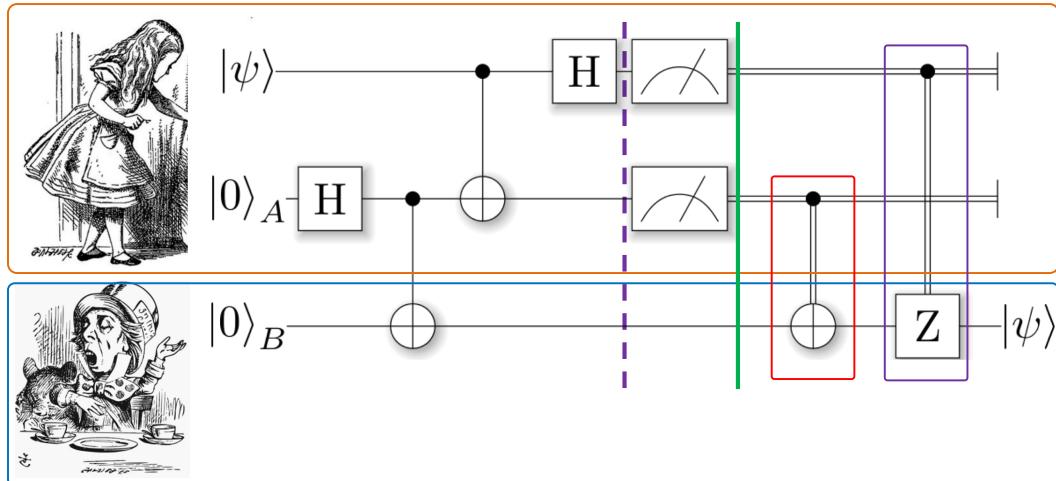
Alice measures her two qubits.

Bob's qubit collapses to one of the four possibilities.

Alice now tells Bob her outcomes (double lines indicate classical communication).

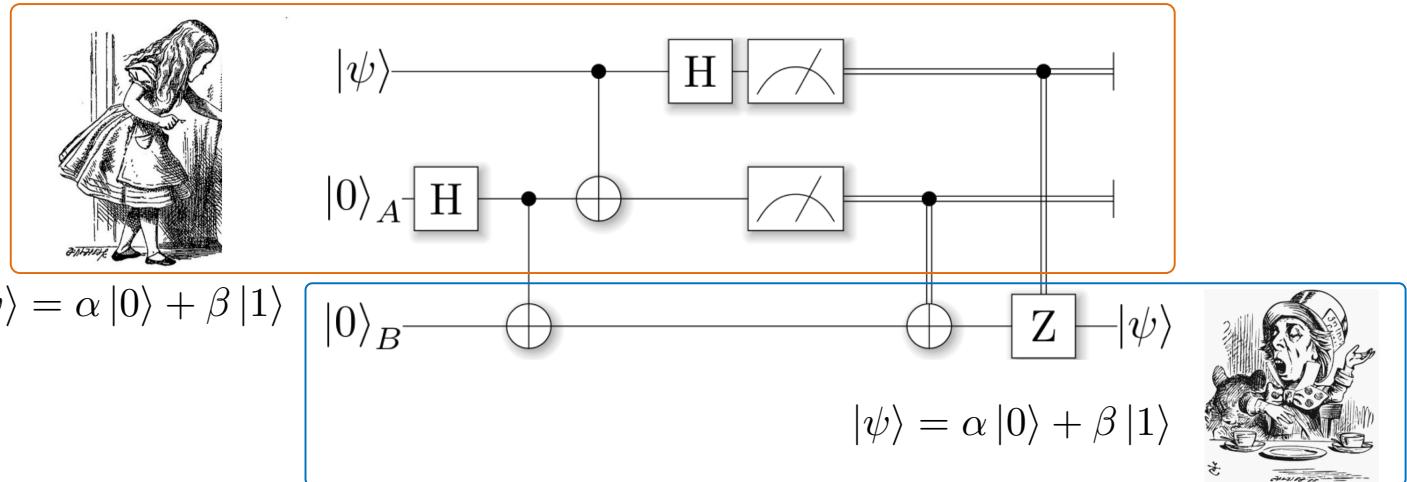
Bob will perform simple corrections shown.

Teleportation



$\frac{ 0\rangle 0\rangle}{2}(\alpha 0\rangle + \beta 1\rangle)$ $+ \frac{ 0\rangle 1\rangle}{2}(\alpha 1\rangle + \beta 0\rangle)$ $+ \frac{ 1\rangle 0\rangle}{2}(\alpha 0\rangle - \beta 1\rangle)$ $+ \frac{ 1\rangle 1\rangle}{2}(\alpha 1\rangle - \beta 0\rangle)$	Alice measures <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center;">Bob's qubit</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0, 0</td> <td style="text-align: center;">$\alpha 0\rangle + \beta 1\rangle$</td> </tr> <tr> <td style="text-align: center;">0, 1</td> <td style="text-align: center;">$\alpha 1\rangle + \beta 0\rangle$ → $\alpha 0\rangle + \beta 1\rangle$</td> </tr> <tr> <td style="text-align: center;">1, 0</td> <td style="text-align: center;">$\alpha 0\rangle - \beta 1\rangle$ → $\alpha 0\rangle + \beta 1\rangle$</td> </tr> <tr> <td style="text-align: center;">1, 1</td> <td style="text-align: center;">$\alpha 1\rangle - \beta 0\rangle$ → $\alpha 0\rangle - \beta 1\rangle$ → $\alpha 0\rangle + \beta 1\rangle$</td> </tr> </tbody> </table>	Bob's qubit		0, 0	$\alpha 0\rangle + \beta 1\rangle$	0, 1	$\alpha 1\rangle + \beta 0\rangle$ → $\alpha 0\rangle + \beta 1\rangle$	1, 0	$\alpha 0\rangle - \beta 1\rangle$ → $\alpha 0\rangle + \beta 1\rangle$	1, 1	$\alpha 1\rangle - \beta 0\rangle$ → $\alpha 0\rangle - \beta 1\rangle$ → $\alpha 0\rangle + \beta 1\rangle$
Bob's qubit											
0, 0	$\alpha 0\rangle + \beta 1\rangle$										
0, 1	$\alpha 1\rangle + \beta 0\rangle$ → $\alpha 0\rangle + \beta 1\rangle$										
1, 0	$\alpha 0\rangle - \beta 1\rangle$ → $\alpha 0\rangle + \beta 1\rangle$										
1, 1	$\alpha 1\rangle - \beta 0\rangle$ → $\alpha 0\rangle - \beta 1\rangle$ → $\alpha 0\rangle + \beta 1\rangle$										

Teleportation



Alice measures	Bob's qubit	i.e. after correction Bob has successfully reconstructed Alice's original state.
0, 0	$\alpha 0\rangle + \beta 1\rangle$	$\alpha 0\rangle + \beta 1\rangle \rightarrow \alpha 0\rangle + \beta 1\rangle$
0, 1	$\alpha 1\rangle + \beta 0\rangle$	$X(\alpha 1\rangle + \beta 0\rangle) \rightarrow \alpha 0\rangle + \beta 1\rangle$
1, 0	$\alpha 0\rangle - \beta 1\rangle$	$Z(\alpha 0\rangle - \beta 1\rangle) \rightarrow \alpha 0\rangle + \beta 1\rangle$
1, 1	$\alpha 1\rangle - \beta 0\rangle$	$ZX(\alpha 1\rangle - \beta 0\rangle) \rightarrow \alpha 0\rangle + \beta 1\rangle$

Week 2 so far

Lecture 3

- 3.1 Two qubit systems and operations
- 3.2 Entanglement

Lecture 4

- 4.1 Dense coding
- 4.2 Teleportation

Lab 2

Two qubit operations, entanglement, dense coding, teleportation

PHYC90045 Introduction to Quantum Computing

Week 3

Lecture 5
Universality in quantum computing, Reversible computation, one qubit adder, the Deutsch-Josza algorithm

Lecture 6
Two basic quantum algorithms: Bernstein-Vazirani and Simon's Algorithms

Lab 3
Logical statements, Reversible logic, Adder, Deutsch-Josza algorithm

PHYC90045 Introduction to Quantum Computing

Overview

In this lecture we will discuss reversible logic and our first quantum algorithm – the Deutsch-Josza algorithm,

1. Reversible (classical) logic
2. The Deutsch-Josza algorithm
3. Aside: Universality in quantum computing

Along the way we will encounter common patterns often turn up in quantum algorithms, and will highlight them because they will help make sense of what of future quantum circuits.

See:

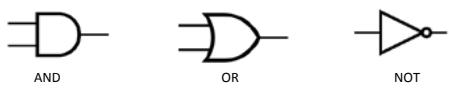
Kaye, 1.5, 6.1-6.4
 Nielsen and Chuang, 1.4, 3.1
 Reiffel, 6, 7.3-7.5

PHYC90045 Introduction to Quantum Computing

Universality in classical logic

A set of (classical) gates is said to be *functionally complete* (or “universal”) if every possible truth table (ie. Boolean function) can be expressed using members of the set.

For example, in classical logic: {AND, OR, NOT} is functionally complete.



AND OR NOT

Every logical circuit can be made from combinations of these gates. In fact, the NAND gate alone is universal. However, we cannot implement these gates directly in a quantum computer. AND/OR are **not reversible**.

PHYC90045 Introduction to Quantum Computing

Irreversible Functions

We cannot implement AND or OR because these functions are irreversible.



A	B	A B
0	0	0
0	1	0
1	0	0
1	1	1

We cannot determine the inputs from the output.

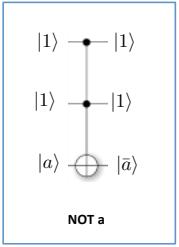
Irreversible functions are *not unitary*.

PHYC90045 Introduction to Quantum Computing

Reversible Logic

Classically, if we would like to calculate some Boolean function, f , we can construct a circuit out of AND, OR and NOT gates. However, AND and OR gates are **not reversible**, and so can't be implemented in a quantum computer.

But by use of reversible circuits and additional bits/qubits, we can express everything in terms of reversible gates (such as Toffoli):



PHYC90045 Introduction to Quantum Computing

Reversible Logic and the Toffoli Gate

a AND b			a XOR b			(NOT a) OR (NOT b)		
$ a\rangle$	\bullet	$ a\rangle$	$ 1\rangle$	\bullet	$ 1\rangle$	$ a\rangle$	\bullet	$ a\rangle$
$ b\rangle$	\bullet	$ b\rangle$	$ a\rangle$	\bullet	$ a\rangle$	$ b\rangle$	\bullet	$ b\rangle$
$ 0\rangle$	\oplus	$ a \wedge b\rangle$	$ b\rangle$	\oplus	$ a \oplus b\rangle$	$ 1\rangle$	\oplus	$ \neg a \vee \neg b \rangle$

a	b	a \wedge b	a	b	a \oplus b	a	b	$\neg a \vee \neg b$
0	0	0	0	0	0	0	0	1
0	1	0	0	1	1	0	1	1
1	0	0	1	0	1	1	0	1
1	1	1	1	1	0	1	1	0

The Toffoli gate is **universal and reversible**. In principle every classical boolean function can be written in terms of reversible gates (such as Toffoli) which can be implemented on a quantum computer.

PHYC90045 Introduction to Quantum Computing

One Bit Adder

We can, for example, implement a one bit adder using only reversible gates:

We will now explain this circuit and in the lab we will extend this to a two bit adder.

PHYC90045 Introduction to Quantum Computing

1+1 Quantum Style

Here is what happens when we add together numbers in superposition:

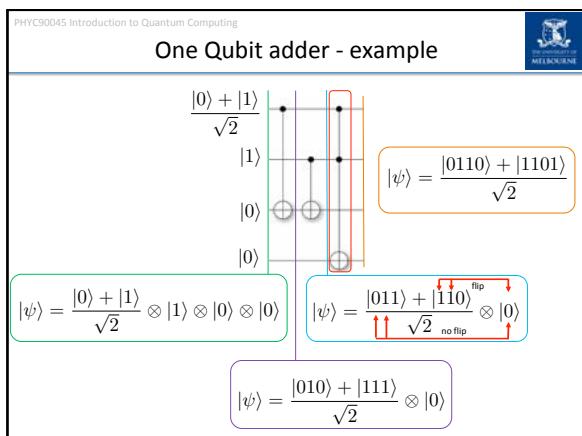
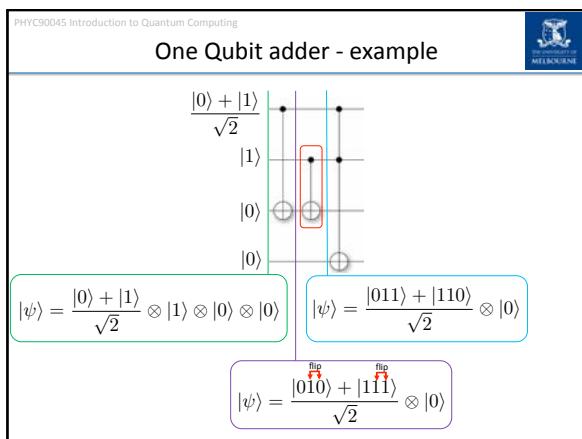
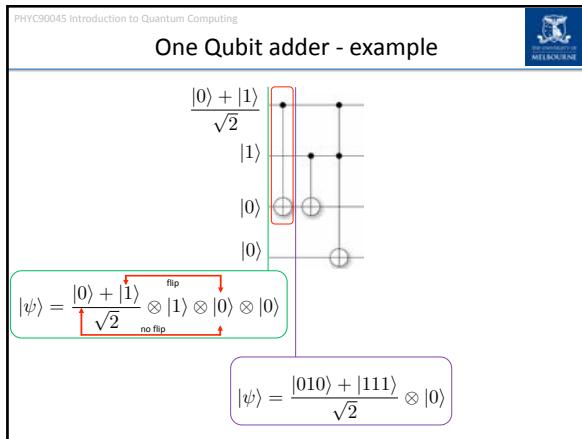
$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad " + " \quad |1\rangle$$

Let's do the walkthrough...

PHYC90045 Introduction to Quantum Computing

One Qubit adder - example

$|\psi\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |1\rangle \otimes |0\rangle \otimes |0\rangle$



PHYC90045 Introduction to Quantum Computing

One Qubit adder - example

$|\psi\rangle = \frac{|0110\rangle + |1101\rangle}{\sqrt{2}}$

Or, considering the last two registers as a two-qubit binary register (msb last):

$$|\psi\rangle = \frac{|0110\rangle + |1101\rangle}{\sqrt{2}}$$

$$|\psi\rangle = \frac{|0\rangle|1\rangle|1\rangle + |1\rangle|1\rangle|2\rangle}{\sqrt{2}}$$

0+1=1 1+1=2

Note: this is an entangled state.

PHYC90045 Introduction to Quantum Computing

Implementing irreversible functions

We cannot compute **irreversible** functions directly (not unitary). One strategy which you often see to make an irreversible function reversible is simply to propagate the input to the output:

PHYC90045 Introduction to Quantum Computing

One Bit Adder

The adder is an example. Given $a + b$, we can't uniquely determine a and b . So we used this trick:

PHYC90045 Introduction to Quantum Computing

Implementing irreversible functions

One strategy which you often see to make an irreversible function reversible is simply to propagate the inputs:

Must deal with all possible inputs, not just 0.

Add (bit-by-bit modulo 2) input and calculated $f(x)$

You will see this pattern in several of the quantum algorithms we will study.

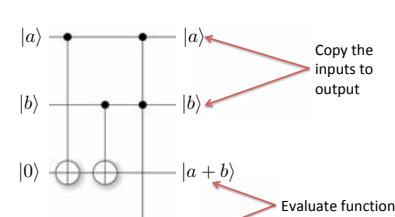
PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF MELBOURNE

One Bit Adder

The adder is an example. Given $a+b$, we can't uniquely determine a and b .
 So we used this trick:



Copy the inputs to output

Evaluate function

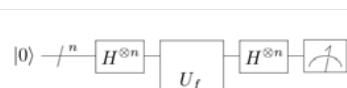
In the lab we will extend this to a two bit adder.

PHYC90045 Introduction to Quantum Computing



Deutsch-Josza algorithm

- Given a boolean function, f , determine if:
 f is constant (always gives the same result), or
 f is balanced (gives equal numbers of 0s and 1s)
- Classical algorithm (worst case) needs $2^n/2+1$ queries**
- Quantum algorithm needs just 1 query.**



PHYC90045 Introduction to Quantum Computing

Deutsch-Josza algorithm (2)

Let's take the example with just one bit/qubit input. There are 4 choices of function:

	$ x\rangle$	$ y\rangle$	U_f	$ x\rangle$	$ y \oplus f(x)\rangle$
--	-------------	-------------	-------	-------------	-------------------------

CONSTANT	$f_0(0) = 0$	$f_0(1) = 0$	X
	$f_1(0) = 1$	$f_1(1) = 1$	
BALANCED	$f_0(0) = 0$	$f_0(1) = 1$	X
	$f_1(0) = 1$	$f_1(1) = 0$	
	$f_1(0) = 0$	$f_1(1) = 0$	

PHYC90045 Introduction to Quantum Computing

Example of a constant function

$|x\rangle = |x\rangle$

$|y\rangle = X |y \oplus 1\rangle$

$\left. \begin{array}{l} f(0) = 1 \\ f(1) = 1 \end{array} \right\}$

The circuit always flips (ie. NOT) the second qubit, regardless of the input.
This function is **constant**, since the output is always 1.

PHYC90045 Introduction to Quantum Computing

Deutsch algorithm: constant function

$|0\rangle$

$|1\rangle$

The function, U , is implemented by the gates inside this box

PHYC90045 Introduction to Quantum Computing

Deutsch algorithm: walkthrough

$|\psi\rangle = |0\rangle \otimes |1\rangle$

PHYC90045 Introduction to Quantum Computing

Deutsch algorithm: walkthrough

$|\psi\rangle = H|0\rangle \otimes H|1\rangle$

$$= \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$= |+\rangle \otimes |-\rangle$$

PHYC90045 Introduction to Quantum Computing

Deutsch algorithm: walkthrough

$|\psi\rangle = |+\rangle \otimes X|-\rangle$

$$= \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes X \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$= \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|1\rangle - |0\rangle}{\sqrt{2}}$$

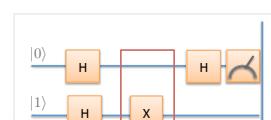
$$= -|+\rangle \otimes |-\rangle \quad \text{Global phase is unmeasurable}$$

PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

Deutsch algorithm: walkthrough



$$|\psi\rangle = -|0\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

We will measure “0” with 100% probability. This indicates (with a single evaluation of the function) that the function is **constant**.

PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

Example of a balanced function



$|x\rangle$ $|x\rangle$
 $|y\rangle$ $|y \oplus x\rangle$

$f(0) = 0$
 $f(1) = 1$

The circuit only flips (ie. NOT) the second qubit, if the input is a 1. This function is **balanced**, since the output has equal numbers of 0 and 1 output.

PHYC90045 Introduction to Quantum Computing

Deutsch algorithm: balanced function

The function, U, is implemented by the gates inside this box

PHYC90045 Introduction to Quantum Computing

Deutsch algorithm: walkthrough

$|\psi\rangle = |0\rangle \otimes |1\rangle$

PHYC90045 Introduction to Quantum Computing

Deutsch algorithm: walkthrough

$$\begin{aligned} |\psi\rangle &= H|0\rangle \otimes H|1\rangle \\ &= \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \\ &= |+\rangle \otimes |- \rangle \end{aligned}$$

PHYC90045 Introduction to Quantum Computing

Deutsch algorithm: walkthrough

$$|\psi\rangle = CNOT|+\rangle|-\rangle$$

This is a common pattern called "phase kickback"

PHYC90045 Introduction to Quantum Computing

Phase kickback

Consider X applied to the output register:

$$\begin{aligned} X|-\rangle &= X \frac{|0\rangle - |1\rangle}{\sqrt{2}} \\ &= \frac{|1\rangle - |0\rangle}{\sqrt{2}} \\ &= -|-\rangle \end{aligned}$$

If we were to apply an X gate to the target qubit we get a *global phase change*. Otherwise the state is unchanged.

What would happen when apply a *control-X* gate?

PHYC90045 Introduction to Quantum Computing

Phase kickback

Now with a control-X gate:

$$\text{CNOT}|+\rangle|-\rangle = \text{CNOT} \frac{|0\rangle + |1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \frac{|0\rangle|0\rangle - |1\rangle|1\rangle}{\sqrt{2}\sqrt{2}} + \frac{|1\rangle|1\rangle - |0\rangle|0\rangle}{\sqrt{2}\sqrt{2}}$$

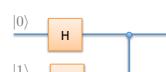
$$\text{CNOT}|+\rangle|-\rangle = \frac{|0\rangle|-\rangle - |1\rangle|-\rangle}{\sqrt{2}}$$

If we were to apply a control-X gate then any control states which apply the X-gate receive a phase change. The "1" state receives the (relative) phase change in this case.

PHYC90045 Introduction to Quantum Computing



Phase kickback



$$\begin{aligned} CNOT |+ \rangle |-\rangle &= \frac{|0\rangle |-\rangle - |1\rangle |-\rangle}{\sqrt{2}} \\ &= \frac{|0\rangle - |1\rangle}{\sqrt{2}} |-\rangle \end{aligned}$$

$$= |-\rangle |-\rangle$$

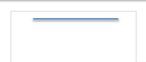
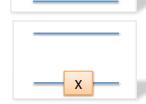
This causes the **phase** to be applied to the *control qubit*. This is known as phase kickback.

The state of the target qubit remains unchanged.

Deutsch algorithm: walkthrough

$|\psi\rangle = CNOT|+\rangle|- \rangle = |- \rangle|-\rangle$

Phase kickback changes the state of the control qubit.

Phase kickback for balanced functions	
Constant functions	 <p>The control qubit is unchanged by the constant functions.</p>
Balanced functions	 <p>The same phase kickback pattern that we just saw applies to both balanced functions</p>

PHYC90045 Introduction to Quantum Computing



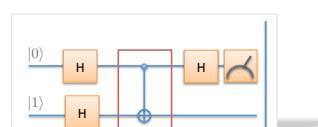
Deutsch algorithm: walkthrough

$$\begin{aligned} |\psi\rangle &= H|-\rangle \otimes |-\rangle \\ &= |1\rangle \otimes |-\rangle \end{aligned}$$

PHYC90045 Introduction to Quantum Computing



Deutsch algorithm: walkthrough



$$|\psi\rangle = |1\rangle \otimes |- \rangle$$

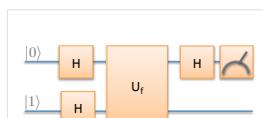
We will measure “1” with 100% probability. This indicates (with a single evaluation of the function) that the function is **balanced**.

PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

Deutsch-Josza (3)



A quantum circuit diagram showing two parallel paths. The top path starts with a blue line labeled $|0\rangle$, followed by a Hadamard gate (H), then a large orange block labeled U_f , and finally a Hadamard gate (H) before a measurement symbol. The bottom path starts with a blue line labeled $|1\rangle$, followed by a Hadamard gate (H), then the same orange U_f block, and finally a Hadamard gate (H) before a measurement symbol.

		Measured
CONSTANT	$f_1(0) = 0$ $f_1(1) = 0$	0
	$f_1(0) = 1$ $f_1(1) = 1$	0
BALANCED	$f_1(0) = 0$ $f_1(1) = 1$	1
	$f_1(0) = 1$ $f_1(1) = 0$	1

The Deutsch-Josza algorithm determines in just *one query* whether the function is constant or balanced.

Classically, this would require *two queries*.

PHYC90045 Introduction to Quantum Computing

Multiple qubits: Deutsch-Josza

This means that there are multiple qubits in the register

Only a single qubit here

PHYC90045 Introduction to Quantum Computing

Example of multi-qubit constant function

x	f(x)
000	1
001	1
010	1
011	1
100	1
101	1
110	1
111	1

$|x\rangle$ $|y\rangle$ $|y \oplus f(x)\rangle$

PHYC90045 Introduction to Quantum Computing

Example of multi-qubit balanced function

x	f(x)
000	0
001	1
010	1
011	0
100	1
101	0
110	1
111	0

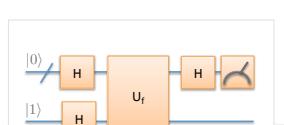
$|x\rangle$ $|y\rangle$ $|y \oplus f(x)\rangle$

PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

Multiple qubits: Deutsch-Josza



The diagram shows a quantum circuit with two qubits. The top qubit starts in state $|0\rangle$ and passes through a Hadamard gate (H). It then enters a controlled unitary block U_f , which consists of a sequence of gates: another Hadamard gate (H), followed by a CNOT gate, and finally a third Hadamard gate (H). The bottom qubit starts in state $|1\rangle$ and also passes through the same sequence of gates within the U_f block: a Hadamard gate (H), a CNOT gate, and a final Hadamard gate (H). The circuit concludes with a measurement of both qubits.

Let's walkthrough this circuit...

PHYC90045 Introduction to Quantum Computing

Recap: binary and decimal representations

The University of
Auckland

n qubits

shorthand notation

$$|\psi\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \dots \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

$$|\psi\rangle = \left[\frac{1}{\sqrt{2}} \right]^n (|00...0\rangle + \dots + |11...1\rangle)$$

i.e. even superposition over binary rep of integers: $i = 0$ to $2^n - 1$

In general we use two representations in the QUI ($N = 2^n$):

"binary"
 $|\psi\rangle = a_{0..00}|0\dots00\rangle + a_{0..01}|0\dots01\rangle + a_{0..10}|0\dots10\rangle + \dots + a_{1..11}|1\dots11\rangle$

"decimal"
 $|\psi\rangle = a_0|0\rangle + a_1|1\rangle + a_2|2\rangle + \dots + a_{N-1}|N-1\rangle$

e.g. $a_{101}|101\rangle$

PHYC90045 Introduction to Quantum Computing

Deutsch-Josza Walkthrough

After the initial Hadamard gates, the state is (n qubits, N = 2ⁿ):

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Equal (even) superposition of states

PHYC90045 Introduction to Quantum Computing

General Function Phase Kickback

Using phase kickback, after the function has been applied:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x (-1)^{f(x)} |x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

If the function evaluates to "1" then the target qubit is flipped, and we pick up a phase. Otherwise, there is no phase applied. This is a simple way to write that.

PHYC90045 Introduction to Quantum Computing

Deutsch-Josza Walkthrough

Using phase kickback, after the function has been applied:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x (-1)^{f(x)} |x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

If the function evaluates to "1" then the target qubit is flipped, and we pick up a phase. Otherwise, there is no phase applied. This is a simple way to write that.

PHYC90045 Introduction to Quantum Computing

Hadamard applied to a general state

Amplitude $a_z \rightarrow$ how many times does the binary representation of z and x have 1's in the same location?

$$H^{\otimes n} |x\rangle = \frac{1}{\sqrt{N}} \sum_{z=0}^{N-1} a_z |z\rangle$$

Shorthand for the bitwise dot product is: $x \cdot z = \sum_{j=0}^n x_j z_j$

When 1's in the same location, we get a sign change $\rightarrow (-1)^{x \cdot z}$

Hadamards applied to a general state (n qubits, $N = 2^n$):

$$H^{\otimes n} |x\rangle = \frac{1}{\sqrt{N}} \sum_{z=0}^{N-1} (-1)^{x \cdot z} |z\rangle$$

PHYC90045 Introduction to Quantum Computing

e.g. Hadamard applied to a general state

$|x\rangle \xrightarrow{\text{H}^{\otimes n}} H^{\otimes n}|x\rangle$

$H^{\otimes 3}|000\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \frac{|0\rangle + |1\rangle}{\sqrt{2}}$

$H^{\otimes 3}|000\rangle = \frac{1}{\sqrt{2^3}}(|000\rangle + |001\rangle + \dots + |111\rangle)$

$H^{\otimes 3}|x=0\rangle = \frac{1}{\sqrt{2^3}} \sum_{z=0}^{2^3-1} (-1)^{x \cdot z} |z\rangle$ bitwise: $x \cdot z = 0$

$H^{\otimes 3}|x=4\rangle = \frac{1}{\sqrt{2^3}} \sum_{z=0}^{2^3-1} (-1)^{x \cdot z} |z\rangle$

$H^{\otimes 3}|x=4\rangle = \frac{1}{\sqrt{2^3}} \sum_{z=0}^{2^3-1} (-1)^{x \cdot z} |z\rangle$ General for $n=3$

PHYC90045 Introduction to Quantum Computing

Deutsch-Josza Walkthrough

$|0\rangle, |1\rangle \xrightarrow{\text{H}} U_f \xrightarrow{\text{H}}$

$H^{\otimes n}|x\rangle = \frac{1}{\sqrt{N}} \sum_{z=0}^{N-1} (-1)^{x \cdot z} |z\rangle$

And after the final Hadamard gates:

$|\psi\rangle = H^{\otimes n} \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} (-1)^{f(x)} |x\rangle$

$|\psi\rangle = \frac{1}{N} \sum_{x=0}^{N-1} (-1)^{f(x)} \sum_{z=0}^{N-1} (-1)^{x \cdot z} |z\rangle$

PHYC90045 Introduction to Quantum Computing

Constant function

For a constant function ($f(x) = 0$ for all x , or $f(x) = 1$ for all x):

$$\begin{aligned} |\psi\rangle &= \frac{1}{N} \sum_{x=0}^{N-1} (-1)^{f(x)} \sum_{z=0}^{N-1} (-1)^{x \cdot z} |z\rangle \\ &= \frac{(-1)^{f(0)}}{N} \sum_{x=0}^{N-1} \sum_{z=0}^{N-1} (-1)^{x \cdot z} |z\rangle \quad \sum_{x=0}^{N-1} (-1)^{x \cdot z} = \begin{cases} N, & z = 0 \\ 0, & z \neq 0 \end{cases} \\ &= \frac{(-1)^{f(0)}}{N} \sum_{z=0}^{N-1} \left(\sum_{x=0}^{N-1} (-1)^{x \cdot z} \right) |z\rangle \\ &= (-1)^{f(0)} |z=0\rangle \end{aligned}$$

So for a constant function "0" will always be measured (global phase is unimportant).

PHYC90045 Introduction to Quantum Computing

Balanced Function



$$|\psi\rangle = H^{\otimes n} \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} (-1)^{f(x)} |x\rangle$$

$$|\psi\rangle = \frac{1}{N} \sum_{x=0}^{N-1} (-1)^{f(x)} \sum_{z=0}^{N-1} (-1)^{x \cdot z} |z\rangle$$

For a balanced function (equal number of $f(x) = 0$ and $f(x) = 1$):

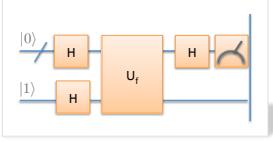
$$|\psi\rangle = \frac{1}{N} \sum_{z=0}^{N-1} \left(\sum_{x, f(x)=0} (-1)^{x \cdot z} - \sum_{x, f(x)=1} (-1)^{x \cdot z} \right) |z\rangle$$

Which has zero amplitude for the $|z=0\rangle$ state, and non-zero for other states.

PHYC90045 Introduction to Quantum Computing

Deutsch-Josza Walkthrough





If 0 is measured, then the function is constant.
If any other value is measured, then the function is balanced.

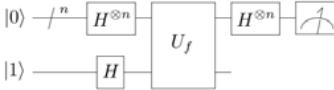
The Deutsch-Josza algorithm evaluates if a function is constant or balanced with a single query. Classically we would require $O(2^n)$ queries.

Of course, there are classical probabilistic algorithms with establish with high probability in few queries, but only with high probability of success not with certainty.

PHYC90045 Introduction to Quantum Computing

Deutsch-Josza algorithm





- Given a Boolean function, f , determine if:
 - f is constant (always gives the same result)
 - f is balanced (gives equal numbers of 0s and 1s)
- Classical algorithm (worst case) needs $2^n/2+1$ queries
- Quantum algorithm needs just 1 query.

PHYC90045 Introduction to Quantum Computing

Aside: Universality in Quantum Computing

In classical computing the NAND gate is **universal**: every Boolean function can be implemented as a sequence of NAND (NOT AND) gates

In quantum computing every quantum circuit can be expressed as a sequence of:

CNOT + Single Qubit Rotations

PHYC90045 Introduction to Quantum Computing

Aside: Universality in Quantum Computing

In classical computing the NAND gate is **universal**: every Boolean function can be implemented as a sequence of NAND (NOT AND) gates

In quantum computing every quantum circuit can be approximated as a sequence of:

CNOT + H + T
Hadamard T Gate (Z axis, pi/4)

PHYC90045 Introduction to Quantum Computing

Aside: Outline of Universal Gate Set Proof

(1) The following sequence of gates creates an irrational fraction of 2π angle rotation gate:
 THTH

Specifically, you can show by direct multiplication that it is a rotation around

$$\vec{n} = \left(\cos \frac{\pi}{8}, \sin \frac{\pi}{8}, \cos \frac{\pi}{8} \right)$$

By an angle defined by: $\cos \frac{\theta}{2} = \cos^2 \frac{\pi}{8}$

(2) You can use this to approximate (within some error, δ) **every rotation around \vec{n}** . This takes $\sim 2n/\delta$ applications of THTH.

PHYC90045 Introduction to Quantum Computing

Aside: Outline of Universal Gate Set Proof

(3) The following sequence of gates creates an irrational fraction of 2π

$$HTHT$$

Around a different axis:

$$\vec{m} = \left(\cos \frac{\pi}{8}, -\sin \frac{\pi}{8}, \cos \frac{\pi}{8} \right)$$

By an angle defined by: $\cos \frac{\theta}{2} = \cos^2 \frac{\pi}{8}$

(4) You can use this to approximate (within some error, ϵ) **every rotation around \mathbf{m}** . This takes $\sim 2\pi/\epsilon$ applications of HTHT.

PHYC90045 Introduction to Quantum Computing

Aside: Outline of Universal Gate Set Proof

(5) These gates can (approximately) implement any rotation around two different axes, you can use this to approximate any single qubit rotation.

(6) Single qubit rotations plus the CNOT gate is universal

(7) Therefore {H, T, CNOT} is universal.

Full proof (carefully keeping track of approximations) is found in Nielsen and Chuang 4.5.3

PHYC90045 Introduction to Quantum Computing

Week 3

Lecture 5
Universality in quantum computing, Reversible computation, one qubit adder, the Deutsch-Josza algorithm

Lecture 6
Two basic quantum algorithms: Bernstein-Vazirani and Simon's Algorithms

Lab 3
Logical statements, Reversible logic, Adder, Deutsch-Josza algorithm

PHYC90045 Introduction to Quantum Computing

Week 3

Lecture 5
Reversible computation, One qubit adder, the Deutsch-Josza algorithm

Lecture 6
Two basic quantum algorithms: Bernstein-Vazirani and Simon's Algorithms

Lab 3
Logical statements, Reversible logic, Adder, Deutsch-Josza algorithm

PHYC90045 Introduction to Quantum Computing

Simple Quantum Algorithms:
Simon
and Bernstein-Vazirani

Physics 90045
Lecture 6

PHYC90045 Introduction to Quantum Computing

Overview

In this lecture we will discuss some of the early quantum algorithms,

1. Bernstein-Vazirani algorithm
2. Simon's algorithm

These algorithms can be taken as simple demonstrations of quantum computation, even if they are of limited practical use.

See:

Kaye, Chapter 6
Nielsen and Chuang, Chapters 1 & 4
Reiffel, 7.1-7.5

PHYC90045 Introduction to Quantum Computing

Bernstein-Vazirani Problem

Given a Boolean function, f :

$$f(x) = x \cdot s \mod 2$$

find s .

Recall, bitwise product: $x \cdot s = \sum_i x_i s_i$

PHYC90045 Introduction to Quantum Computing

Example: Linear Boolean function

Example:

$$f(x) = x \cdot 5 \mod 2$$

Remember, in binary, $5 = 101$.

x	f(x)
000	0
001	1
010	0
011	1
100	1
101	0
110	1
111	0

Given a black-box which calculates this function, find $s=5$.

PHYC90045 Introduction to Quantum Computing

Solving BV Problem Classically

$f(x) = x \cdot 5 \mod 2$

x	f(x)
000	0
001	1
010	0
011	1
100	1
101	0
110	1
111	0

Input one single digit "1" at a time.

Can determine s using n queries.

PHYC90045 Introduction to Quantum Computing

Bernstein-Vazirani Problem

Given a Boolean function, f :

$$f(x) = x \cdot s \mod 2$$

find s .

Recall: bitwise product: $x \cdot s = \sum_i x_i s_i$

- Classical algorithm needs n queries
- Quantum algorithm needs just 1 query.

PHYC90045 Introduction to Quantum Computing

Bernstein-Vazirani algorithm

The circuit is the same as for the Deutsch-Josza algorithm:

The guarantees on f are different:

$$f(x) = x \cdot s \mod 2$$

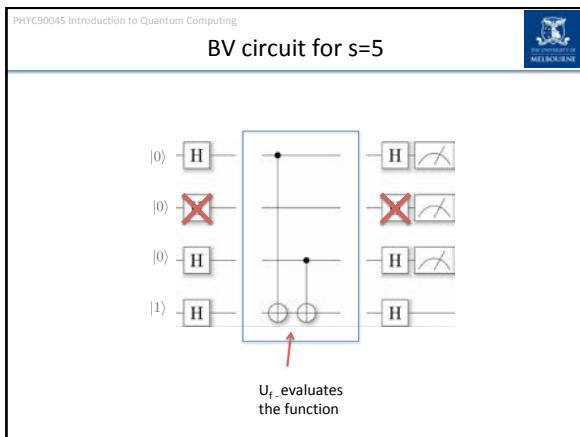
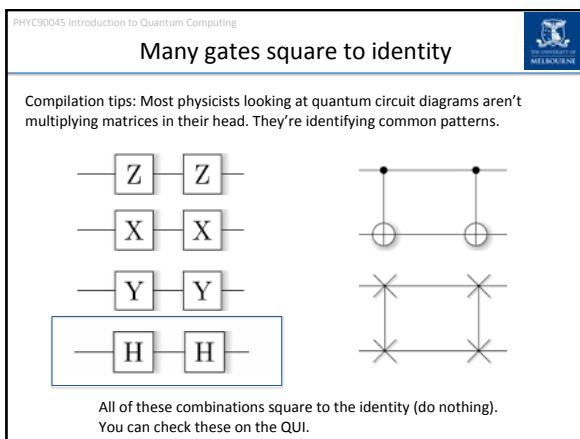
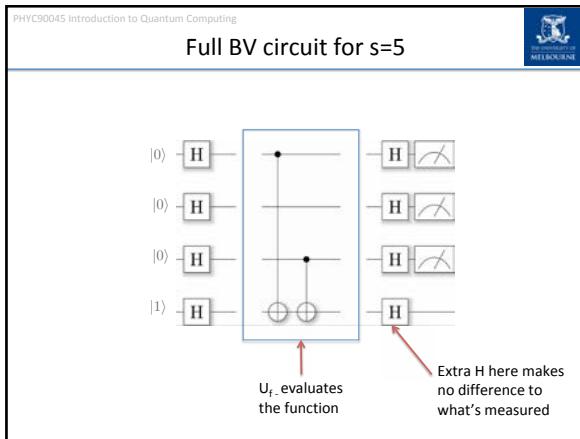
Recall: Deutsch-Josza algorithm required the function to either be constant or balanced.

PHYC90045 Introduction to Quantum Computing

Implementing a Linear Boolean Function

For $s = 5 = 101_2$ the function is evaluated using this circuit:

The bits of s determine the location of the CNOTs.
Every linear Boolean function has a circuit of the same form.



PHYC90045 Introduction to Quantum Computing

Circuit identity: Inverted CNOT

Exercise: You can verify this by writing out the matrices and multiplying!

PHYC90045 Introduction to Quantum Computing

Simple explanation of BV

Hadamard gates “conjugating” CNOT:

We can determine s with just one query, by making use of quantum superposition.

PHYC90045 Introduction to Quantum Computing

Simplifying circuit

If in doubt, check using QUI

PHYC90045 Introduction to Quantum Computing

BV Solution

This circuit will measure:
 $101 = 5$
which is correct ($s=5$)

A similar reduction would work for any s , but let us prove that formally.

PHYC90045 Introduction to Quantum Computing

Bernstein-Vazirani algorithm

The circuit is the same as for the Deutsch-Josza algorithm:

The guarantees on f are different:
 $f(x) = x \cdot s \mod 2$

Recall: Deutsch-Josza algorithm required the function to either be constant or balanced.

PHYC90045 Introduction to Quantum Computing

BV algorithm explained

State after the initial Hadamard gates:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Sum of all computational basis states (again)

PHYC90045 Introduction to Quantum Computing

Recall: General Function Phase Kickback

Using phase kickback, after the function has been applied:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x (-1)^{f(x)} |x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

If the function evaluates to "1" then the target qubit is flipped, and we pick up a phase. Otherwise, there is no phase applied. This is a simple way to write that.

PHYC90045 Introduction to Quantum Computing

BV algorithm explained

Using phase kickback, after the function has been applied:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} (-1)^{f(x)} |x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Phase kickback

$$= \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} (-1)^{x \cdot s} |x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Since $f(x) = x \cdot s \pmod{2}$

PHYC90045 Introduction to Quantum Computing

Recall: Hadamard applied to a general state

Amplitude $a_z \rightarrow$ how many times does the binary representation of z and x have 1's in the same location?

$$|x\rangle \rightarrow H^{\otimes n} |x\rangle = \frac{1}{\sqrt{N}} \sum_{z=0}^{N-1} a_z |z\rangle$$

Shorthand for the bitwise dot product is: $x \cdot z = \sum_{j=0}^n x_j z_j$

When 1's in the same location, we get a sign change $\rightarrow (-1)^{x \cdot z}$

Hadamards applied to a general state (n qubits, $N = 2^n$):

$$H^{\otimes n} |x\rangle = \frac{1}{\sqrt{N}} \sum_{z=0}^{N-1} (-1)^{x \cdot z} |z\rangle$$

PHYC90045 Introduction to Quantum Computing

BV algorithm explained

Considering the upper register only:

$$\begin{aligned} |\psi\rangle &= \text{H}^{\otimes n} \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} (-1)^{x \cdot s} |x\rangle \\ &= \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} (-1)^{x \cdot s} \frac{1}{\sqrt{N}} \sum_{z=0}^{N-1} (-1)^{x \cdot z} |z\rangle \quad \text{H's applied to basis state} \\ &= \frac{1}{N} \sum_{x=0}^{N-1} \sum_{z=0}^{N-1} (-1)^{x \cdot (s \oplus z)} |z\rangle = \frac{1}{N} \sum_{z=0}^{N-1} \left(\sum_{x=0}^{N-1} (-1)^{x \cdot (s \oplus z)} \right) |z\rangle \\ &\underline{x \oplus z = x_0 + z_0 \mod 2, x_1 + z_1 \mod 2, \dots} \end{aligned}$$

PHYC90045 Introduction to Quantum Computing

BV algorithm explained

Simplifying the sum:

$$\begin{aligned} |\psi\rangle &= \frac{1}{N} \sum_{z=0}^{N-1} \left(\sum_{x=0}^{N-1} (-1)^{x \cdot (s \oplus z)} \right) |z\rangle \\ &= \frac{1}{N} \sum_{z=0}^{N-1} (-1)^0 |s\rangle \quad \text{This sum (over x) is zero unless } s \oplus z = 0 \\ &= |s\rangle \quad \text{That is, z and s are bitwise identical, ie. } z = s \end{aligned}$$

We will therefore measure s with certainty – the aim of the algorithm.

PHYC90045 Introduction to Quantum Computing

Bernstein-Vazirani Algorithm

Given a Boolean function, f :

$$f(x) = x \cdot s \mod 2$$

find s .

$$x \cdot s = \sum_i x_i s_i$$

- Classical algorithm needs n queries
- Quantum algorithm needs just 1 query.

PHYC90045 Introduction to Quantum Computing



Third Quantum Algorithm: Simon's algorithm

PHYC90045 Introduction to Quantum Computing



Simon's Problem

Given a 2-to-1 function, f , such that

$$f(x) = f(x \oplus a)$$

Find a .

Unlike the previous two examples, here the range of $f(x)$ is \mathbb{Z} , integers.
 Simon's algorithm is an example of a "Hidden (Abelian) subgroup problem" (HSP) and was the inspiration for Shor's factoring algorithm.

PHYC90045 Introduction to Quantum Computing



Example of a hidden a

x	$f(x)$
000	0
001	1
010	2
011	3
100	2
101	3
110	0
111	1

$f(001) = f(111)$

We would like to find the hidden ' a ' s.t.

$$f(x) = f(x \oplus a)$$

In this case:
 $a = 110_2 = 6$

PHYC90045 Introduction to Quantum Computing

Solving Simon's problem classically

Just try different inputs until you see a collision:

$$\begin{aligned} f(000) &= 0 \\ f(011) &= 3 \\ \textcolor{red}{f(111)} &= 1 \\ f(010) &= 2 \\ f(001) &= 1 \end{aligned}$$

Actually this is equivalent to the famous “birthday” problem, and takes fewer queries than you might expect. Probabilistically, if there are N different inputs we need

$$O(\sqrt{N})$$

Evaluations of the function before we find a collision.

Simon's algorithm does the same with $O(n)$ queries.

PHYC90045 Introduction to Quantum Computing

Simon's algorithm circuit

Randomly measure a result of the function. Collapse to a superposition of inputs which give that value. Send these through Hadamard gates, and measure:

x {
 $|0\rangle$ H U_f H } Measure to find a
 $|0\rangle$
 $f(x_0), f(x_0 \oplus a)$

PHYC90045 Introduction to Quantum Computing

Simon's algorithm

After the initial Hadamard gates:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle |0\rangle$$

PHYC90045 Introduction to Quantum Computing

Simon's algorithm

After evaluation of the function:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x U_f |x\rangle |0\rangle$$

$$= \frac{1}{\sqrt{N}} \sum_x |x\rangle |f(x)\rangle$$

PHYC90045 Introduction to Quantum Computing

Simon's algorithm

It's easiest to consider that the bottom register is measured first. Before measurement the state is:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle |f(x)\rangle$$

Some value, $f(x_0)$ will be measured at random, and the top register collapses to:

$$|\psi\rangle = \frac{|x_0\rangle + |x_0 \oplus a\rangle}{\sqrt{2}}$$

PHYC90045 Introduction to Quantum Computing

Example: Measuring function

$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle |f(x)\rangle$

$$= \frac{1}{\sqrt{8}} (|0\rangle |0\rangle + |1\rangle |1\rangle + |2\rangle |2\rangle + |3\rangle |3\rangle + |4\rangle |2\rangle + |5\rangle |3\rangle + |6\rangle |0\rangle + |7\rangle |1\rangle)$$

If we measure the second register, and measure obtain "3", the state collapses to only those states compatible with this measurement:

$$|\psi'\rangle = \frac{|3\rangle |3\rangle + |5\rangle |3\rangle}{\sqrt{2}}$$

$$= \frac{|3\rangle + |5\rangle}{\sqrt{2}} \otimes |3\rangle$$

First register: $|\psi\rangle = \frac{|x_0\rangle + |x_0 \oplus a\rangle}{\sqrt{2}}$

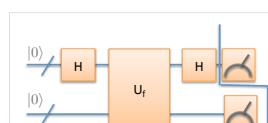
x	f(x)
000	0
001	1
010	2
011	3
100	2
101	3
110	0
111	1

PHYC90045 Introduction to Quantum Computing



 The University of
 MELBOURNE

Simon's algorithm

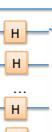


We now apply Hadamard to the top register:

$$|\psi\rangle = H^{\otimes n} \frac{|x_0\rangle + |x_0 \oplus a\rangle}{\sqrt{2}}$$

PHYS90045 Introduction to Quantum Computing

Hadamard applied to a general state



Amplitude $a_y >$ how many times does the binary representation of y and x have 1's in the same location?

$$|x\rangle \xrightarrow{\text{H}^{\otimes 5}} \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} a_y |y\rangle = x_0 y_0 + x_1 y_1 + x_2 y_2 + \dots + x_n y_n$$

Shorthand for the bitwise dot product is: $x \cdot y = \sum_{j=0}^n x_j y_j$

When 1's in the same location, we get a sign change $\rightarrow (-1)^{x \cdot y}$

Hadamards applied to a general state (n qubits, $N = 2^n$):

$$H^{\otimes n} |x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} (-1)^{x \cdot y} |y\rangle$$

(changed dummy index to y)

PHYC90045 Introduction to Quantum Computing



Simon's algorithm

$$\begin{aligned}
 |\psi\rangle &= H^{\otimes n} \frac{|x_0\rangle + |x_0 \oplus a\rangle}{\sqrt{2}} \\
 &= \frac{1}{\sqrt{2^{n+1}}} \sum_y \left((-1)^{x_0 \cdot y} + (-1)^{(x_0 \oplus a) \cdot y} \right) |y\rangle \\
 &= \frac{1}{\sqrt{2^{n+1}}} \sum_y (-1)^{x_0 \cdot y} (1 + (-1)^{a \cdot y}) |y\rangle
 \end{aligned}$$

The amplitude of any state, y , is zero unless:

$$a \cdot y = 0 \mod 2$$

Therefore, the state therefore becomes:

$$|\psi\rangle = \frac{1}{\sqrt{2^{n-1}}} \sum_{a \cdot y = 0} (-1)^{x_0 \cdot y} |y\rangle$$

PHYC90045 Introduction to Quantum Computing

Simon's algorithm

$$|\psi\rangle = \frac{1}{\sqrt{2^{n-1}}} \sum_{a \cdot y = 0} (-1)^{x_0 \cdot y} |y\rangle$$

$$a \cdot y = 0 \pmod{2}$$

Each time we measure, we randomly measure a "y" which is orthogonal to "a":

Obtain n random y's this way and **perform Gauss/Jordan elimination** to obtain "a"

PHYC90045 Introduction to Quantum Computing


THE UNIVERSITY OF
MELBOURNE

Example of Simon's algorithm

x	f(x)
000	0
001	1
010	2
011	3
100	2
101	3
110	0
111	1

We would like to find the hidden 'a' s.t.

$$f(x) = f(x \oplus a)$$

In this case, $a=110_2=6$

PHYC90045 Introduction to Quantum Computing



Running the circuit



The circuit consists of two horizontal lines representing qubits. The top qubit starts with a $|0\rangle$ state, passes through a Hadamard gate (H), then a unitary U_f , and another Hadamard gate (H). The bottom qubit starts with a $|0\rangle$ state, passes through a unitary U_f , and then a measurement symbol. A red arrow points from the text $a \cdot y = 0 \pmod{2}$ to the measurement result of the bottom qubit.

We run the circuit, and at random, obtain measure the results:

001
110
111

We want to find,
 $a \cdot 110_2 = 6$

PHYC90045 Introduction to Quantum Computing

In matrix form

We know that $a \cdot y = 0 \pmod{2}$

We have three values of 'y' for which this is true, so we can write a system of linear equations for the bits of 'a':

$$\mathbf{Y}\vec{a} = \vec{0}$$

Measured values	001	110	111	\rightarrow	$\left[\begin{array}{ccc c} 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right]$
-----------------	-----	-----	-----	---------------	---

Solving for a

Solution is degenerate. $a=(0,0,0)$ or $a_1=a_2=1$ ie. $a=(1,1,0)$

PHYC90045 Introduction to Quantum Computing

Simon's Algorithm

Given a 2-to-1 function, f , such that

$$f(x) = f(x \oplus a)$$

Find a .

Classical algorithm: $O(\sqrt{N})$ Queries to the oracle (probabilistically)

Quantum algorithm: $O(n)$ Queries to the oracle

PHYC90045 Introduction to Quantum Computing

This Week

Lecture 1
Quantum search – introduction to Grover's algorithm for amplitude amplification, geometric interpretation

Lecture 2
Optimality, Succeeding with Certainty, Quantum Counting

Lab
Grover's algorithm

PHYC90045 Introduction to Quantum Computing

Grover's Algorithm

Physics 90045
Lecture 7

PHYC90045 Introduction to Quantum Computing

Introduction to Grover's algorithm

• This lecture: Grover's search algorithm

- Grover's algorithm
- Worked Example
- Geometric interpretation

References:
Reiffel, Chapter 9.1-9.2
Kaye, Chapter 8.1-8.2
Nielsen and Chuang, Chapter 6.1-6.2

PHYC90045 Introduction to Quantum Computing

Reminder: Outer Product

For two quantum states $|\psi\rangle = \begin{bmatrix} a \\ b \end{bmatrix}$, $|\phi\rangle = \begin{bmatrix} c \\ d \end{bmatrix}$

We can define an outer product between them:

$$|\psi\rangle \langle \phi| = \begin{bmatrix} a \\ b \end{bmatrix} \otimes \begin{bmatrix} c^* & d^* \end{bmatrix} = \begin{bmatrix} ac^* & ad^* \\ bc^* & bd^* \end{bmatrix}$$

$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ $|1\rangle \langle 2|$
For number basis states,
this specifies a matrix
with a single "1" in the
location 1,2. In general:
 $|\text{row}\rangle \langle \text{column}|$

PHYC90045 Introduction to Quantum Computing

Unordered Search

Grover's algorithm performs a similar* problem to this: You are given a telephone book

And a phone number: 23675

Your task:
Find the name which goes with that number...



Part of Norfolk Island's telephone book, with people listed by nickname (Photo: Wikicommons)

* Not all that similar, better examples later....

PHYC90045 Introduction to Quantum Computing

Quantum search – Grover's problem

Given an black box (oracle), U_f , which computes the function:
 $f: \{0,1\}^n \rightarrow \{0,1\}$

Find an x s.t. $f(x) = 1$

PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

Grover's Algorithm (1996)

- Unordered search, find one marked item among many
- Classically, this requires $N/2$ queries to the oracle
- Quantum mechanically, requires only $O(\sqrt{N})$ queries.

Simple problem = search for one integer marked by the oracle.

High level structure:

The diagram illustrates the high-level structure of Grover's algorithm. It shows a quantum circuit with four qubits. The first three qubits follow a similar sequence of operations: they start with a Hadamard (H) gate, followed by an Oracle block, then another H gate, and finally a sequence of H gates. The fourth qubit follows a different sequence: it starts with an Inversion block (labeled $(I - 2|0\rangle\langle 0|)$), followed by a sequence of H gates. The circuit concludes with three measurement blocks at the end of each line.

Lov Grover

PHYC90045 Introduction to Quantum Computing

Two basic steps in Grover's algorithm

Quantum database: $|\Phi\rangle = \frac{1}{\sqrt{N}} \sum_i |i\rangle$ (i.e. all integers 0 to N-1)

The diagram illustrates a quantum circuit with five qubits. The first four qubits represent the input state $|\Phi\rangle$, which is a superposition of all states from 0 to N-1. The fifth qubit represents the marked state $|m\rangle$. The circuit consists of two main parts enclosed in a green box:

- Oracle:** A blue block containing a sequence of operations: Hadamard (H) gates followed by a control operation, then another sequence of Hadamard gates.
- Inversion:** A grey block labeled $(|2>-|0>)$ representing an "Inversion about the mean".

These two operations are repeated multiple times, indicated by three orange curly braces on the right side of the green box. Below the circuit, text explains the oracle identifies a particular marked state, m, and the inversion step is "Inversion about the mean". A callout box on the right says "Repeat these two operations O(N) times".

PHYS90045 Introduction to Quantum Computing

The Oracle

The task of recognizing the correct solution goes to the “oracle”.

Binary function, or “oracle”
Identifying a marked state, m

$|x\rangle$ $|x\rangle$
 $|y\rangle$ $|y \oplus f(x)\rangle$

U_f

Oracle

Designed to flip the last bit if the input, j , is a solution

The oracle is just a Boolean function (as seen in previous lectures)

PHYC90045 Introduction to Quantum Computing

Phase kickback for Boolean function

Binary function, or "oracle"

After the function has been applied:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} (-1)^{f(i)} |i\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

If the oracle function evaluates to "1" then the target qubit is flipped, and we pick up a phase (associated with the control qubit state). Otherwise, there is no phase applied. This is a simple way to write that.

PHYC90045 Introduction to Quantum Computing

Example: Oracle recognizing the state "2 = |10>"

The effect on each of the 4 states in the 2-qubit control register, x:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = I - 2|10\rangle\langle10|$$

PHYC90045 Introduction to Quantum Computing

The marked state

Initially in Grover's algorithm, we will be searching for a single (integer) solution, m . In that case the effect of the oracle on the control register is:

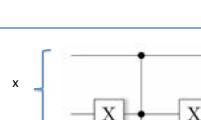
$$I - 2|m\rangle\langle m| \quad (\text{in decimal ket notation})$$

As a matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Here, as in future slides, we are only writing out the control qubits (in this case 2 qubits only).

Example: Oracle recognizing the state "2 = $|10\rangle$ "



The circuit consists of two horizontal lines representing qubits. A control box labeled 'X' is placed on the top line. A target box labeled 'X' is placed on the bottom line. A vertical line connects the two boxes. A bracket labeled 'x' is positioned to the left of the boxes. Below the boxes, a circle with a diagonal line through it is placed between the two lines. To the left of this circle, the expression $\frac{|0\rangle - |1\rangle}{\sqrt{2}}$ is written.

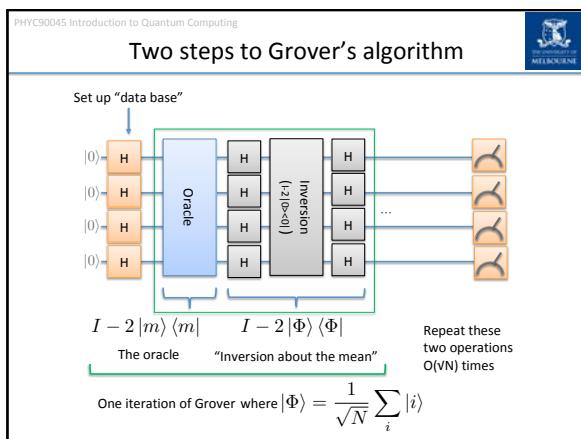
Phase kickback

$ 00\rangle \rightarrow 00\rangle$
$ 01\rangle \rightarrow 01\rangle$
$ 10\rangle \rightarrow - 10\rangle$
$ 11\rangle \rightarrow 11\rangle$

In practice we can implement the oracle without the check qubit using a controlled-Z gate (ex. Show the circuit right marks the state $|10\rangle$, i.e. $|m=2\rangle$).



The simplified circuit shows two horizontal lines. The first line has a box labeled 'X'. The second line has a box labeled 'Z'. A vertical line connects the two boxes. A bracket labeled 'x' is positioned to the left of the boxes. This represents a controlled-Z gate where the Z gate is applied only if the control qubit is in state |1>.



PHYS90045 Introduction to Quantum Computing

The University of MELBOURNE

Unpicking the details: “Inversion” operation

The “Inversion” part is just applying a phase to the zero state:

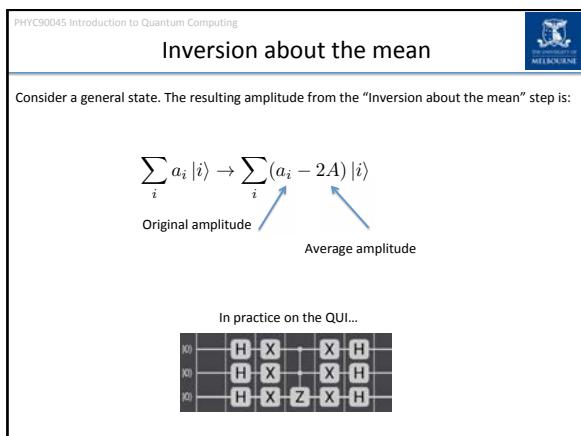
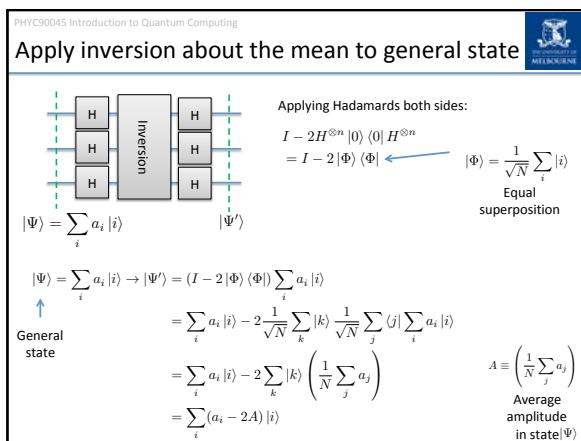
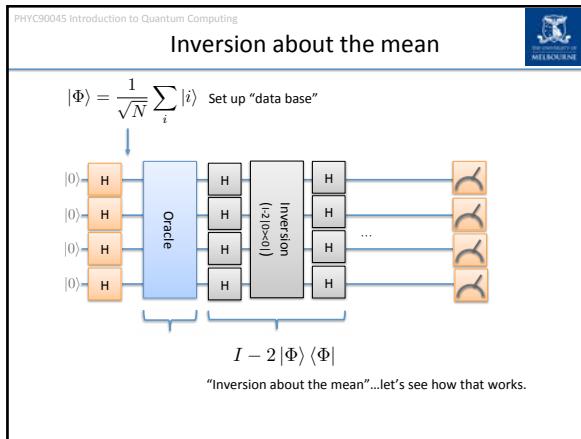


$I - 2|0\rangle\langle 0| = \begin{bmatrix} -1 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & \dots \\ 0 & 0 & 0 & 1 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}$

How? Recall outer product etc: $|\psi\rangle\langle\phi| = \begin{bmatrix} a \\ b \end{bmatrix} \otimes \begin{bmatrix} c^* & d^* \end{bmatrix} = \begin{bmatrix} ac^* & ad^* \\ bc^* & bd^* \end{bmatrix}$

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad |0\rangle\langle 0| = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \otimes (10\dots 0) = \begin{pmatrix} 10\dots 0 \\ 00\dots 0 \\ \vdots \\ 00\dots 0 \end{pmatrix} \quad I = \begin{pmatrix} 100\dots 0 \\ 010\dots 0 \\ 001\dots 0 \\ \vdots \\ 000\dots 1 \end{pmatrix}$$

$$I - 2|0\rangle\langle 0| = \begin{pmatrix} 100\dots 0 \\ 010\dots 0 \\ 001\dots 0 \\ \vdots \\ 000\dots 1 \end{pmatrix} - 2 \begin{pmatrix} 10\dots 0 \\ 00\dots 0 \\ \vdots \\ 00\dots 0 \end{pmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & \dots \\ 0 & 0 & 0 & 1 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$



PHYC90045 Introduction to Quantum Computing

Inversion about the mean

Amplitudes of the state, before and after:

The diagram shows two horizontal number lines representing amplitude. The top line has a central point labeled 'Mean'. Above the mean, there is a double-headed vertical arrow labeled $a_i - A$. Below the mean, there is another double-headed vertical arrow labeled $a_i - A$. To the right of the mean, there is a double-headed vertical arrow labeled A . The total length between the two outermost arrows is labeled $2A - a_i$. The bottom line has a single vertical arrow labeled a_i .

When the state undergoes this transformation:

$$\sum_i a_i |i\rangle \rightarrow -\sum_i (2A - a_i) |i\rangle$$

PHYC90045 Introduction to Quantum Computing

Effect of inversion about the mean

Amplitude

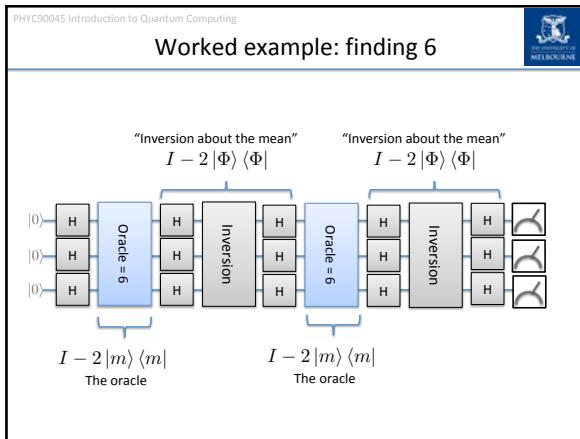
The diagram shows two horizontal number lines representing amplitude. The top line is labeled 'State' and has a red horizontal line labeled 'The mean'. A blue vertical tick labeled 'The marked state' is positioned below the mean. A blue arrow points down to the bottom line, which is also labeled 'State'. The bottom line has a red horizontal line labeled 'The mean'. The same blue tick labeled 'The marked state' is now positioned above the mean. A red arrow points from the text 'Increased amplitude of the marked state' to this tick.

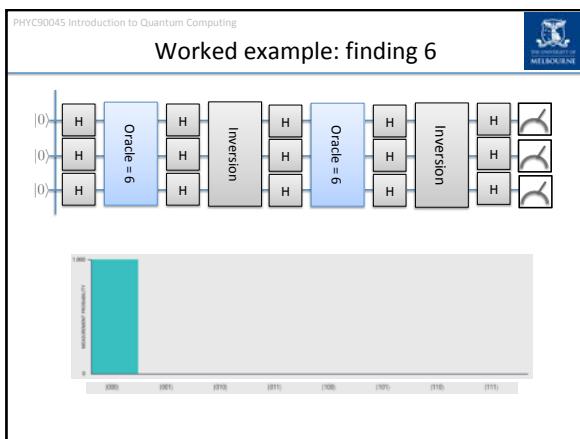
Inversion about the mean

PHYC90045 Introduction to Quantum Computing

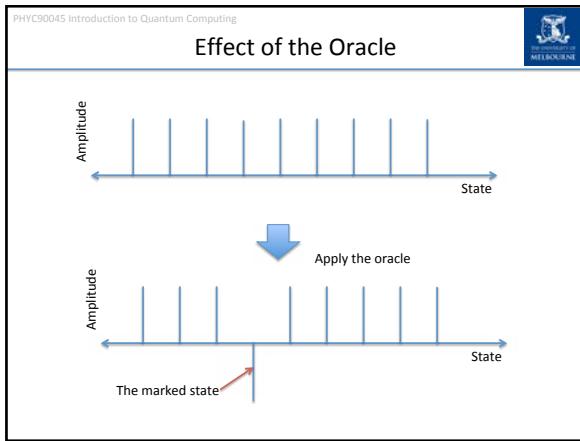
Interactive Example

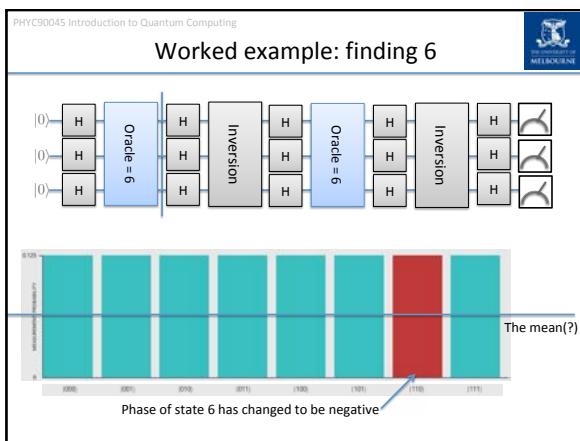
<https://codepen.io/samtonetto/full/BVOGmW>

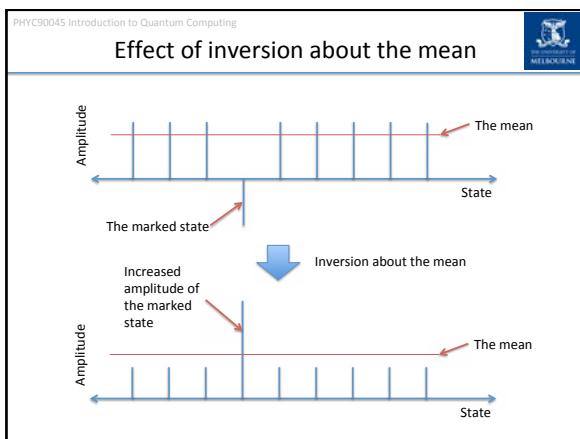




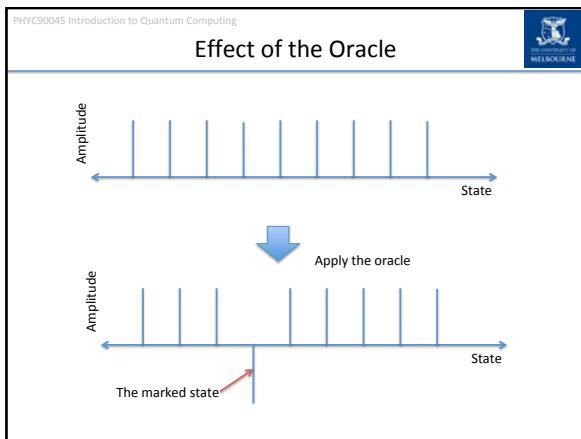




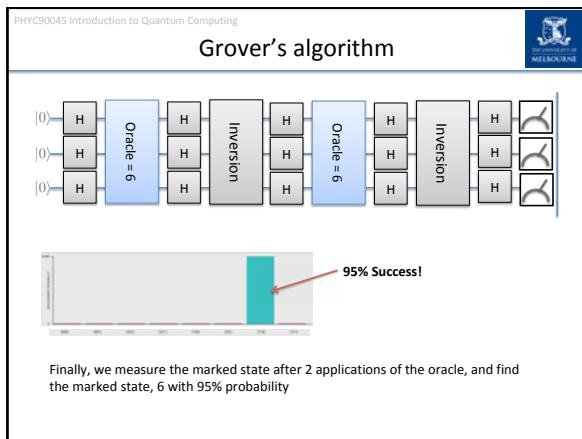
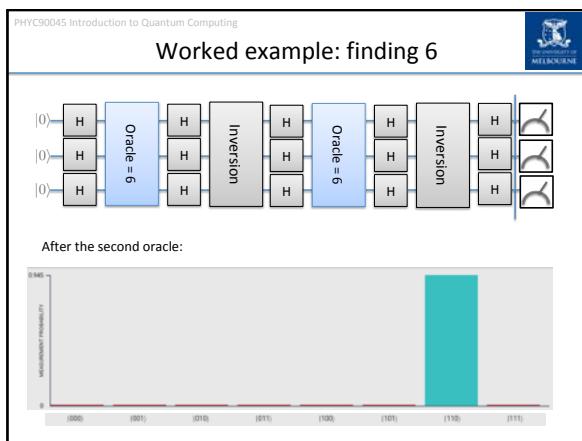
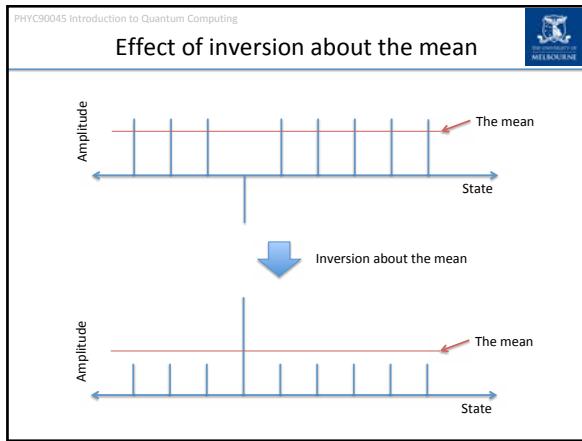












PHYC90045 Introduction to Quantum Computing

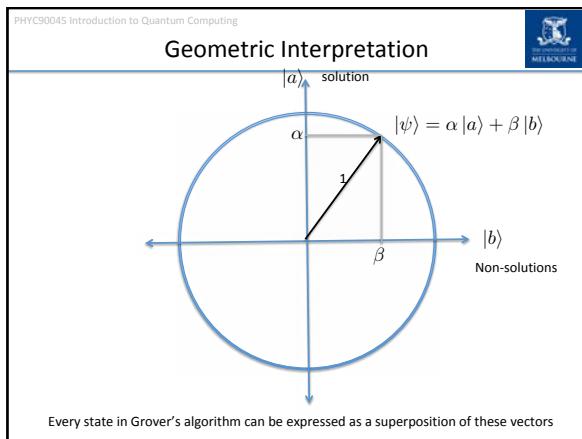
Geometric interpretation of Grover's algorithm

A very useful basis:

$$|a\rangle = |m\rangle \quad \text{Solution!} \quad \text{Smiley face icon}$$

$$|b\rangle = \frac{1}{\sqrt{N-1}} \sum_{i \notin \text{solutions}} |i\rangle \quad \text{Non-solutions...} \quad \text{No entry icon}$$

We only need to consider the amplitude of these two states in Grover's algorithm.
Every operation is also real, so we can plot on a circle.



PHYC90045 Introduction to Quantum Computing

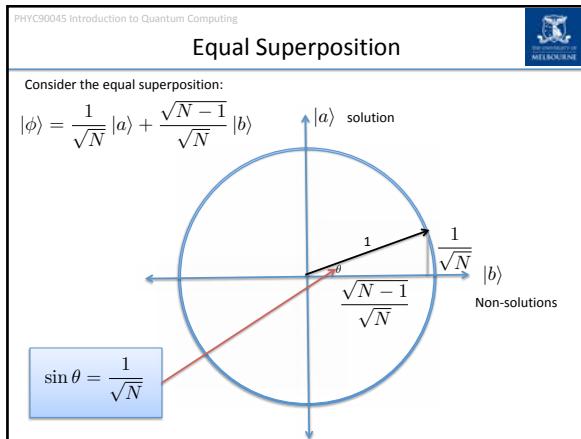
Equal superposition

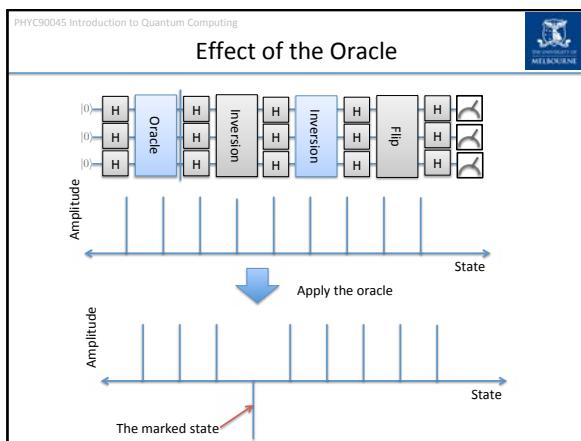
Equal superposition state:

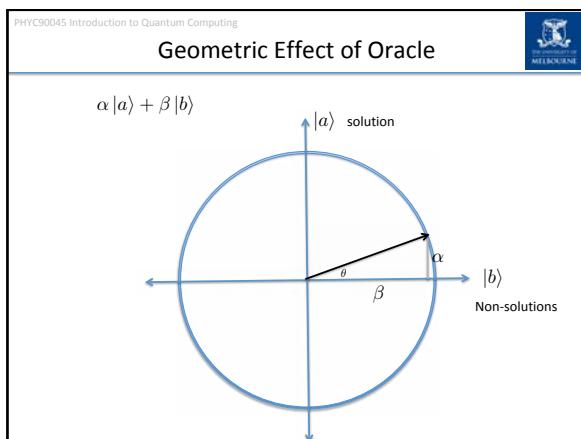
$$|\Phi\rangle = \frac{1}{\sqrt{N}} \sum_i |i\rangle$$

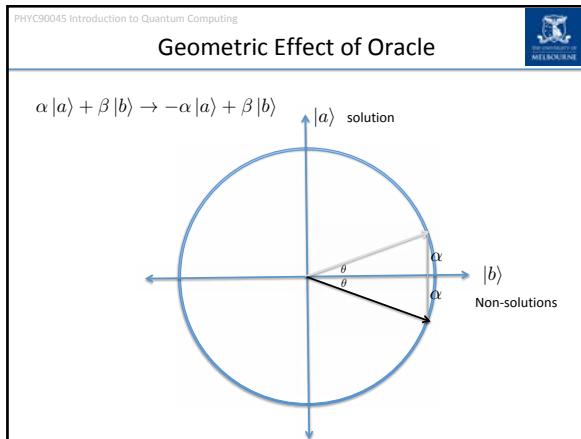
$$= \frac{1}{\sqrt{N}} |a\rangle + \frac{\sqrt{N-1}}{\sqrt{N}} |b\rangle$$

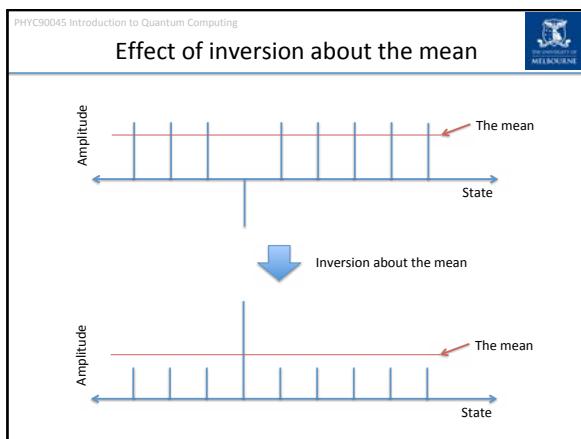
$$|a\rangle = |m\rangle \quad |b\rangle = \frac{1}{\sqrt{N-1}} \sum_{i \notin \text{solutions}} |i\rangle$$

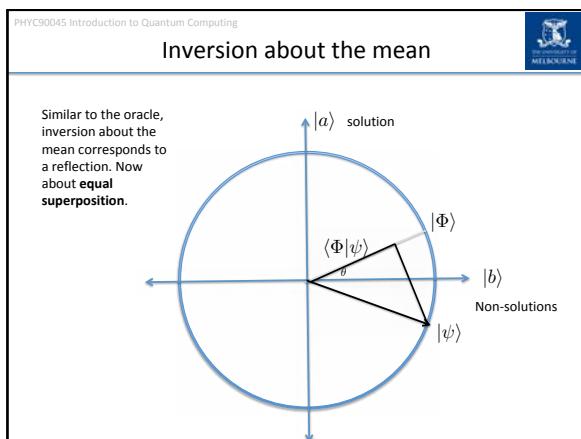


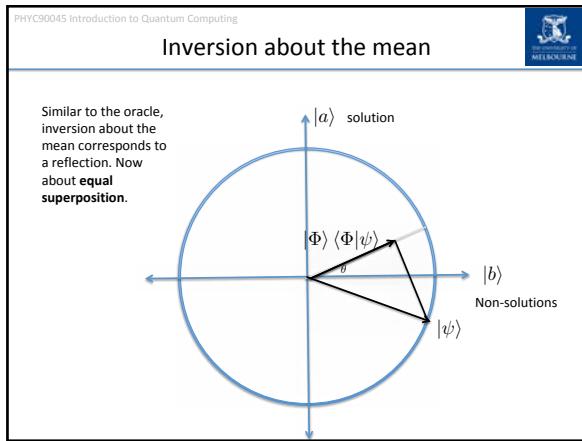


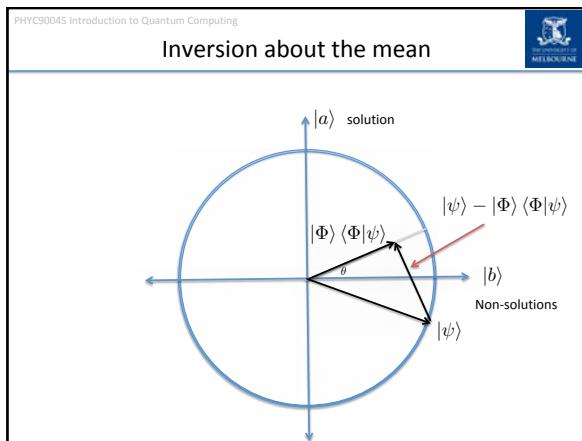


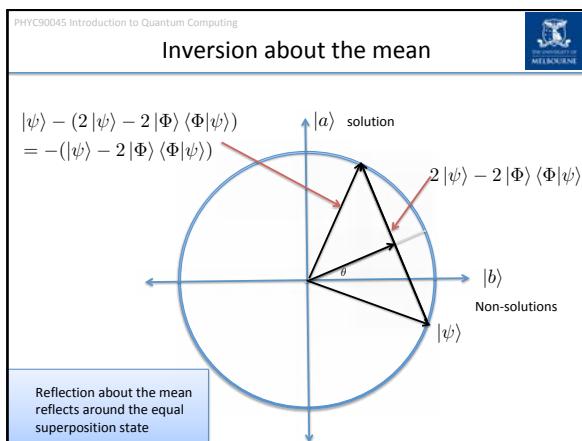












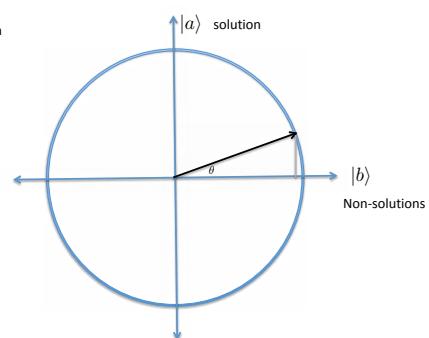
PHYS90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

Geometric Effect of both Oracle and Inversion

Combining both effects:





PHYC90045 Introduction to Quantum Computing

Geometric Effect of both Oracle and Inversion

Combining both effects:



PHYC90045 Introduction to Quantum Computing

Geometric Effect of both Oracle and Inversion

Combining both effects:

$|a\rangle$ solution

$|b\rangle$

Non-solutions

2θ

θ

θ



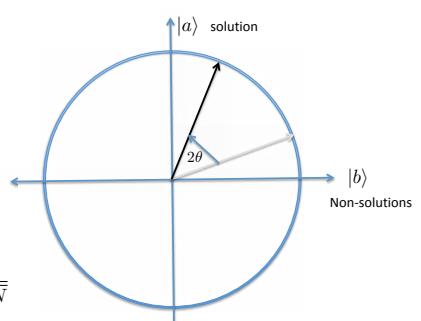
PHYS90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

Total effect of one Grover iteration

Product of two reflections is a rotation.



A diagram illustrating the effect of two successive reflections on a state vector. The horizontal axis is labeled $|b\rangle$ Non-solutions and the vertical axis is labeled $|a\rangle$ solution. A blue circle represents the unit circle. A vector starts from the origin, passes through the point on the positive horizontal axis, and then reflects off the circle to end at a point on the positive vertical axis. The angle between the initial vector and the positive horizontal axis is labeled 2θ .

$$\sin \theta = \frac{1}{\sqrt{N}}$$

PHYC90045 Introduction to Quantum Computing

THE UNIVERSITY OF
 MELBOURNE

Many Grover iterations

Product of two reflections is a rotation.

Product of two reflections is a rotation.

$|a\rangle$ solution

$|b\rangle$ Non-solutions

$\sin \theta = \frac{1}{\sqrt{N}}$

PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

How many iterations required?

$$\sin \theta = \frac{1}{\sqrt{N}}$$

For small angles,

$$\theta \approx \frac{1}{\sqrt{N}}$$

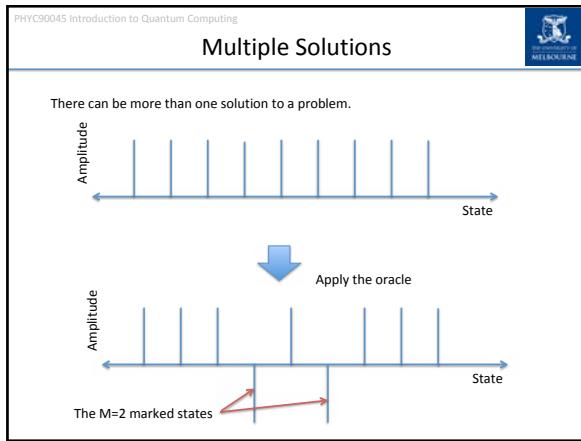
After n iterations, we rotate to have only marked solutions:

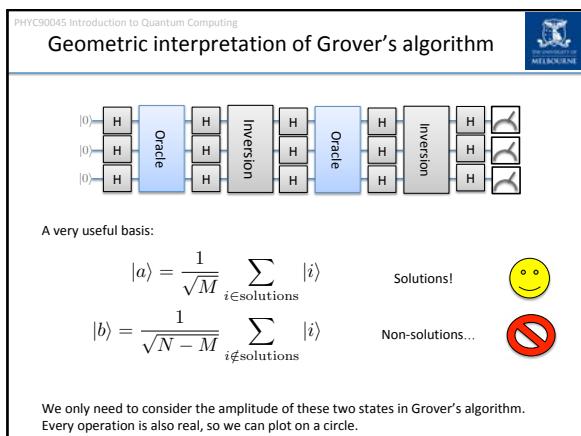
$$(2n + 1)\theta = \frac{\pi}{2}$$

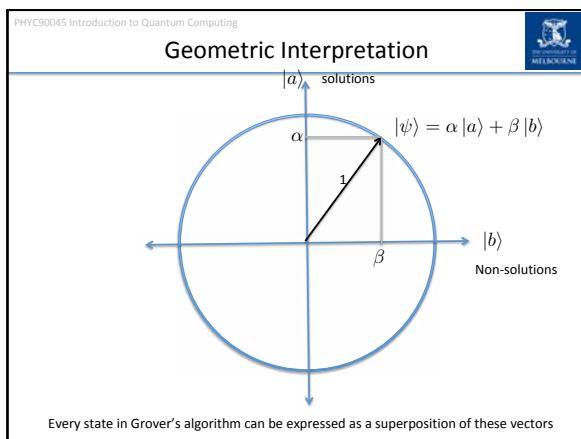
$$n \approx \frac{\pi}{4} \sqrt{N}$$

The number of steps, n , required scales as $O(VN)$, and not with N as it would classically.

This is a “polynomial” rather than an “exponential” speedup.





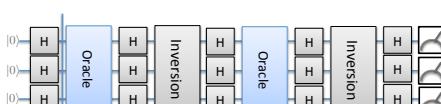


PHYC90045 Introduction to Quantum Computing



 The University of
 MELBOURNE

Equal superposition



A quantum circuit diagram showing a sequence of operations on three qubits. The circuit starts with three initial states: $|0\rangle$, $|0\rangle$, and $|0\rangle$. It consists of the following sequence of gates:

- First two qubits pass through a Hadamard (H) gate.
- All three qubits pass through an **Oracle** block.
- The first qubit passes through an **Inversion** block.
- The second qubit passes through an **Inversion** block.
- The third qubit passes through an **Inversion** block.
- All three qubits pass through another **Oracle** block.
- The first two qubits pass through another **Inversion** block.
- The final state of each qubit is measured, resulting in three meter symbols.

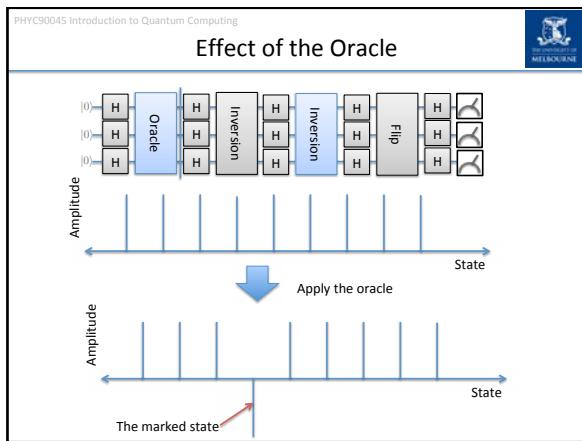
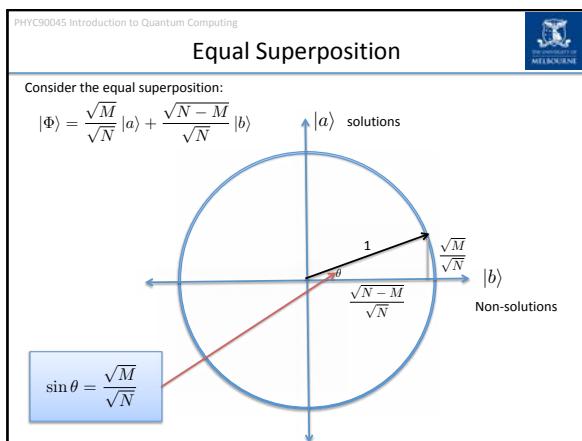
Equal superposition state:

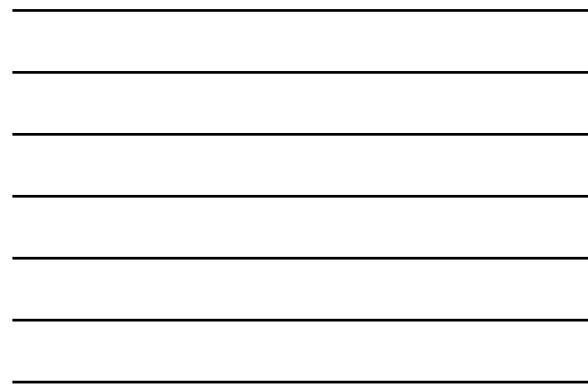
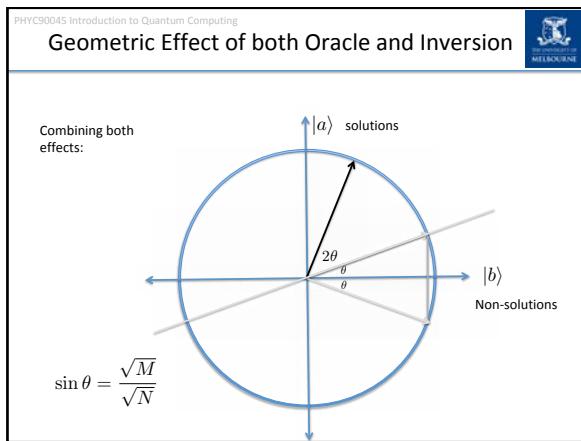
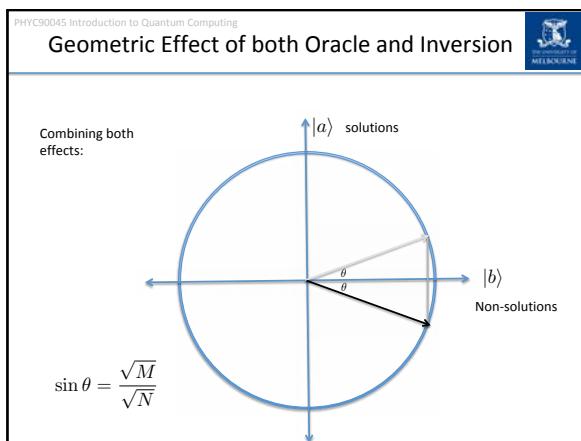
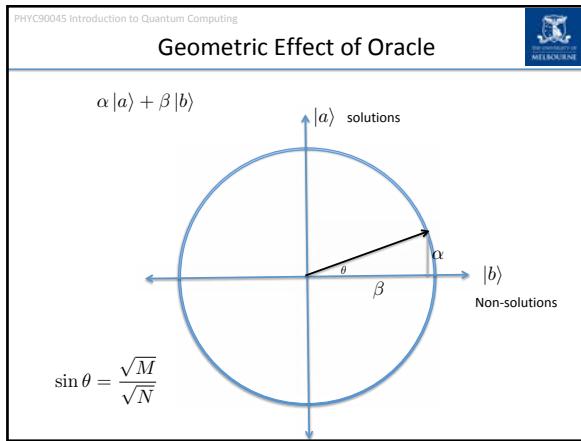
$$|\Phi\rangle = \frac{1}{\sqrt{N}} \sum_i |i\rangle$$

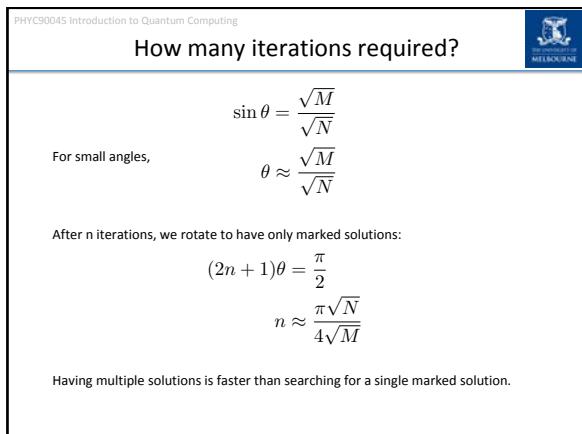
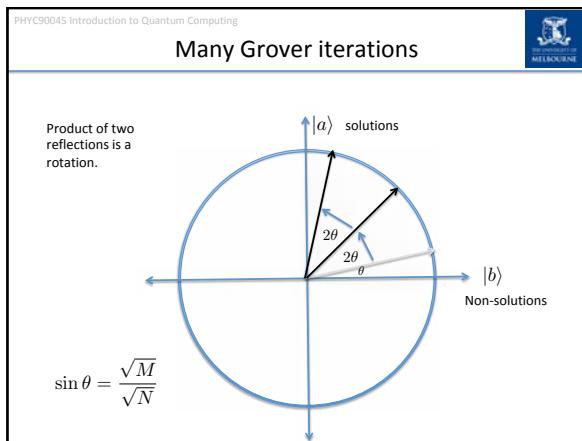
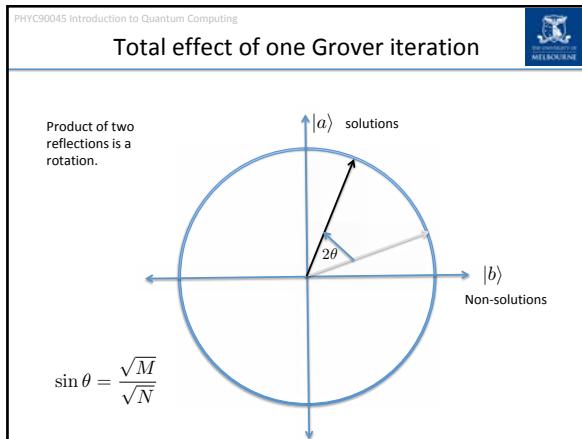
$$|\Phi\rangle = \frac{\sqrt{M}}{\sqrt{N}} |a\rangle + \frac{\sqrt{N-M}}{\sqrt{N}} |b\rangle$$

$$|a\rangle = \frac{1}{\sqrt{M}} \sum_{i \in \text{solutions}} |i\rangle$$

$$|b\rangle = \frac{1}{\sqrt{N-M}} \sum_{i \notin \text{solutions}} |i\rangle$$







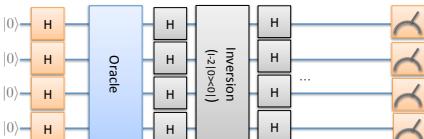
PHYC90045 Introduction to Quantum Computing

Grover's Algorithm



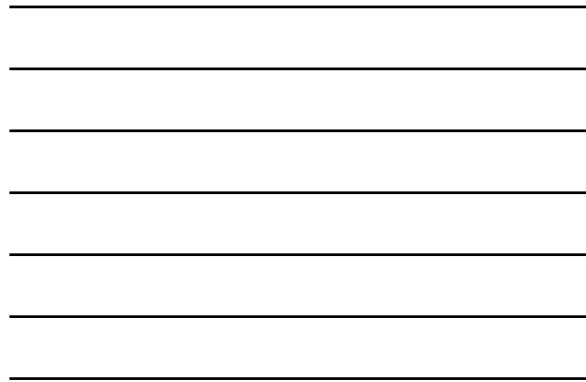
- Unordered search, find one marked item among many
- Classically, this requires $N/2$ uses of the oracle
- Quantum mechanically, requires only $O(\sqrt{N})$.


Lov Grover



The diagram shows a quantum circuit with four horizontal lines representing qubits. Each qubit starts in the $|0\rangle$ state. The circuit consists of the following sequence of operations from left to right:

- Four Hadamard gates (H) act on each qubit.
- A blue rectangular box labeled "Oracle".
- Four Hadamard gates (H) act on each qubit.
- A grey rectangular box labeled "Inversion" with the text " $\{z_j | j < n\}$ " inside.
- Four Hadamard gates (H) act on each qubit.
- A vertical ellipsis "..." indicates the circuit continues for more than four qubits.
- Four final Hadamard gates (H) act on each qubit.
- Each qubit ends in a curly brace symbol, indicating they are all in the same superposition state.



PHYC90045 Introduction to Quantum Computing

This Week

Lecture 7
Introduction to Grover's algorithm for amplitude amplification, geometric interpretation

Lecture 8
Amplitude Amplification, Succeeding with Certainty, Quantum Counting

Lab
Grover's algorithm

PHYC90045 Introduction to Quantum Computing

Amplitude Amplification

PHYC90045 Introduction to Quantum Computing
Lecture 8

PHYC90045 Introduction to Quantum Computing

Amplitude Amplification

- This lecture: Amplitude amplification
 - Amplitude Amplification
 - Succeeding with certainty
 - Quantum Counting

References:

Rieffel, Chapter 9.1-9.2
Kaye, Chapter 8.1-8.2
Nielsen and Chuang, Chapter 6.1-6.2

PHYC90045 Introduction to Quantum Computing

Grover's Algorithm (1996)

• Unordered search, find one marked item among many
 • Classically, this requires $N/2$ queries to the oracle
 • Quantum mechanically, requires only $O(\sqrt{N})$ queries.
 Simple problem = search for one integer marked by the oracle.

High level structure:

Lov Grover

PHYC90045 Introduction to Quantum Computing

Some notation

$S_G = I - 2|m\rangle\langle m|$

$S_0 = I - 2|0\rangle\langle 0|$

PHYC90045 Introduction to Quantum Computing

Some notation

Q

n is the number of input qubits.
N is the total dimension ($N=2^n$).
M is the number of solutions.

PHYC90045 Introduction to Quantum Computing

Oracles for NP-problems

The phone book isn't a great example: Adding in all the names would take $O(N)$ time.

In general though, many problems (specifically those in the class NP) can have easily **checkable** solutions even if it is hard to solve the problem originally. Examples:

- Factoring
- Travelling Salesman with route less than distance d
- Hamiltonian cycle

Straightforward application of Grover's algorithm provides a **polynomial** improvement over random guessing... and potentially a better (but still polynomial speedup) known as **amplitude amplification**.



Part of Norfolk Island's telephone book, with people listed by nickname (Photo: Wikicommons)

PHYC90045 Introduction to Quantum Computing

Oracle for a hash function

A hash function whose output is hard to predict based on the input.

The oracle recognises the 'correct' solution, but does not know in advance which input leads to the correct solution.

"Uncompute" the hash function – ensures the input register remains unchanged.

PHYC90045 Introduction to Quantum Computing

Amplitude amplification

What happens if we replace the Hadamard gates with some other U ? Perhaps, for example, we can create a U which gives the correct outcome with probability greater than $1/N$. Can we get any advantage?

Inversion about the mean Apply general U instead of H

PHYC90045 Introduction to Quantum Computing

New inversion step

Apply general U instead of H

$$|\phi\rangle = U|0\rangle$$

Then we can break this up as:

$$|\phi\rangle = g_0|\phi_g\rangle + b_0|\phi_b\rangle$$

Good: In the subspace spanned by all solutions

Bad: Not in the subspace spanned by all solutions

PHYC90045 Introduction to Quantum Computing

Maths of the Geometric Interpretation

where $|\phi\rangle = U|0\rangle$
 $|\phi\rangle = g_0|\phi_g\rangle + b_0|\phi_b\rangle$

$$\begin{aligned} Q|\phi\rangle_g &= -US_0U^\dagger S_G|\phi_g\rangle \\ &= US_0U^\dagger|\phi_g\rangle \\ &= |\phi_g\rangle - 2g_0^*U|0\rangle \\ &= |\phi_g\rangle - 2g_0^*g_0|\phi_g\rangle - 2g_0^*b_0|\phi_b\rangle \\ &= (1-2t)|\phi_g\rangle - 2\sqrt{t(1-t)}|\phi_b\rangle \end{aligned}$$

$t = |g_0|^2$

PHYC90045 Introduction to Quantum Computing

Maths of Amplitude Amplification

Similarly, $Q|\phi_b\rangle = (1-2t)|\phi_b\rangle + 2\sqrt{t(1-t)}|\phi_g\rangle$

And from previous slide: $Q|\phi_g\rangle = (1-2t)|\phi_g\rangle - 2\sqrt{t(1-t)}|\phi_b\rangle$

Q recursive step:

$$Q = \begin{bmatrix} (1-2t) & -2\sqrt{t(1-t)} \\ 2\sqrt{t(1-t)} & (1-2t) \end{bmatrix}$$

Compare to a rotation matrix:

$$R(2\theta) = \begin{bmatrix} \cos 2\theta & -\sin 2\theta \\ \sin 2\theta & \cos 2\theta \end{bmatrix}$$

$\sin \theta = \sqrt{t} = g_0$

PHYC90045 Introduction to Quantum Computing

Grover vs Amplitude Amplification

Grover: A sequence of operations: three Hadamard gates (H) on the first three wires, followed by an 'Inversion' gate, and then three more Hadamard gates (H) on the same three wires.

Amplitude Amplification: A sequence of operations: an 'U' gate, an 'Inversion' gate, and an 'U†' gate.

Angle of rotation:

$$\sin \theta = \frac{\sqrt{M}}{\sqrt{N}}$$

If you can construct a U with a higher probability of success than random guessing $1/N$, then amplitude amplification can help.

PHYC90045 Introduction to Quantum Computing

How to achieve 100% Success

The optimal, 100% probability of measuring marked can be missed.

Can we modify the algorithm to obtain 100% probability of success?

$\sin \theta = \frac{\sqrt{M}}{\sqrt{N}}$

A circular diagram representing a 2D vector space. The vertical axis is labeled $|a\rangle$ solutions and the horizontal axis is labeled $|b\rangle$ Non-solutions. Several blue arrows originate from the center (0,0) and point towards the upper-left quadrant ($|a\rangle$ solutions). One arrow points directly along the $|a\rangle$ axis.

PHYC90045 Introduction to Quantum Computing

Grover with 100% success

This step gives 100% probability of finding the marked state

Idea: reduce the size of each step (intentionally) so that a whole number of steps is required.

Using amplitude amplification

A circular diagram representing a 2D vector space. The vertical axis is labeled $|a\rangle$ solutions and the horizontal axis is labeled $|b\rangle$ Non-solutions. Several blue arrows originate from the center (0,0) and point towards the upper-left quadrant ($|a\rangle$ solutions). One arrow points directly along the $|a\rangle$ axis. Red arrows indicate the path of the algorithm, showing a series of steps that have been intentionally reduced in size to result in a whole number of steps.

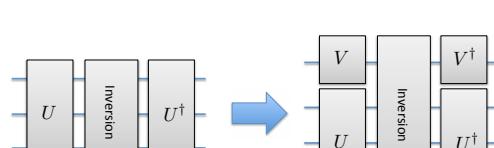
PHYC90045 Introduction to Quantum Computing



Reducing the angle

We want to reduce the angle of rotation in Grover's algorithm/amplitude amplification so that we require a whole number of steps to achieve 100% probability of success.

Trick: Introduce a new qubit.



The diagram illustrates a quantum circuit transformation. On the left, a sequence of three boxes labeled U , "Inversion", and U^\dagger is shown, representing a standard Grover iteration. A blue arrow points to the right, indicating a transformation. On the right, the circuit is modified: the first two boxes (U and "Inversion") remain, but the third box (U^\dagger) is split into two separate boxes: V above and U below. The "Inversion" box remains between them. This modification effectively halves the rotation angle of the original U^\dagger operation, as the total effect is equivalent to applying V , then U , then "Inversion".

PHYC90045 Introduction to Quantum Computing

How much reduction?

Previously:

$$U |0\rangle = g_0 |\phi_g\rangle + b_0 |\phi_b\rangle$$

With new qubit:

$$V \otimes U |0\rangle = V |0\rangle \otimes (g_0 |\phi_g\rangle + b_0 |\phi_b\rangle)$$

If we arrange so that:

$$V |0\rangle = \sqrt{1 - \left(\frac{g'_0}{g_0}\right)^2} |0\rangle + \frac{g'_0}{g_0} |1\rangle$$

e.g. Y-rotation by an angle: $\cos \frac{\alpha}{2} = \frac{g'_0}{g}$

PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

New rotation angle

$$V \otimes U |0\rangle = V|0\rangle \otimes (g_0 |\phi_g\rangle + b_0 |\phi_b\rangle)$$

$$V|0\rangle = \sqrt{1 - \left(\frac{g'_0}{g_0}\right)^2}|0\rangle + \frac{g'_0}{g_0}|1\rangle$$

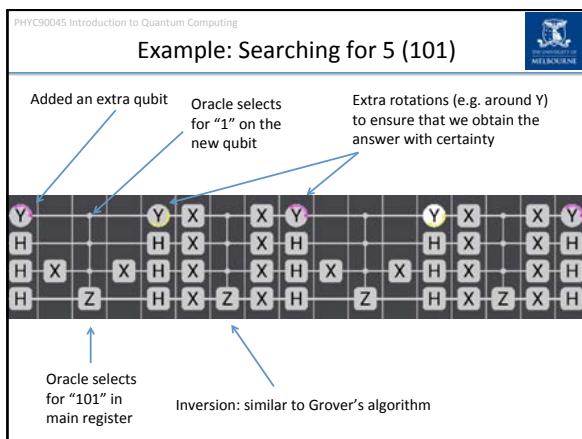
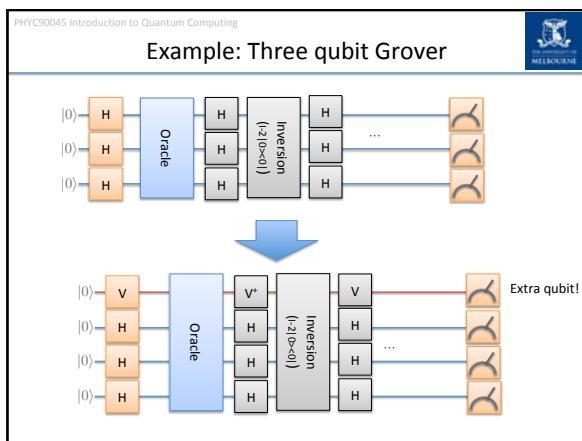
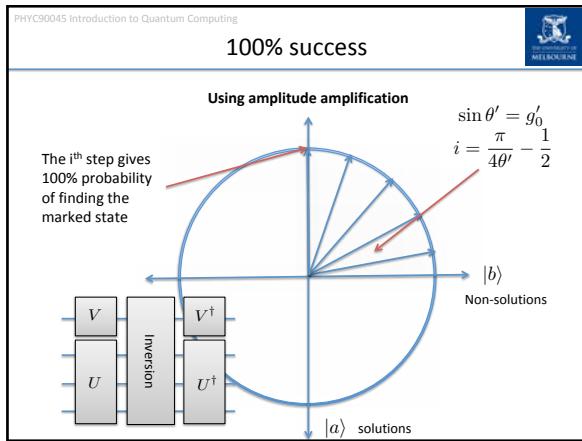
Gives:

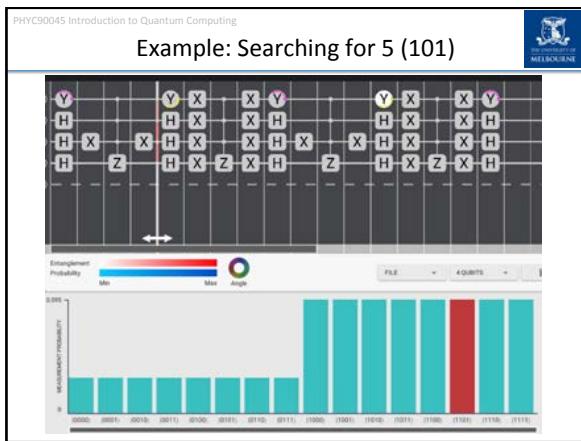
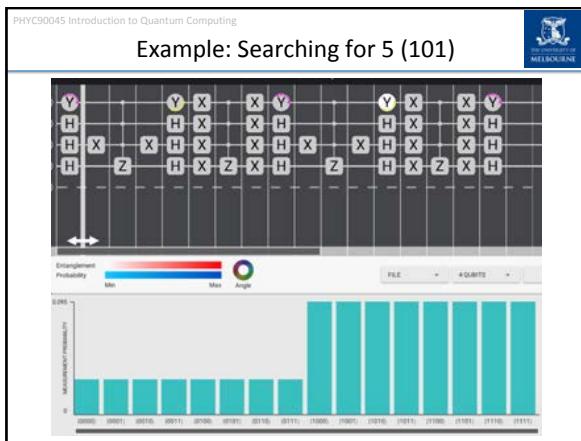
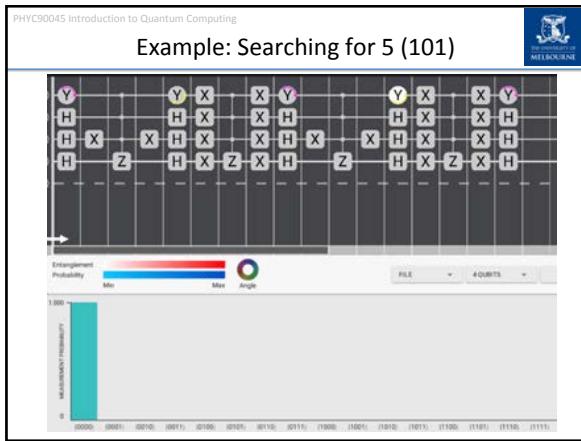
$$V \otimes U |0\rangle = g'_0 |1\rangle |\phi_g\rangle + \dots$$

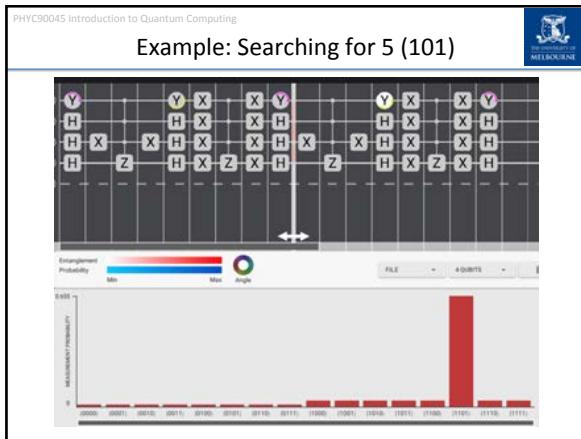
We can choose the initial amplitude to be anything value less than the original

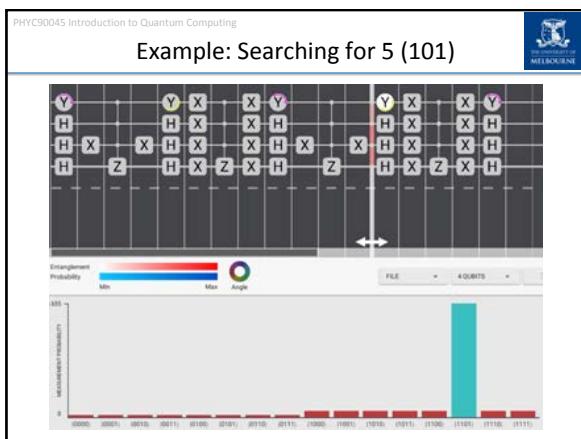
Our new "good" states, but now have a preceding "1" on the extra qubit we added

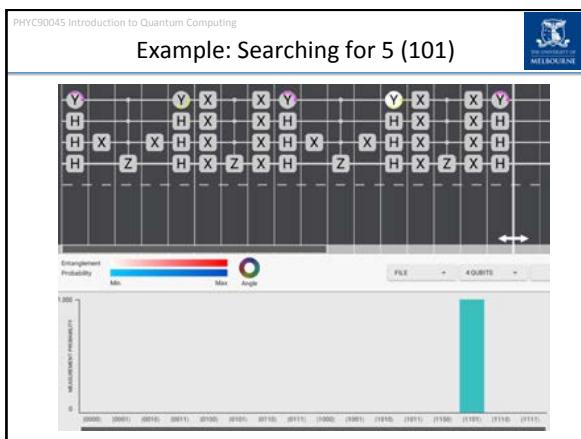
Choose g'_0 's.t. $i = \frac{\pi}{4\theta'} - \frac{1}{2}$ is a whole number











PHYC90045 Introduction to Quantum Computing

Amplitude Amplification

Given a black box (oracle), U_f , which computes the function $f: \{0,1\}^n \rightarrow \{0,1\}$
Find an x s.t. $f(x) = 1$

- Unordered search, generalisation of Grover's algorithm
- Classically, this requires $N/2$ uses of the oracle
- Quantum mechanically, requires only $O(\sqrt{N})$.

PHYC90045 Introduction to Quantum Computing

Amplitude Amplification is optimal

Proof in your textbooks.

Grover's algorithm is optimal in terms of the number of applications of the oracle.
For many oracle problems the required number of uses of the oracle scales like:

$$O(\sqrt{N})$$

This means that for a broad range of problems the best speedup we can achieve using a quantum computer is **not** exponential, but polynomial (which can be quite significant).

For problems with identifiable structure, we might hope for more speedup.
More on this next week.

PHYC90045 Introduction to Quantum Computing

Quantum Counting

Will show you this algorithm now, but will leave some of the details until after next week's lectures/lab.

Given a black box (oracle), U_f , which computes the function $f: \{0,1\}^n \rightarrow \{0,1\}$
How many x s.t. $f(x) = 1$?

PHYC90045 Introduction to Quantum Computing

Equivalent question

What angle rotation does Q make?

$$\sin \theta = \frac{\sqrt{M}}{\sqrt{N}}$$

PHYC90045 Introduction to Quantum Computing

Plotting amplitude as function of step number

After k steps: $\theta_k = (2k + 1)\theta$ $\sin \theta = \frac{\sqrt{M}}{\sqrt{N}}$

Number of solutions is reflected in the period/frequency

Incorrect solutions would follow cosine, rather than sin

Amplitude at step 'k' is: $g_k = \sin(2k + 1)\theta$

PHYC90045 Introduction to Quantum Computing

Finding the period of a periodic function

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle \otimes Q^x |0\rangle$$

Control register, x x steps of Grover's algorithm Quantum Fourier Transform (next week)

PHYC90045 Introduction to Quantum Computing

Finding the period of a periodic function

$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle \otimes Q^x |0\rangle$

$$= \frac{1}{\sqrt{N}} \sum_x |x\rangle \otimes (\sin(2x+1)\theta |\psi_g\rangle + \cos(2x+1)\theta |\psi_b\rangle)$$

If we measure the second register

PHYC90045 Introduction to Quantum Computing

After measurement of the second register

$|\psi\rangle = \sum_x \sin(2x+1)\theta |x\rangle \otimes |\psi_g\rangle$ (not normalized)

Next step: Use Quantum Fourier Transformation to find the period

PHYC90045 Introduction to Quantum Computing

After the Fourier transformation

After Fourier transforming a periodic function, we get a good approximation to theta. If we measure value "j":

$$\theta = \frac{j\pi}{N'} \quad \sin \theta = \frac{\sqrt{M}}{\sqrt{N}}$$

Which we can solve to obtain the number of solutions, M

PHYC90045 Introduction to Quantum Computing

Phase Estimation and HSP Problems

A quantum circuit diagram showing two horizontal lines representing qubits. The top qubit starts with a state $|0\rangle$, followed by a Hadamard gate (H), then a control point, then a Quantum Fourier Transform gate (QFT), and finally a meter. The bottom qubit starts with a question mark $?$, followed by a control point, and then a meter. The two meters are connected by a dashed line.

The Hadamard and Fourier transform part is known as **phase estimation**, and extremely useful for period functions (and eigenvalues which are periodic).

As we will see in the next lecture, this pattern is often repeated.

PHYC90045 Introduction to Quantum Computing

This Week

Lecture 7
Introduction to Grover's algorithm for amplitude amplification, geometric interpretation

Lecture 8
Amplitude Amplification, Succeeding with Certainty, Quantum Counting

Lab
Grover's algorithm

PHYC90045 Introduction to Quantum Computing

This Week

Lecture 9
Fourier Transformations, Regular Fourier Transform, Fourier Transform as a matrix, Quantum Fourier Transform, QFT examples, Inverse QFT

Lecture 10
Shor's Quantum Factoring algorithm, Shor's algorithm for factoring and discrete logarithm, HSP Problem

Lab-5
QFT and Shor's algorithm

PHYC90045 Introduction to Quantum Computing

Lecture 9 overview

• Fourier Transformations
 – Regular Fourier Transform
 – Fourier Transform as a matrix
 – Quantum Fourier Transform (QFT)
 – QFT examples
 – Inverse QFT

Reiffel, Chapter 8
 Kaye, Chapter 7
 Nielsen and Chuang, Chapter 5

PHYC90045 Introduction to Quantum Computing

Last lecture: Quantum Counting

Dimension: N'

Dimension: N

$$|\psi\rangle = \sum_x \sin(2x + 1)\theta |x\rangle \otimes |\psi_g\rangle$$

After Fourier transforming a periodic function, we get a good approximation to frequency $\theta \rightarrow \frac{\sqrt{M}}{\sqrt{N}}$

Number of solutions

PHYC90045 Introduction to Quantum Computing

Fourier Transform in Quantum Computing

In QC the equivalent of the Fourier Transform – quantum Fourier Transform (QFT) – is important in a number of algorithms, most notably Shor’s Factoring algorithm...

Hence, before we can cover Shor’s algorithm we need to understand the QFT and how to implement it in a QC (and on the QUI)...

The screenshot shows a quantum circuit editor and a bitvector register viewer.

Quantum Circuit:

- Registers: $|0\rangle$, $|0\rangle$, $|0\rangle$, $|+\rangle$.
- Gates:
 - On $|0\rangle$: H , Z , Z .
 - On $|0\rangle$: X .
 - On $|0\rangle$: X .
 - On $|+\rangle$: H .
- Control: A control gate (represented by a box with a dot) controls the H gate on $|+\rangle$.
- Measurement: A measurement meter is positioned after the H gate on $|+\rangle$.

Bitvector Register:

A 4-bit register labeled "bitvector" with values: 1000, 0100, 0010, 0001.

The diagram illustrates the Fourier Transform process. On the left, the "Time domain" plot shows a periodic square wave signal with an amplitude ranging from -1.0 to 1.0 over time. On the right, the "Frequency domain" plot shows two sharp peaks at approximately ±0.4 units of frequency, with an amplitude ranging from 0 to 80000. A large blue arrow labeled "Fourier Transform" points from the Time domain graph to the Frequency domain graph.

PHYC90045 Introduction to Quantum Computing

Discrete Fourier Transform

Maps a vector: $(x_0, x_1, \dots, x_{N-1}) \in \mathbb{C}^N$ to a vector: $(y_0, y_1, \dots, y_{N-1}) \in \mathbb{C}^N$

According to: $y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi i j k / N}$

N.B.
 $i = \text{sqrt}(-1)$
 j and k are integers

e.g. Frequency Domain

Frequency	y_k
-1.0	0
-0.5	0
0.0	0
0.5	120000
1.0	0

e.g. Time Domain

Time	x_j
-1.0	1.0
-0.5	0.5
0.0	0.0
0.5	-0.5
1.0	-1.0
1.5	-0.5
2.0	0.0

PHYS90045 Introduction to Quantum Computing

Example: Fourier transform of periodic function

Imagine that we had a periodic function:

$$x_j = \exp\left(-2\pi i \frac{u_j}{N}\right)$$

Complex number, $i^2=-1$

The frequency, u

$0 \leq j < N$

$u=1$

j	Real (blue)	Imaginary (orange)
0	1.00	-0.80
20	0.50	-0.50
40	-0.50	0.50
60	-0.50	0.50
80	0.50	0.50
100	0.50	-0.50
110	1.00	-0.80

$u=2$

j	Real (blue)	Imaginary (orange)
0	1.00	-0.50
20	0.50	-0.80
40	-0.50	0.50
60	-0.50	0.50
80	0.50	0.50
100	0.50	-0.80
110	1.00	-0.50

PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

Example: Periodic function

$$\begin{aligned}
 y_k &= \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \exp \left(2\pi i \frac{jk}{N} \right) \\
 &= \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \exp \left(-2\pi i \frac{uj}{N} \right) \exp \left(2\pi i \frac{jk}{N} \right) \\
 &= \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \exp \left(-2\pi i \frac{j(k-u)}{N} \right)
 \end{aligned}$$

If $k=u$ then

$y_u = \sqrt{N}$

PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

Example: Periodic function

For any other value of k,

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \exp \left(-2\pi i \frac{j(k-u)}{N} \right)$$

Recall, for a geometric series, $1 + r + r^2 + \dots r^{N-1} = \frac{1 - r^N}{1 - r}$

Where for us, $r = \exp \left(-2\pi i \frac{k-u}{N} \right)$

And therefore since $k \neq u$ (but difference is an integer): $r^N = 1$

Except for $k=u$,

$$y_k = 0$$

i.e. just one non-zero amplitude $y_u \leftrightarrow$ frequency, u

PHYC90045 Introduction to Quantum Computing

Fourier Transform as a Matrix

The University of Melbourne

We define the Fourier transformation matrix as follows:

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi i j k / N}$$

$$y_k = \sum_j F_{kj} x_j \quad \text{where} \quad F_{kj} = \frac{1}{\sqrt{N}} e^{2\pi i j k / N}$$

For example:

$$\text{N=2: } F = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \text{N=4: } F = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}$$

We will see that the quantum Fourier transform for one qubit is a Hadamard gate!

PHYC90045 Introduction to Quantum Computing

Quantum Fourier Transform (QFT)

The Fourier transform, written in this matrix form is unitary. It can make a valid quantum operation:

$$|\psi\rangle = \sum_{j=0}^{N-1} x_j |j\rangle \xrightarrow{\text{QFT}} |\psi'\rangle = \sum_{j=0}^{N-1} y_j |j\rangle \quad \text{with} \quad y_k = \sum_{j=0}^{N-1} F_{kj} x_j$$

$$F_{kj} = \frac{1}{\sqrt{N}} e^{2\pi i j k / N}$$

On an individual basis state $|a\rangle$ (i.e. $j = a$ only non-zero x_j) we have:

$$|a\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} y_k |k\rangle, \quad y_k = \sum_{j=0}^{N-1} F_{kj} x_j = F_{ka} = \frac{1}{\sqrt{N}} e^{2\pi i k a / N}$$

i.e. $\text{QFT } |a\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi i}{N} k a} |k\rangle$ (more familiar form relating variables a and k by Fourier transform -> puts a into the phase)

Question: How can we systematically make this operation using quantum gates?

PHYC90045 Introduction to Quantum Computing

Product Form of QFT

The Fourier transform can be expressed in a product notation:

$$|j_1, \dots, j_n\rangle \rightarrow \frac{|0\rangle + e^{2\pi i 0 j_n} |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i 0 j_{n-1} j_n} |1\rangle}{\sqrt{2}} \otimes \dots \otimes \frac{|0\rangle + e^{2\pi i 0 j_1 j_2 \dots j_{n-1} j_n} |1\rangle}{\sqrt{2}}$$

(this is not obvious – see appendix at end)

Where the notation $0.j_1 j_2 \dots j_n = \frac{j_1}{2} + \frac{j_2}{2^2} + \dots + \frac{j_{n-1}}{2^{n-1}} + \frac{j_n}{2^n}$

is shorthand for writing a fraction in binary notation. That is,

$$0.1 = \frac{1}{2}$$

$$0.11 = \frac{1}{2} + \frac{1}{2^2} = \frac{3}{4}$$

$$0.101 = \frac{1}{2} + \frac{1}{2^3} = \frac{5}{8} \quad \text{etc}$$

PHYC90045 Introduction to Quantum Computing

Product Form: One Qubit



$$|j_1, \dots, j_n\rangle \rightarrow \frac{|0\rangle + e^{2\pi i 0.j_n}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle}{\sqrt{2}} \otimes \dots \otimes \frac{|0\rangle + e^{2\pi i 0.j_1j_2\dots j_{n-1}j_n}|1\rangle}{\sqrt{2}}$$

For one qubit (i.e. n=1, N=2): $|j_1\rangle \rightarrow \frac{|0\rangle + e^{2\pi i 0.j_1}|1\rangle}{\sqrt{2}}$ $j_1 = 0, 1$

$$|0\rangle \rightarrow \frac{|0\rangle + e^{2\pi i 0.0}|1\rangle}{\sqrt{2}} = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

Beware binary fraction!
 $0.\underline{1} = 1/2$ etc

$$|1\rangle \rightarrow \frac{|0\rangle + e^{2\pi i \underline{0.1}}|1\rangle}{\sqrt{2}} = \frac{|0\rangle + e^{\pi i}|1\rangle}{\sqrt{2}} = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

As before, we get:
 $F = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

(i.e. a Hadamard)

PHYC90045 Introduction to Quantum Computing

Product Form: Two Qubits



$$|j_1, \dots, j_n\rangle \rightarrow \frac{|0\rangle + e^{2\pi i 0.j_n}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle}{\sqrt{2}} \otimes \dots \otimes \frac{|0\rangle + e^{2\pi i 0.j_1j_2\dots j_{n-1}j_n}|1\rangle}{\sqrt{2}}$$

$|j_1j_2\rangle \rightarrow \frac{|0\rangle + e^{2\pi i 0.j_2}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i 0.j_1j_2}|1\rangle}{\sqrt{2}}$

$$|00\rangle \rightarrow \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2}$$

$$|01\rangle \rightarrow \frac{|0\rangle + e^{i2\pi 0.1}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{i2\pi 0.01}|1\rangle}{\sqrt{2}} = \frac{|00\rangle + i|01\rangle - |10\rangle - i|11\rangle}{2}$$

$$|10\rangle \rightarrow \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{i2\pi 0.1}|1\rangle}{\sqrt{2}} = \frac{|00\rangle - |01\rangle + |10\rangle - |11\rangle}{2}$$

$$|11\rangle \rightarrow \frac{|0\rangle + e^{i2\pi 0.1}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{i2\pi 0.11}|1\rangle}{\sqrt{2}} = \frac{|00\rangle - i|01\rangle - |10\rangle + i|11\rangle}{2}$$

PHYC90045 Introduction to Quantum Computing

Product Notation: Two Qubits



$$|j_1, \dots, j_n\rangle \rightarrow \frac{|0\rangle + e^{2\pi i 0.j_n}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle}{\sqrt{2}} \otimes \dots \otimes \frac{|0\rangle + e^{2\pi i 0.j_1j_2\dots j_{n-1}j_n}|1\rangle}{\sqrt{2}}$$

$|00\rangle \rightarrow \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2}$

$$|01\rangle \rightarrow \frac{|0\rangle + e^{i2\pi 0.1}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{i2\pi 0.01}|1\rangle}{\sqrt{2}} = \frac{|00\rangle + i|01\rangle - |10\rangle - i|11\rangle}{2}$$

$$|10\rangle \rightarrow \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{i2\pi 0.1}|1\rangle}{\sqrt{2}} = \frac{|00\rangle - |01\rangle + |10\rangle - |11\rangle}{2}$$

$$|11\rangle \rightarrow \frac{|0\rangle + e^{i2\pi 0.1}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{i2\pi 0.11}|1\rangle}{\sqrt{2}} = \frac{|00\rangle - i|01\rangle - |10\rangle + i|11\rangle}{2}$$

As before: $F = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}$

PHYC90045 Introduction to Quantum Computing

Pick it apart...

Look a little bit more closely:

$$|j_1, \dots, j_n\rangle \rightarrow \frac{|0\rangle + e^{2\pi i 0.j_n}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle}{\sqrt{2}} \otimes \dots \otimes \frac{|0\rangle + e^{2\pi i 0.j_1j_2\dots j_{n-1}j_n}|1\rangle}{\sqrt{2}}$$

Very similar to equal superposition.
All qubits have an equal amplitude,
just not an equal phase.

Each qubit acquires a phase
dependent on (the original
state of) all prior qubits.

$$\begin{aligned} |0\rangle + e^{2\pi i 0.j_1j_2\dots j_n}|1\rangle &= \frac{|0\rangle + e^{2\pi i [\frac{j_1}{2} + \frac{j_2}{2^2} + \dots + \frac{j_n}{2^n}]}}{\sqrt{2}}|1\rangle \\ &= \frac{|0\rangle + e^{2\pi i \frac{j_1}{2}} e^{2\pi i \frac{j_2}{2^2}} \dots e^{2\pi i \frac{j_n}{2^n}}}{\sqrt{2}}|1\rangle \end{aligned}$$

Product of phases applied, i.e. of the form $\begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i j_k/2^k} \end{pmatrix}$ e.g. rotation by $\theta = 2\pi/2^k$ controlled by j_k

PHYC90045 Introduction to Quantum Computing

Circuit for QFT

Look carefully at the product form:

$$|j_1, \dots, j_n\rangle \rightarrow \frac{|0\rangle + e^{2\pi i 0.j_n}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle}{\sqrt{2}} \otimes \dots \otimes \frac{|0\rangle + e^{2\pi i 0.j_1j_2\dots j_{n-1}j_n}|1\rangle}{\sqrt{2}}$$

Suggests an efficient circuit implementation – e.g. for n=4:

Controlled rotations with: $R_{z_k} = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i j_k/2^k} \end{pmatrix}$

Notice how the required QFT form is recovered by re-labelling qubits

PHYC90045 Introduction to Quantum Computing

One qubit QFT circuit

Look at the pattern of the circuit:

For one qubit we have just an H-gate:

$$|j_1\rangle \rightarrow \frac{|0\rangle + e^{2\pi i 0.j_1}|1\rangle}{\sqrt{2}}$$

$|0\rangle \rightarrow \frac{|0\rangle + e^{2\pi i 0.0}|1\rangle}{\sqrt{2}} = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$

$|1\rangle \rightarrow \frac{|0\rangle + e^{2\pi i 0.1}|1\rangle}{\sqrt{2}} = \frac{|0\rangle + e^{\pi i}|1\rangle}{\sqrt{2}} = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$

PHYC90045 Introduction to Quantum Computing

Two Qubit QFT circuit

$$R_{z_k} = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix} \quad R_{z_2} = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^2} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{pmatrix}$$

QUI gates:

$$\begin{aligned} R_Z(\theta_R) &= e^{i\theta_g} \left[I \cos \frac{\theta_R}{2} - iZ \sin \frac{\theta_R}{2} \right] = e^{i\theta_g} \left[\begin{pmatrix} \cos \frac{\theta_R}{2} & 0 \\ 0 & \cos \frac{\theta_R}{2} \end{pmatrix} - i \begin{pmatrix} \sin \frac{\theta_R}{2} & 0 \\ 0 & -\sin \frac{\theta_R}{2} \end{pmatrix} \right] \\ &= e^{i\theta_g} \begin{pmatrix} \cos \frac{\theta_R}{2} - i \sin \frac{\theta_R}{2} & 0 \\ 0 & \cos \frac{\theta_R}{2} + i \sin \frac{\theta_R}{2} \end{pmatrix} \\ &= e^{i\theta_g} \begin{pmatrix} e^{-i\theta_R/2} & 0 \\ 0 & e^{+i\theta_R/2} \end{pmatrix} \\ &= e^{i\theta_g} e^{-i\theta_R/2} \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta_R} \end{pmatrix} \end{aligned}$$

$$R_{z_2} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{pmatrix} \equiv R_Z \left(\frac{\pi}{2} \right) \quad \text{with } \theta_g = \frac{\pi}{4} \text{ Global phase cancels prefactor}$$

PHYC90045 Introduction to Quantum Computing

Two Qubit QFT circuit - walkthrough

$$|j_1, \dots, j_n\rangle \rightarrow \frac{|0\rangle + e^{2\pi i 0.j_n}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle}{\sqrt{2}} \otimes \dots \otimes \frac{|0\rangle + e^{2\pi i 0.j_1j_2\dots j_{n-1}j_n}|1\rangle}{\sqrt{2}}$$

$$R_{z_2} = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^2} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{pmatrix}$$

Check it gives the product form:

$$|\psi_a\rangle = \frac{1}{\sqrt{2}} (|0\rangle + e^{i\pi j_1}|1\rangle) \otimes |j_2\rangle \quad \text{Hadamard has negative sign on } |1\rangle \text{ if } j_1 = 1$$

$$|\psi_b\rangle = \frac{1}{\sqrt{2}} (|0\rangle + e^{i\pi j_1} e^{i(\pi/2)j_2}|1\rangle) \otimes |j_2\rangle \quad R_{z_2} \text{ applied only when } j_2 = 1$$

$$|\psi_c\rangle = \frac{1}{\sqrt{2}} (|0\rangle + e^{i\pi j_1} e^{i(\pi/2)j_2}|1\rangle) \otimes \frac{1}{\sqrt{2}} (|0\rangle + e^{i\pi j_2}|1\rangle) \quad \text{Hadamard on } |j_2\rangle$$

Binary fractions: $e^{i\pi j_1} e^{i(\pi/2)j_2} = e^{2\pi i(j_1/2 + j_2/4)} = e^{2\pi i 0.j_1j_2}$

$$|\psi_c\rangle = \frac{1}{\sqrt{2}} (|0\rangle + e^{2\pi i 0.j_1j_2}|1\rangle) \otimes \frac{1}{\sqrt{2}} (|0\rangle + e^{2\pi i 0.j_2}|1\rangle) \quad \text{i.e. circuit gives product form with } j_1 \text{ and } j_2 \text{ order reversed}$$

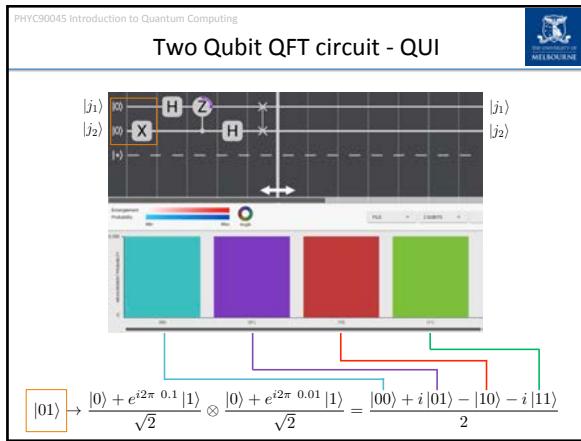
PHYC90045 Introduction to Quantum Computing

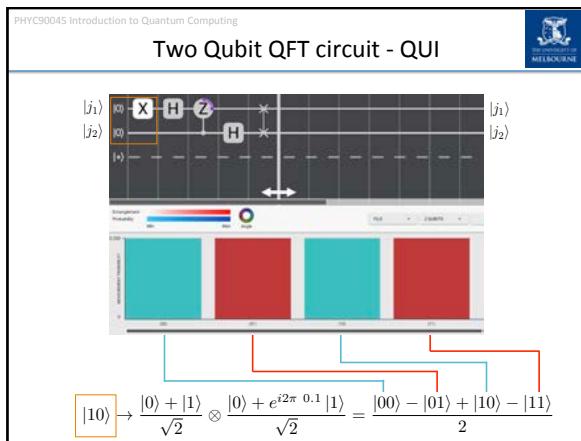
Two Qubit QFT circuit - QUI

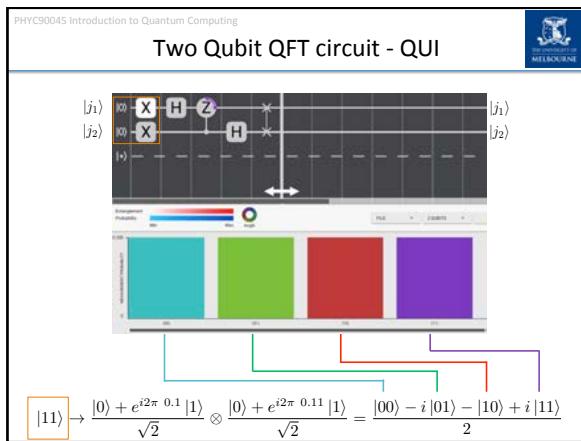
Note: inserted SWAP gate so ordering is same c/f ket expression

All phases zero

$$|00\rangle \rightarrow \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2}$$







PHYC90045 Introduction to Quantum Computing

Three Qubit QFT circuit

Rotation gates in the QFT:

$$R_{z2} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{pmatrix} \equiv R_Z \left(\frac{\pi}{2} \right) \quad \text{with } \theta_g = \frac{\pi}{4}$$

$$R_{z3} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \equiv R_Z \left(\frac{\pi}{4} \right) \quad \text{with } \theta_g = \frac{\pi}{8}$$

SWAP gate reverses order, so same as input

PHYC90045 Introduction to Quantum Computing

Three Qubit QFT - QUI

Example: $|011\rangle$

N.B. same as $\pi/4$ etc

$$|011\rangle \rightarrow \left(\frac{1}{\sqrt{2}}\right)^3 \left(|000\rangle + e^{3\pi i/4} |001\rangle + e^{3\pi i/2} |010\rangle + e^{9\pi i/4} |011\rangle + e^{i\pi} |100\rangle + e^{7\pi i/4} |101\rangle + e^{5\pi i/2} |110\rangle + e^{13\pi i/4} |111\rangle \right)$$

PHYC90045 Introduction to Quantum Computing

Step back for a moment

After all that, let's check on what we were trying to achieve:

On a single basis state $\text{QFT } |a\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi i}{N} ka} |k\rangle$

e.g. $|011\rangle \rightarrow \left(\frac{1}{\sqrt{2}}\right)^3 \left(|000\rangle + e^{3\pi i/4} |001\rangle + e^{3\pi i/2} |010\rangle + e^{9\pi i/4} |011\rangle + e^{i\pi} |100\rangle + e^{7\pi i/4} |101\rangle + e^{5\pi i/2} |110\rangle + e^{13\pi i/4} |111\rangle \right)$

i.e. $3=011$ $|3\rangle \rightarrow \left(\frac{1}{\sqrt{2}}\right)^3 \left(|0\rangle + e^{3\pi i/4} |1\rangle + e^{3\pi i/2} |2\rangle + e^{9\pi i/4} |3\rangle + e^{i\pi} |4\rangle + e^{7\pi i/4} |5\rangle + e^{5\pi i/2} |6\rangle + e^{13\pi i/4} |7\rangle \right)$

It obeys: $\text{QFT } |3\rangle = \frac{1}{\sqrt{8}} \sum_{k=0}^{N-1} e^{\frac{2\pi i}{8} 3k} |k\rangle$ (check it!)

PHYC90045 Introduction to Quantum Computing

Programing the Inverse QFT

As with any circuit: invert the QFT by inverting every gate and reversing the order:

$$R_{z_2} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{pmatrix} \equiv R_Z\left(\frac{\pi}{2}\right) \quad \text{with } \theta_g = \frac{\pi}{4}$$

$$R_{z_3} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \equiv R_Z\left(\frac{\pi}{4}\right) \quad \text{with } \theta_g = \frac{\pi}{8}$$

$$\begin{aligned} R_Z(\theta_R) &= e^{i\theta_R} \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta_R/2} - iZ^{\otimes n} e^{i\theta_R/2} \end{pmatrix} \\ &= e^{i\theta_R} e^{-i\theta_R/2} \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta_R} \end{pmatrix} \end{aligned}$$

$$R_z^\dagger(\theta_R) = e^{-i\theta_R} e^{+i\theta_R/2} \begin{pmatrix} 1 & 0 \\ 0 & e^{-i\theta_R} \end{pmatrix}$$

i.e. Reverse signs of θ_R and θ_g

e.g. $|011\rangle$

$$\begin{aligned} |011\rangle &\xrightarrow{\text{QFT}} \left(\frac{1}{\sqrt{2}}\right)^3 \left(|000\rangle + e^{3\pi i/4} |001\rangle + e^{3\pi i/2} |010\rangle + e^{9\pi i/4} |011\rangle \right. \\ &\quad \left. + e^{i\pi} |100\rangle + e^{7\pi i/4} |101\rangle + e^{5\pi i/2} |110\rangle + e^{13\pi i/4} |111\rangle \right) \xrightarrow{\text{QFT}^\dagger} |011\rangle \end{aligned}$$

PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

Adding using QFT

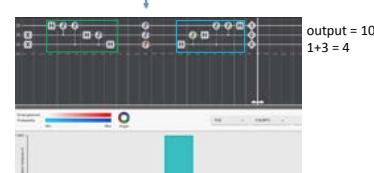
Now we can see how to add numbers using phase...

modify phase to add b = 001



$$a = 011 = 3$$

MSB top



output = 100
 $1+3 = 4$

Details next lecture...

PHYC90045 Introduction to Quantum Computing

This Week



The University of
MELBOURNE

PHYC90045 Introduction to Quantum Computing



Appendix: proof of the product form

In case you want to go through it at your leisure

$$\begin{aligned}
 |j\rangle &\rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k / N} |k\rangle \\
 &= \frac{1}{\sqrt{N}} \sum_{k_1=0}^1 \dots \sum_{k_n=0}^1 e^{2\pi i j \sum_l k_l 2^{-l}} |k_1 \dots k_n\rangle \\
 &= \frac{1}{\sqrt{N}} \sum_{k_1=0}^1 \dots \sum_{k_n=0}^1 \otimes_l e^{2\pi i j k_l 2^{-l}} |k_l\rangle \\
 &= \frac{1}{\sqrt{N}} \otimes_l \left[|0\rangle + e^{2\pi i j 2^{-l}} |1\rangle \right] \\
 &= \frac{|0\rangle + e^{2\pi i 0.j_n} |1\rangle}{\sqrt{2}} \otimes \dots \otimes \frac{|0\rangle + e^{2\pi i 0.j_1 j_2 \dots j_n} |1\rangle}{\sqrt{2}}
 \end{aligned}$$

PHYC90045 Introduction to Quantum Computing

This Week

Lecture 9
Fourier Transformations, Regular Fourier Transform, Fourier Transform as a matrix, Quantum Fourier Transform, QFI examples, Inverse QFT

Lecture 10
Shor's Quantum Factoring algorithm, Shor's algorithm for factoring and discrete logarithm, HSP Problem

Lab 5
QFT and Shor's algorithm

PHYC90045 Introduction to Quantum Computing

Quantum Factoring Algorithm

Physics 90045
Lecture 10

PHYC90045 Introduction to Quantum Computing

Quantum factoring algorithm

- Shor's Factoring algorithm
 - Shor's algorithm for factoring and discrete logarithm
 - HSP Problem
 - RSA cryptography

Reiffel, Chapter 8
Kaye, Chapter 7
Nielsen and Chuang, Chapter 5

PHYC90045 Introduction to Quantum Computing

Shor's algorithm



- Efficient quantum algorithms for **factoring** semiprime numbers
- Best known classical algorithm is number field sieve (exponential in bit-length).
- Underpins the RSA cryptosystem
- Hidden Subgroup Problems (eg. Discrete logarithm) similar.



Peter Shor



Shor, Proc 35th Ann Symp of Comp Sci, 26, (1995)

PHYC90045 Introduction to Quantum Computing

Factoring and Period Finding



We want to factor N=15. Take a number a=2 (say) relatively-prime to N (ie. no prime factors in common) and find the **order** r of a. That is the least r, such that $a^r \equiv 1 \pmod{15}$:

$2^0 = 1 \pmod{15}$
$2^1 = 2 \pmod{15}$
$2^2 = 4 \pmod{15}$
$2^3 = 8 \pmod{15}$
$2^4 = 1 \pmod{15}$

After which the pattern repeats.
Formally, we say: the **order** of 2 mod 15 is 4. Or, if we defined a function:

$$f(k) = a^k \pmod{N}$$

We would say that the **period** of f is r, since $f(x+r) = f(x)$.

PHYC90045 Introduction to Quantum Computing

Example of finding factors from a period



In our case, we have a=2, N=15 and r=4. Happily r=4 is even. We can rearrange:

$$\begin{aligned} a^r &= 1 \pmod{N} \\ a^r - 1 &= 0 \pmod{N} \\ (a^{r/2} + 1)(a^{r/2} - 1) &= 0 \pmod{N} \end{aligned}$$

In our case,

$$\begin{aligned} a^{r/2} - 1 &= 2^{4/2} - 1 = 3 \\ a^{r/2} + 1 &= 2^{4/2} + 1 = 5 \end{aligned}$$

and

3 × 5 = 15

PHYC90045 Introduction to Quantum Computing

Divisors of N

In our case 3 and 5 divide N=15 exactly, but we're not guaranteed that always, only that:

$$(a^{r/2} + 1)(a^{r/2} - 1) = 0 \pmod{N}$$

i.e. that

$$(a^{r/2} + 1)(a^{r/2} - 1) = kN$$

As long as neither factor is a multiple of N, then both will have non-trivial factors with N. To find these factors, we find the greatest common divisors (for which the Euclidean algorithm is efficient):

$$\gcd(a^{r/2} + 1, N)$$

$$\gcd(a^{r/2} - 1, N)$$

These give a **non-trivial factor of N**.

If r is even or if the factors found are trivial, we repeat the algorithm with a different choice of a.

PHYC90045 Introduction to Quantum Computing

TLDR: Factoring and Period finding

If we can find the period of $f(k) = a^k \pmod{N}$ efficiently, we can factor efficiently.

Shor's algorithm finds this period efficiently, and we can then use classical techniques to factor semi-prime numbers into their prime factors.

PHYC90045 Introduction to Quantum Computing

Shor's algorithm

Two registers*:

(1) Equal superposition

(2) Calculate function:
 $f(x) = a^x \pmod{N}$

(3) QFT

(4) Measure result

* L = number of bits in N

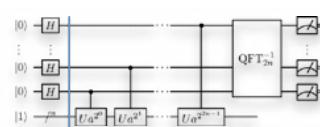
Shor, Proc 35th Ann Symp of Comp Sci, 26, (1995)

PHYC90045 Introduction to Quantum Computing



 The University of
 MELBOURNE

Shor's algorithm explained



The diagram illustrates a quantum circuit for Shor's algorithm. It features two horizontal lines representing registers. The top register has four qubits, each initialized to |0⟩. A Hadamard gate (H) is applied to the first qubit. This is followed by three control dots, indicating the sequence continues for the remaining qubits. The bottom register has one qubit initialized to |1⟩. A sequence of gates follows: a Hadamard gate (H), then a series of controlled phase shift gates labeled $U_{a^0}, U_{a^2}, U_{a^4}, \dots, U_{a^{2n-1}}$. After this sequence, there are three more control dots. Finally, a Quantum Fourier Transform (QFT) gate is applied to the bottom register, followed by three more control dots.

After the Hadamard gates, the top register is in the equal superposition:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle |1\rangle$$

PHYC90045 Introduction to Quantum Computing

Modular Exponentiation

Most Significant Bit

Least Significant Bit

2L qubits

L qubits

Multiplication by $a^1, a^2, a^4 \dots \text{mod } N$

For example if the top register contained $x = 101$, and $a=2$, and $N=15$ then we would:

- Start with 1
- Multiply by $a^1=2^1=2$ giving $2 \text{ mod } 15$
- Not multiply by $a^2=2^2=4$
- Multiply by $a^4=2^4=16$ giving $32 \text{ or } 2 \text{ mod } 15$

Top register in superposition, so bottom register is correlated (entangled) with the top register

PHYC90045 Introduction to Quantum Computing

Example of Modular Exponentiation

THE UNIVERSITY OF MELBOURNE

After modular exponentiation:

$$|\psi\rangle = \sum_x |\langle x | a^x \mod N \rangle |x\rangle$$

e.g. For $a=2, N=15$:

$$\begin{aligned} |\psi\rangle &= (|0\rangle + |4\rangle + |8\rangle + |12\rangle) \otimes |1\rangle \\ &\quad + (|1\rangle + |5\rangle + |9\rangle + |13\rangle) \otimes |2\rangle \\ &\quad + (|2\rangle + |6\rangle + |10\rangle + |14\rangle) \otimes |4\rangle \\ &\quad + (|3\rangle + |7\rangle + |11\rangle + |15\rangle) \otimes |8\rangle \end{aligned}$$

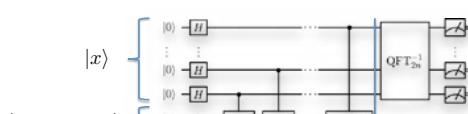
Note: States are unnormalized! (for simplicity)

PHYC90045 Introduction to Quantum Computing



 The University of
 MELBOURNE

Shor's algorithm explained



The diagram illustrates a quantum circuit for Shor's algorithm. It features two horizontal register lines. The top register has four qubits, labeled $|x\rangle$, with controls for a sequence of gates: U , followed by three dashed boxes representing intermediate operations, then another U , and finally a QFT_{2n}^{-1} block. The bottom register has four qubits, labeled $|a^x \bmod N\rangle$, with controls for a sequence of gates: U , followed by three dashed boxes representing intermediate operations, then another U , and finally a Ua^{2^n-1} gate.

If, at this point the bottom register is measured to be 2 (at random), we may collapse to the state:

$$|\psi\rangle = (|1\rangle + |5\rangle + |9\rangle + |13\rangle) \otimes |2\rangle$$

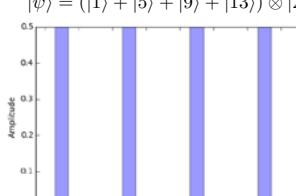
PHYC90045 Introduction to Quantum Computing


 THE UNIVERSITY OF
MELBOURNE

The Fourier Transform

Imagine we measure the bottom register, and plot the amplitudes in the top register:

$$|\psi\rangle = (|1\rangle + |5\rangle + |9\rangle + |13\rangle) \otimes |2\rangle$$



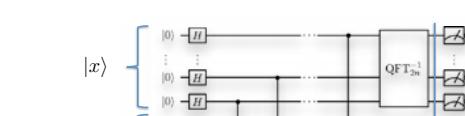
Index	Amplitude
0	0.0
1	0.5
2	0.0
3	0.0
4	0.0
5	0.5
6	0.0
7	0.0
8	0.5
9	0.5
10	0.0
11	0.0
12	0.0
13	0.5
14	0.0
15	0.0

This function is periodic, with a period of r .

PHYC90045 Introduction to Quantum Computing



Taking the QFT



The diagram illustrates a quantum circuit for performing a Quantum Fourier Transform (QFT) on a state $|x\rangle$ mod N . The circuit consists of two main parts: a preparation section on the left and a transformation section on the right.

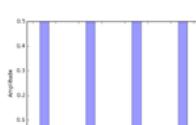
Preparation Section: This section takes the input state $|x\rangle$ mod N and prepares it as a superposition of basis states. It uses controlled operations where the control is on the second register and the target is on the first register. The controls are labeled $|a^x \bmod N\rangle$, which is expanded into its binary representation $|a^x \bmod N\rangle = \sum a_i |i\rangle$. Each term $|i\rangle$ is controlled by a Hadamard gate (H) on the second register. The first term $|0\rangle$ is controlled by a single H gate on the second register. Subsequent terms $|1\rangle, |2\rangle, \dots, |n-1\rangle$ are controlled by multi-controlled operations where the control is on the entire first register and the target is on the second register. These controls are labeled $|a^x \bmod N\rangle = \sum a_i |i\rangle$.

Transformation Section: This section contains a sequence of unitary gates $U_{a^2^0}, U_{a^2^1}, \dots, U_{a^2^{n-1}}$ applied to the first register. Following this, a large block labeled QFT_{2n}^{-1} is applied to both registers. Finally, measurement devices are shown on both registers.

PHYC90045 Introduction to Quantum Computing



Inverse QFT for N=15, a=2



Index	Amplitude
0	~0.7
4	~0.7
8	~0.7
12	~0.7
Others	0.0

To find the period, we will take the Quantum Fourier Transform to reveal r (and so also, if r is even, factors of N).

The result of taking a Fourier transform is a “spectrum” peaked around (for integer, k):

$$k \frac{2^n}{r}$$

n ($2L$) is number of qubits in the top register r is the period being determined



Frequency	Amplitude
0	~0.7
4	~0.7
8	~0.7
12	~0.7
Others	0.0

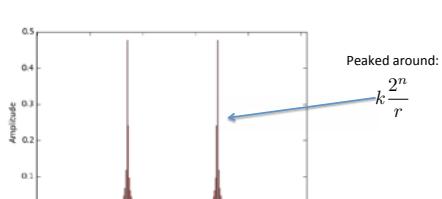
PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

When r doesn't divide evenly

What happens when r doesn't divide evenly into the top register? Then we still get a very peaked distribution around the same values:



Peaked around:
 $k \frac{2^n}{r}$

Here is an example for $r=3$ and $2^n=256$.

PHYC90045 Introduction to Quantum Computing

Measurement

Measurement will randomly give one of these values, close to:

Frequency	Probability
0	~0.48
4	~0.48
8	~0.48
12	~0.48

$$m = k \frac{2^n}{r}$$

$$\text{or} \quad \frac{k}{r} = \frac{m}{2^n}$$

We need a rational approximation of $m/2^n$ to find r .

PHYC90045 Introduction to Quantum Computing



Example for $a=2$, $N=15$

In our example:

We might randomly measure $m=4$

$$\frac{k}{r} = \frac{m}{2^n} \text{ and in this case: } \frac{m}{2^n} = \frac{4}{16}$$

$$= \frac{1}{4}$$

Since this is equal to k/r ,
We have correctly found
 $r=4$

Note: This step might only reveal a factor of r , and so might have to be repeated...

PHYC90045 Introduction to Quantum Computing

The logo of The University of Melbourne, featuring a coat of arms with a central shield, a crest with a lion, and a banner below it.

Continued Fractions

The result of taking a Fourier transform is a spectrum peaked around (for integer, k):

$$k \frac{2^n}{r}$$

Unless r divides 2^n exactly, we will only get an approximation to $k2^n/r$ when measured.

Most of the time 2^n and r will be relatively prime. The problem then is to find good approximations to the measured value $m/2^n = k/r$. The "correct" approximation yields the period, r, as the denominator.

A good method for making *rational approximations* is to use the **continued fractions** method.

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots \cfrac{1}{a_n}}}}$$

PHYL9004S Introduction to Quantum Computing



 The University of
 MELBOURNE

Continued Fraction of Pi

As an example, let's try to make a rational approximation to pi. Our first approximation is

$$\pi \approx 3 \quad (a_0 = 3)$$

The remaining decimal part is $0.14159265\dots = 1/7.0625\dots$. This gives a second approximation:

$$\pi \approx 3 + \frac{1}{7} \quad (a_1 = 7)$$

The remaining decimal part $0.0625 = 1/15.9966\dots$. This gives a third approximation:

$$\pi \approx 3 + \frac{1}{7 + \frac{1}{15}} \quad (a_2 = 15)$$

And so on. This method can be used to find good rational approximations to $\sqrt{2}$ and find r .

PHYC90045 Introduction to Quantum Computing

Example: Factoring the number

$a^{r/2} - 1 = 2^{4/2} - 1 = 3$

$a^{r/2} + 1 = 2^{4/2} + 1 = 5$

Not really necessary here, but in general you'd have to evaluate:

$$\begin{aligned} \gcd(3, 15) &= 3 \\ \gcd(5, 15) &= 5 \end{aligned}$$

And so we've found two non-trivial factors of 15:

3 × 5 = 15

PHYC90045 Introduction to Quantum Computing

Shor's algorithm Summary

1. Randomly pick integer $0 < a < N$ (and check a is not a factor of N)

2. Apply the circuit above, using modular exponentiation to calculate a^x , $\text{QFT } x$.

3. Measure to obtain an approximation to $v = k 2^n/r$

4. Use continued fractions of $v/2^n$ to obtain even r

5. Use Euclidean algorithm to find common factors of N with $(a^{r/2}+1)$ and $(a^{r/2}-1)$

6. Repeat if necessary

PHYC90045 Introduction to Quantum Computing

Shor's algorithm

- Efficient quantum algorithms for **factoring** semiprime numbers
- Best known classical algorithm is number field sieve (exponential in bit-length).
- Underpins the RSA cryptosystem
- Hidden Subgroup Problems (eg. Discrete logarithm) similar.

Peter Shor

Shor, Proc 35th Ann Symp of Comp Sci, 26, (1995)

PHYC90045 Introduction to Quantum Computing

Private Key Cryptography

Much of internet security relies on 'public key cryptography'.

RSA cryptography relies on the difficulty of factoring large semi-primes.

The best known **classical algorithm** is the number field sieve:

$$O(e^{1.9(\log N)^{1/3}(\log \log N)^{2/3}})$$

Shor's factoring **quantum algorithm** solves the same problem in poly-log time:

$$O((\log N)^2(\log \log N)(\log \log \log N))$$

RSA Factoring Challenge			
			
RSA-190	190	938	May 8, 2010 [edit history]
RSA-191	190	629	November 8, 2010 [edit history]
RSA-440	193	640	December 2, 2006 [edit history]
RSA-390	200	963	May 9, 2005 [edit history]
RSA-717	210	696	September 26, 2013 [edit history]
RSA-107	210	696	April 2009 [edit history]
RSA-754	312	754	\$30,000 USD [edit history]
RSA-201	220	729	May 13, 2016 [edit history]
RSA-200	230	762	August 15, 2016 [edit history]
RSA-332	232	768	Samuel S. Gross, Nostra, Inc. [edit history]
RSA-191	232	768	Thorsten Kleinjung et al. [edit history]
RSA-240	240	793	

RSA Factoring Challenge, Wikipedia

PHYC90045 Introduction to Quantum Computing

The University of
MELBOURNE

Discrete Logarithm

A closely related class of problems which are important for cryptography are solving discrete logarithm problems:

Given, **a**, **b** and **N**, st.

$$a = b^t \bmod N$$

find **t**.

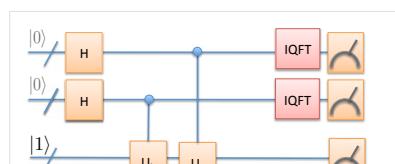
RSA is based on factoring. Diffie-Hellman key exchange, El Gamal and elliptic curve cryptography rely on discrete logarithm being a hard problem.

PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

Circuit for Discrete Logarithm



The diagram shows a quantum circuit with three horizontal wires representing qubits. The top wire starts in state $|0\rangle$, the middle in $|0\rangle$, and the bottom in $|1\rangle$.
 - On the top wire, there is a Hadamard gate (H) followed by a control point for a CNOT gate.
 - On the middle wire, there is a Hadamard gate (H) followed by a control point for a CNOT gate.
 - On the bottom wire, there are two gates: U_b followed by U_a .
 - A CNOT gate connects the top wire to the middle wire.
 - Another CNOT gate connects the middle wire to the bottom wire.
 - After the bottom wire passes through U_a , it enters a block labeled "IQFT" which includes three controlled operations: a CNOT gate with control on the bottom wire and target on the top wire, a phase gate (blue diamond), and another CNOT gate with control on the bottom wire and target on the middle wire.
 - The middle wire then passes through an IQFT block, which includes a CNOT gate with control on the middle wire and target on the top wire, a phase gate, and another CNOT gate with control on the middle wire and target on the bottom wire.
 - Finally, both the top and middle wires pass through a measurement meter (indicated by a curved arrow).

Measurement of the second register reveals: k/r

Measurement of the first register reveals: $kt \bmod r/r$

Note: same $k!$

At least in principle we can know r by Shor's factoring algorithm, so only t is unknown, and can easily found:

$$k^{-1}kt = t \bmod r$$

PHYC90045 Introduction to Quantum Computing

The logo of the University of Melbourne, featuring a coat of arms with a lion and a unicorn flanking a shield, topped by a crown, with the words "UNIVERSITY OF MELBOURNE" below it.

Hidden Subgroup Problems

The generalisation of Shor's algorithm to arbitrary groups is known as the Hidden Subgroup Problem:

Let G be a group. Suppose a subgroup $H \leq G$ is implicitly defined by f on G s.t. f is a constant (and distinct) on every coset of H . Find the generators of H .

Simon's algorithm and Shor's algorithm are examples of Hidden Subgroup Problems (HSPs).

PHYC90045 Introduction to Quantum Computing		
Addition with QFT		
<p>Now we need to build the basic arithmetical operations to implement Shor's algorithm.</p> <p>Addition using the QFT (more in Lab-5):</p>		
a_1	QFT	$a = a_1 2^{n-1} + a_2 2^{n-2} + \dots + a_{n-2} 2 + a_n$
a_2	QFT	$b = b_1 2^{n-1} + b_2 2^{n-2} + \dots + b_{n-2} 2 + b_n$
a_3	QFT	$s = a + b = s_1 2^{n-1} + s_2 2^{n-2} + \dots + s_{n-2} 2 + s_n$
$a_1 = \frac{1}{\sqrt{2}}(0\rangle + e^{2\pi i (b_1 + a_1)/2^n} 1\rangle)$ $a_2 = \frac{1}{\sqrt{2}}(0\rangle + e^{2\pi i (b_2 + a_2)/2^n} 1\rangle)$ $a_3 = \frac{1}{\sqrt{2}}(0\rangle + e^{2\pi i (b_3 + a_3)/2^n} 1\rangle)$		$R_Z\left(\frac{\pi}{2^n}\right) R_Z\left(\frac{\pi}{2^n}\right) R_Z\left(\frac{\pi}{2^n}\right) R_Z\left(\frac{\pi}{2^n}\right)$ $R_Z\left(\frac{\pi}{2^n}\right) R_Z\left(\frac{\pi}{2^n}\right)$ $R_Z\left(\frac{\pi}{2^n}\right)$
$b_1 = \frac{1}{\sqrt{2}}(0\rangle + e^{2\pi i (a_1 + b_1)/2^n} 1\rangle)$ $b_2 = \frac{1}{\sqrt{2}}(0\rangle + e^{2\pi i (a_2 + b_2)/2^n} 1\rangle)$ $b_3 = \frac{1}{\sqrt{2}}(0\rangle + e^{2\pi i (a_3 + b_3)/2^n} 1\rangle)$		$\frac{1}{\sqrt{2}}(0\rangle + e^{2\pi i (a_1 + b_1 + a_2 + b_2 + a_3 + b_3)/2^n} 1\rangle)$ $\frac{1}{\sqrt{2}}(0\rangle + e^{2\pi i (a_1 + b_1 + a_2 + b_2)/2^n} 1\rangle)$ $\frac{1}{\sqrt{2}}(0\rangle + e^{2\pi i (a_1 + b_1)/2^n} 1\rangle)$
$s_1 = \frac{1}{\sqrt{2}}(0\rangle + e^{2\pi i (a_1 + b_1 + a_2 + b_2 + a_3 + b_3)/2^n} 1\rangle)$ $s_2 = \frac{1}{\sqrt{2}}(0\rangle + e^{2\pi i (a_1 + b_1 + a_2 + b_2)/2^n} 1\rangle)$ $s_3 = \frac{1}{\sqrt{2}}(0\rangle + e^{2\pi i (a_1 + b_1)/2^n} 1\rangle)$		
$R_Z\left(\frac{\pi}{2^n}\right) \frac{1}{\sqrt{2}}(0\rangle + C 1\rangle) = \frac{1}{\sqrt{2}}(0\rangle + C e^{2\pi i k/2^n} 1\rangle) = \frac{1}{\sqrt{2}}(0\rangle + C e^{2\pi i a_n/2^n} 1\rangle) = \frac{1}{\sqrt{2}}(0\rangle + C e^{2\pi i b_n/2^n} 1\rangle)$		

The figure shows a quantum circuit simulation interface. The circuit consists of six qubits. The first three qubits (q0, q1, q2) are initialized to 1, indicated by the value 3=011₂. The next two qubits (q3, q4) are initialized to 2, indicated by the label "Phase rotations representing 2". The final qubit (q5) is initialized to 0. The circuit includes a sequence of operations: a QFT on qubits q0 through q4, followed by an inverse QFT on qubits q3 through q5. The interface features a timeline at the bottom, a color bar for probability, and various controls for parameters like rotation angles.

PHY90045 Introduction to Quantum Computing

Addition using QFT: First qubit

$$|j_1, \dots, j_n\rangle \rightarrow \frac{|0\rangle + e^{2\pi i \cdot 0 \cdot j_n}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i \cdot 0 \cdot j_{n-1} \cdot j_n}|1\rangle}{\sqrt{2}} \otimes \dots \otimes \frac{|0\rangle + e^{2\pi i \cdot 0 \cdot j_1 \cdot j_2 \cdot \dots \cdot j_{n-1} \cdot j_n}|1\rangle}{\sqrt{2}}$$

Local phase as a fraction of 2π

$$0.011 = 3/8$$

Rotation by
0.010 = 2/8

Total phase rotation:
5/8 = 0.101

Same as
for
5=101!

PHYC90045 Introduction to Quantum Computing

Addition using QFT: Second qubit

$|j_1, \dots, j_n\rangle \rightarrow \frac{|0\rangle + e^{2\pi i \cdot 0 \cdot j_n}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i \cdot 0 \cdot j_{n-1} \cdot j_n}|1\rangle}{\sqrt{2}} \otimes \dots \otimes \frac{|0\rangle + e^{2\pi i \cdot 0 \cdot j_1 \cdot j_2 \dots j_{n-1} \cdot j_n}|1\rangle}{\sqrt{2}}$

Leave off the first qubit!

Local phase as a fraction of 2π

Rotation by $0.10 = 2/4$

Same as for $5=10!$

0.11 = 3/4

Rotation by $0.11 = 3/4$

Total phase rotation: $5/4 = 1/4 = 0.01$

PHYC90045 Introduction to Quantum Computing

Addition using QFT: Third qubit

$|j_1, \dots, j_n\rangle \rightarrow \frac{|0\rangle + e^{2\pi i \cdot 0 \cdot j_n}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i \cdot 0 \cdot j_{n-1} \cdot j_n}|1\rangle}{\sqrt{2}} \otimes \dots \otimes \frac{|0\rangle + e^{2\pi i \cdot 0 \cdot j_1 \cdot j_2 \dots j_{n-1} \cdot j_n}|1\rangle}{\sqrt{2}}$

Leave off the first two qubits!
Local phase as a fraction of 2π

Rotation by
 $0.0 = 0$

$2\pi/2=\pi$

Total phase rotation:
 $0 + \frac{1}{2} = 1/2 = 0.1$
 $2\pi/2=\pi$

Same as for
 $5=1011$

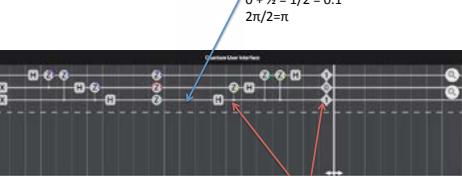
PHYC90045 Introduction to Quantum Computing

Addition using QFT: Third qubit

$$|j_1, \dots, j_n\rangle \rightarrow \frac{|0\rangle + e^{2\pi i \cdot 0 \cdot j_n}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i \cdot 0 \cdot j_{n-1} \cdot j_n}|1\rangle}{\sqrt{2}} \otimes \dots \otimes \frac{|0\rangle + e^{2\pi i \cdot 0 \cdot j_1 \cdot j_2 \cdot \dots \cdot j_{n-1} \cdot j_n}|1\rangle}{\sqrt{2}}$$

Leave off the first two qubits!

Total phase rotation:
 $0 + \frac{\pi}{2} = 1/2 = 0.1$
 $2n/2 = \pi$



π rotation means H makes this state "1"

PHYS90045 Introduction to Quantum Computing

Addition using QFT: Second qubit

$|j_1, \dots, j_n\rangle \rightarrow \frac{|0\rangle + e^{2\pi i \cdot 0 \cdot j_n}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i \cdot 0 \cdot j_{n-1} \cdot j_n}|1\rangle}{\sqrt{2}} \otimes \dots \otimes \frac{|0\rangle + e^{2\pi i \cdot 0 \cdot j_1 \cdot j_2 \cdot \dots \cdot j_{n-1} \cdot j_n}|1\rangle}{\sqrt{2}}$

Leave off the first qubit!

Local phase as a fraction of 2π

Total phase rotation:
 $\frac{5}{4}/\frac{1}{4} = 0.04\pi$

Controlled operation cancels the $2\pi/4$ rotation

No remaining phase, leaves qubit in 0 state

PHYC90045 Introduction to Quantum Computing

Addition using QFT: First qubit

$|j_1, \dots, j_n\rangle \rightarrow \frac{|0\rangle + e^{2\pi i \cdot 0 \cdot j_n}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i \cdot 0 \cdot j_{n-1} \cdot j_n}|1\rangle}{\sqrt{2}} \otimes \dots \otimes \frac{|0\rangle + e^{2\pi i \cdot 0 \cdot j_1 \cdot j_2 \dots j_{n-1} \cdot j_n}|1\rangle}{\sqrt{2}}$

Local phase as a fraction of 2π :
 $5/8 = 0.101$ Total phase rotation:
 Gives the answer,
 $3+2=5$

Quantum Interface

Controlled operation cancels the $2\pi/8$ rotation
 Remaining π phase, leaves qubit in 1 state

PHYC90045 Introduction to Quantum Computing

Multiplier

$a \cdot b = (a_n 2^n + a_{n-1} 2^{n-1} + \dots + a_2 4 + a_1 \cdot 2 + a_0 \cdot 1)b$
 $= a_n 2^n b + a_{n-1} 2^{n-1} b + \dots + a_2 4b + a_1 2b + a_0 b$

Add $2^n b$ iff $a_n = 1$. Key idea: Use a_n as a control qubit for addition

PHYC90045 Introduction to Quantum Computing

Multiplication (2x3) using QFT

$2 = 010_1$

Add 3 iff the ones bit is a 1

Quantum User Interface

Entanglement Probability: Min Max Avg

FILE N QUBITS

The screenshot shows a quantum circuit on the Quantum User Interface. The circuit consists of two qubits. The first qubit starts with a Hadamard gate, followed by three CNOT gates with controls on the second qubit. The second qubit starts with a CNOT gate with control on the first qubit, followed by three Z gates, another CNOT gate with control on the first qubit, and finally a Hadamard gate. A text annotation above the circuit reads "Add 6 iff the two's bit is a 1". A blue arrow points from the text to the third CNOT gate on the second qubit. On the left, the input state $2 = 010_1$ is shown as a vector with a red component at index 1. Below the circuit, an "Entanglement Probability" slider is set to Max, and a color wheel indicates the current angle. The interface also includes a FILE menu and a 5 QUBITS setting.

PHYC90045 Introduction to Quantum Computing

Multiplication (2x3) using QFT

Add 4 ($-12 \bmod 8$) iff the fours bit is a 1

$2 = 010_1$

Quantum User Interface

Entanglement Probability

FILE

9 QUBITS

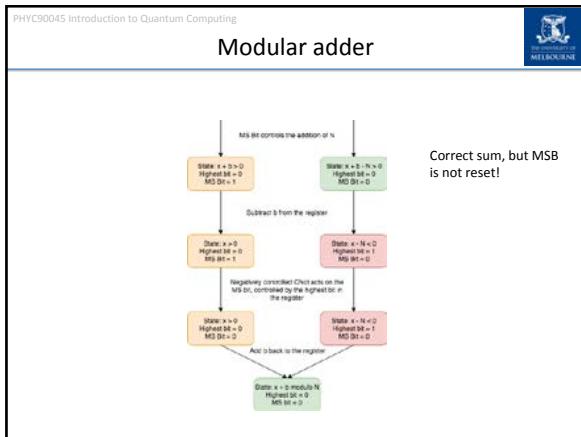
Measurement Probability

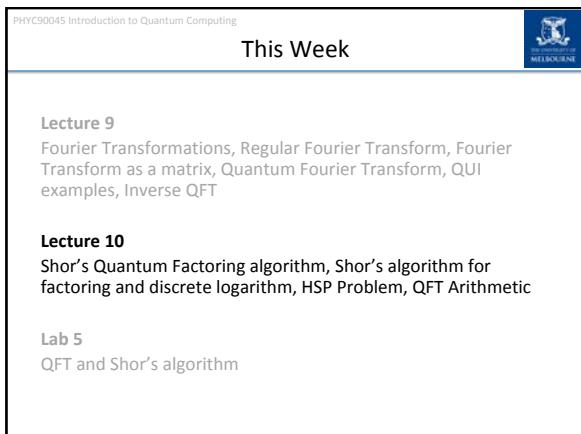
PHYC90045 Introduction to Quantum Computing

Modular adder

```

graph TD
    Start((Start with x)) --> Perform[Perform x = 0 > N]
    Perform --> Left[State: x = 0 > N - 1  
Highest bit = 1]
    Perform --> Right[State: x = 0 > N - 1  
Highest bit = 0]
    Left --> Ctrl[Ctrl acts on the MS bit, controlled by the highest bit in the register]
    Ctrl --> StateN1[State: x = 0 > N - 1  
MS bit = 1]
    Right --> StateN1
    StateN1 --> Add[MS bit controls the addition of N  
x+b < N]
    Add --> Final[State: x = 0 > N  
MS bit = 1]
    Final --> Result[x+b > N]
  
```





PHYC90045 Introduction to Quantum Computing

Week 6

Lecture 11 - Quantum Supremacy

- 11.1 Boson Sampling
- 11.2 IQP Problem
- 11.3 Google's pseudorandom circuits

Lecture 12 - Errors

- 12.1 Quantum errors: unitary and stochastic errors
- 12.2 Randomized Benchmarking
- 12.3 Purity

Lab 6

Quantum Supremacy and Errors

PHYC90045 Introduction to Quantum Computing

Quantum Supremacy

Physics 90045
Lecture 11

PHYC90045 Introduction to Quantum Computing

Determining supremacy?



Gary Kasparov

vs



Deep Blue

On February 10, 1996, Deep Blue beat Kasparov under tournament regulations. In the subsequent 1997 rematch, Deep Blue won the series.

PHYC90045 Introduction to Quantum Computing

Quantum supremacy in the news

Google, Alibaba Spar Over Timeline for 'Quantum Supremacy' - NewScientist

Google just made it much harder to build a serious quantum computer



Is a quantum revolution near? ALFRED HIRSCHEN/SCIENCE PHOTO LIBRARY

By Chelsea Whyte

Google is racing to create the first quantum computer capable of solving a problem ordinary computers cannot – and it has just made that challenge much harder.

Achieving "quantum supremacy", as it is known, involves building a device that can solve a problem faster than any non-quantum computer.

<https://www.wired.com/story/google-alibaba-spar-over-timeline-for-quantum-supremacy/>

<https://www.newscientist.com/article/2176375-google-just-made-it-much-harder-to-build-a-serious-quantum-computer/>

PHYC90045 Introduction to Quantum Computing

THE UNIVERSITY OF MELBOURNE

What is quantum supremacy?

Quantum supremacy is using a quantum computer to solve a problem which classical computers **practically** cannot.

PHYC90045 Introduction to Quantum Computing

UNIVERSITY OF MELBOURNE

Algorithms for quantum supremacy

The race is on to build a quantum computer which will achieve quantum supremacy. Implementing large scale factoring would demonstrate quantum supremacy, but that would require a very large (potentially millions of qubits) quantum computer. In the short term we will only have access to NISQ devices.

Noisy
Intermediate Scale (50-100 qubits)
Quantum devices

Three quantum algorithms that might be able to demonstrate quantum supremacy with 50-100 qubits:

- Boson Sampling
- Instantaneous Quantum Polynomial-Time circuits (IQP)
- Pseudorandom circuits

PHYC90045 Introduction to Quantum Computing

HOWTO quantum supremacy

Pick a problem which is:

- As easy as possible for a quantum computer
- As hard as possible for a classical computer to simulate

PHYC90045 Introduction to Quantum Computing

Boson Sampling

PHYC90045 Introduction to Quantum Computing

A little physics experiment...

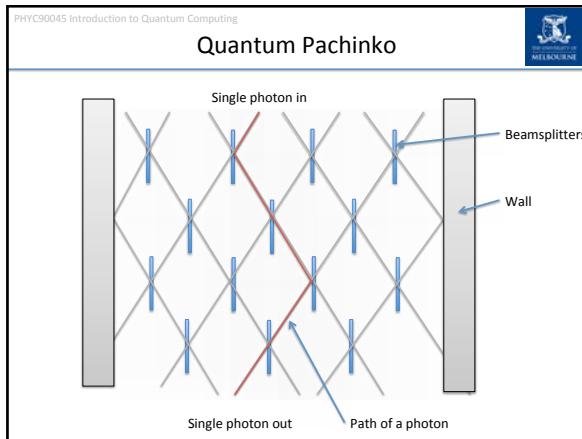
n single photon sources

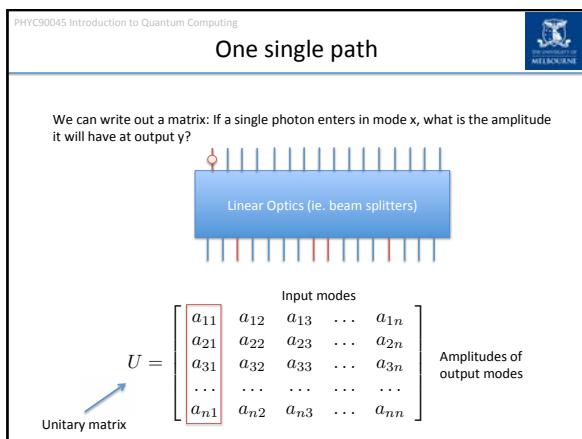
Linear Optics (ie. beam splitters)

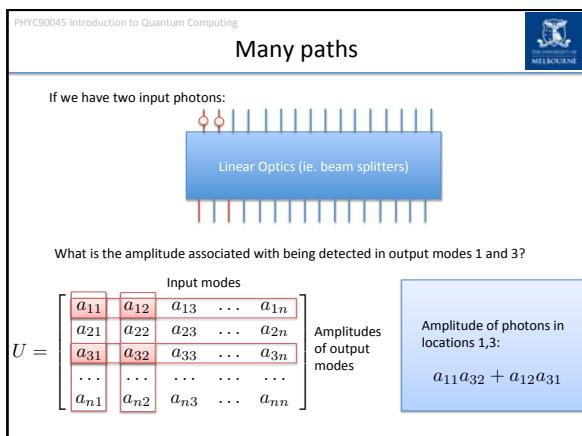
n^2 output modes.

You won't need to know the physics of this device.

Can a classical computer produce **samples** from the output which mimic the quantum device?







PHYC90045 Introduction to Quantum Computing

Many paths

In general take the submatrix corresponding to a particular input and output, and find its permanent:

$$U = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix}$$

Submatrix defined by the input modes and output modes

The resulting amplitude is the **permanent** of the submatrix:

$$\text{perm}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^n a_{i,\sigma(i)}$$

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad \text{perm}(A) = ad + bc$$

Same as determinant, but with no subtraction, all addition.

PHYC90045 Introduction to Quantum Computing

Complexity of finding permanent

Unlike determinants, finding a permanent of a matrix is a *surprisingly difficult* computational problem.

Finding the permanent is a #P complete problem.

#P is the set of counting problems associated with decision problems in NP.

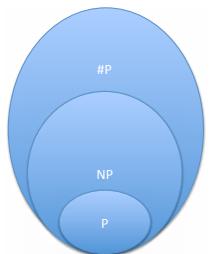
NP: Is there as a satisfying assignment of variables to this 3SAT problem?
#P: How many satisfying assignments of variables are there to this 3SAT problem?

NP: Is there a travelling salesman path with distance less than d ?
#P: How many travelling salesman paths are there with a distance less than d .

Calculating amplitudes and probabilities for Boson sampling is a hard classical problem!

PHYC90045 Introduction to Quantum Computing

Some classical complexity classes



Informally:

P: Problems which can be solved in polynomial time

NP: Problems which can be checked in polynomial time (ie. they have an efficiently verifiable proof)

#P: Problems which count the number of solutions in NP

PHYC90045 Introduction to Quantum Computing

The polynomial hierarchy

Very quick introduction: Given an **oracle** in some complexity class which evaluates instantly, what problems can we now evaluate in polynomial time?

$P^P = P$

Polynomial time algorithm
With access to an oracle which can instantaneously evaluate functions in P

$NP^P = NP$

But a polynomial time algorithm with an NP oracle appears to be more powerful than both P and NP :

P^{NP}

We can recursively define complexity classes this way, with oracles which increase in strength at each level. This whole hierarchy is known as the Polynomial Hierarchy, **PH**.

If, at some level, providing the oracle didn't lead to a superset of problems, the polynomial hierarchy would "collapse". Computer scientists don't think this happens.

PHYC90045 Introduction to Quantum Computing

Sampling is also hard to simulate

Calculating amplitudes and probabilities for Boson sampling is a hard classical problem!

Calculate the **permanent** of the submatrix:

$$\text{perm}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^n a_{i,\sigma(i)} \quad \text{perm} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad + bc$$

We don't technically have to calculate the probabilities explicitly. Maybe we can *sample* from the probability distribution?

No – this would result in a collapse of the Polynomial Hierarchy.

Not proven, but like $P=NP$, computer scientists generally don't expect the polynomial hierarchy collapses.

PHYC90045 Introduction to Quantum Computing

HOWTO quantum supremacy

Boson Sampling is a problem which:

- "Easy" to implement using linear optics
- Hard for a classical computer to simulate –
Polynomial Hierarchy would collapse

PHYC90045 Introduction to Quantum Computing



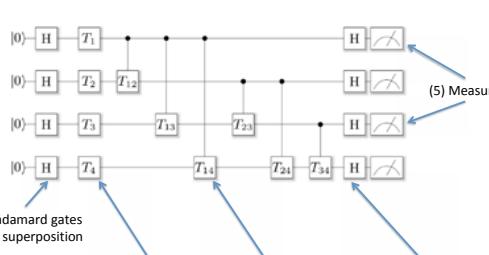
IQP Circuits

Instantaneous Quantum Polynomial-Time

PHYC90045 Introduction to Quantum Computing



IQP Circuits



(1) Hadamard gates
Equal superposition

(2) Random single qubit phases

(3) Random two qubit phase gates
between every pair

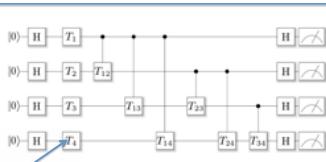
(4) Interfere all the resulting states

(5) Measure

PHYC90045 Introduction to Quantum Computing



Random Phases



Each of these T_m gates is a rotation (around z) by a multiple of $\pi/4$:

$$T_m = \cos\left(\frac{k_m \pi}{8}\right) I + i \sin\left(\frac{k_m \pi}{8}\right) Z_m$$

$$T_m = R_z\left(-\frac{k_m \pi}{4}\right) \quad \text{on the } m^{\text{th}} \text{ qubit}$$

Where k_m is an integer chosen uniformly at random between 0 and 7. This is equivalent (up to a global phase) of applying a T gate k_m times.

PHYC90045 Introduction to Quantum Computing

Random Joint Phases

Each of these T_{mn} gates is a joint phase rotation by a multiple of $\pi/8$:

$$T_{mn} = \cos\left(\frac{k_{mn}\pi}{8}\right) I + i \sin\left(\frac{k_{mn}\pi}{8}\right) Z_m Z_n$$

Where k_m is an integer chosen uniformly at random between 0 and 7.

In the lab we can implement a similar algorithm with controlled T_{mn} gates.

PHYC90045 Introduction to Quantum Computing

"Instantaneous"

All of these phase gates commute

The order which you apply the single and two qubit phase gates doesn't matter. They commute with each other, so can be applied in any order.

Eg. $ZT_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix}$

$T_2Z = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix}$

Diagonal gates commute

PHYC90045 Introduction to Quantum Computing

Collapse of the polynomial hierarchy

Aim: To sample from the output of this circuit. Easy for a quantum computer.

If this could be done efficiently using a classical computer, it would imply the collapse of the polynomial hierarchy (and so isn't expected to be possible).

Practically, classical simulations are limited to <50-70 qubits (for low error rates).

PHYC90045 Introduction to Quantum Computing

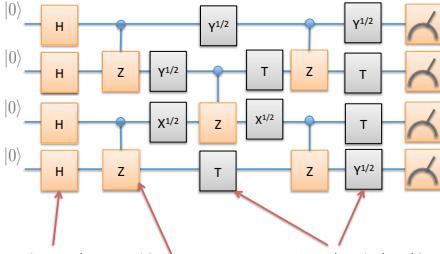


Pseudorandom Circuits

PHYC90045 Introduction to Quantum Computing



The circuit



Hadamards to give equal superposition

Controlled-Z gates to provide entanglement in a regular pattern

A random single qubit gate between every pair of CZ gates.

PHYC90045 Introduction to Quantum Computing



Square Root X and Y

In the previous slide, we simply have that

$$X^{1/2} = R_x \left(\frac{\pi}{2} \right)$$

and similarly,

$$Y^{1/2} = R_y \left(\frac{\pi}{2} \right)$$

PHYC90045 Introduction to Quantum Computing

Schedule of CZ

FIG. 6. Layouts of CZ gates in a 6×6 qubit lattice. It is currently not possible to perform two CZ gates simultaneously in two neighboring superconducting qubits [33, 34, 49, 52]. We iterate over these arrangements sequentially, from 1 to 8.

From Boixo et al, <https://arxiv.org/pdf/1608.00263.pdf>, 2016.

PHYC90045 Introduction to Quantum Computing

Sampling is hard

Once again, the aim of the algorithm is to sample from the measured values.
If this were possible to do efficiently classically, it would imply a collapse of the polynomial hierarchy.
In practice, simulating ~ 45 qubits for this problem is hard (note: need high depth circuit)

PHYC90045 Introduction to Quantum Computing

What do you need for quantum supremacy?

- Many qubits (~ 50)
- Large depth circuit (~ 100)
- Entanglement, high T gate count
- Low error rates (<1%)

PHYC90045 Introduction to Quantum Computing

60 Qubit Simulations of Shor's algorithm

Using MPS, Aidan Dang wrote parallel code were able to do large scale simulations of Shor's algorithm.

l	r	α	β	n_{node}	t_U	t_{max}	t_{QPT}	t_{total}
16	28140	2	7035	2	1538	353	4290	6181
17	57516	2	14379	24	1694	406	4544	6644
20	479568	4	29973	216	4271	1496	20236	26003

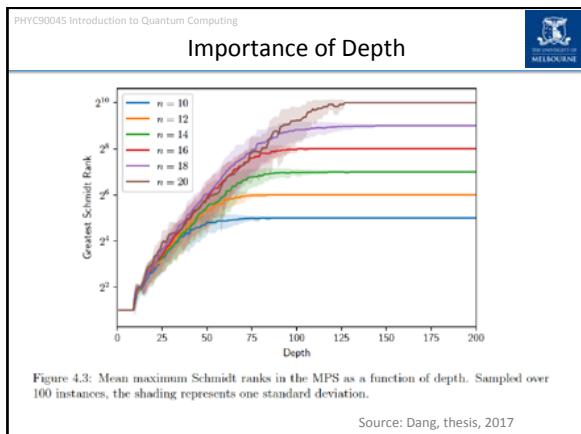
Table 3.2: Further QCMPs benchmarks, this time across multiple nodes of a supercomputer. Each node has 24 cores and 64 GB of RAM. With n_{node} nodes, we simulated the three cases $l = 16$, $N = 56759$, $a = 2$; $l = 17$, $N = 124631$, $a = 2$; and also $l = 20$, $N = 961307$, $a = 5$.

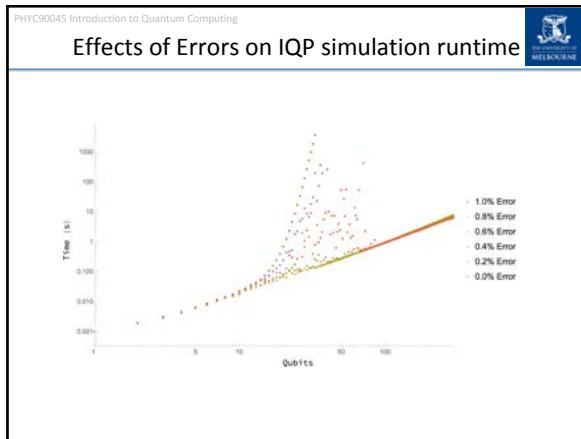
Source: Dang, thesis, 2017

PHYC90045 Introduction to Quantum Computing

Simulating Google's Pseudo-Random Circuits

Depth of the circuit is equivalent to the number of timesteps





PHYC90045 Introduction to Quantum Computing

What do you need for quantum supremacy?

- Many qubits (~ 50)
- Large depth circuit (~ 100)
- Entanglement, high T gate count
- Low error rates (<1%)

PHYC90045 Introduction to Quantum Computing

Week 6

Lecture 11 - Quantum Supremacy

- 11.1 Boson Sampling
- 11.2 IQP Problem
- 11.3 Google's pseudorandom circuits

Lecture 12 - Errors

- 12.1 Quantum errors: unitary and stochastic errors
- 12.2 Randomized Benchmarking
- 12.3 Tomography

Lab 6

Quantum Supremacy and Errors

PHYC90045 Introduction to Quantum Computing

Week 6



Lecture 11 - Quantum Supremacy

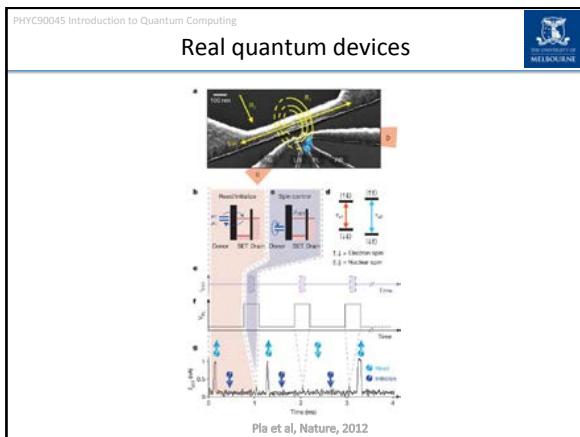
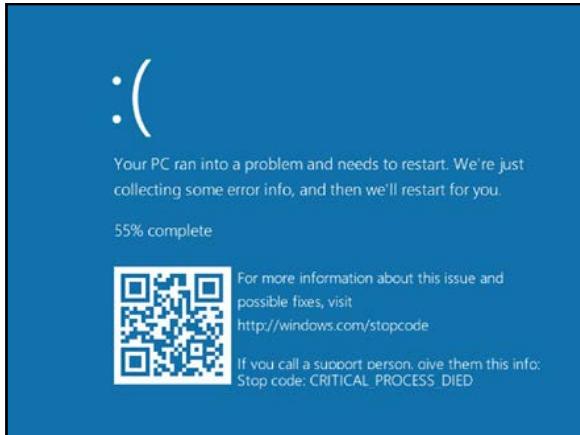
- 11.1 Boson Sampling
- 11.2 IQP Problem
- 11.3 Google's pseudorandom circuits

Lecture 12 - Errors

- 12.1 Quantum errors: unitary and stochastic errors
- 12.2 Purity
- 12.3 Tomography
- 12.4 Randomized Benchmarking

Lab 6

Quantum Supremacy and Errors



PHYC90045 Introduction to Quantum Computing

Two types of errors



Quantum computers are extremely fragile, and vulnerable to noise and errors. While errors occur in classical computing too, we're accustomed to very low error rates – our hard drives rarely forget what they store.

Two types of error:

- (1) **Systematic unitary errors** – eg. Control pulse error
- (2) **Random noise** – eg. Decoherence

PHYC90045 Introduction to Quantum Computing

Control Errors



Control of qubits requires high precision, and errors can sneak in. For example:

- Variations in magnetic fields across the sample, or variations in material properties.
- Stray electric fields, charge traps, strain.
- Applying a microwave pulse where the strength of the pulse is slightly too strong or too weak causes a systematic over-rotation or under-rotation.
- Cross-talk between gates.
- Unwanted interaction between qubits.

 IBM image, Flickr

PHYC90045 Introduction to Quantum Computing

Systematic Errors in the QUI



The QUI is effectively a pristine qubit environment, but we can introduce such effects systematically and investigate how quantum gate errors affect the output of quantum circuits.

We will consider rotation errors around the cartesian axes in the QUI using the R-gate. For example, a Z-rotation error (or just "Z-error") is a gate δZ defined as:

$$\delta Z \equiv \begin{pmatrix} e^{-i\epsilon/2} & 0 \\ 0 & e^{i\epsilon/2} \end{pmatrix}$$

where the level of error is governed by the angle ϵ (assumed to be small). Similarly, we could consider small rotations around other axes:

$$\delta X = R_X(\epsilon), \quad \delta Y = R_Y(\epsilon), \quad \delta Z = R_Z(\epsilon)$$

In the lab on Friday, we will consider the effect of these errors on the success of quantum circuits.

PHYC90045 Introduction to Quantum Computing

Decoherence

In realistic quantum systems there will always be (unwanted) interaction between the qubits and the environment (electrons, spins, phonons, charge traps).

This causes a type of noise on the system (ie. qubits we want to protect) which we call decoherence.

PHYC90045 Introduction to Quantum Computing

Modeling Stochastic Errors

Stochastic errors can be modeled in the QUI by randomly applying bit and phase flips to the qubits.

Dephasing noise: Apply Z gate with some probability p .
Depolarizing noise: Apply either X, Y or Z gates, each with with probability $p/3$.
 Perform many “Monte Carlo” simulations, where errors are placed randomly on each run, and average the measurement results.

PHYC90045 Introduction to Quantum Computing

Pure and Mixed States

Pure states have no errors, and are perfectly coherent.

For example, the state $|\psi\rangle = |0\rangle$ is a coherent, quantum state.

Mixed states may have errors, and are less coherent.

Imagine we took system in a pure quantum state, and noise – with say 20% probability- flipped the qubit around the X axis. That state would then be a “mixed” state.

PHYC90045 Introduction to Quantum Computing

Superposition vs mixed states

Consider the pure state,

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

and a mixed state which is

$$50\% |0\rangle \quad 50\% |1\rangle$$

Is it possible to tell these two states apart in experiment?

Consider what happens if we apply a Hadamard gate, then measure:

A quantum circuit diagram illustrating the effect of a Hadamard gate. It consists of three main components: a blue rectangular box labeled $|\psi\rangle$ representing an input state, a blue rectangular box labeled H representing the Hadamard gate, and a blue rectangular box with a curved arrow labeled $\langle \psi|$ representing the measurement operator. The circuit shows the flow of information from the input state through the Hadamard gate to the final measurement.

PHYC90045 Introduction to Quantum Computing

THE UNIVERSITY OF MELBOURNE

Superposition vs mixed states

For the mixed state if $|0\rangle$ were prepared :

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

50% of the time we will measure 0

50% of the time we will measure 1

For the mixed state if $|1\rangle$ were prepared :

$$H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

50% of the time we will measure 0

50% of the time we will measure 1

So if the mixed state is prepared:

50% of the time we will measure 0

50% of the time we will measure 1

For the pure state: $H|+\rangle = |0\rangle$ so **100%** of the time we will measure 0.

Purity on the Bloch Sphere

Pure states lie on the surface of the Bloch Sphere

PHYC90045 Introduction to Quantum Computing

Mixed states on the Bloch Sphere

Mixed states lie inside the Bloch Sphere.

The closer to the origin, the more mixed.

PHYC90045 Introduction to Quantum Computing

Purity for one qubit

If the distance from the origin to the state is measured to be r , the purity is:

$$P = \frac{1 + r^2}{2}$$

Maximum purity of 1 for all pure states.

Minimum purity of $\frac{1}{2}$ for a completely mixed state.

Note: There's a more technical definition of purity in terms of density matrices, which we won't cover in this course.

PHYC90045 Introduction to Quantum Computing

Reminder: Calculating expectation values

Example of calculating an expectation value for X ,

$$\begin{aligned} \langle X \rangle &= \langle \psi | X | \psi \rangle \\ &= (a^* \langle 0 | + b^* \langle 1 |) X (a | 0 \rangle + b | 1 \rangle) \\ &= a^* a \langle 0 | X | 0 \rangle + b^* b \langle 1 | X | 1 \rangle + a^* b \langle 0 | X | 1 \rangle + b^* a \langle 1 | X | 0 \rangle \\ &= a^* b + b^* a \end{aligned}$$

PHYC90045 Introduction to Quantum Computing



 The University of
 MELBOURNE

Example: 20% error

Eg. Consider a state $|0\rangle$ present in a noisy system. 20% of the time, a bit flip X has applied to it.

$$\begin{aligned}\langle X \rangle &= 0.80 \langle 0 | X | 0 \rangle + 0.2 \langle 1 | X | 1 \rangle \\ &= 0 + 0 \\ &= 0\end{aligned}$$

$$\begin{aligned}\langle Y \rangle &= 0.80 \langle 0 | Y | 0 \rangle + 0.2 \langle 1 | Y | 1 \rangle \\ &= 0 + 0 \\ &= 0\end{aligned}$$

$$\begin{aligned}\langle Z \rangle &= 0.80 \langle 0 | Z | 0 \rangle + 0.2 \langle 1 | Z | 1 \rangle \\ &= 0.8 - 0.2 \\ &= 0.6\end{aligned}$$

PHYC90045 Introduction to Quantum Computing

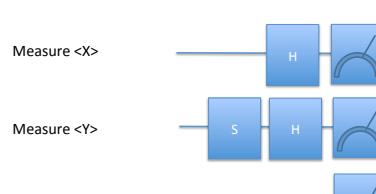
PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

Measuring purity in the QUI

Run many trials – on each trial choosing a different random set of errors.



The diagram illustrates three measurement scenarios:

- Measure $\langle X \rangle$:** A single blue box labeled "H" (representing a Hadamard gate) is followed by a meter symbol.
- Measure $\langle Y \rangle$:** Two blue boxes are shown in sequence: the first is labeled "S" (representing a phase shift gate), and the second is labeled "H" (representing a Hadamard gate). This is followed by a meter symbol.
- Measure $\langle Z \rangle$:** A single blue box representing a quantum circuit (labeled with a question mark) is followed by a meter symbol.

$$P = \frac{1 + \langle X \rangle^2 + \langle Y \rangle^2 + \langle Z \rangle^2}{2}$$

Then calculate the purity:

PHYC90045 Introduction to Quantum Computing

Quantum State Tomography

“Tomography” is quantum computing jargon for measuring/determining the quantum state, as well as possible. For one qubit, this is just measuring:

$$\langle X \rangle, \langle Y \rangle, \langle Z \rangle$$

For two qubits, we need to accurately measure correlations between the qubits as well. We measure the 15 parameters:

$$\langle XX \rangle, \langle XY \rangle, \langle XZ \rangle, \langle XI \rangle$$

$$\langle YX \rangle, \langle YY \rangle, \langle YZ \rangle, \langle YI \rangle$$

$$\langle ZX \rangle, \langle ZY \rangle, \langle ZZ \rangle, \langle ZI \rangle$$

$$\langle IX \rangle, \langle IY \rangle, \langle IZ \rangle$$

Because of counting statistics the states can be unphysical (eg. radius greater than 1). However, these measurements can be used to estimate the closest (mixed) quantum state.

PHYS90045 Introduction to Quantum Computing

THE UNIVERSITY OF
 MELBOURNE

Two qubit example

Measuring $\langle XX \rangle$:

A quantum circuit diagram showing two qubits. The top qubit starts with a unitary U , followed by a Hadamard gate H , and ends with a measurement in the $\{|+\rangle, |-\rangle\}$ basis. The bottom qubit starts with a Hadamard gate H , followed by a measurement in the $\{|+\rangle, |-\rangle\}$ basis.

Measurement results:

- $m_1 = \pm 1$
- $m_2 = \pm 1$

Find the product of these, $m = m_1 m_2$

Average over many runs of the experiment, with different locations/errors on each run to determine $\langle XX \rangle$.

PHYC90045 Introduction to Quantum Computing

Randomized Benchmarking

How good are our gates individual gate? We want a number for how much error doing each operation is. One way of determining this is to perform **randomized benchmarking**.

$|0\rangle$

X H $Z^{1/2}$ Y ... U^{-1}

Apply a random sequence of gates

Make a measurement:
Should measure 0

PHYC90045 Introduction to Quantum Computing

The Clifford Gates (for one qubit)

Typically this random sequence is chosen from a small gate set. One common choice is called the Clifford gates: These gates only rotate between the states which lie along +/- x, y and z axes.

$$|0\rangle, |1\rangle, \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \frac{|0\rangle - |1\rangle}{\sqrt{2}}, \frac{|0\rangle + i|1\rangle}{\sqrt{2}}, \frac{|0\rangle - i|1\rangle}{\sqrt{2}}$$

All of the preset gates except T are Clifford: X, Y, Z, S.

PHYC90045 Introduction to Quantum Computing

Randomized Benchmarking

At the last step we apply the inverse of the preceding sequence. So if we started in the state $|0\rangle$

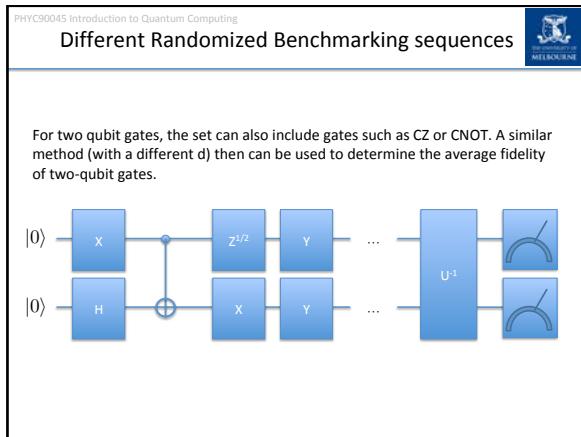
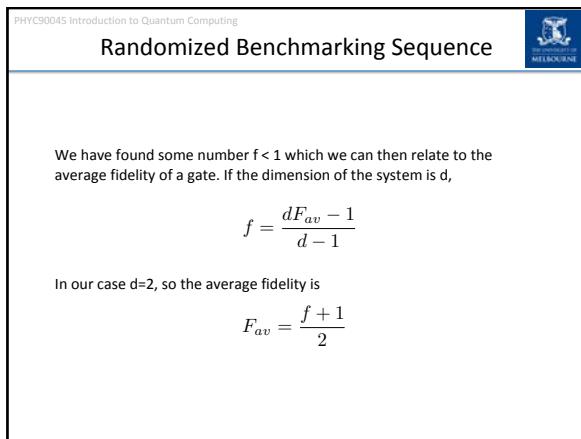
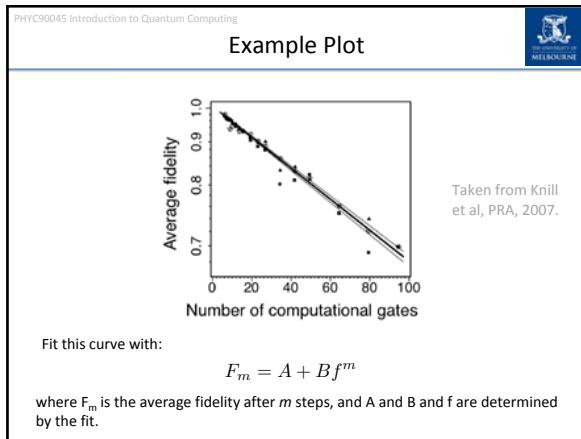
We should also end up in that state. If there were no errors, should measure 0, with certainty. However, with errors, the fidelity of the final measurement drops.

PHYC90045 Introduction to Quantum Computing

The Clifford Gates (for one qubit)

Repeat sequences of the same length many times. For each length of sequence, average over many runs of the sequences, to work out the probability of measuring the correct result.

We can plot the fidelity (i.e. the probability of getting the correct answer) against the length of sequence.



PHYC90045 Introduction to Quantum Computing

Interleaved Randomized Benchmarking

To determine the fidelity of an individual gate, interleave it throughout the randomized benchmarking. Eg. X

This gives an indication of the error in individual gates.

More advanced schemes also exist, eg. Adaptive versions of randomized benchmarking pinpoint where and what type of errors are occurring, rather than just giving a single number of average error per gate.

PHYC90045 Introduction to Quantum Computing

Quantum Process Tomography

Just as we can do tomography to determine a (mixed) quantum state, in principle we can measure what happens in a quantum process. Technically we are determining a completely positive (CP) map.

General strategy for one qubit

For each possible input states:

$$|0\rangle, |1\rangle, \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \frac{|0\rangle - |1\rangle}{\sqrt{2}}, \frac{|0\rangle + i|1\rangle}{\sqrt{2}}, \frac{|0\rangle - i|1\rangle}{\sqrt{2}}$$

Act the operation, U, on each input states

Do complete state tomography on each output (ie. $\langle X \rangle, \langle Y \rangle, \langle Z \rangle$)

Similar process for multiple qubits - QPT requires many measurements!

PHYC90045 Introduction to Quantum Computing

Week 6

Lecture 11 - Quantum Supremacy

- 11.1 Boson Sampling
- 11.2 IQP Problem
- 11.3 Google's pseudorandom circuits

Lecture 12 - Errors

- 12.1 Quantum errors: unitary and stochastic errors
- 12.2 Purity
- 12.3 Quantum state Tomography
- 12.3 Randomized Benchmarking

Lab 6

Quantum Supremacy and Errors



IBM

QUANTUM COMPUTING AT IBM

Anna Phan
September 10, 2019



\$6B R&D Budget
3000 Researchers
12 labs worldwide
Innovation That Matters
3 Nobel Prizes
6 Turing Awards
25 years of Patent Leadership

Almaden Yorktown

1955

Austin
1995

Dublin

2011

Zurich

1956

Haifa

1972

China

1995

India

1998

Tokyo

1962

Africa

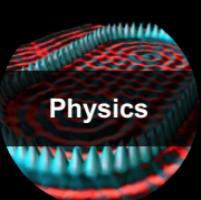
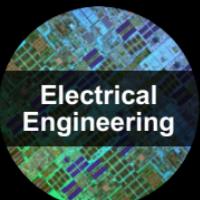
2012

Brazil

2010

Australia

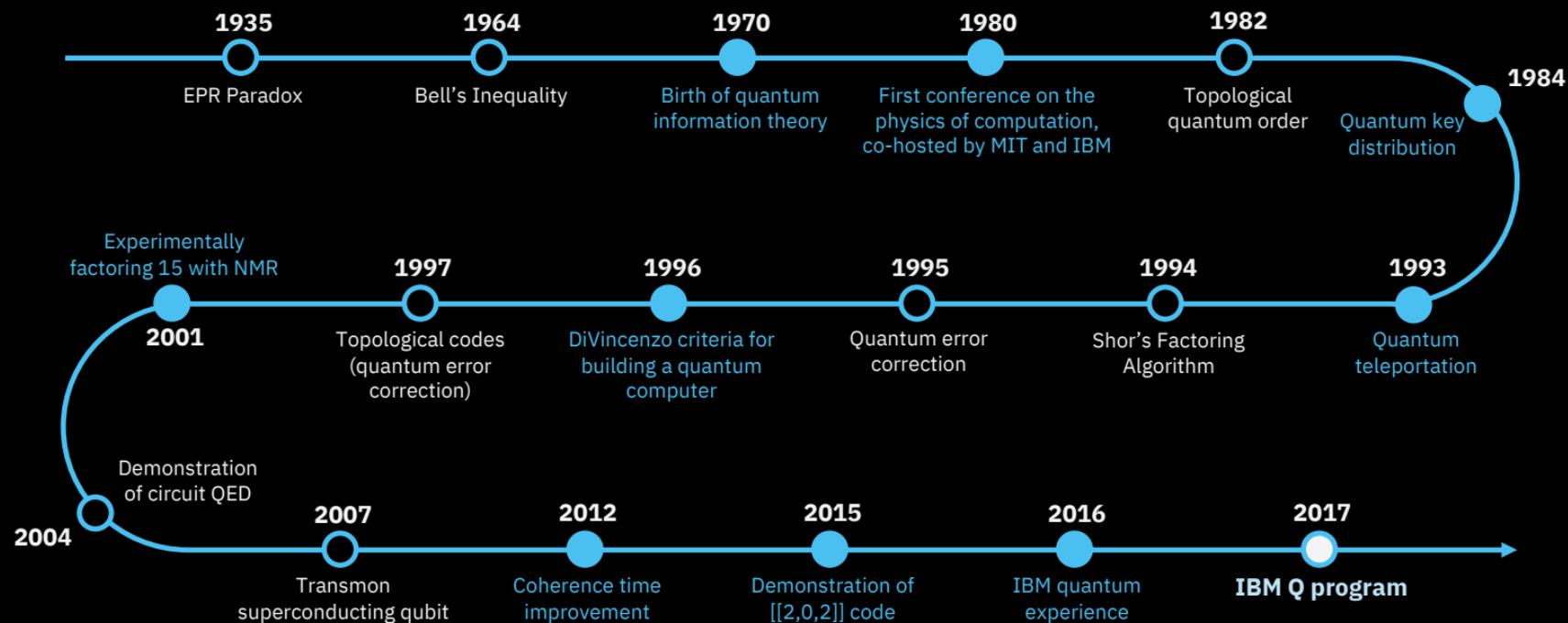
2011



IBM Research & Quantum Computing

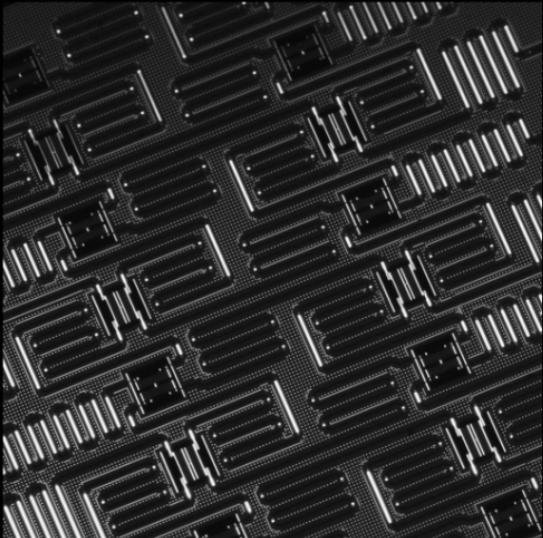
IBM Q

"Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy." - Richard Feynman, Physics of Computation Conference, co-hosted by MIT and IBM, 1981



IBM Quantum Computing Program

IBM Q



**Hardware
&
Engineering**

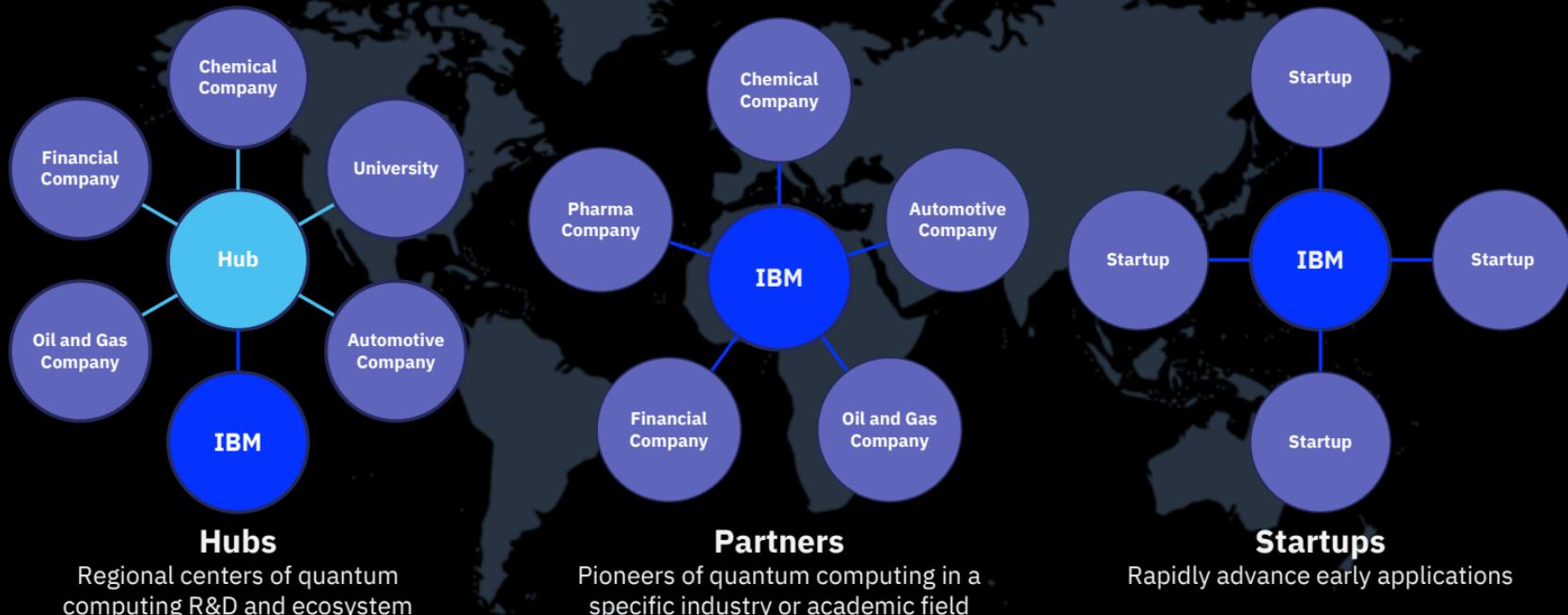


**Software
&
Ecosystem**



**Algorithms
&
Applications**

A collaboration with leading Fortune 500 companies and research institutions with the shared mission of **advancing quantum computing**, launching the **first commercial applications**, and **educating and preparing** the future workforce.



IBM Q Network

77 members

- 7 industry partners
- 9 hubs
- 15 members
- 20 startups
- 26 academic partners

Industry Partners

ExxonMobil

JP Morgan Chase & Co.

Samsung

Daimler

JSR Corporation

Accenture

US Air Force Research Lab

Hubs

The University of Melbourne

Oak Ridge National Laboratory

Keio University

NC State University

University of Oxford

University of Bundeswehr Munich

National Taiwan University

Iberian Nanotechnology Laboratory

CSIC Spain

Members

Barclays

Mizuho

MUFG

Mitsubishi Chemical

Argonne Lab

Fermilab

Berkeley Lab

Brookhaven Lab

ITRI

III Taiwan

CERN

University of Minho

Honda

Hitachi Metals

Nagase

Startups

QC Ware

Grid

Quemix

CQC

1QBit

Zapata

Strange Works

Q-CTRL

Quantum Benchmark

MDR

Qu&Co

JoS Quantum

SolidStateAI

ProteinQure

Labber Quantum

MaxKelsen

Netramark

Entropica

Boxcat

Rahko

Academic Partners

MIT

EDX.org

Virginia Tech

U. Montpellier

Notre Dame

Harvard

Princeton

Florida State

U. Stony Brook

U. Chicago

U. Tokyo

Duke

UC Boulder

U. Waterloo

U. Illinois

Northwestern

NYU

Wits

Aalto University

U. of Turku

U. Basque Country

U. of Innsbruck

EPFL

Chalmers University

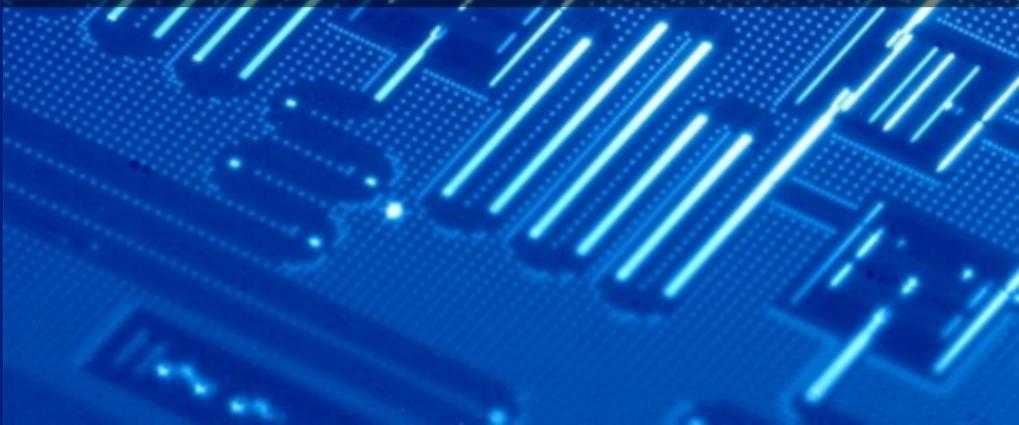
ETH Zurich

Saarland University



IBM

QUANTUM COMPUTING HARDWARE



Quantum computing technology examples

IBM Q

Classical Bits



Relays



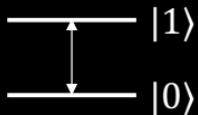
Vacuum
Tube



Transistor

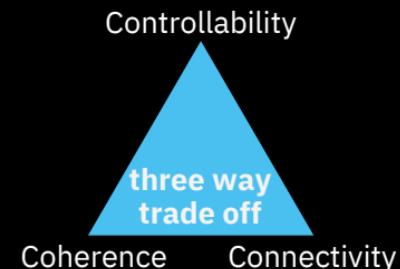
Quantum Bits

Two-Level Systems



Example:

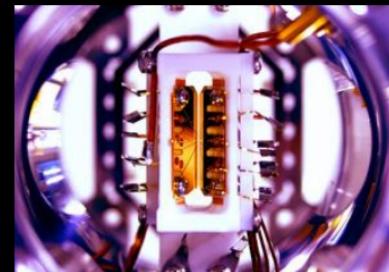
Atom orbitals with different energetic levels



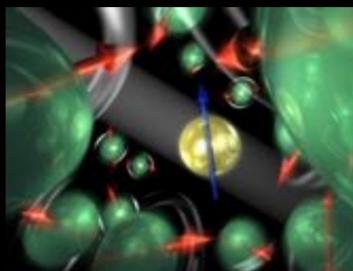
Photons



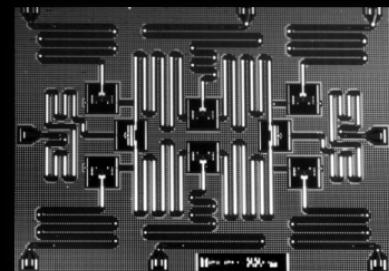
Trapped Ions



Solid State Defects



Superconducting Circuits



Superconducting quantum processor

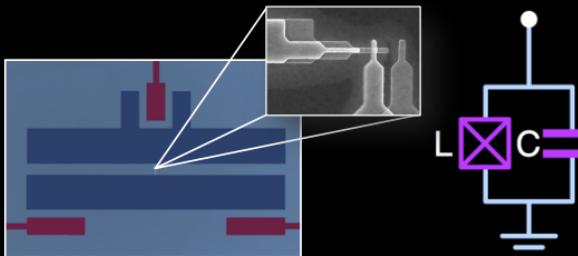
IBM Q

≈ 5 mm



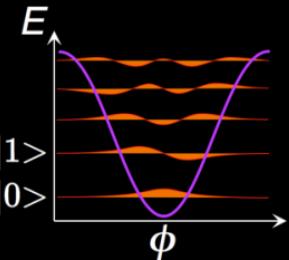
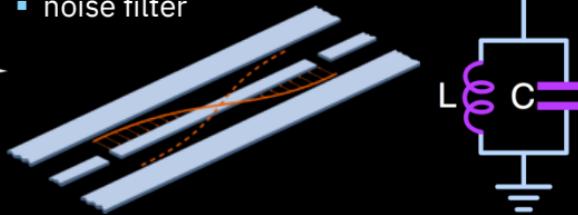
Superconducting transmon qubit:

- non-linear Josephson Junction (inductance)
- anharmonic energy spectrum => qubit
- nearly dissipationless

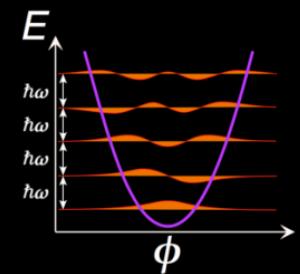


Microwave resonator as:

- read-out of qubit states
- multi-qubit quantum bus
- noise filter



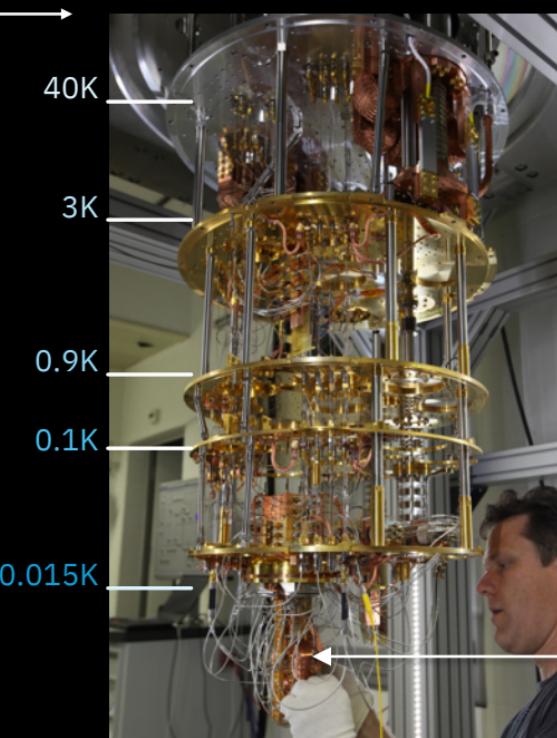
$$E_{01} \approx 5 \text{ GHz} \approx 240 \text{ mK}$$



Challenging engineering environment

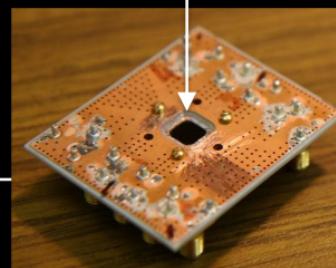
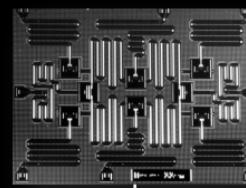
IBM Q

Microwave electronics



Refrigerator to cool qubits to 10 - 15 mK with a mixture of ^3He and ^4He

Chip with superconducting qubits and resonators

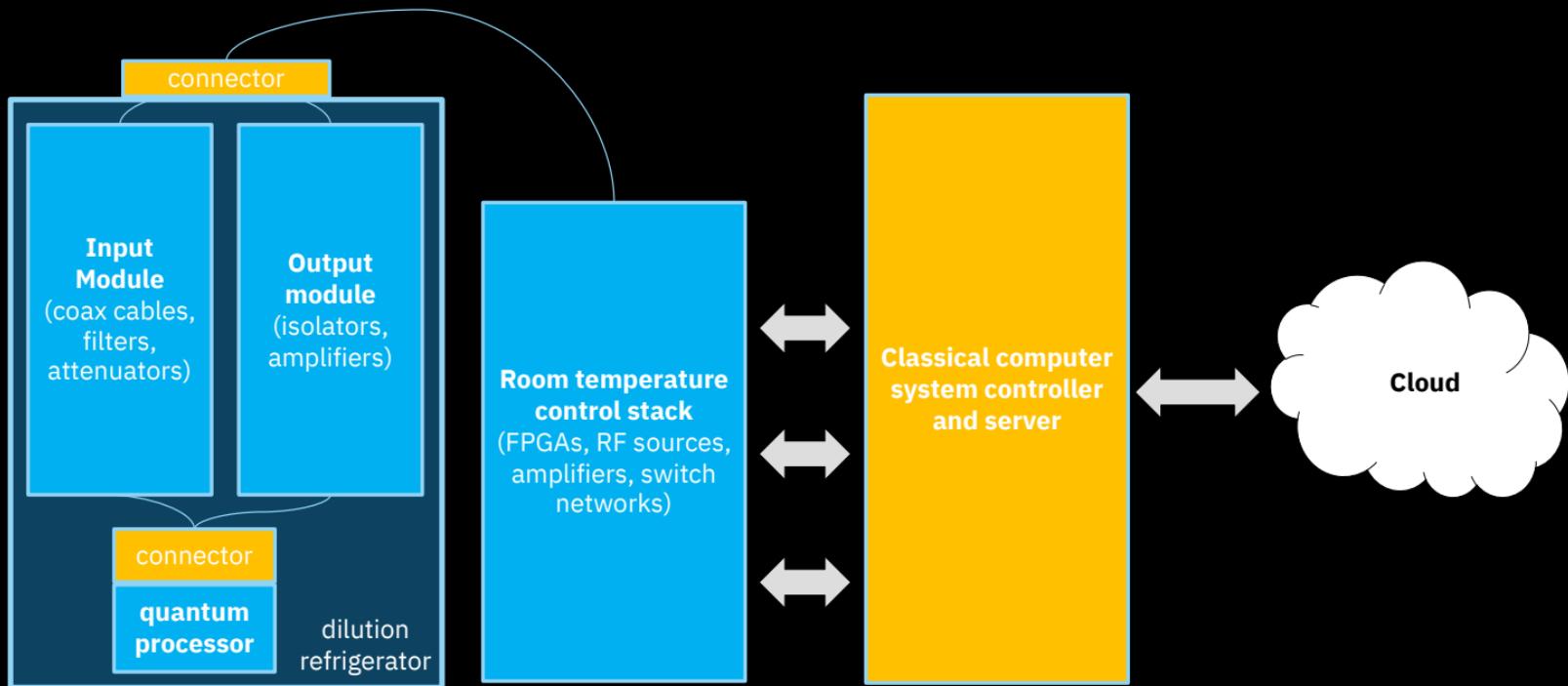


Printed circuit board with the qubit chip at 15 mK

Protected from the environment by multiple shields

The complete system

IBM Q

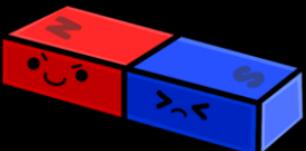
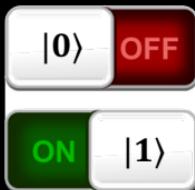




IBM

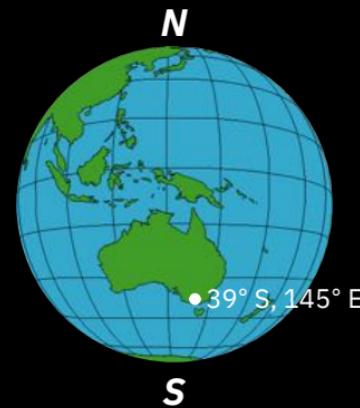
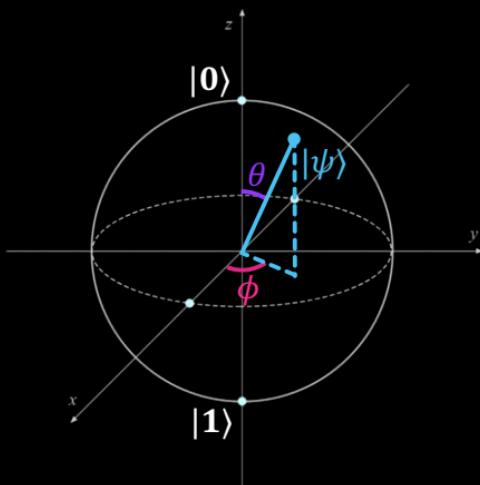
QUANTUM COMPUTING SOFTWARE

Classical Bits:



Quantum Bits:

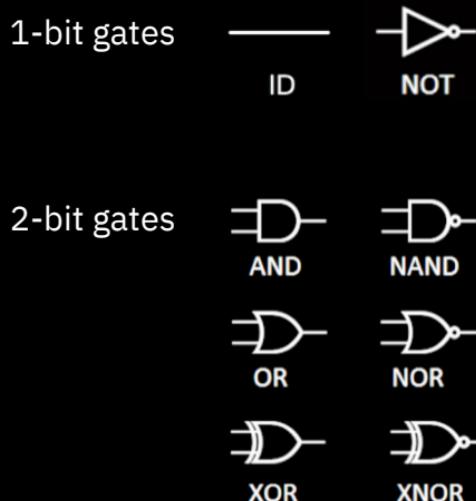
- The state of a qubit, $|\psi\rangle$, can be an arbitrary point on the surface of a sphere.
- The state of a qubit is therefore defined by two angles like longitude, θ , and latitude, ϕ , on a globe.



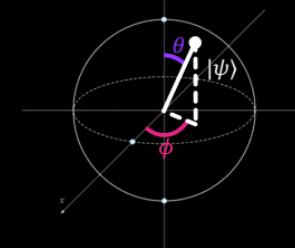
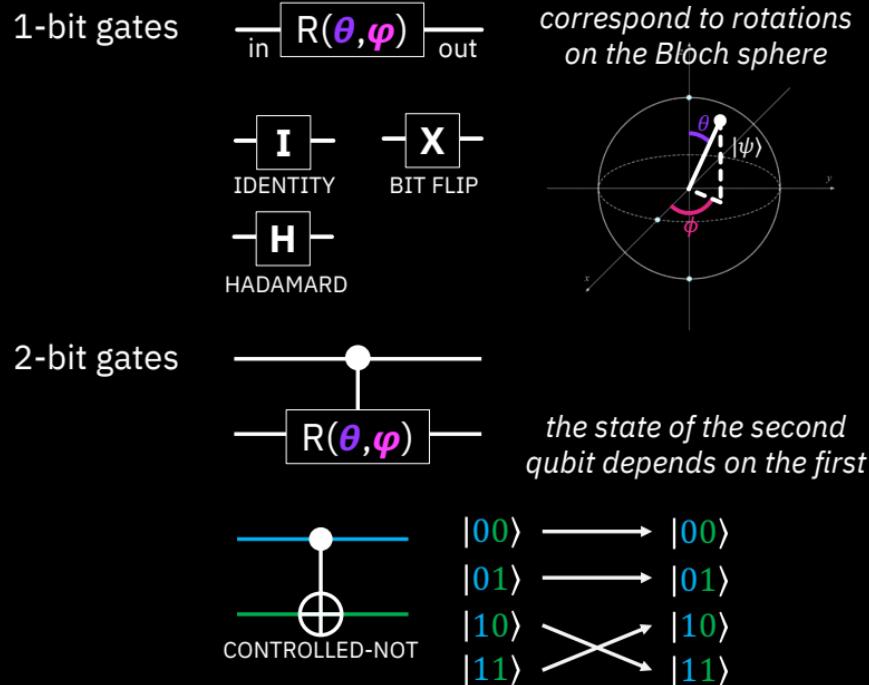
Classical and Quantum Logic

IBM Q

Classical logic



Quantum logic



Classical and Quantum Computation

IBM Q

C++, Java, Python, Swift, SQL,
Javascript, Ruby, PHP, Go, R,
Scala, Rust, Julia, Haskell ...

HIGH-LEVEL
PROGRAMMING
LANGUAGE

?

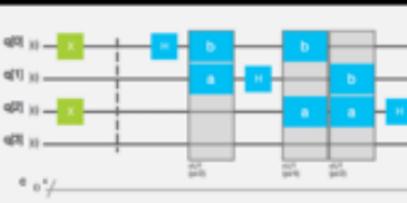
LOW-LEVEL PROGRAMMING LANGUAGE

```
449a: ff40 4e00 0400 mov.b    #0x4a, 0x4(r15)
44a0: ff40 2f00 0500 mov.b    #0x2f, 0x5(r15)
44a6: ff40 2900 0600 mov.b    #0x29, 0x6(r15)
44ac: cf43 0700 mov.b    #0x0, 0x7(r15)
44b0: 3041 ret
44b2 <get_password>
44b2: 3e40 6400 moy    #0x64, r14
44b6: b012 8445 call    #0x4584 <getan>
44ba: 3041 ret
```

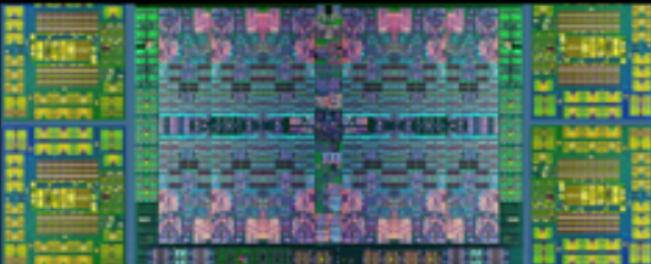
```

INMQASM 2.0;
include "qelib1.inc";
greg q[4];
cgreg c[4];
x q[0];
x q[1];
barrier q;
h q[0];
cui(p0/2) q[1],q[0];
h q[1];
mult(q0/4),q[2],q[0];

```



HARDWARE



- First quantum computers on the cloud
- **150k+** users
- On all **7** continents
- **10M+** experiments
- **200+** external research papers



Qiskit is an open source software development kit providing libraries, documentation, a simulator, and connections to IBM Q devices



Quantum Experience Devices

IBM Q

- **5 qubit device (ourense)**

- T configuration
- 4 bidirectional CNOTS available



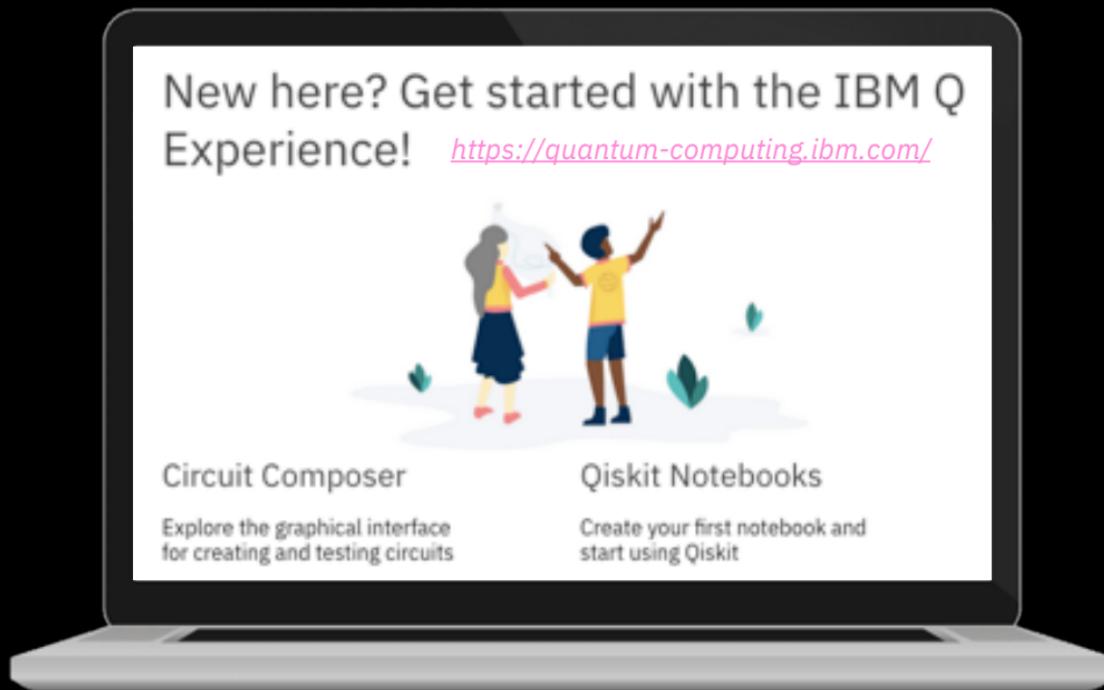
- **14 qubit device (melbourne)**

- Ladder configuration
- 18 unidirectional CNOTS available
- Qiskit API access only



Demonstration

IBM Q



References



- Quantum Experience: <https://quantum-computing.ibm.com/>
- Quantum Experience Device Information: <https://github.com/Qiskit/qiskit-backend-information/>
- OpenQASM: <https://github.com/Qiskit/openqasm>
- Qiskit: <https://qiskit.org/>
- Qiskit GitHub: <https://github.com/Qiskit>
- Hello Quantum: <http://helloquantum.mybluemix.net/>
- Hello Quantum Blog Post: <https://medium.com/qiskit/hello-quantum-2c1c00fe830c>

Q

IBM



THANK YOU

PHYC90045 Introduction to Quantum Computing

Week 7

Lecture 13 – Introduction to IBM Quantum Experience
Introduction to IBM Quantum Experience: Guest Lecture

Lecture 14 – IBM and Optimizations
14.1 Rotation operators: QUI and IBM conversion
14.2 QASM and QISKit
14.3 Optimizing circuits

Lab 7
Using the IBM Q system

PHYC90045 Introduction to Quantum Computing

IBM Q system and Optimization

Physics 90045
Lecture 14

PHYC90045 Introduction to Quantum Computing

The IBM Q System

quantum-computing.ibm.com

IBM Q Experience

Sign in to IBM Q Experience

What is IBM Q Experience? [Learn more](#)

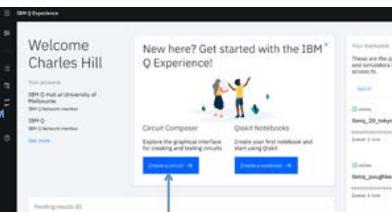
[IBM](#) [GitHub](#) [Google](#) [G+](#) [GitHub](#) [Twitter](#) [LinkedIn](#) [Email](#)

[IBM Q](#) [Privacy](#) [Terms of Use](#) [IBM Q End User Agreement](#) [IBM Q Privacy Policy](#) [Cookie Preferences](#) v1.2.2

Sign up using your university email before Thursday/Friday!

PHYC90045 Introduction to Quantum Computing

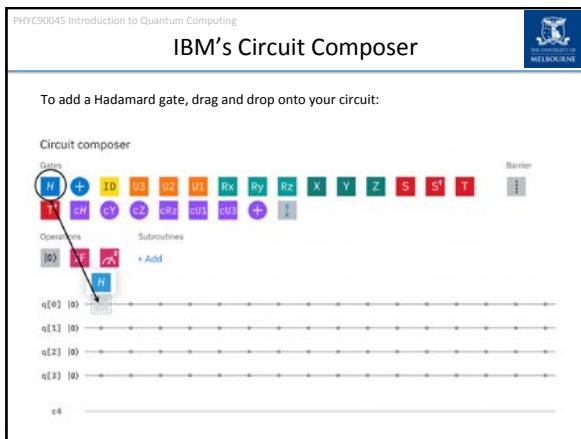
Starting a new circuit



Can also access circuit composer through menu here:



Click here to create a new circuit



PHYC90045 Introduction to Quantum Computing



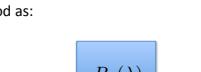
 THE UNIVERSITY OF
 MELBOURNE

U1

U1 is a rotation around Z by angle lambda, which is equivalent to a rotation around the z-axis by an angle lambda

$$U_1 = \begin{bmatrix} 1 & 0 \\ 0 & \exp i\lambda \end{bmatrix}$$

Most easily understood as:



In the QUI, to emulate these z-rotations, use a global phase of lambda/2.
 No global phase for the y-rotation.

PHYC90045 Introduction to Quantum Computing



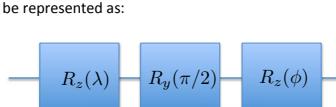
 THE UNIVERSITY OF
 MELBOURNE

U2

The U2 operation is given by

$$U_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -\exp(i\lambda) \\ \exp(i\phi) & \exp(i\lambda + i\phi) \end{bmatrix}$$

Which can be represented as:



```

    graph LR
      W1(( )) --> Rz1[R_z(<math>\lambda</math>)]
      Rz1 --> Ry1[R_y(<math>\pi/2</math>)]
      Ry1 --> Rz2[R_z(<math>\phi</math>)]
      Rz2 --- W2(( ))
  
```

In the QUI, to emulate these z-rotations, use a global phase of theta/2.
 No global phase for the y-rotation.

PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

U3

The matrix of a U3 rotation is:

$$U_3 = \frac{1}{\sqrt{2}} \begin{bmatrix} \cos \theta/2 & -\exp(i\lambda) \sin(\theta/2) \\ \exp(i\phi) \sin(\theta/2) & \exp(i\lambda + i\phi) \cos(\theta/2) \end{bmatrix}$$

As a circuit:



```

    graph LR
      A(( )) --> B[R_z(λ)]
      B --> C[R_y(θ)]
      C --> D[R_z(ϕ)]
  
```

PHYC90045 Introduction to Quantum Computing

Euler Angle Decomposition



PHYC90045 Introduction to Quantum Computing

Converting to and from Euler angles

General form of arbitrary rotation about an unit axis $n=(n_x, n_y, n_z)$:

$$R_n(\alpha) = \cos \frac{\alpha}{2} I - i \sin \frac{\alpha}{2} \hat{n} \cdot \sigma$$

$$= \begin{bmatrix} \cos \frac{\alpha}{2} - i n_z \sin \frac{\alpha}{2} & \sin \frac{\alpha}{2} (-i n_x - n_y) \\ \sin \frac{\alpha}{2} (-i n_x + n_y) & \cos \frac{\alpha}{2} + i n_z \sin \frac{\alpha}{2} \end{bmatrix}$$

Euler angle rotations (with global phase = 0):

$$U_3 = \begin{bmatrix} e^{-i(\lambda+\phi)/2} \cos(\theta/2) & -e^{i(\lambda-\phi)/2} \sin(\theta/2) \\ e^{i(-\lambda+\phi)/2} \sin(\theta/2) & e^{i(\lambda+\phi)/2} \cos(\theta/2) \end{bmatrix}$$

Write out the matrix and equate elements.

PHYC90045 Introduction to Quantum Computing

Product of single qubit unitaries



Euler angle rotations (with global phase = 0):

$$U_3 = \begin{bmatrix} e^{-i(\lambda+\phi)/2} \cos(\theta/2) & -e^{i(\lambda-\phi)/2} \sin(\theta/2) \\ e^{i(-\lambda+\phi)/2} \sin(\theta/2) & e^{i(\lambda+\phi)/2} \cos(\theta/2) \end{bmatrix}$$

Write out the matrix and equate elements.

PHYC90045 Introduction to Quantum Computing

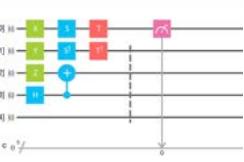
QASM – Quantum Assembly language

Backend: ibmqx4 • My Units: 15 • Experiment Units: 3

```

1 #include "qelib1.inc"
2 qreg q[5];
3 creg c[5];
4
5 x q[0];
6 y q[1];
7 z q[2];
8 h q[3];
9
10 s q[0];
11 msdg q[1];
12
13 cx q[3],q[2];
14 t q[0];
15 tdg q[1];
16
17 barrier q[1],q[2],q[3],q[4];
18 measure q[0] -> c[0];
19

```



Import QASM Download QASM

PHYC90045 Introduction to Quantum Computing

QASM Syntax

The diagram illustrates the QASM syntax with annotations:

- Semi-Colons:** Points to the semicolon at the end of line 1.
- Comment:** Points to the double slashes in line 3.

```

1 include "qelib1.inc";
2 // This is a comment
3 qreg q[5];
4 creg c[5];
5
6 x q[0];
7 y q[1];
8 z q[2];
9 h q[3];
10 s q[0];
11 sdg q[1];
12
13 cx q[3],q[2];
14 t q[0];
15 tdg q[1];
16
17 barrier q[1],q[2],q[3],q[4];
18 measure q[0] -> c[0];
19
20
21

```

PHYC90045 Introduction to Quantum Computing

QASM

The diagram illustrates the OPENQASM 2.0 syntax with annotations:

- Include standard definitions:** Points to the first line of code.
- Declare quantum register:** Points to line 2.
- Declare classical register:** Points to line 3.
- Single qubit gates:** Points to lines 5-8.
- CNOT gate (control first parameter, target second):** Points to line 13.
- Dagger indicated by "dg":** Points to line 15.
- Barrier (don't optimize across it):** Points to line 17.
- Measure qubits to classical register:** Points to line 18.

```

Hidden first line: OPENQASM 2.0;
1 include "qelib1.inc";
2 qreg q[5];
3 creg c[5];
4
5 x q[0];
6 y q[1];
7 z q[2];
8 h q[3];
9
10 s q[0];
11 sdg q[1];
12
13 cx q[3],q[2];
14 t q[0];
15 tdg q[1];
16
17 barrier q[1],q[2],q[3],q[4];
18 measure q[0] -> c[0];
19

```

PHYC90045 Introduction to Quantum Computing

Defining a new Function/Gate

The diagram shows the definition of a new gate:

```

Keyword: gate
    // Controlled Phase gate
    gate cz a,b
    {
        h b;
        cx a,b;
        h b;
    }

```

Annotations explain:

- "cz" is name of gate
- a and b are parameters

This gate can then be used like a native gate:

```

cz q[3],q[2];

```

PHYC90045 Introduction to Quantum Computing

QASM Header File



```

// Quantum Experience (QE) Standard header
// File: qelib1.inc

// --- QE Hardware primitives ---

// 3-parameter 2-pulse single qubit gate
gate u3(theta,phi,lambda) a { u(theta,phi,lambda) a; }

// 2-parameter 1-pulse single qubit gate
gate u2(theta,lambda) a { u3(theta,pi/2,lambda) a; }

// 1-parameter R-pulse single qubit gate
gate u1(lambda) a { u(0,0,lambda) a; }

// controlled-NOT
gate cx c,t { cx c,t; }

// idle gate (identity)
gate id a { u(0,0,0) a; }

// --- QE Standard Gates ---

// Pauli gate: bit-flip
gate x a { u3(0,0,pi) a; }

// Pauli gate: bit and phase flip
gate y a { u3(0,pi/2,pi/2) a; }

// Pauli gate: phase flip
gate z a { u3(0,pi,0) a; }

// Clifford gate: Hadamard
gate h a { u2(0,pi) a; }

// Clifford gate: square-root phase gate

```

Also defines:

Rotations
RX, RY, RZ

Toffoli
CCX

Controlled rotations
cu1, cu2, cu3, crz, ch

PHYC90045 Introduction to Quantum Computing

QISKit



Secure https://qiskit.org

Qiskit

An open-source quantum computing framework for leveraging today's quantum processors and conducting research

Logout Join the Slack community Try it out

Introducing VSCode extension!

Simplifying Qiskit to make developing quantum circuits and applications faster.

More information

Getting started with Qiskit

In this episode Doug McClure, Qiskitter at IBM, introduces us to Qiskit and its functions. You'll learn all about how to run your first quantum program on real IBM Q hardware.

Lots of examples in the github repository.

PHYC90045 Introduction to Quantum Computing

QISKit



There is also a Python interface to IBM Quantum Experience.

It is required to make use of the larger machines.

You can:

- Authenticate with the system
- Construct circuits (ie. python which translates to QASM)
- Submit jobs, and check for results
- Receive the results of jobs

Python works well with Jupyter interface.

We will use this later when we use the 16 qubit quantum computer.

PHYC90045 Introduction to Quantum Computing



Python Primer (if required)

PHYC90045 Introduction to Quantum Computing



Some Python Basics

```
In [2]: a=6
       b=7
       life = a*b
       life
Out[2]: 42
```

Similar to many other imperative languages you may know for numerical work:
(C/C++, MATLAB, R, FORTRAN, Julia) and often used for data processing.

PHYC90045 Introduction to Quantum Computing



Defining and calling Functions

```
def square(x):
    # This is a comment
    return x*x
```

def keyword indicates a new function
Whitespace is significant in python.
Indentation indicates a new block.

No types on parameters
No semicolons.
Newline is the end of a statement

Colon
Comment

Calling a function:

```
square(4)
square(x=4)
```

Named parameters

PHYC90045 Introduction to Quantum Computing

Lists and for loops



PHYC90045 Introduction to Quantum Computing

THE UNIVERSITY OF MELBOURNE

Dictionaries

Dictionaries store key-value pairs.

Curly braces indicate a dictionary

```
me = {"name": "Charles", "height":1.79, "favourite_food": "pizza"}  
me["favourite_food"]  
  
'pizza'
```

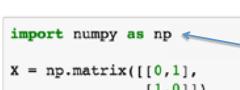
key

value

```
me["favourite_food"] = "sweet and sour pork"  
me["favourite_food"]  
  
'sweet and sour pork'
```

PHYC90045 Introduction to Quantum Computing

Importing other libraries



The diagram shows a box containing Python code:

```
import numpy as np
X = np.matrix([[0,1],
               [1,0]])
```

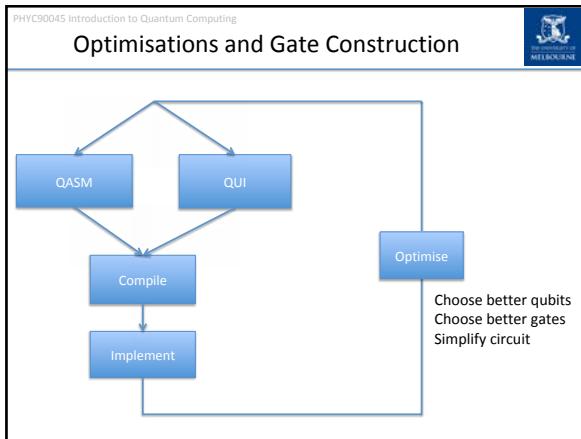
Annotations explain the code:

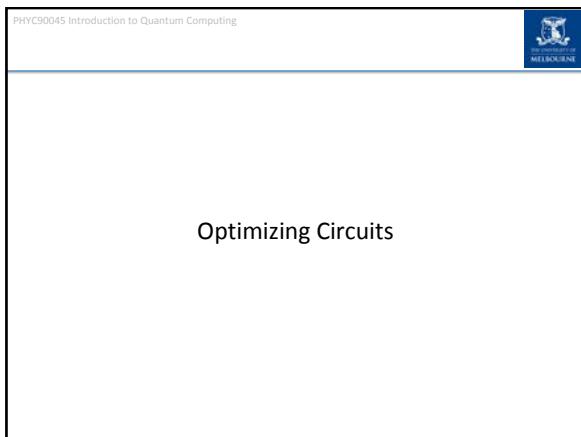
- An arrow points to the line `import numpy as np` with the text "Importing a module ("as np" is optional). numpy gives similar functionality to MATLAB".
- An arrow points to the line `X = np.matrix([[0,1], [1,0]])` with the text "Calling functions from that module. Here creating an X matrix."

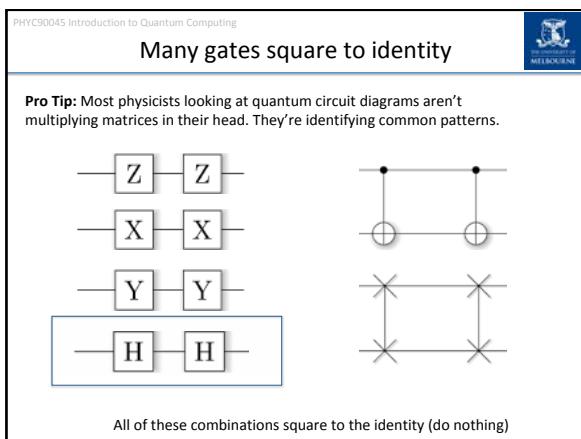
Or import individual functions and classes:

```
from qiskit import QuantumProgram
from qiskit import available_backends, execute, get_backend, compile
from qiskit import QuantumCircuit, ClassicalRegister, QuantumRegister, QISKitError
```

qiskit is an Python library/API for interacting with IBM's quantum computers remotely.



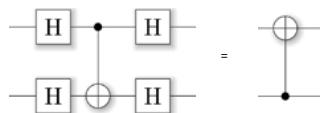




PHYC90045 Introduction to Quantum Computing

Circuit identity: Inverted CNOT

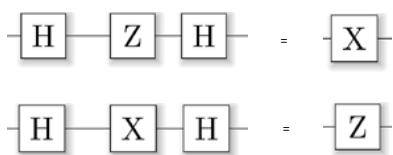
The diagram illustrates the circuit identity for an inverted CNOT gate. On the left, a standard CNOT gate is followed by Hadamard gates H on both qubits. On the right, the resulting circuit is shown as a single controlled phase gate with a phase of -1.



Exercise: You can verify this by writing out the matrices and multiplying!



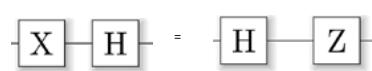
PHYC90045 Introduction to Quantum Computing

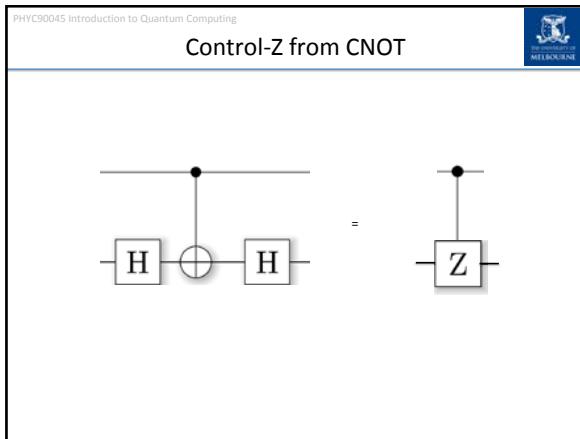


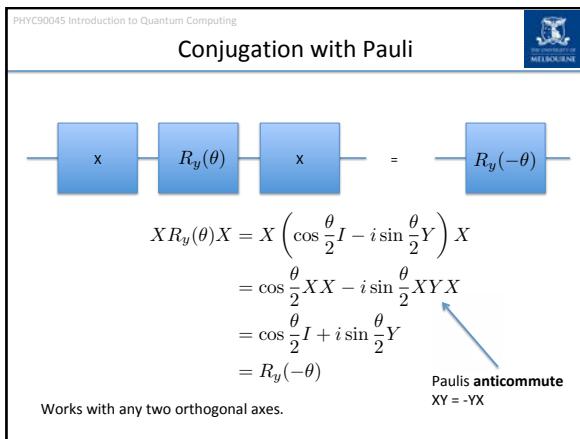
PHYC90045 Introduction to Quantum Computing

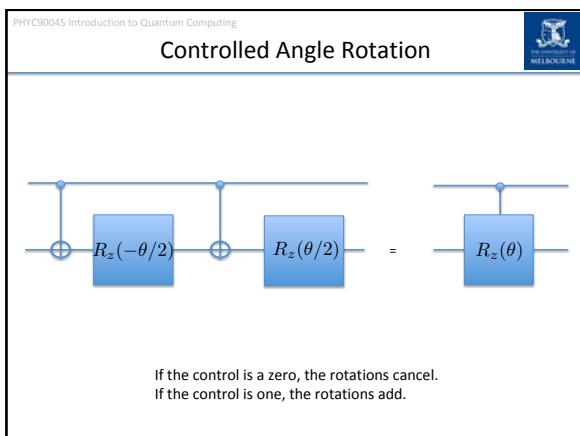
Commuting through Hadamard

The diagram illustrates two commutation relations between the Hadamard (H) gate and other single-qubit gates (X and Z).
Top row: $XH = HZ$
Bottom row: $ZH = ZX$







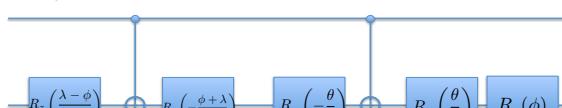


PHYC90045 Introduction to Quantum Computing

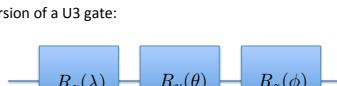


Any Controlled U

For U_3 Euler angle rotation (on IBM's system):



Controlled version of a U_3 gate:



PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

Conjugation with Rotation

Conjugation with a rotation:



Changes the axis of rotation, but not the rotation angle.

$$\begin{aligned}
 SR_x(\theta)S^\dagger &= \cos\left(\frac{\theta}{2}\right)I - i\sin\left(\frac{\theta}{2}\right)SX S^\dagger \\
 &= \cos\left(\frac{\theta}{2}\right)I - i\sin\left(\frac{\theta}{2}\right)Y \\
 &= R_y(\theta)
 \end{aligned}$$

This rotates
the axis itself

Conjugation with Hadamard is a special case of this.

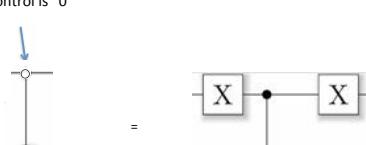
PHYC90045 Introduction to Quantum Computing



 The University of
MELBOURNE

Control from “0” state

Open circle =
 Only apply when
 the control is “0”



The diagram illustrates the decomposition of a CNOT gate with a control of 0. On the left, a standard CNOT gate symbol is shown with its control circle open. An arrow points to this open circle with the label "Only apply when the control is ‘0’". To the right of an equals sign, the CNOT gate is shown as a sequence of three operations: first, a single-qubit rotation gate labeled "X" (represented by a square box) is applied to the control qubit; second, a two-qubit CNOT gate (represented by a square box with a black dot in the center) acts on both qubits; third, another single-qubit rotation gate labeled "X" is applied to the target qubit.

We've seen this trick in labs: for example in the oracle for Grover's algorithm.

PHYC90045 Introduction to Quantum Computing

Swap gate from three CNOTs

Let's check:

- 00 -> 00
- 01 -> 10
- 10 -> 01
- 11 -> 11

PHYC90045 Introduction to Quantum Computing

Square root of SWAP

SWAP

$$U_{Swap} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Square root of SWAP

$$U_{SS} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1+i}{2} & \frac{1-i}{2} & 0 \\ 0 & \frac{1-i}{2} & \frac{1+i}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

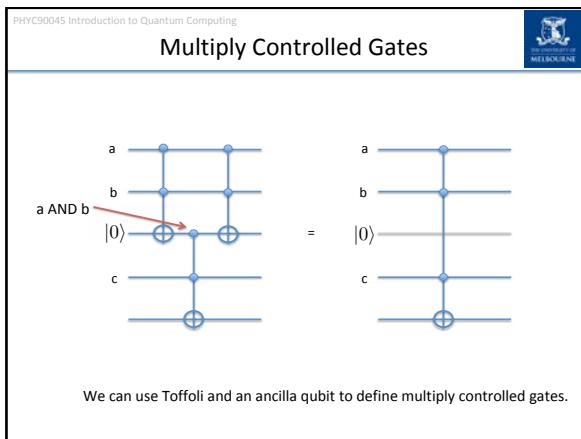
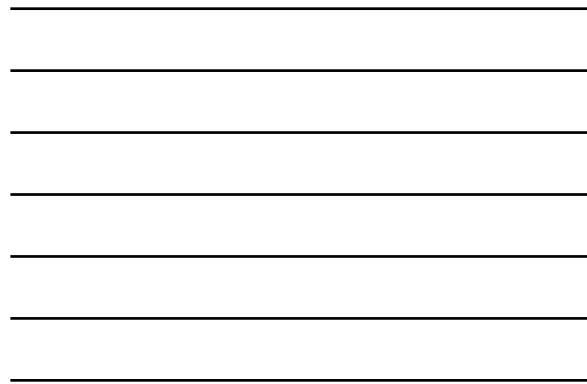
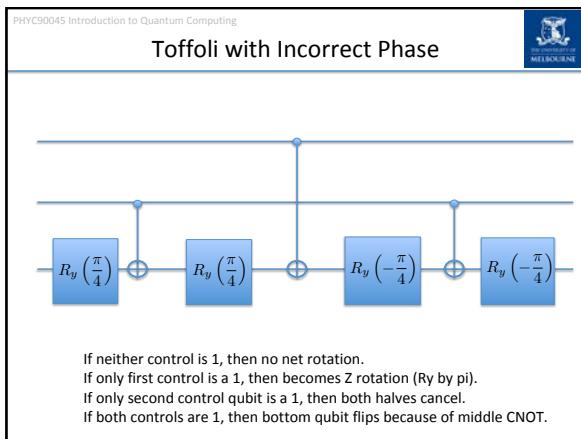
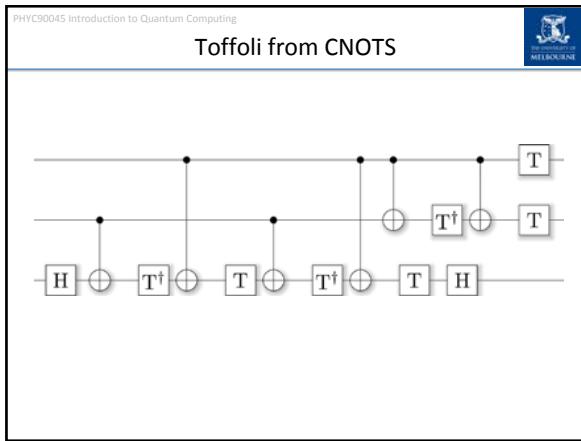
PHYC90045 Introduction to Quantum Computing

Square Root Swap Construction

$R_x\left(\frac{\pi}{2}\right)$

Similar to SWAP

More general version of Gray Code construction.



PHYC90045 Introduction to Quantum Computing

Mocking up gates

If you use controlled operations on all qubits except the target, then you isolate a single 2x2 subspace, with the rest of the matrix untouched.

$$CU = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & c & d \end{bmatrix}$$

Or controlled off the zero state:

$$C_0U = \begin{bmatrix} a & b & 0 & 0 \\ c & d & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

PHYC90045 Introduction to Quantum Computing

Using Gray codes

Imagine we wanted a 2x2 matrix between the 000 and 111 states:

A	B	C
0	0	0
0	0	1
0	1	1
1	1	1

Gray code

Corresponding sequence:

In this way, complicated multi-qubit gates can be built, piece by piece.

PHYC90045 Introduction to Quantum Computing

Week 7

Lecture 13 – Introduction to IBM Quantum Experience
Introduction to IBM Quantum Experience: Guest Lecture

Lecture 14 – IBM and Optimizations

- 14.1 Rotation operators: QUI and IBM conversion
- 14.2 QASM
- 14.3 Optimizing circuits

Lab 7
Using the IBM Q system

PHYC90045 Introduction to Quantum Computing

Week 8

Lecture 15
Simple classical error correction codes, Quantum error correction codes, stabilizer formalism, 5-qubit code, 7-qubit Steane code

Lecture 16
The more advanced quantum error correction codes, Fault Tolerance, surface code.

Lab 8
Quantum error correction

PHYC90045 Introduction to Quantum Computing

Introduction to Quantum Error Correction

Physics 90045
Lecture 15

PHYC90045 Introduction to Quantum Computing

Overview

This lecture we will introduce error correction for quantum computers:

- Overview of need for quantum error correction
- Simple classical error correction codes
- Quantum error correction codes
- The stabilizer formalism
- The five qubit code and seven qubit Steane code

Reiffel, Chapter 11
Kaye, Chapter 10
Nielsen and Chuang, Chapter 10

PHYC90045 Introduction to Quantum Computing

Decoherence and control errors

Qubit: Bloch sphere (or fragile "quantum bubble")

Decoherence: interaction with environment affects quantum state/operation

Even if you get decoherence under control...

Control: imprecise control leads to error in quantum state/operation

Impact of errors on fidelity of Shor's quantum factoring algorithm at logic level:

qubits

circuit time

Decomposed QFT

Measure

PHYC90045 Introduction to Quantum Computing

Decoherence and control errors

Qubit: Bloch sphere (or fragile "quantum bubble")

Decoherence: interaction with environment affects quantum state/operation

Even if you get decoherence under control...

Control: imprecise control leads to error in quantum state/operation

Impact of errors on fidelity of Shor's quantum factoring algorithm at logic level:

qubits

circuit time

Some error locations are very sensitive...it doesn't take much to rattle an algorithm...

Even after reducing physical errors, a quantum computer needs error correction...

PHYC90045 Introduction to Quantum Computing

Classical Error Correction

The simplest example of a classical error correction code is a repetition code:

$0 \rightarrow 000$	Logical "0"
$1 \rightarrow 111$	Logical "1"
"Codewords"	

If an error occurs (ie. bit flip) then using the *redundant information* we can still correct by simply taking the majority:

$0 \left\{ \begin{array}{l} 000 \\ 001 \\ 010 \\ 100 \end{array} \right.$	$1 \left\{ \begin{array}{l} 111 \\ 110 \\ 101 \\ 011 \end{array} \right.$
---	---

With one error, we can correct the error and continue the computation.

PHYC90045 Introduction to Quantum Computing

Code distance

The **distance** of the code is the (minimum) number of logical errors between codewords.

3 bit-flips takes 000 to 111, so the distance of the 3-bit repetition code is 3.

For classical codes, the distance is simply the minimum Hamming distance between any two codewords.

Often (for linear codes) you will see the notation:

The three-bit repetition code is a [3, 1, 3] code.

PHYC90045 Introduction to Quantum Computing

Code failure

Too many errors can overwhelm an error correction code. For example if we have two distinct errors on the codeword, 000:

$$000 \rightarrow 101$$

Which we would (wrongly) decode as “1”.

A distance d code can correct $\left\lfloor \frac{d-1}{2} \right\rfloor$ errors.

PHYC90045 Introduction to Quantum Computing

Quantum Error Correction

Similar to classical error correction codes, we can have a quantum repetition code:

$$\begin{aligned} |0\rangle &\rightarrow |000\rangle && \text{"Logical 0"} \\ |1\rangle &\rightarrow |111\rangle && \text{"Logical 1"} \end{aligned}$$

In particular, a quantum superposition would be encoded as:

$$\alpha |0\rangle + \beta |1\rangle \rightarrow \alpha |000\rangle + \beta |111\rangle$$

Two key differences between quantum and classical error correction codes:

1. Cannot measure the codewords directly; would collapse the state
2. Phase errors

PHYC90045 Introduction to Quantum Computing

Syndrome Measurements

If we measured our qubits, we would collapse the state. For example, if we had the three qubit error correction code, and measured the first qubit as "0" then we would collapse:

$$\alpha |000\rangle + \beta |111\rangle \rightarrow |000\rangle$$

We do not measure the qubits individually, but instead measure correlations between qubits. The measurements are known as **syndrome** measurements.

[Recall: $Z|0\rangle = +1|0\rangle$ $Z|1\rangle = -1|1\rangle \rightarrow Z_1Z_2|01\rangle = (+1) \times (-1)|01\rangle = -|01\rangle$]

We measure: Z_1Z_2 Z_2Z_3

"Are the first two qubits the same?" and "Are the second two qubits the same?" If an X-error has occurred, we can tell that an error has happened, and where it is, but we have not measured any information about the encoded state.

PHYC90045 Introduction to Quantum Computing

Syndrome Measurement example

We have an encoded (logical) qubit:

$$\alpha |000\rangle + \beta |111\rangle$$

An X-error occurs on the first physical qubit:

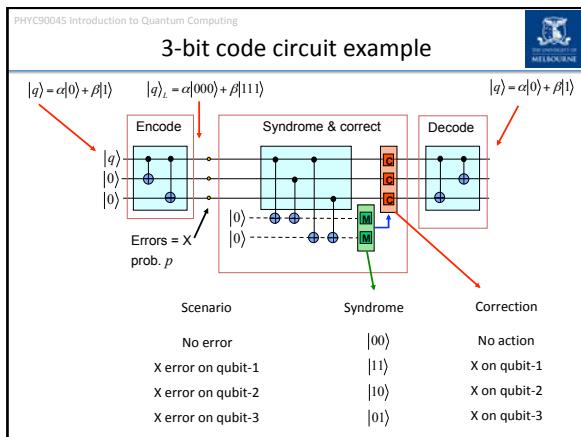
$$\alpha |100\rangle + \beta |011\rangle$$

We measure:

$Z_1Z_2 = -1$	First two qubits different
$Z_2Z_3 = +1$	Second two qubits same

From this we can deduce that an error has occurred on the first qubit, and correct (with an X gate we apply):

$$X_1(\alpha |100\rangle + \beta |011\rangle) = \alpha |000\rangle + \beta |111\rangle$$



PHYC90045 Introduction to Quantum Computing

Phase errors

In QM, bit flips are not the only type of errors which can occur. We can also have phase errors (and in practice these are more common).

$$Z_1 (\alpha |000\rangle + \beta |111\rangle) = \alpha |000\rangle - \beta |111\rangle$$

We have seen in the labs these errors are just as detrimental as bit flip errors!

We can make a phase-flip repetition code:

$$\begin{aligned} |0\rangle &\rightarrow |+++ \rangle & |+\rangle &= \frac{|0\rangle + |1\rangle}{\sqrt{2}} & X|\pm\rangle &= \frac{1}{\sqrt{2}}(X|0\rangle \pm X|1\rangle) \\ |1\rangle &\rightarrow |--- \rangle & \text{where} & |-\rangle &= \frac{|0\rangle - |1\rangle}{\sqrt{2}} & = \frac{1}{\sqrt{2}}(|1\rangle \pm |0\rangle) \\ && & & & = \pm |\pm\rangle \\ && & & & \rightarrow X_1 X_2 |+-\rangle = -|+-\rangle \end{aligned}$$

The syndrome measurements we make are:

$$X_1 X_2 \quad X_2 X_3$$

This code detects and corrects phase flip errors, but does not detect bit flip errors. Quantum error correction codes need to do both!

PHYC90045 Introduction to Quantum Computing

Phase flip code example

We have an encoded (logical) qubit:

$$\alpha |+++ \rangle + \beta |--- \rangle$$

An Z-error occurs on the third physical qubit:

$$\alpha |++-\rangle + \beta |-+ \rangle$$

We measure:

$$\begin{aligned} X_1 X_2 &= +1 & \text{First two qubits same} & X|\pm\rangle = \pm |\pm\rangle \\ X_2 X_3 &= -1 & \text{Second two qubits different} & \rightarrow X_1 X_2 |--\rangle = +|--\rangle \end{aligned}$$

From this we can deduce that a phase error has occurred on the third qubit, and correct (with an Z gate we apply):

$$Z_3 (\alpha |++-\rangle + \beta |-+ \rangle) = \alpha |+++ \rangle + \beta |--- \rangle$$

PHYC90045 Introduction to Quantum Computing

The Bacon-Shor Code

Codes exist which correct **both** phase flips, and bit flips, such as the Bacon-Shor 9-qubit code:

$$\begin{aligned} |0_L\rangle &= \frac{1}{\sqrt{8}} (|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle) \\ |1_L\rangle &= \frac{1}{\sqrt{8}} (|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle) \end{aligned}$$

Syndrome measurements are a combination the bit-flip and phase-flip codes. First as if this is three bit flip codes:

$$Z_1 Z_2, Z_2 Z_3, Z_4 Z_5, Z_5 Z_6, Z_7 Z_8, Z_8 Z_9$$

Then treating it as three logical qubits of three qubits each, and checking for a bit flip on any of these:

$$X_1 X_2 X_3 X_4 X_5 X_6, X_4 X_5 X_6 X_7 X_8 X_9$$

PHYC90045 Introduction to Quantum Computing

Stabilizer Formalism



Instead of specifying the codewords, we will specify the syndrome measurements which should give a “+1” result. From this we can derive the codewords/codespace.

An operator, S , is a stabilizer of the state $|\psi\rangle$ if

$$S|\psi\rangle = |\psi\rangle$$

Similarly, an operator S is a stabilizer of a subspace, if it stabilizes every basis state of that subspace.

For example:

$$Z|0\rangle = |0\rangle \quad X\frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

For our purposes, the stabilizers will all be tensor products of Pauli operators and the identity.

PHYC90045 Introduction to Quantum Computing

Aside: The Stabilizer Group



Mathematically, the stabilizers of a state (or a subspace) form a group, known as the stabilizer group, S . Verifying the four group axioms:

$I|\psi\rangle = |\psi\rangle$

If S_1, S_2 and S_3 stabilize $|\psi\rangle$ then:

$$S_1 S_2 |\psi\rangle = S_1 |\psi\rangle = |\psi\rangle$$

Associativity:

$$(S_1 S_2) S_3 = S_1 (S_2 S_3)$$

If S stabilizes $|\psi\rangle$ then

$$S^{-1}|\psi\rangle = S^{-1}S|\psi\rangle = |\psi\rangle$$

Typically (and for all of these lectures) we will choose the stabilizer group to be a subset of the Pauli group, and it is **Abelian** (ie. $AB=BA$).

We can specify the stabilizer group by writing its generators ($S_1, S_2, S_3, \dots, S_k$).

PHYC90045 Introduction to Quantum Computing

Stabilizers and QEC



For the bit-flip code, the “stabilizers” (generators of the stabilizer group) of the code are:

$$\begin{cases} Z_1 Z_2 \\ Z_2 Z_3 \end{cases}$$

The codewords are stabilized by these operators:

$$Z_1 Z_2 |000\rangle = |000\rangle \quad Z_1 Z_2 |111\rangle = |111\rangle$$

$$Z_2 Z_3 |000\rangle = |000\rangle \quad Z_2 Z_3 |111\rangle = |111\rangle$$

Any linear combination is also stabilized by these operators:

$$\alpha|000\rangle + \beta|111\rangle$$

PHYC90045 Introduction to Quantum Computing

Commutation of Pauli operators

The University of Melbourne

Commutation properties of the Pauli operators X, Y and Z are very useful at this point. We get the relations by considering actions on an arbitrary state.

For an arbitrary state we have different Pauli operators anti-commute (a negative sign when they are switched in order):

$$XZ |\psi\rangle = -ZX |\psi\rangle \rightarrow XZ = -ZX$$

$$XY |\psi\rangle = -YX |\psi\rangle \rightarrow XY = -YX$$

$$ZY |\psi\rangle = -YZ |\psi\rangle \rightarrow ZY = -YZ$$

Operators on different qubits commute (self evident):

$$X_1 Z_2 |\psi\rangle = Z_2 X_1 |\psi\rangle \rightarrow X_1 Z_2 = Z_2 X_1$$

But even products of operators commute:

$$X_1 X_2 Z_1 Z_2 |\psi\rangle = Z_1 Z_2 X_1 X_2 |\psi\rangle \rightarrow X_1 X_2 Z_1 Z_2 = Z_1 Z_2 X_1 X_2$$

PHYC90045 Introduction to Quantum Computing

Error And Stabilizers

The University of Melbourne

If an error **anti-commutes** with a syndrome measurement operator (ie. stabilizer generator) then the measurement result changes sign.

No Error

For example, consider the three qubit code, for which

$$Z_1 Z_2 |\psi\rangle = +1 |\psi\rangle$$

The syndrome measurement outcome is +1 (since the system is in the +1 eigenstate)

X Error

After an X-error on the first qubit:

$$Z_1 Z_2 |\psi'\rangle = Z_1 Z_2 X_1 |\psi\rangle = -X_1 Z_1 Z_2 |\psi\rangle = -X_1 |\psi\rangle = -|\psi'\rangle$$

The syndrome measurement outcome is -1 (since the system is in the -1 eigenstate)

PHYC90045 Introduction to Quantum Computing

Error and Stabilizers

The University of Melbourne

Error	State	$Z_1 Z_2$	$Z_2 Z_3$
I	$\alpha 000\rangle + \beta 111\rangle$	+1	+1
X_1	$\alpha 100\rangle + \beta 011\rangle$	-1	+1
X_2	$\alpha 010\rangle + \beta 101\rangle$	-1	-1
X_3	$\alpha 001\rangle + \beta 110\rangle$	+1	-1

Unique syndromes means that we can identify which error has occurred.

PHYC90045 Introduction to Quantum Computing

The University of
MELBOURNE

The Five Qubit Code

The smallest d=3 code to identify both bit and phase flips has five qubits.

Optimal

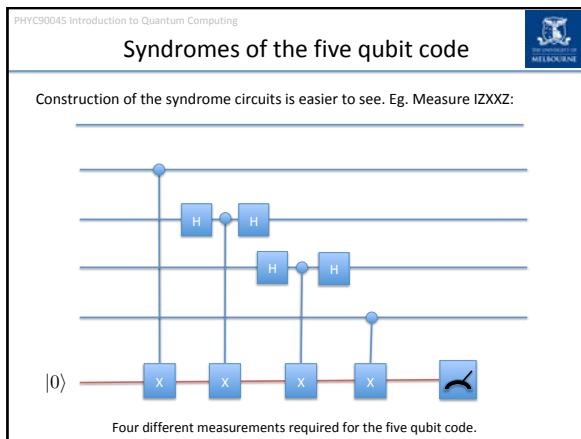
5 (qubits) x 3 (possible {X, Y or Z} errors on each qubit) + 1 (no error) = 16 syndromes

$2^4 = 16$ possible syndromes from four measurements.

The stabilizers of this code are:

$$\left\{ \begin{array}{l} IXZZX \\ XIXZZ \\ ZXIXZ \\ ZZXIX \end{array} \right.$$

Exercise: Write out all 15 single qubit errors and check that their syndromes are unique



PHYC90045 Introduction to Quantum Computing

Seven Qubit Steane Code

7 qubit “Steane” code. It is also known as the seven qubit “colour” code (which is a topological code – more next lecture). Stabilizers of this code are:

$$\left\{ \begin{array}{ccccccc} I & I & I & X & X & X & X \\ I & X & X & I & I & X & X \\ X & I & X & I & X & I & X \\ I & I & I & Z & Z & Z & Z \\ I & Z & Z & I & I & Z & Z \\ Z & I & Z & I & Z & I & Z \end{array} \right.$$

Exercise: Check that every single qubit error produces a unique syndrome!

PHYC90045 Introduction to Quantum Computing

Logical States of the Steane code



Logical States

$$|0_L\rangle = \frac{1}{\sqrt{8}}(|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle)$$

$$|1_L\rangle = \frac{1}{\sqrt{8}}(|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle + |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle)$$

We want to operate on these states while remaining protected ie. without decoding.

Logical X Operator

$$X_L = XXXXXX$$

(see this by operating directly on logical states above)

PHYC90045 Introduction to Quantum Computing

Logical Operators Commute with Stabilizers



Example Stabilizers:

$$\begin{pmatrix} I & I & I & X & X & X & X \\ I & X & X & I & I & X & X \\ X & I & X & I & X & I & X \\ I & I & I & Z & Z & Z & Z \\ I & Z & Z & I & I & Z & Z \\ Z & I & Z & I & Z & I & Z \end{pmatrix}$$

Logical X: $X_L = XXXXXX$

-> X operators all commute with themselves, and even number of Z commute, so logical operator commutes with the stabilisers and so code states are stabilised by the logical X operator.

PHYC90045 Introduction to Quantum Computing

Other logical operators



$$|0_L\rangle = \frac{1}{\sqrt{8}}(|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle)$$

$$|1_L\rangle = \frac{1}{\sqrt{8}}(|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle + |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle)$$

Logical 0 has zero or four 1's. Logical 1 has three or seven ones. So

$$Z_L = ZZZZZZZ$$

$$S_L = S^\dagger S^\dagger S^\dagger S^\dagger S^\dagger S^\dagger S^\dagger$$

$$i^4 = 1, i^3 = -i$$

PHYC90045 Introduction to Quantum Computing

Other logical operators

$|0_L\rangle = \frac{1}{\sqrt{8}}(|000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle)$

$|1_L\rangle = \frac{1}{\sqrt{8}}(|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle + |111000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle)$

Logical 0 has zero or four 1's. Logical 1 has three or seven. So

$Z_L = ZZZZZZZZ$

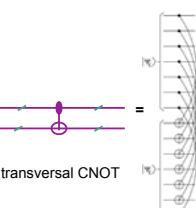
$S_L = S^\dagger S^\dagger S^\dagger S^\dagger S^\dagger S^\dagger S^\dagger$ $i^4 = 1, i^3 = -i$

PHYC90045 Introduction to Quantum Computing

Transversal Gates

Other Steane code 7-qubit code gates include H and CNOT.

Transversal gates:



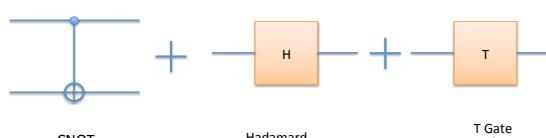
Hadamard on a single logical qubit

Can also implement the T gate (but this is not transversal)

PHYC90045 Introduction to Quantum Computing

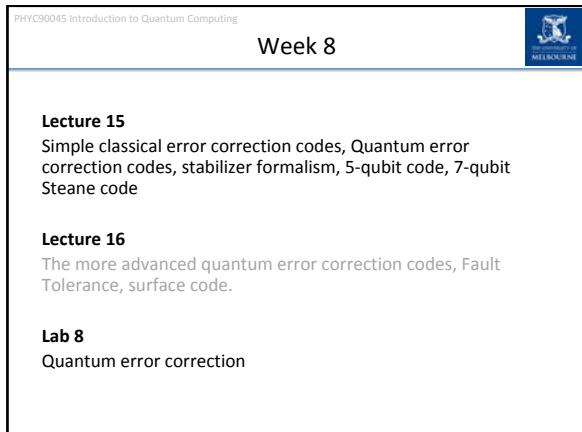
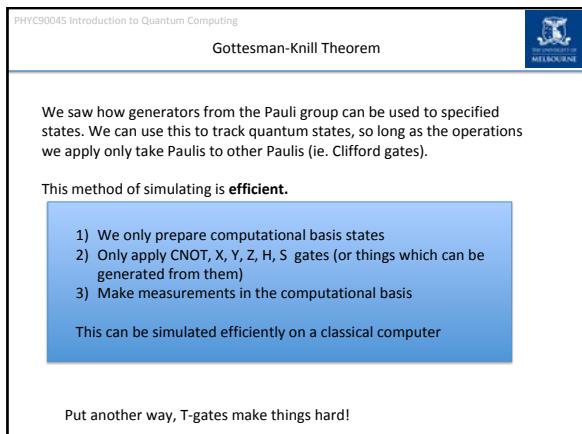
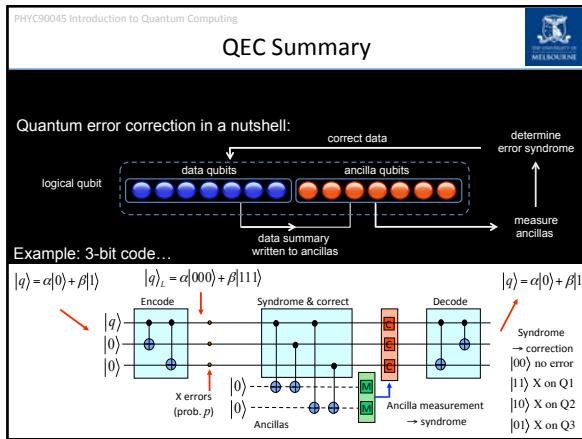
Fault Tolerant Universal Gate Set

In quantum computing every quantum circuit can be expressed as a sequence of:



CNOT Hadamard T Gate ($\pi/8$)

These gates can be implemented "fault tolerantly" using quantum error codes.



PHYC90045 Introduction to Quantum Computing

Week 8



Lecture 15
Simple classical error correction codes, Quantum error correction codes, stabilizer formalism, 5-qubit code, 7-qubit Steane code

Lecture 16
The more advanced quantum error correction codes, Fault Tolerance, QEC threshold, surface code.

Lab 8
Quantum error correction

PHYC90045 Introduction to Quantum Computing



Fault Tolerance and Topological Error Correction

Physics 90045
Lecture 16

PHYC90045 Introduction to Quantum Computing

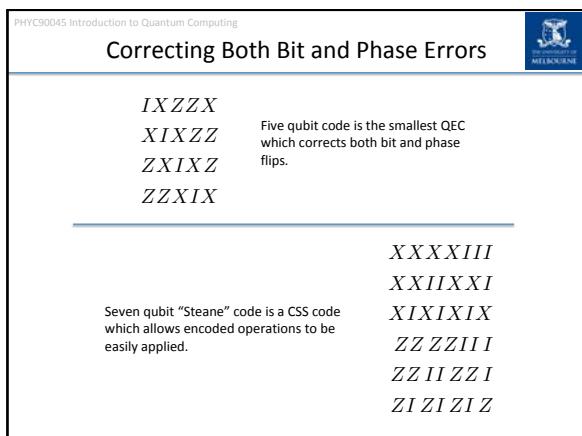
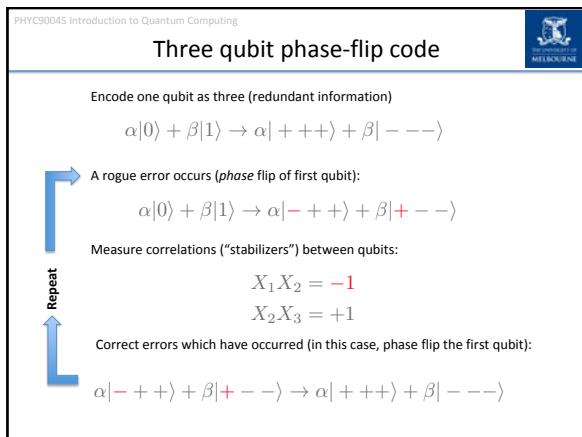
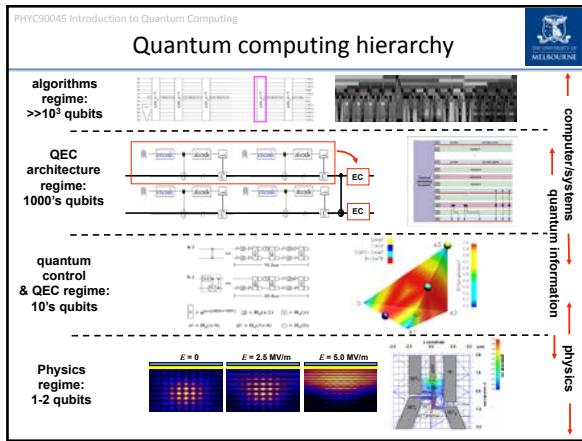
Overview



This lecture we will introduce more advanced error correction for quantum computers:

- Review some of the concepts from last lecture
- Fault Tolerance
- Concatenating quantum error correction codes
- The “threshold”
- Topological quantum error correction: The surface code

Reiffel, Chapter 11
Kaye, Chapter 10
Nielsen and Chuang, Chapter 10



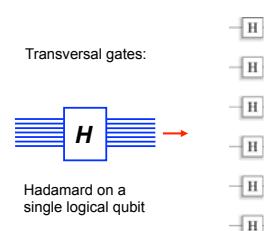
PHYC90045 Introduction to Quantum Computing

Logical Gates



The University of
MELBOURNE

Transversal gates:



Hadamard on a single logical qubit

This gate can be operated while leaving the logical qubit encoded, protected by the QEC code.

PHYC90045 Introduction to Quantum Computing

Logical CNOT

The diagram illustrates a quantum circuit with two horizontal lines representing 'Encoded Qubit 1' and 'Encoded Qubit 2'. A blue bracket on the left groups these two lines. A vertical blue line with a horizontal bar at its midpoint connects the two qubits. Five control points (represented by small circles) are placed along this vertical line, with three on Qubit 1 and two on Qubit 2. The circuit consists of two main parts: a sequence of single-qubit operations (represented by small circles with dots) and a sequence of multi-qubit operations (represented by small circles with crosses). The multi-qubit operations are controlled by the points on the vertical line.

Encoded Qubit 1

Encoded Qubit 2

Can also implement CZ,
Swap transversally

Danger! CNOTs can propagate errors. We need to make sure this happens in a controlled way.

PHYC90045 Introduction to Quantum Computing

Fault Tolerance

The logo of The University of Melbourne, featuring a coat of arms with a lion and a unicorn flanking a shield, topped by a crown, all within a blue square border.

Strategy: Take the original circuit and replace it with the *logical* version. In doing so we need to control the **spread** of errors. Doing this in a way which controls the spread of errors is known as fault tolerance:

Fault tolerant: a single error in any of the QEC procedures causes at most one error in the block of encoded qubits (which can be corrected)

A single error (on a physical qubit) should not propagate to two errors on the same logical qubit, otherwise we would not be able to correct that qubit.

PHYC90045 Introduction to Quantum Computing

Transversal CNOT is Fault Tolerant

The diagram shows a grid of horizontal lines representing the state of two encoded qubits. On the left, vertical blue brackets group the lines into 'Encoded Qubit 1' and 'Encoded Qubit 2'. A blue starburst indicates an error on the top line of Encoded Qubit 1 with probability p . This error propagates to the top line of Encoded Qubit 2 with probability p^2 , as indicated by another blue starburst. The bottom bracket of Encoded Qubit 2 is labeled 'Probability of two (uncorrectable) errors in this block is still proportional to p^2 '. The University of Melbourne logo is in the top right corner.

An error here
With probability p

Encoded Qubit 1

Encoded Qubit 2

Propagates to an error here, but is still correctable. Neither logical qubit has more than one error.

Probability of two (uncorrectable) errors in this block is still proportional to p^2 .

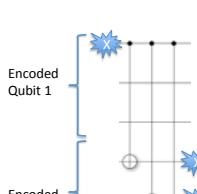
PHYS90045 Introduction to Quantum Computing



 The University of
MELBOURNE

NOT Fault Tolerant

Consider the following CNOT gate for the 3-qubit bit flip code (000/111)



Although this circuit is "correct" (the operation it performs – assuming no error is an encoded CNOT)...

... a physical single error can cause the second encoded qubit to be uncorrectable. It is **not** fault tolerant!

Care needed! Every operation (including measurement of syndromes) can have errors. Not only do X errors propagate, so do Z errors.

Transversal gates are Fault Tolerant				
Logical X	Logical Z	Logical S	Logical H	
				
				
				
				
				
				
				
				
Logical X	Logical Z	Logical S	Logical H	Logical CNOT

PHYC90045 Introduction to Quantum Computing

Larger distance codes

We have seen some simple error correction codes which correct one error (distance 3 codes). How can we construct quantum error correction codes which correct more than one error?

$$|0_L\rangle \rightarrow |00000\rangle \quad |1_L\rangle \rightarrow |11111\rangle$$

Distance 5 bit flip code

More errors needed before an uncorrectable, leading to a logical error.
More physical qubits give more locations for potential errors.

PHYC90045 Introduction to Quantum Computing

Concatenated codes

Systematic way to increase the distance of a code. Feed the code back into itself:

$$|0_{L2}\rangle = \frac{1}{\sqrt{8}}(|0_10_0_0_0_0_0_0_0\rangle + |1_10_1_0_1_0_1_0\rangle + |0_11_1_0_0_1_0_1\rangle + |1_11_0_0_1_0_1_0\rangle + |0_0_0_1_0_1_1_1_0\rangle + |1_0_1_0_1_1_1_1_0\rangle + |0_1_0_1_1_0_1_1_0\rangle + |1_1_0_1_1_0_1_1_0\rangle)$$

$$|1_{L2}\rangle = \frac{1}{\sqrt{8}}(|1_11_1_1_1_1_1_1_1\rangle + |0_11_0_1_1_0_1_1_0\rangle + |1_10_0_1_1_1_0_1_0\rangle + |0_10_1_1_1_0_1_0_1\rangle + |1_11_1_0_0_1_0_0_1\rangle + |0_11_0_0_1_0_0_1_0\rangle + |1_10_0_0_1_0_0_1_0\rangle + |0_10_0_0_1_0_0_1_0\rangle)$$

$|0_L\rangle = \frac{1}{\sqrt{8}}(|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle)$

$|1_L\rangle = \frac{1}{\sqrt{8}}(|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle + |111000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle)$

This method is known as “concatenation” of error correction codes.

PHYC90045 Introduction to Quantum Computing

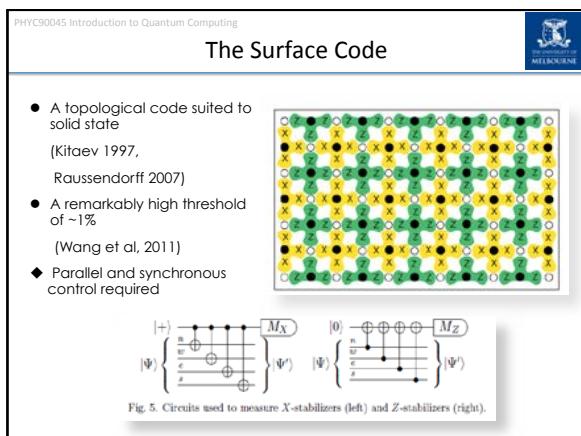
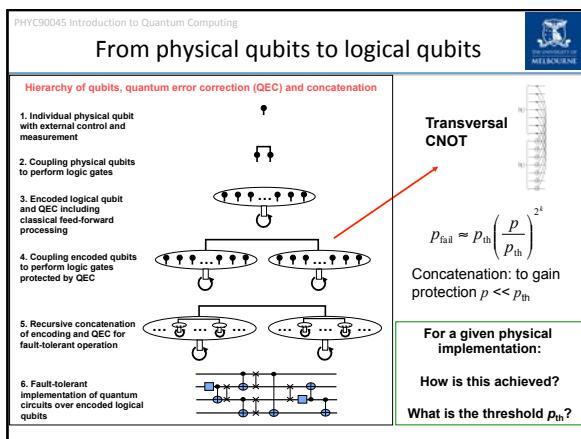
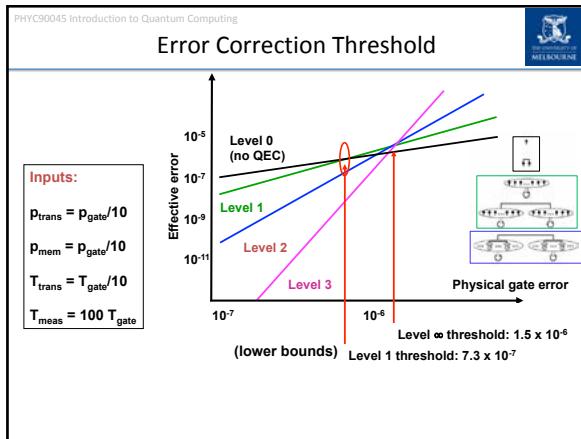
Error after different levels of encoding

Logical error rate achieved

$$\rho_{\text{fail}} = \rho_{\text{th}}(\rho/\rho_{\text{th}})^{2^k}$$

$$\rho_{\text{th}} = 10^{-5}, \rho = 10^{-6}$$

k=1: $\rho_{\text{fail}} = 10^{-7}$
k=2: $\rho_{\text{fail}} = 10^{-9}$
k=3: $\rho_{\text{fail}} = 10^{-13}$



PHYC90045 Introduction to Quantum Computing

Errors on the surface code



 THE UNIVERSITY OF
 MELBOURNE

PHYS90045 Introduction to Quantum Computing

Chains of Errors



THE UNIVERSITY OF
MELBOURNE

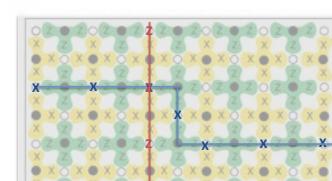
- Errors form chains, can only see syndrome changes (-1) at the ends.
- **Minimum weight matching** determines the most likely errors.
- Chains greater than half way across the surface can cause failure.

PHYS90045 Introduction to Quantum Computing

Logical Operators on the surface code



 THE UNIVERSITY OF
 MELBOURNE



A logical X operation is a chain of X operations, left to right

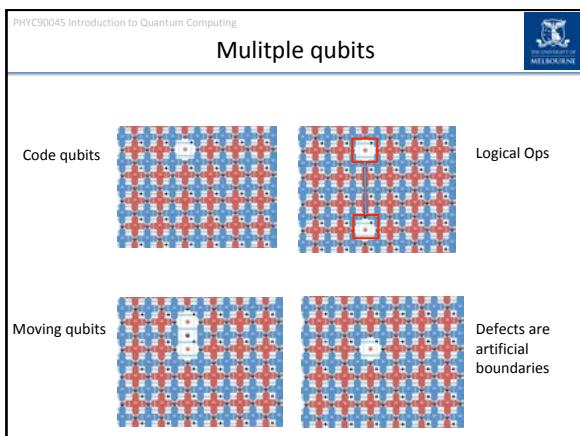
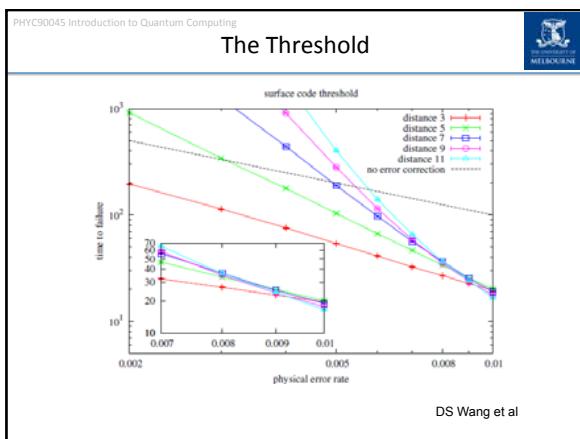
A logical Z operation is a chain of Z operations, top to bottom

Logical operations anti-commute (as they should)

PHYC90045 Introduction to Quantum Computing

Distance of the Surface Code

- Distance of the code is equal to the length of a side.
- Scale up by simply making larger patch of surface code (concatenation not required)
- Topologically defined, so easy to map onto physical architectures



PHYC90045 Introduction to Quantum Computing

Requirements for an Error Corrected Shor

PHYSICAL REVIEW A 86, 012324 (2012)

Surface codes: Towards practical large-scale quantum computation

Austin G. Fowler
Centre for Quantum Computation and Communication Technology, School of Physics, The University of Melbourne, Victoria 3010, Australia
Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland
Department of Physics, University of California, Santa Barbara, California 93106-9530, USA
and California Nanosystems Institute, University of California, Santa Barbara, California 93106-9530, USA
(Received 2 August 2012; published 18 September 2012)

Bits in factored number	2000
Number of Logical qubits required	4000
Number of qubits in surface code	~20 million qubits
Time for one measurement	100 ns
Total time required	26 hours

Research topic: Bring these requirements down!

CENTRE FOR QUANTUM COMPUTATION & COMMUNICATION TECHNOLOGY
UNIVERSITY OF MELBOURNE, VICTORIA, AUSTRALIA

Experimental proposal

High-level:
Qubit: uniform nuclear spin
Addressing: electron load (not gate defined waveform)
Gates: electron load and global ESR/NMR control
Operation: parallel, 60 MHz loading pulses, robust to local variations

Criss-cross gate array → parallel shared control of qubit addressing (robust)
For N qubits, # control lines scales as \sqrt{N}
Established ESR/NMR spin control (Morton et al Nature 2008, Pla et al Nature 2013)
3D STM fabrication of array (McKibbin Nanotechnology 2013)
Initial proposal: CNOT dipole coupling (slow) → developing faster gates (MHz regime)

C. Hill et al., Science Advances 2015

Lecture 15
Simple classical error correction codes, Quantum error correction codes, stabilizer formalism, 5-qubit code, 7-qubit Steane code

Lecture 16
The more advanced quantum error correction codes, Fault Tolerance, QEC threshold, surface code.

Lab 8
Quantum error correction

PHYC90045 Introduction to Quantum Computing

Week 9



Lecture 17
Optimization problems, Encoding problems as energies,
Quadratic Binary Optimization (QUBO), Problem embedding

Lecture 18
Adiabatic Quantum Computing

Lab 9
Using DWave machines

PHYC90045 Introduction to Quantum Computing



Quantum Optimization

Physics 90045
Lecture 17

PHYC90045 Introduction to Quantum Computing

Overview



This lecture we will talk about mapping Optimization Problems to physical systems:

- Cooling as an optimization algorithm
- QUBO problems
- Examples of QUBO problems:
 - Subset sum, Graph Partitioning and Travelling Salesman
- Embedding problems in quantum computing architectures

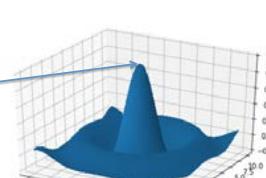
Kaye 8.5
Rieffel 13.4.2

PHYC90045 Introduction to Quantum Computing

The University of
MELBOURNE

Optimization Problems

Given some cost-function or “objective function” we would like to maximize/minimize. Often the inputs/parameters are constrained.



A 3D surface plot representing a cost function. The vertical axis is labeled "Cost function" and ranges from -0.2 to 1.0. The horizontal axes are labeled "Parameters" and range from -10.0 to 10.0, with tick marks at -10.0, -7.5, -5.0, -2.5, 0.0, 2.5, 5.0, 7.5, 10.0. A single, sharp peak is centered at approximately (0, 0) with a height of about 0.9. A blue arrow points from the label "Optimal value" towards the peak, indicating the direction of optimization.

PHYC90045 Introduction to Quantum Computing

Physics and computation: Cooling

The diagram shows a vertical blue arrow labeled "Energy" pointing upwards. To its right, several horizontal grey lines represent energy levels. At the bottom left, under the label "High temperature", there are two pairs of lines: one pair is blue and labeled "Occupied", and another pair is grey and labeled "Unoccupied". Red arrows point from the "Occupied" levels down to the "Unoccupied" levels. At the bottom right, under the label "Low temperature", all levels are now grey and labeled "Unoccupied". Red arrows point from the "Occupied" levels at high temperature down to the "Unoccupied" levels at low temperature. A label "Ground state" points to the lowest energy level.

Physical systems naturally perform optimization problems. For example, if you cool a system towards absolute zero, it will populate the the lowest energy level.

PHYC90045 Introduction to Quantum Computing

Reminder: Measuring Observables

IN QM an observable (eg. energy) is represented by a Hermitian matrix. The values measured when measuring the observable are the **eigenvalues** of the corresponding matrix.

In the simplest case, the matrix is a diagonal matrix and the eigenvalues are listed down the diagonal:

H stands for “Hamiltonian”
The total energy in the system.

$$H = Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Energy of the $|0\rangle$ state
Energy of the $|1\rangle$ state

In more complicated cases you may have to *diagonalize* the matrix first, using an eigenvalue decomposition.

PHYC90045 Introduction to Quantum Computing

Encoding Problems into the Energy

In building a quantum computer, we have built a perfectly controllable quantum system. We can do different algorithms to try to find the state with the lowest energy.

How can we encode an objective function into the energy of a system?

For a single qubit:

Z	+1 energy for 0 -1 energy for 1
EZ	+E energy for 0 -E energy for 1
$-EZ$	+E for the 1 state -E for the 0 state

PHYC90045 Introduction to Quantum Computing

Measuring Observables (multiple qubits)

Each observable (eg. energy) is represented by a Hermitian matrix. The values measured of when measuring the observable are the **eigenvalues** of the corresponding matrix.

In the simplest case, the matrix is a diagonal matrix and the eigenvalues are listed down the diagonal:

H stands for "Hamiltonian"
The total energy in the system. $\rightarrow H = Z \otimes Z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ Energy of the $|00\rangle$ state
Energy of the $|01\rangle$ state
Energy of the $|10\rangle$ state
Energy of the $|11\rangle$ state

In more complicated cases you may have to *diagonalize* the matrix first, using an eigenvalue decomposition.

PHYC90045 Introduction to Quantum Computing

Two Qubit Ising Interactions

"Ising spin" couplings only involve two qubits, and just Z operators:

$Z_1 Z_2$	+1 for 00 or 11 state -1 for 01 or 10 state	Anti-ferromagnetic
$-Z_1 Z_2$	-1 for 00 or 11 state +1 for 01 or 10 state	

PHYC90045 Introduction to Quantum Computing

Total Energy of the System

Consider a system that has an energy function:

$$E = J_{12}z_1z_2 + J_{23}z_2z_3 + J_{13}z_1z_3 + B_1z_1 + B_2z_2 + B_3z_3$$

where the z_i are $+/-1$, and the J 's and B 's are specific parameters defining the particular problem at hand.

To get ready to map to a QC, we write the total energy as the operator "H" (which physicists would call the "Hamiltonian") on a system of qubits as a sum of these terms with $z_i \rightarrow Z_i$ (where Z_i is the Z operator on the i th qubit):

$$H = J_{12}Z_1Z_2 + J_{23}Z_2Z_3 + J_{13}Z_1Z_3 + B_1Z_1 + B_2Z_2 + B_3Z_3$$

Pairwise interactions between qubits Bias on individual qubit

PHYC90045 Introduction to Quantum Computing

Mapping the Spin Glass form to QC

Optimisation problems can often be cast into an equivalent "spin glass" form:

$$E = \sum_{i \neq j} J_{ij}z_iz_j + \sum_i B_iz_i$$

This is convert to a convenient form to map onto a quantum computer:

$$H = \sum_{i \neq j} J_{ij}Z_iZ_j + \sum_i B_iZ_i$$

Ising coupling local "field"

In QM, energy is represented as a matrix!

The Z_i are now operators defined as per our definitions with eigenvalues $+/-1$ (which can be mapped to binary variables 0/1)

PHYC90045 Introduction to Quantum Computing

QUBO Problems

QUBO stands for "Quadratic Unconstrained Binary Optimization"

The cost function (which we want to minimize) is:

$$E(x_1, \dots, x_n) = \sum_i c_i x_i + \sum_{i,j} Q_{ij} x_i x_j$$

Where x_i are Boolean (binary) variables, either 0 or 1.

N.B. we will use x for binary, z for $+/-1$

Quadratic term

PHYC90045 Introduction to Quantum Computing

Binary to energy

Typically when we write such energy functions we write in terms of the Z variables:

z_i

(lower case z)

But the binary variables in terms of 0 or 1:

$x_i = 0 \quad \text{or} \quad x_i = 1$

Can convert between x_i and z_i using:

$x_i \rightarrow \frac{z_i + 1}{2}$

PHYC90045 Introduction to Quantum Computing

Example: Number partitioning

Given the numbers:

1, 3, 8, 10, 6, 5, 5

Is there a way to partition these numbers into two disjoint partitions, such that the sum of the elements in both partitions is the same?

Yes (in this case): {1, 8, 10} and {3, 6, 5, 5}

PHYC90045 Introduction to Quantum Computing

Graph Partitioning to QUBO

1, 3, 8, 10, 6, 5, 5

We assign a qubit to each number.
The qubit being zero indicates it is one partition.
The qubit being one indicates it is in the other.

Qubits are $|0\rangle$ if they're in partition 0, $|1\rangle$ if they're in partition 1

PHYC90045 Introduction to Quantum Computing

Number partitioning as a QUBO problem

1, 3, 8, 10, 6, 5, 5

As an optimization problem: We want

$$\sum_i w_i z_i = 0$$

The i^{th} number ± 1

Unfortunately if we just minimize this, all the qubits will end up with $z_i = -1$

PHYC90045 Introduction to Quantum Computing

Number partitioning as a QUBO problem

But if we square, we should get a positive solution (or zero). We want to find the assignment of spins which has the minimum energy (ie. closest to zero):

$$H = \left(\sum_i w_i Z_i \right)^2 = \sum_{i \neq j} 2w_i w_j Z_i Z_j + \sum_i w_i^2 I$$

Coupling is the product of numbers

Eg. For the set {1, 2, 3}:

$H = 4Z_1Z_2 + 6Z_1Z_3 + 12Z_2Z_3 + 14I$

Finding minimum energy state will solve the problem!

PHYC90045 Introduction to Quantum Computing

Solution for our Number Partitioning

$H = 4Z_1Z_2 + 6Z_1Z_3 + 12Z_2Z_3 + 14I$

Two degenerate solutions: $|110\rangle$ $|001\rangle$

$E = 4 - 6 - 12 + 14 = 0$

And of course, they correctly partition the numbers: $1+2=3$
Other combinations go worse, eg, $|111\rangle$

$E = 4 + 6 + 12 + 14 = 36$

PHYC90045 Introduction to Quantum Computing

Example: Graph Partitioning

What is a partition of the set of vertices, V , into two subsets of *equal size* such that the number of edges connecting the two partitions is minimized?

PHYC90045 Introduction to Quantum Computing

Example: Graph Partitioning

What is a partition of the set of vertices, V , into two subsets of *equal size* such that the number of edges connecting the two partitions is minimized?

PHYC90045 Introduction to Quantum Computing

Graph Partitioning to QUBO

Qubits are $|0\rangle$ if they're in partition 0, $|1\rangle$ if they're in partition 1

PHYC90045 Introduction to Quantum Computing

Even numbers in each subset



$$H_A = \left(\sum_i Z_i \right)^2$$

As before, having an equal number of terms in each partition will evaluate to zero. Unequal numbers result in a positive value.

PHYC90045 Introduction to Quantum Computing

Number of edges joining the subsets



$$H_B = \sum_{i,j \in E} \frac{I - Z_i Z_j}{2}$$

Evaluates to 0 if the edge is in the same partition, but +1 if the edge goes between partitions. In total H_B counts the number of edges between the two partitions.

PHYC90045 Introduction to Quantum Computing

Total Hamiltonian



$$H_A = \left(\sum_i Z_i \right)^2$$

Same number of vertices in each partition.

$$H_B = \sum_{i,j \in E} \frac{I - Z_i Z_j}{2}$$

Number of edges between partitions.

In total then, with $A \gg B$:

$$H = AH_A + BH_B$$

PHYC90045 Introduction to Quantum Computing

Travelling Salesman Problem (TSP)

Given several (n) cities and the distances between them, find the shortest path (Hamiltonian cycle) which visits all of them once and returns to the original city.

- TSP is an example of an NP-Complete problem (Karp, 1972) and best known classical algorithms require exponential time.
- Has many direct practical applications including planning, logistics, DNA sequencing, and astronomy.
- Largest solved tour is approximately 85,000 sites with heuristic methods able to find solutions (for million site tours) within 2-3% of the optimal tour.

PHYC90045 Introduction to Quantum Computing

Mapping TSP to QUBO problem

$x_{i,v} = \begin{cases} 1, & \text{tour passes through city } v \text{ at step } i \\ 0, & \text{otherwise} \end{cases}$

PHYC90045 Introduction to Quantum Computing

Energy Penalties

Each city appears exactly once in the cycle:

$$H_{city} = \sum_v \left(1 - \sum_i x_{v,i} \right)^2$$

Each step has exactly one city:

$$H_{step} = \sum_i \left(1 - \sum_v x_{v,i} \right)^2$$

Only paths between cities with edges between them are taken:

$$H_{cycle} = \sum_i \sum_{u \rightarrow v \notin E} x_{u,i} x_{v,i+1}$$

Any path making a cycle will have energy, $E=0$, with all other combinations having higher energy.

PHYC90045 Introduction to Quantum Computing

Shortest cycle

To find the shortest cycle, add an energy penalty of the length of the cycle:

$$H_{length} = \sum_i \sum_{u \rightarrow v \in E} W_{uv} x_{u,i} x_{v,i+1}$$

Finding the lowest energy arrangement of $x_{v,i}$ of

$$H_{TSP} = A(H_{city} + H_{site} + H_{cycle}) + BH_{length}$$

will solve the corresponding TSP problem ($A \gg B$).

We can make the problem quantum mechanical by mapping classical variables $x_{v,i}$ to qubits (and operators):

$$x_{v,i} \rightarrow \frac{I + Z_{v,i}}{2}$$

The problem is now to find the ground state of a quantum mechanical Hamiltonian
-> several ways to do this by coding the problem onto a QC
i.e. digital (e.g. QAOA/VQE next lecture) or analogue (e.g. AQC, later on)

PHYC90045 Introduction to Quantum Computing

Quantum Computer Layouts

$K_{4,4}$ and $K_{2,2}$ Chimera Subgraphs

Two-level Grid Subgraph

Kuratowski Subgraph

Sparse Kuratowski Subgraph

From: S. Tonetto MSc thesis

PHYC90045 Introduction to Quantum Computing

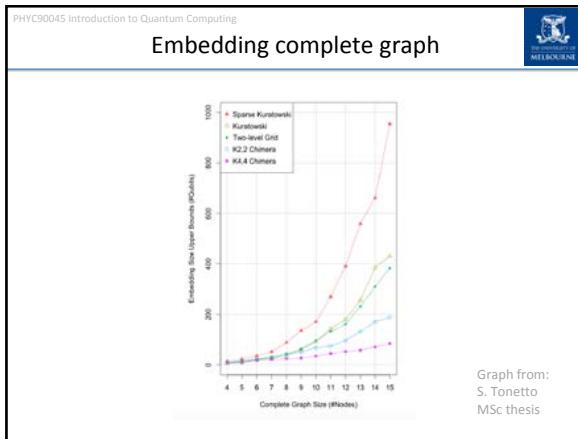
Embedding computational problems

Logical Ising Graph

Minor embedding

Ferromagnetic couplings, both qubits the same

From: S. Tonetto MSc thesis



PHYC90045 Introduction to Quantum Computing

Speed up of optimization algorithms

It is proven (Aharonov, Kempe, et al) that you can map any circuit onto an equivalent optimization problem, with only a polynomial difference in resources (including time required to solve).

Their cost functions involve more than just “Z” – carefully worked out “gadgets”.

Just because we can find an encoding of a problem in a way which a quantum computer could solve it, doesn't say anything about the speed up.

Typically, we when considering hard problems such as NP-Complete problems (e.g. TSP) we expect to achieve a quadratic speedup in accordance with quantum search on unstructured problems.

Given the power of classical heuristics to attack such problems, the development and power of “quantum heuristics” is an open question.

PHYC90045 Introduction to Quantum Computing

Week 9

Lecture 17
Optimization problems, Encoding problems as energies,
Quadratic Binary Optimization (QUBO), Problem embedding

Lecture 18
Adiabatic Quantum Computing

Lab 9
Using DWave machines

PHYC90045 Introduction to Quantum Computing

Week 9

Lecture 17
Optimization problems, Encoding problems as energies, Quadratic Binary Optimization (QUBO), Problem embedding

Lecture 18
Adiabatic Quantum Computing

Lab 9
Using DWave machines

PHYC90045 Introduction to Quantum Computing

Adiabatic Quantum Computation

Physics 90045
Lecture 21

PHYC90045 Introduction to Quantum Computing

Overview

In this lecture we will cover/review

- Quantum Adiabatic Processes
- The problem Hamiltonian
- Avoided crossings and energy gap
- The adiabatic theorem

PHYC90045 Introduction to Quantum Computing

D-Wave systems



The University of Melbourne

Image: D-Wave Systems

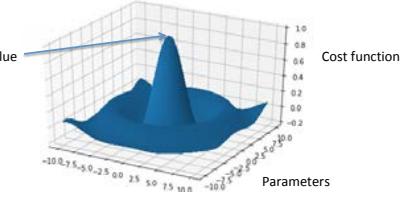
- 2000 qubit quantum "annealers"
- Sold quantum computers to Lockheed Martin, Google, NASA, Los Alamos National Laboratory (\$15 million each)
- We will be using Dwave machines (remotely) during the labs

PHYC90045 Introduction to Quantum Computing

Review: Optimization Problems

The University of Melbourne

Given some cost-function or "objective function" we would like to maximize/minimize. Often the inputs/parameters are constrained.



We have seen in previous lecture how to map these onto QUBO problem Hamiltonians.

PHYC90045 Introduction to Quantum Computing

Adiabatic Processes

The University of Melbourne

- Start in the known ground state of a simple Hamiltonian
- Slowly change the Hamiltonian to the problem Hamiltonian
- Provided the change has been "slowly enough" the system will remain in the ground state, and we will have found the ground state of the problem Hamiltonian.

$$H(s) = (1-s)H_x + sH_p \quad 0 \leq s \leq 1$$

$$s = t/T$$

Bad analogy: Moving a glass full of water slowly enough means that you keep the water in the glass. If you move it too fast all the water slops out.

Note: Quantum adiabatic process is not the same as thermodynamic adiabatic process ($Q=0$)

PHYC90045 Introduction to Quantum Computing

Hamiltonian of transverse field



$$H_x \propto B_x \sum_i X_i$$

Transverse field Hamiltonian

Eg. Three electrons in a transverse field:

$$H_x = g\mu_B B_x (X_1 + X_2 + X_3)$$

We call this the transverse Hamiltonian, H_x .

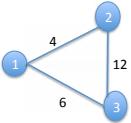
PHYC90045 Introduction to Quantum Computing

Problem Hamiltonian



We have seen many examples of problem Hamiltonians in our discussions of QUBO Problems. Two-body because that's what nature gives us.

Eg. Number partitioning for the set {1, 2, 3}:



$$H = 4Z_1Z_2 + 6Z_1Z_3 + 12Z_2Z_3 + 14I$$

Finding minimum energy state will solve the problem!

We call this the problem Hamiltonian, H_p .

PHYC90045 Introduction to Quantum Computing

Quantum “Annealing”



Definition according to Google:

Annealing: heat (metal or glass) and allow it to cool slowly, in order to remove internal stresses and toughen it. "Copper tubes must be annealed after bending or they will be brittle"

$$H(s) = (1 - s)H_x + sH_p$$

Transverse field plays the role of temperature. Causes excitations, strength is slowly being lowered.

Problem Hamiltonian defines the energy landscape of the problem we want to solve.

PHYC90045 Introduction to Quantum Computing

Example: Just one Qubit

$H(s) = (1-s)H_x + sH_p$

$s = t/T$

(Lowest) Eigenvalues

$H_x = X$

$H_p = Z$

“Landau-Zener” avoided crossing

System is originally in the state:
 $|-\rangle$

Time

System ends up in the state:
 $|1\rangle$

PHYS90045 Introduction to Quantum Computing

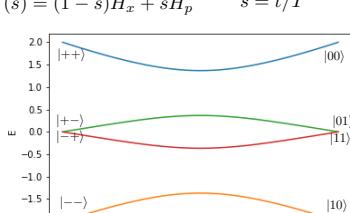


 THE UNIVERSITY OF
 MELBOURNE

Two qubit example

$$\begin{aligned}H_p &= Z_1 Z_2 + Z_1 \\H_x &= X_1 + X_2\end{aligned}$$

$$H(s) = (1-s)H_x + sH_p \quad s = t/T$$



s	$ +>$	$ 01>$	$ +->$	$ -->$	$ 11>$	$ 10>$
0.0	2.0	2.0	0.0	-1.8	-0.2	-1.8
0.2	1.8	1.8	0.2	-1.7	-0.2	-1.7
0.4	1.6	1.6	0.4	-1.6	-0.2	-1.6
0.6	1.8	1.8	0.2	-1.7	-0.2	-1.7
0.8	2.0	2.0	0.0	-1.8	-0.2	-1.8
1.0	2.0	2.0	0.0	-1.8	-0.2	-1.8

(Lowest) Eigenvalues

Energy

Time

$\Delta = 0.17023$

Avoided crossing
Energy gap

PHYC90045 Introduction to Quantum Computing

Adiabatic Theorem

How slowly is slowly enough? Adiabatic criterion.

Time derivative of Hamiltonian

$$\sum_{m \neq n} \frac{\hbar |\langle m | \dot{H} | n \rangle|}{|E_n - E_m|^2} = \sum_{m \neq n} \left| \frac{\hbar \langle m | \dot{n} \rangle}{E_n - E_m} \right| \ll 1$$

m, nth element

Energy of eigenstates:
n is the ground state,
m is every other state

PHYC90045 Introduction to Quantum Computing

Roughly speaking...

$$\sum_{m \neq n} \frac{\hbar |\langle m | \dot{H} | n \rangle|}{|E_n - E_m|^2} = \sum_{m \neq n} \left| \frac{\hbar \langle m | \dot{n} \rangle}{E_n - E_m} \right| \ll 1$$

For ground state, largest contribution from the smallest two eigenvalues. $E_n - E_m$

$$|E_n - E_m| > \Delta$$

For our linear schedule:

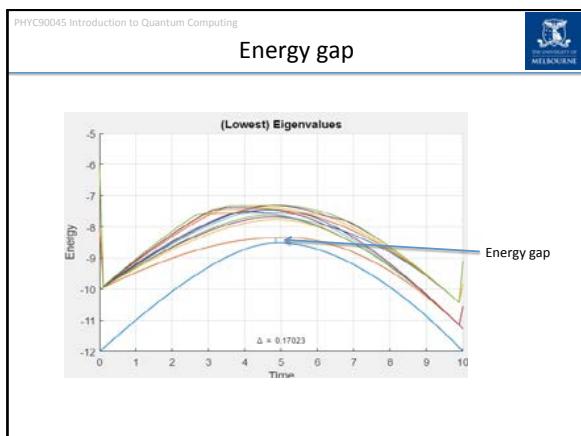
$$\dot{H}_{mn} \approx \frac{H_{mn}}{T}$$

Energy gap between ground state and first excited Eigenstate

Time required scales inversely proportional to the gap:

$$T \propto \frac{1}{\Delta}$$

For a given problem: How big is the gap? Difficult to work out in general!

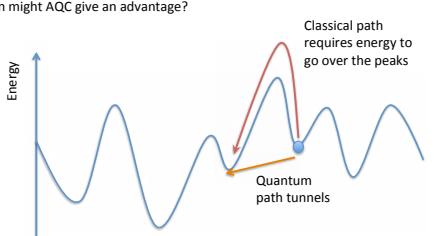


PHYC90045 Introduction to Quantum Computing



Quantum Tunneling

When might AQC give an advantage?



The figure shows a plot of Energy versus Configuration. A blue line represents the potential energy landscape, characterized by several peaks and valleys. A red line, labeled "Classical path requires energy to go over the peaks", follows the top of the peaks. An orange line, labeled "Quantum path tunnels", follows the valleys, indicating that a quantum system can find lower-energy states even if they are separated by small peaks.

Quantum annealing has been shown to outperform classical annealing in the case where the barriers are high, but thin.

PHYC90045 Introduction to Quantum Computing

The University of
MELBOURNE

Equivalent of Grover's algorithm

Roland and Cerf demonstrated that AQC could be used to implement an unordered search:

$$\begin{aligned}
 H_x &= I - 2|\phi\rangle\langle\phi| \\
 H_p &= I - 2|m\rangle\langle m|
 \end{aligned}$$

$|\phi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle$
 Marked state, m

x	Blue Curve (E)	Orange Curve (E)
0.0	0.35	0.05
0.5	0.30	0.02
1.0	0.25	0.01
1.5	0.20	0.005
2.0	0.15	0.002
2.5	0.10	0.001
3.0	0.05	0.0005
3.5	0.00	0.0002

Optimization, since the energy spectrum is known:
 Change Hamiltonian faster when the gap is larger, faster when it is smaller.

This achieves the same $O(\sqrt{N})$ speedup as Grover's algorithm.

The figure displays five different subgraph types used in quantum computing:

- K_{4,4} and K_{2,2} Chimera Subgraphs:** Two separate subgraphs consisting of four nodes arranged in a square pattern, with each node connected to its neighbors via multiple wires.
- Two-level Grid Subgraph:** A rectangular grid of nodes where every node is connected to its immediate horizontal and vertical neighbors.
- Kuratowski Subgraph:** A complex, dense subgraph formed by several overlapping cycles and triangles, representing a Kuratowski graph.
- Sparse Kuratowski Subgraph:** A simplified version of the Kuratowski graph, showing fewer connections between nodes.

PHYS90045 Introduction to Quantum Computing

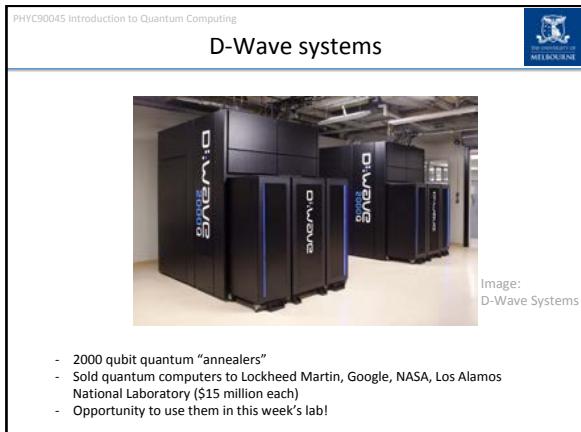
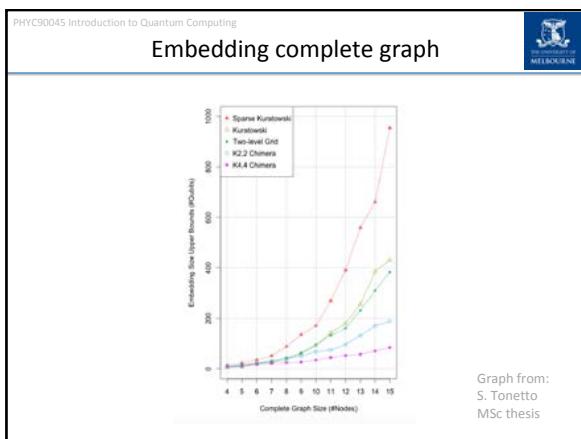
Embedding computational problems

Logical Ising Graph

Minor embedding

**Ferromagnetic couplings,
both qubits the same**

From:
S. Tonetto
MSc thesis



PHYC90045 Introduction to Quantum Computing

Adiabatic from Start to Finish

(1) Map computational problem to QUBO/Hamiltonian
 (2) Embed problem on physical architecture
 (3) Execute the adiabatic algorithm
 (4) Read the ground state configuration for the answer to the problem

PHYC90045 Introduction to Quantum Computing

MAX-Cut Problem

Partition the nodes into two disjoint subsets (not necessarily with equal numbers of nodes in each!) so that there is the maximum number of edges between the two subsets.

PHYC90045 Introduction to Quantum Computing

MAX-Cut Problem

Partition the nodes into two disjoint subsets (not necessarily with equal numbers of nodes in each!) so that there is the maximum number of edges between the two subsets.

PHYC90045 Introduction to Quantum Computing

Graph Partitioning to QUBO

Qubits are $|0\rangle$ if they're in subset 0, $|1\rangle$ if they're in subset 1

PHYC90045 Introduction to Quantum Computing

Map Problem to QUBO/Hamiltonian

Hamiltonian which counts the edges between subsets:

$$H = \sum_{i,j \in E} \frac{Z_i Z_j - I}{2}$$

Score -1 if edges are in different subsets
Score 0 if the edges are in the same subset

Most edges between subsets will have the minimum energy.
Ground state gives the answer to the MAX-CUT problem.

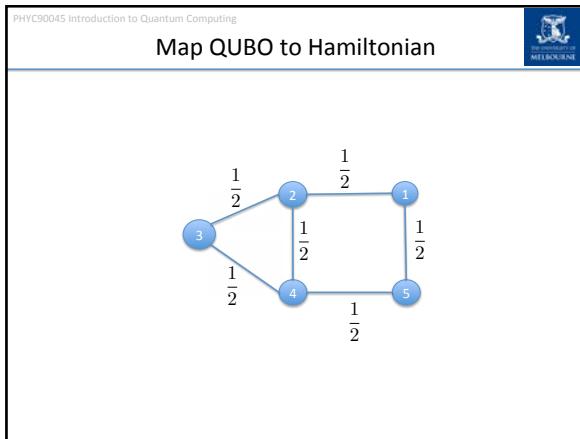
PHYC90045 Introduction to Quantum Computing

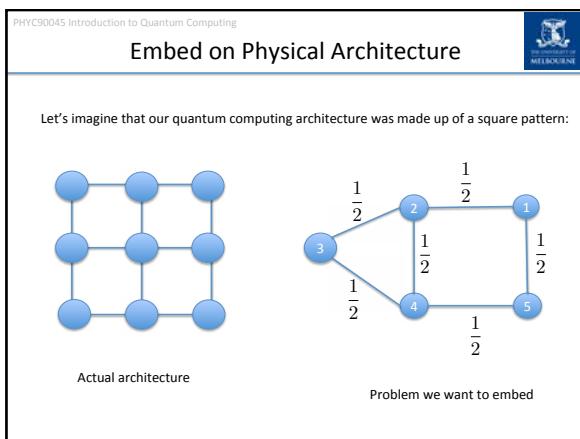
Map QUBO to Hamiltonian

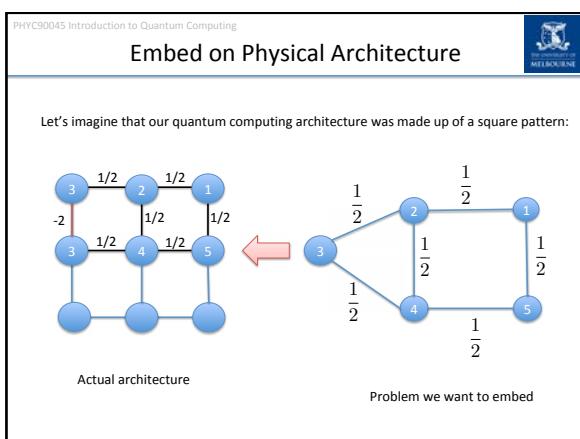
$$H = \sum_{i,j \in E} \frac{Z_i Z_j - I}{2}$$

In our case:

$$H = \frac{Z_1 Z_2 - I}{2} + \frac{Z_2 Z_3 - I}{2} + \frac{Z_3 Z_4 - I}{2} + \frac{Z_4 Z_5 - I}{2} + \frac{Z_5 Z_1 - I}{2} + \frac{Z_2 Z_4 - I}{2}$$







PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

Execute algorithm

At this point you would program the couplings into your physical quantum computer, and physically perform the annealing schedule:

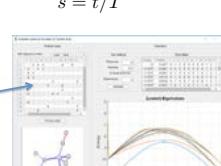
$$H(s) = (1 - s)H_x + sH_p$$

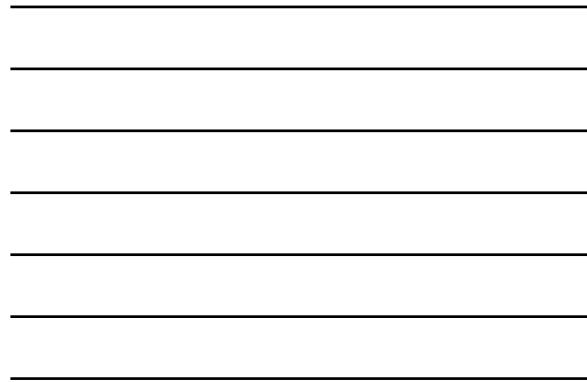
$$0 \leq s \leq 1$$

$$s = t/T$$

In our case, we will enter the couplings in our MATLAB environment







PHYC90045 Introduction to Quantum Computing

Energy for our MAX-CUT example

s	Line 1 (Orange)	Line 2 (Blue)	Line 3 (Green)	Line 4 (Yellow)	Line 5 (Red)	Line 6 (Purple)	Line 7 (Brown)	Line 8 (Grey)	Line 9 (Light Blue)	Line 10 (Dark Blue)
0.0	4.5	3.0	-1.0	1.0	0.5	0.5	1.0	0.5	-5.0	0.5
0.2	3.8	2.8	-0.5	0.8	0.2	0.2	0.8	0.2	-4.5	0.2
0.4	3.5	2.5	-0.2	0.6	0.1	0.1	0.6	0.1	-4.0	0.1
0.6	3.2	2.3	-0.1	0.4	0.0	0.0	0.4	0.0	-3.5	0.0
0.8	3.0	2.1	-0.05	0.2	0.05	0.05	0.2	0.05	-3.0	0.05
1.0	3.2	2.3	-0.1	0.4	0.0	0.0	0.4	0.0	-4.5	0.0

Prepare in $|-\rangle^{\otimes 5}$ state

In ground state,
solving computational problem



PHYS90045 Introduction to Quantum Computing

The University of
MELBOURNE

Read Ground State configuration

After the adiabatic evolution we read the state of the quantum computer. Provided we have changed our Hamiltonian slowly enough, we will be in the ground state:

$$|01001\rangle \quad \text{or} \quad |10110\rangle$$



PHYC90045 Introduction to Quantum Computing

Adiabatic from Start to Finish

(1) Map computational problem to QUBO/Hamiltonian
 (2) Embed problem on physical architecture
 (3) Execute the adiabatic algorithm
 (4) Read the ground state configuration for the answer to the problem

PHYC90045 Introduction to Quantum Computing

MATLAB environment in the lab

PHYC90045 Introduction to Quantum Computing

D-Wave systems

- Opportunity to use them in this week's lab!
- Note: Friday lab times changed for this week only.

PHYC90045 Introduction to Quantum Computing

Week 9



Lecture 17
Optimization problems, Encoding problems as energies,
Quadratic Binary Optimization (QUBO), Problem embedding

Lecture 18
Adiabatic Quantum Computing

Lab 9
Using DWave machines

PHYC90045 Introduction to Quantum Computing

Week 10

Lecture 19
Quantum Approximate Optimization Algorithm (QAOA),
Variational Quantum Eigensolver (VQE), classical feedback

Lecture 20
Exponentials, and Quantum Optimization

Lab 10
Optimization problems

PHYC90045 Introduction to Quantum Computing

Hybrid Quantum/Classical Optimization Algorithms

Physics 90045
Lecture 19

PHYC90045 Introduction to Quantum Computing

Overview

This lecture we will talk about two algorithms to find the minimum energy of a quantum system:

- Quantum Approximate Optimization Algorithm (QAOA) algorithm
- Variational Quantum Eigen-solver (VQE) algorithm

Both algorithms are closely related, combining classical optimization with quantum mechanical states.

Kaye 8.5
Rieffel 13.4.2

PHYC90045 Introduction to Quantum Computing

Review of last lecture

The cost function is visualized as a smooth, single-peaked surface. An arrow labeled "Encode" points from the plot to a mathematical equation below.

Ising coupling

local "field"

$$H = \sum_{i \neq j} J_{ij} Z_i Z_j + \sum_i B_i Z_i$$

We can encode problems in the energy of the system, but we had no way of minimizing the energy. Today we will see a hybrid technique which allows us to minimize the energy of a quantum system.

PHYC90045 Introduction to Quantum Computing

Recall Total Energy of the System

Consider a system that has an energy function:

$$E = J_{12}z_1z_2 + J_{23}z_2z_3 + J_{13}z_1z_3 + B_1z_1 + B_2z_2 + B_3z_3$$

where the z_i are +/-1, and the J's and B's are specific parameters defining the particular problem at hand.

To get ready to map to a QC, we write the total energy as the operator "H" (which physicists would call the "Hamiltonian") on a system of qubits as a sum of these terms with $z_i \rightarrow Z_i$ (where Z_i is the Z operator on the i th qubit):

$$H = J_{12}Z_1Z_2 + J_{23}Z_2Z_3 + J_{13}Z_1Z_3 + B_1Z_1 + B_2Z_2 + B_3Z_3$$

PHYC90045 Introduction to Quantum Computing

Mapping the Spin Glass form to QC

Optimisation problems can often be cast into an equivalent "spin glass" form:

$$E = \sum_{i \neq j} J_{ij} z_i z_j + \sum_i B_i z_i$$

-> convert to a convenient form to map onto a quantum computer:

Ising coupling

local "field"

$$H = \sum_{i \neq j} J_{ij} Z_i Z_j + \sum_i B_i Z_i$$

The Z_i are now operators defined as per our definitions with eigenvalues +/-1 (which can be mapped to binary variables 0/1)

PHYC90045 Introduction to Quantum Computing

QUBO Problems

QUBO stands for "Quadratic Unconstrained Binary Optimization"

The cost function (which we want to minimize) is:

$$E(x_1, \dots, x_n) = \sum_i c_i x_i + \sum_{i,j} Q_{ij} x_i x_j$$

Where x_i are Boolean (binary) variables, either 0 or 1.
NB. we will use x for binary, z for +/-1

Quadratic term

PHYC90045 Introduction to Quantum Computing

Binary to energy

Typically when we write such energy functions we write in terms of the Z variables:

$$z_i \quad (\text{lower case } z)$$

But the binary variables in terms of 0 or 1:

$$x_i = 0 \quad \text{or} \quad x_i = 1$$

Can convert between x_i and z_i using:

$$x_i \rightarrow \frac{z_i + 1}{2}$$

PHYC90045 Introduction to Quantum Computing

Example: Number partitioning

Given a set, S , of numbers:

1, 3, 8, 10, 6, 5, 5

Is there a partition of this set of numbers into two disjoint subsets R and $S - R$, such that the sum of the elements in both sets is the same?

Yes (in this case): {1, 8, 10} and {3, 6, 5, 5}

PHYC90045 Introduction to Quantum Computing

Graph Partitioning to QUBO

1, 3, 8, 10, 6, 5, 5

We assign a qubit to each number in the problem set.
The qubit being zero indicates it is one subset.
The qubit being one indicates it is in the other.

Qubits are $|0\rangle$ if they're in subset 0, $|1\rangle$ if they're in subset 1

PHYC90045 Introduction to Quantum Computing

Number partitioning as a QUBO problem

1, 3, 8, 10, 6, 5, 5

As an optimization problem: We want

$$\sum_i w_i z_i = 0$$

The i^{th} number ± 1

Unfortunately if we just minimize this, all the qubits will end up with $z_i = -1$!

PHYC90045 Introduction to Quantum Computing

Number partitioning as a QUBO problem

But if we square, we should get a positive solution (or zero). We want to find the assignment of spins which has the minimum energy (ie. closest to zero):

$$H = \left(\sum_i w_i Z_i \right)^2 = \sum_{i \neq j} 2w_i w_j Z_i Z_j + \sum_i w_i^2 I$$

Coupling is the product of numbers

Eg. For the set {1, 2, 3}:

$H = 4Z_1 Z_2 + 6Z_1 Z_3 + 12Z_2 Z_3 + 14I$

Finding minimum energy state will solve the problem!

PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

Solution for our Number Partitioning

$$H = 4Z_1Z_2 + 6Z_1Z_3 + 12Z_2Z_3 + 14I$$

Two degenerate solutions:

$$|110\rangle \quad |001\rangle$$

1,2

3

$$E = 4 - 6 - 12 + 14 = 0$$

And of course, they correctly partition the numbers: $1+2=3$

Other combinations go worse, eg, $|111\rangle$

$$E = 4 + 6 + 12 + 14 = 36$$

PHYC90045 Introduction to Quantum Computing

PHYC90045 Introduction to Quantum Computing

Structure of Algorithm

QUI/Quantum Computer

- (1) Prepare a trial state $|\psi(\theta)\rangle$
on the quantum computer, where θ represents angles in phase and X rotations.
- (2) Measure the solution in z basis to obtain the energy, E.
- (3) For small depth circuits can analytically calculate optimal values for θ . Original paper found these for a MAX-CUT problem.

The University of Melbourne

PHYC90045 Introduction to Quantum Computing

QAOA Trial state

The QAOA trial state has a particular form:

$$R_x(\theta_k)P(\alpha_k) \dots R_x(\theta_2)P(\alpha_2)R_x(\theta_1)P(\alpha_1) |+ \rangle^{\otimes n}$$

X-rotations on every qubit Phase gate, closely related to the problem Hamiltonian/Energy

Preparation of a trial state, k=2 steps

PHYC90045 Introduction to Quantum Computing

Phase gates in QAOA

For every Z term in the Hamiltonian:

$H = E_Z Z$

For every ZZ term in the Hamiltonian:

$H = E_{ZZ} Z Z Z$

Angles all proportional to their term in the Hamiltonian:

$$\theta_{ZZ} = \alpha E_{ZZ} \quad \theta_Z = \alpha E_Z \quad \text{etc}$$

PHYC90045 Introduction to Quantum Computing

X-rotations

The QAOA trial state has a particular form:

$$R_x(\theta_k)P(\alpha_k) \dots R_x(\theta_2)P(\alpha_2)R_x(\theta_1)P(\alpha_1) |+ \rangle^{\otimes n}$$

X-rotations on every qubit

Preparation of a trial state, k=1 steps

PHYC90045 Introduction to Quantum Computing

X-rotations

The QAOA trial state has a particular form:

$$R_x(\theta_k)P(\alpha_k)\dots R_x(\theta_2)P(\alpha_2)R_x(\theta_1)P(\alpha_1)|+\rangle^{\otimes n}$$

X-rotations on every qubit

Preparation of a trial state, $k=2$ steps

PHYC90045 Introduction to Quantum Computing

Measuring the energy

The Hamiltonian of a QUBO problem can be expressed as a sum of several terms:

$$H = \sum_{i \neq j} J_{ij} Z_i Z_j + \sum_i B_i Z_i$$

Ising coupling local "field"

For example:

$$H = B_1 Z_1 + B_2 Z_2 + J_{12} Z_1 Z_2$$

A better approximation to the ground state (depending on the number of steps, k , and the choice of angles) will have a higher probability of measuring the minimum energy, and the lowest energy state.

PHYC90045 Introduction to Quantum Computing

Measure to find energy

Measure Z:

In the QUI you can look directly at probabilities rather than measure and building statistics!

Unlike for QEC codes, you don't need to add an extra qubit to measure correlations. Eg. For ZZ you can just multiply yourself for a given state!

PHYC90045 Introduction to Quantum Computing

Multiple measurements



Original function we would like to minimize:

$$H = B_1 Z_1 + B_2 Z_2 + J_{12} Z_1 Z_2$$

Take many samples, to determine:

$$\langle H \rangle = B_1 \langle Z_1 \rangle + B_2 \langle Z_2 \rangle + J_{12} \langle Z_1 Z_2 \rangle$$

PHYC90045 Introduction to Quantum Computing

Structure of Algorithm



Q#/Quantum Computer

(1) Prepare a trial state $|\psi(\theta)\rangle$
on the quantum computer, where θ represents angles in phase and X rotations.

(2) Measure the solution in z basis to obtain the energy, E.

(3) For small depth circuits can analytically calculate optimal values for θ . Original paper found these for a MAX-CUT problem. However, prohibitative to calculate analytically for larger depth circuits.

PHYC90045 Introduction to Quantum Computing

Structure of Algorithm



Q#/Quantum Computer

(1) Prepare a trial state $|\psi(\theta)\rangle$
on the quantum computer, where θ can be any adjustable gate parameter.

(2) Measure the expectation value of the energy, E.

(3) Use a classical optimization technique such as the Nelder–Mead simplex method, determine new values of θ that decrease E.

Repeat these steps until the value of the energy converges

Classical

PHYC90045 Introduction to Quantum Computing

VQE overview

VQE = Variational Quantum Eigensolver

We need not confine ourselves to Ising/Z terms. Many interesting quantum systems have a Hamiltonian, H, which includes terms in X and Y (and might not even come from a qubit system at all!).

Or perhaps we have a Hermitian matrix, and we just want to find the smallest eigenvalue. How do we do that?

- 1) Pick a convenient basis to express H
- 2) Write out the total energy, H, as a matrix
- 3) Convert your matrix a linear combination of Pauli matrices
- 4) Parameterize an “Ansatz” wavefunction candidate solution to problem
- 5) Use hybrid classical/quantum optimization to find the lowest energy and lowest energy state.

PHYC90045 Introduction to Quantum Computing

Structure of Algorithm

Quantum Computer

- (1) Prepare a trial state $|\psi(\theta)\rangle$ on the quantum computer, where θ can be any adjustable gate parameter.
- (2) Measure the expectation value of the energy, E.
- (3) Use a **classical optimization** technique such as the Nelder–Mead simplex method, determine new values of θ that decrease E.

Repeat these steps until the value of the energy converges

Classical

PHYC90045 Introduction to Quantum Computing

Hamiltonian as linear combination of Pauli operators

Always possible decompose a matrix as a sum of Paulis. If you have a matrix only:

$$E_i = \frac{\text{Tr} [\sigma_i H]}{d}$$

Where d is the dimension of the system, H is the Hamiltonian and σ_i is the Pauli. If the matrix is Hermitian, the co-efficients you find, E_i , should be real.

Express the Hamiltonian as a sum of Paulis:

$$H = \sum_i E_i \sigma_i$$

For example:

$$H = B_1 X_1 + B_2 X_2 + J_{12} Z_1 Z_2$$

PHYC90045 Introduction to Quantum Computing

Find the expectation value of H

We can express the expectation value of the energy as a sum of expectation values of the Paulis:

$$\langle H \rangle = \sum_i E_i \langle \sigma_i \rangle$$

For our example:

$$\langle H \rangle = B_1 \langle X_1 \rangle + B_2 \langle X_2 \rangle + J_{12} \langle Z_1 Z_2 \rangle$$

For a given trial state, these can be found directly from experiment (or through the QUI)

PHYC90045 Introduction to Quantum Computing

Reminder: How to measure Paulis

Measure X:

Measure Y:

Measure Z:

In the QUI you record probabilities rather than measure and building statistics!

Unlike for QEC codes, you don't need to add an extra qubit to measure correlations. Eg. For ZZ you can just multiply yourself given the state which is measured!

PHYC90045 Introduction to Quantum Computing

VQE: Jordan Wigner Transformation

A classic of physics from 1928, by Jordan and Wigner. You've got a system of qubits. You want to use it to simulate fermions (eg. someone's given you a chemistry problem involving electrons). Electrons do not behave the same as qubits.

How do we do this?

Tempting solution:

$$\sigma_j^+ \rightarrow \frac{X_j + iY_j}{2} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}_j = f_j^+ \quad \text{Create a fermion at the } j^{\text{th}} \text{ site?}$$

$$\sigma_j^- \rightarrow \frac{X_j - iY_j}{2} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}_j = f_j^- \quad \text{Destroy a fermion at the } j^{\text{th}} \text{ site?}$$

This is close but **WRONG!** The commutation relations between different sites are wrong (fermions anti-commute). $[f_j, f_k] = 0$

PHYC90045 Introduction to Quantum Computing

Jordan Wigner Transformation



Correct solution:

$$\sigma_j^+ \rightarrow Z_1 Z_2 \dots Z_{j-1} \frac{X_j + iY_j}{2}$$

$$\sigma_j^- \rightarrow Z_1 Z_2 \dots Z_{j-1} \frac{X_j - iY_j}{2}$$

Jordan and Wigner: Images from Wikipedia

PHYC90045 Introduction to Quantum Computing

Picking a VQE Ansatz



QAOA State
Use a combination of X rotations and phase rotations.

Adiabatic Methods
Slowly vary the Hamiltonian, in a parameterized way, to obtain an approximation to the ground state.

Coupled Cluster Methods
First start in a (often unentangled) reference state, which can be calculated classically, eg. with mean field methods.
Consider successively more complicated perturbations away from this reference state: First we consider just (parameterized) single qubit rotations away from the reference state. Then we consider unitaries with both single qubit rotations and two body interactions, then one, two and three body interactions. As with QAOA we can consider several rounds of interaction.

PHYC90045 Introduction to Quantum Computing

Nelder-Mead Classical Optimization



A classical method of optimization.

- Classical optimization technique
- Requires only the calculation of few points at each step

Based on simplex:

A simplex S in n dimension is defined as the convex hull of n+1 vertices: x_0, \dots, x_n

Eg. Triangle in 2D 

PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

Nelder-Mead operations

- Start with an initial simplex
- Repeat until the convergence is reached:
 - Test if we've reached convergence
 - If not then transform the working simplex

Four types of transformation to test:

1) Reflection



Centroid of all other points

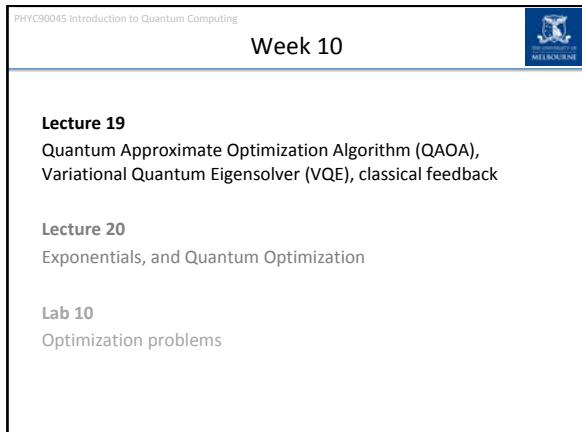
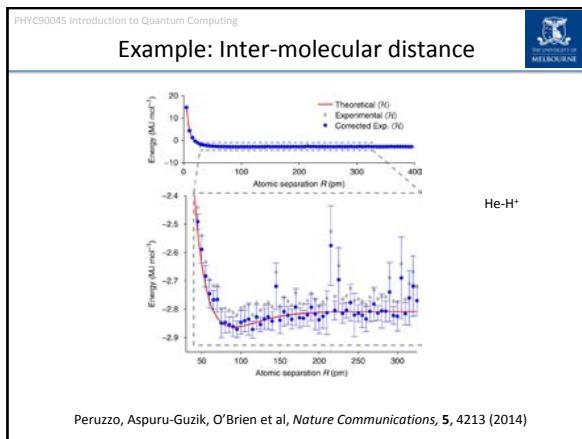
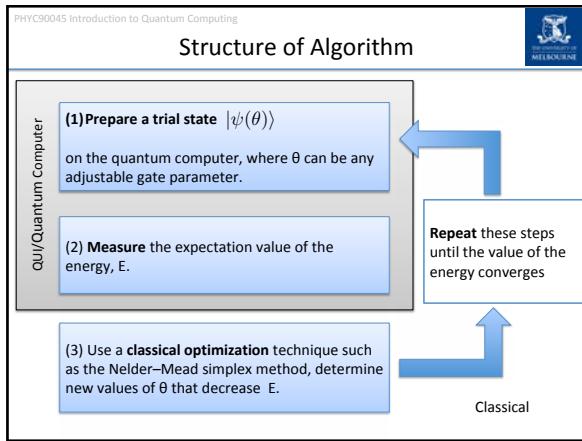
Worst point

PHYC90045 Introduction to Quantum Computing

Nelder-Mead operations

The diagram illustrates three Nelder-Mead operations:

- 2) Expand:** A blue triangle represents a simplex. A red arrow labeled "Centroid of all other points" points from the center of the triangle to a point labeled "Worst point" on one of its edges.
- 3) Contract:** A blue triangle represents a simplex. A red arrow labeled "Centroid of all other points" points from the center of the triangle to a point labeled "Worst point" on one of its edges.
- 4) Shrink:** A blue triangle represents a simplex.



PHYC90045 Introduction to Quantum Computing

Week 10



Lecture 19
Quantum Approximate Optimization Algorithm (QAOA),
Variational Quantum Eigensolver (VQE), classical feedback

Lecture 20
Exponentials, and Quantum Optimization

Lab 10
Optimization problems

PHYC90045 Introduction to Quantum Computing



Matrix Exponentiation

Physics 90045
Lecture 19

PHYC90045 Introduction to Quantum Computing

Overview



This lecture we will introduce useful tools for mapping problems to a quantum computing framework:

- Power series
- Gates as exponentials
- Rotations as exponentials
- BCH Formula
- “Trotter” to add exponents
- Cartan Decomposition

Kaye 8.5
Reiffel 13.4.2

PHYC90045 Introduction to Quantum Computing

Hamiltonians and gates...

We've been using matrices to represent both gates and Hamiltonians.

Gates on the QC

Operators are represented by a **unitary matrix**

$$V^\dagger V = I$$

Operators form a *group*.

Problem Hamiltonians

$$H = J_{12}Z_1Z_2 + J_{23}Z_2Z_3 + J_{13}Z_1Z_3 + B_1Z_1 + B_2Z_2 + B_3Z_3$$

Observables (like total energy) are represented by a **Hermitian matrix**

$$H^\dagger = H$$

Observables form an algebra.

Beware, " H " is used for Hadamard and Hamiltonian, but they are different!

PHYC90045 Introduction to Quantum Computing

Exponentiation of Operators

There is a deep relationship in QM between the Hamiltonian and gates in a circuit.

A circuit represents “evolution” of a quantum state through the action of gate operators.

In general a system evolves via an operator U through the exponentiation of an underlying Hamiltonian:

$$U = \exp(iHt)$$

Evolution operator

e.g. a gate in the circuit

Hamiltonian – two types

1. Of the QC itself to implement a gate, e.g. Z

2. Of the problem being mapped to the QC

We will first look at what exponentiation of operators (matrices) means mathematically...

PHYC90045 Introduction to Quantum Computing

Power series

Power series (Taylor/Maclaurin series):

$$\exp(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

Power series for trigonometric functions:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} \dots$$

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} \dots$$

PHYC90045 Introduction to Quantum Computing

Complex Numbers: Exponential form

We can use this to prove useful things. For example consider:

$$\exp(i\theta) = 1 + i\theta - \frac{\theta^2}{2} - i\frac{\theta^3}{3!} + \frac{\theta^4}{4!} + i\frac{\theta^5}{5!} - \dots$$

Comparing to power series for trigonometric functions:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} \dots$$

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} \dots$$

We can prove that

$$\exp(i\theta) = \cos \theta + i \sin \theta$$

PHYC90045 Introduction to Quantum Computing

Matrix Exponentiation

By analogy we can define the exponential of a matrix to be:

$$\exp(A) = I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \dots$$

Matrix equation

Compare to the equation for real/complex numbers

$$\exp(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

PHYC90045 Introduction to Quantum Computing

Rotations as a matrix exponential

Consider:

$$\exp(i\theta Z) = I + i\theta Z - \frac{\theta^2 Z^2}{2} - i\frac{\theta^3 Z^3}{3!} + \frac{\theta^4 Z^4}{4!} + i\frac{\theta^5 Z^5}{5!} - \dots$$

Using the fact that $Z^2 = I$

$$\begin{aligned} \exp(i\theta Z) &= I + i\theta Z - \frac{\theta^2}{2} I - i\frac{\theta^3}{3!} Z + \frac{\theta^4}{4!} I + i\frac{\theta^5}{5!} Z - \dots \\ &= (1 - \frac{\theta^2}{2} + \frac{\theta^4}{4!} + \dots)I + i(\theta - \frac{\theta^3}{3!} + \dots)Z \\ &= \cos(\theta)I + i\sin(\theta)Z \end{aligned}$$

i.e. the rotation matrix around Z-axis!

PHYC90045 Introduction to Quantum Computing

Z-rotations

$\exp(i\theta Z) = I + i\theta Z - \frac{\theta^2}{2}I - i\frac{\theta^3}{3!}Z + \frac{\theta^4}{4!}I + i\frac{\theta^5}{5!}Z - \dots$

$$= (1 - \frac{\theta^2}{2} + \frac{\theta^4}{4!} + \dots)I + i(\theta - \frac{\theta^3}{3!} + \dots)Z$$

$$= \cos(\theta)I + i\sin(\theta)Z$$

$R_z(\theta) = \exp\left(-\frac{i\theta}{2}Z\right)$

Compare with our R-gate (with zero global phase) $R_z(\theta_R) = e^{i\theta_g} \left(I \cos\left(\frac{\theta_R}{2}\right) - iZ \sin\left(\frac{\theta_R}{2}\right) \right)$

put in -ve sign

PHYC90045 Introduction to Quantum Computing

Rotations around other axes

The only fact we used the fact that Z squares to the identity. Other axes work in a similar way.

$R_x(\theta) = \exp\left(-\frac{i\theta}{2}X\right)$

$R_y(\theta) = \exp\left(-\frac{i\theta}{2}Y\right)$

$R_z(\theta) = \exp\left(-\frac{i\theta}{2}Z\right)$

PHYC90045 Introduction to Quantum Computing

Arbitrary axis rotation as matrix exponential

Consider any unit vector, \hat{n} :

$$(n_x X + n_y Y + n_z Z)^2 = (n_x^2 + n_y^2 + n_z^2)I +$$

$$n_x n_z (XZ + ZX) + n_y n_z (YZ + ZY) + n_x n_z (XZ + ZX)$$

$$= (n_x^2 + n_y^2 + n_z^2)I$$

$$= I$$

Like Z, this squares to the identity.

An arbitrary rotation of one qubit can be expressed:

$R_n(\theta) = \exp\left(-\frac{i\theta}{2}\hat{n} \cdot \sigma\right) = \cos\frac{\theta}{2} - i\sin\frac{\theta}{2}(\hat{n} \cdot \sigma)$

PHYC90045 Introduction to Quantum Computing

Exponentiation of two qubit operators

The University of Melbourne

Not only do single qubit operators square to the identity, we can consider:

If $(Z \otimes Z)(Z \otimes Z) = I$

So just like for a single qubit rotations, we could use a power series to show:

Then $\exp(i\theta Z \otimes Z) = \cos(\theta)I + i\sin(\theta)Z \otimes Z$

PHYC90045 Introduction to Quantum Computing

Eigenvalues and exponentiation

The University of Melbourne

Consider an eigenvalue decomposition of a matrix, A

$A = VDV^\dagger$

Here V is unitary, D is diagonal and real. We can find powers:

$$\begin{aligned} A &= VDV^\dagger \\ A^2 &= VDV^\dagger VDV^\dagger = VD^2V^\dagger \\ A^3 &= \dots = VD^3V^\dagger \\ A^n &= VD^nV^\dagger \end{aligned}$$

And taking powers of a diagonal matrix is the same as taking the power of each of the diagonal entries.

PHYC90045 Introduction to Quantum Computing

Applied to exponentiation

The University of Melbourne

U is unitary, D is diagonal with the entries on the diagonal equal to the eigenvalues.

$$\exp(A) = I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \frac{A^4}{4!} + \dots$$

So, using the powers of A from the previous slide:

$$\begin{aligned} \exp(A) &= I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \frac{A^4}{4!} + \dots \\ &= V \left(I + D + \frac{D^2}{2!} + \frac{D^3}{3!} + \frac{D^4}{4!} + \dots \right) V^\dagger \\ &= V \exp(D) V^\dagger \end{aligned}$$

Simply exponentiate the eigenvalues/diagonal of D matrix.

PHYC90045 Introduction to Quantum Computing

Conjugating with V and V inverse

$\exp(A) = \exp(VDV^\dagger) = V \exp(D)V^\dagger$

Conjugating in the exponent Conjugating with a gate and its inverse
eg. In QUI.

PHYC90045 Introduction to Quantum Computing

Example: CZ gate

$$\begin{aligned} CZ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \\ &= \begin{bmatrix} e^{i0} & 0 & 0 & 0 \\ 0 & e^{i0} & 0 & 0 \\ 0 & 0 & e^{i0} & 0 \\ 0 & 0 & 0 & e^{i\pi} \end{bmatrix} \\ CZ &= \exp \left(i\pi \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \right) \end{aligned}$$

PHYC90045 Introduction to Quantum Computing

Matrix as linear combination of Pauli operators

Always possible decompose a matrix as a sum of Pauli products. If you have a matrix only:

$$E_i = \frac{\text{Tr} [\sigma_i H]}{d} \quad \sigma_i = XI, IX, ... XZ, ... ZZ$$

Where d is the dimension of the system (d=4 for 2 qubits), H is the Hamiltonian and σ_i is the Pauli. If the matrix is Hermitian, the co-efficients you find, E_i , should be real.

Express the Hamiltonian as linear combination of Pauli matrix products:

$$H = \sum_i E_i \sigma_i$$

For example: $H = B_1 X_1 + B_2 X_2 + J_{12} Z_1 Z_2$

PHYC90045 Introduction to Quantum Computing

CZ Gate continued

$CZ = \exp\left(i\pi \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\right)$

$\frac{II - ZI - IZ + ZZ}{4}$

$CZ = \exp\left(i\frac{\pi}{4}II - i\frac{\pi}{4}ZI - i\frac{\pi}{4}IZ + i\frac{\pi}{4}ZZ\right)$

All the terms commute, so

$CZ = \exp\left(i\frac{\pi}{4}II\right) \exp\left(-i\frac{\pi}{4}ZI\right) \exp\left(-i\frac{\pi}{4}IZ\right) \exp\left(+i\frac{\pi}{4}ZZ\right)$

Global phase Single qubit rotations Interaction

PHYC90045 Introduction to Quantum Computing

CZ Circuit

$CZ = \exp\left(i\frac{\pi}{4}II\right) \exp\left(-i\frac{\pi}{4}ZI\right) \exp\left(-i\frac{\pi}{4}IZ\right) \exp\left(+i\frac{\pi}{4}ZZ\right)$

Using $R_z(\theta) = \exp\left(-\frac{i\theta}{2}Z\right)$

PHYC90045 Introduction to Quantum Computing

Example: CNOT gate in exponent form

$CZ = \exp\left(i\frac{\pi}{4}II\right) \exp\left(-i\frac{\pi}{4}ZI\right) \exp\left(-i\frac{\pi}{4}IZ\right) \exp\left(+i\frac{\pi}{4}ZZ\right)$

Global phase Single qubit rotations Interaction

We can work out how CNOT can be expressed as an exponent:

$$\begin{aligned} CNOT &= I \otimes H \quad CZ \quad I \otimes H \\ &= I \otimes H \quad \exp\left(i\pi \frac{II - ZI - IZ + ZZ}{4}\right) \quad I \otimes H \\ &= \exp\left(i\pi \frac{II - ZI - IX + ZX}{4}\right) \end{aligned}$$

PHYC90045 Introduction to Quantum Computing

Cartan Decomposition

Known as the Cartan or KAK decomposition:

The interacting part, V, depends on just three parameters:

$$V = \exp(i\theta_x XX + i\theta_y YY + i\theta_z ZZ)$$

All of these terms commute, so the order doesn't matter. Can be implemented separately.

PHYC90045 Introduction to Quantum Computing

Warning about non-commuting operators

Warning:

$$\exp(A) \exp(B) \neq \exp(A + B)$$

Unless A and B commute! (see CZ example)

PHYC90045 Introduction to Quantum Computing

Baker-Campbell-Hausdorff formula

First few terms of the BCH formula:

$$\exp(A) \exp(B) = \exp \left(A + B + \frac{1}{2}[A, B] + \frac{1}{12}([A, [A, B]] + [B, [B, A]] + \dots) \right)$$

Higher order terms involve commutators

Where the commutator is given by

$$[A, B] = AB - BA$$

PHYC90045 Introduction to Quantum Computing

Question for you



$[X, Y] = ?$
 $[Y, Z] = ?$
 $[Z, X] = ?$

Given:

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

$[A, B] = AB - BA$

PHYC90045 Introduction to Quantum Computing

Commutators of Paulis



$[X, Y] = 2iZ$
 $[Y, Z] = 2iX$
 $[Z, X] = 2iY$

PHYC90045 Introduction to Quantum Computing

Baker-Campbell-Hausdorff formula



First few terms of the BCH formula:

$$\exp(A) \exp(B) = \exp \left(A + B + \frac{1}{2}[A, B] + \frac{1}{12}([A, [A, B]] + [B, [B, A]] + \dots) \right)$$

Higher order terms involve commutators

Where the commutator is given by

$[A, B] = AB - BA$

PHYC90045 Introduction to Quantum Computing

Trotter Approximation

But what if you do want to create the gate:

$$\exp(A + B)$$

We might try:

$$\exp(A + B) \approx \exp(A) \exp(B)$$

$$\exp(A + B) \approx \exp\left(\frac{A}{2}\right) \exp\left(\frac{B}{2}\right) \exp\left(\frac{A}{2}\right) \exp\left(\frac{B}{2}\right)$$

$$\exp(A + B) \approx \left(\exp\left(\frac{A}{n}\right) \exp\left(\frac{B}{n}\right) \right)^n$$

This is called the Trotter (sometimes Trotter-Suzuki) approximation – useful!

PHYC90045 Introduction to Quantum Computing

Lie Algebras and Lie Groups

$U = \exp(iHt)$

$SU(n)$
“Special” unitary Lie group
 $\text{Det}(U) = 1$
 U is unitary
These operations you can implement using QUI

$\mathfrak{su}(n)$
Lie algebra
 iHt is anti-Hermitian
Traceless
Like using phase in complex numbers, this can help!

PHYC90045 Introduction to Quantum Computing

Problem Hamiltonians

Now consider the “Hamiltonian” associated with a particular problem, e.g.

$$H = J_{12}Z_1Z_2 + J_{23}Z_2Z_3 + J_{13}Z_1Z_3$$

(set $t = -\alpha/2$, to absorb factor of 2)

Evolution operator is:

$$U = \exp[-iH\alpha/2] = \exp[-(i\alpha/2)(J_{12}Z_1Z_2 + J_{23}Z_2Z_3 + J_{13}Z_1Z_3)]$$

$$= \exp[-(i\alpha/2)(J_{12}Z_1Z_2)] \exp[-(i\alpha/2)(J_{23}Z_2Z_3)] \exp[-(i\alpha/2)(J_{13}Z_1Z_3)]$$

Equivalent circuit is:

circuit element $R_z(\alpha)$

equivalent Hamiltonian coupling αZZ

PHYC90045 Introduction to Quantum Computing

Problem Hamiltonians

The University of Melbourne

Now consider the "Hamiltonian" associated with a particular problem, e.g.

$$H = J_{12}Z_1Z_2 + J_{23}Z_2Z_3 + J_{13}Z_1Z_3 \quad (\text{set } t = -\alpha/2, \text{ to absorb factor of 2})$$

Evolution operator is:

$$U = \exp[-iH\alpha/2] = \exp[-(i\alpha/2)(J_{12}Z_1Z_2 + J_{23}Z_2Z_3 + J_{13}Z_1Z_3)] \\ = \exp[-(i\alpha/2)(J_{12}Z_1Z_2)] \exp[-(i\alpha/2)(J_{23}Z_2Z_3)] \exp[-(i\alpha/2)(J_{13}Z_1Z_3)]$$

This is the basis for encoding the QAOA trial state for the problem Hamiltonian:

PHYC90045 Introduction to Quantum Computing

Week 10

The University of Melbourne

Lecture 19
Quantum Approximate Optimization Algorithm (QAOA), Variational Quantum Eigensolver (VQE), classical feedback

Lecture 20
Exponentials, and Quantum Optimization

Lab 10
Optimization problems

PHYC90045 Introduction to Quantum Computing

Week 11

Lecture 21
Python, IBM's QISKit

Lecture 22
- Further quantum algorithms

Lab 11
Implementing small algorithms on IBM's 16 qubit machine

PHYC90045 Introduction to Quantum Computing

QISKit and Python

Physics 90045
Lecture 21

PHYC90045 Introduction to Quantum Computing

Overview

In Friday's laboratory we will be programming IBM's 16 qubit quantum computer. This lecture introduces the tools you can use to access it on your computer:

- Python primer
- Jupyter notebooks
- Using QISKit

PHYC90045 Introduction to Quantum Computing

On your own machine: Installing Python

Install Python3

Install Anaconda with **Python 3.6** version, download from:
<https://www.anaconda.com/download/>

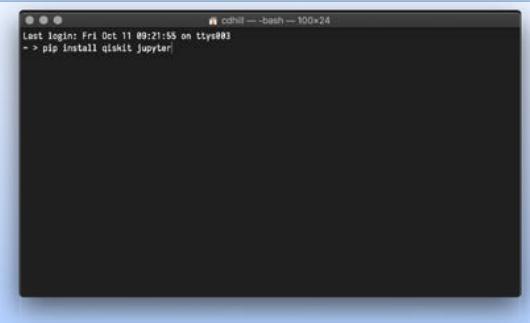
Reference:
<https://www.qiskit.org/documentation/install.html>

Quantum Computing and IBM Q #18M2



PHYC90045 Introduction to Quantum Computing

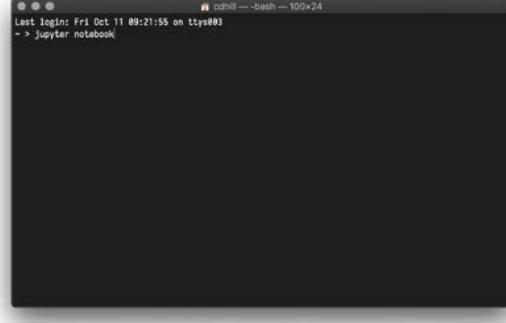
Install Qiskit



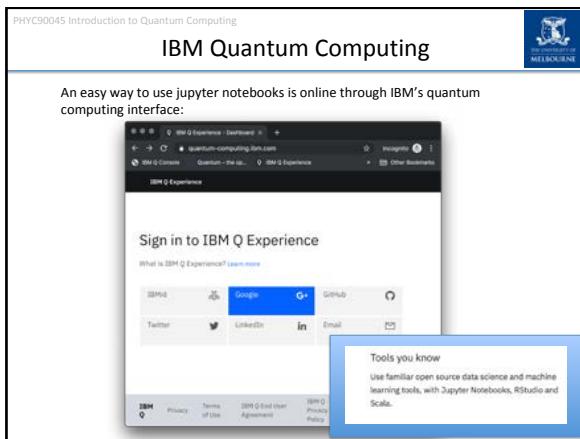
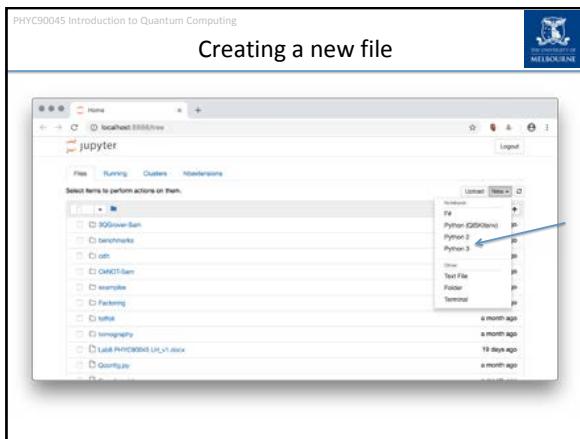
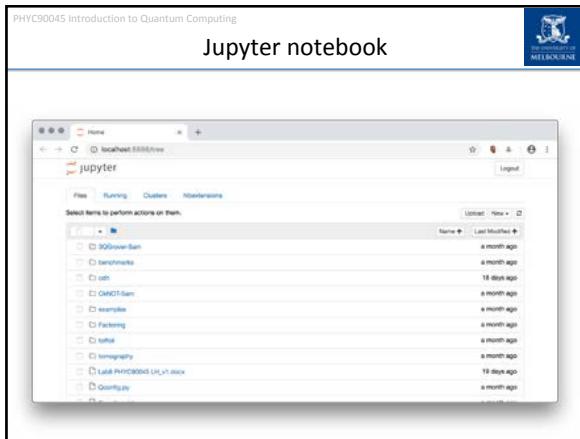
```
cdhil ~ -bash ~ 100x24
Last login: Fri Oct 11 09:21:55 on ttys003
-> pip install qiskit jupyter
```

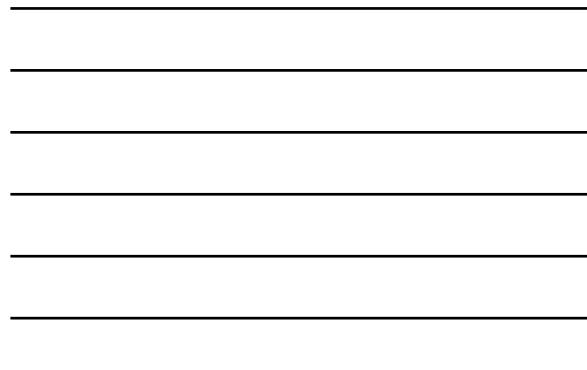
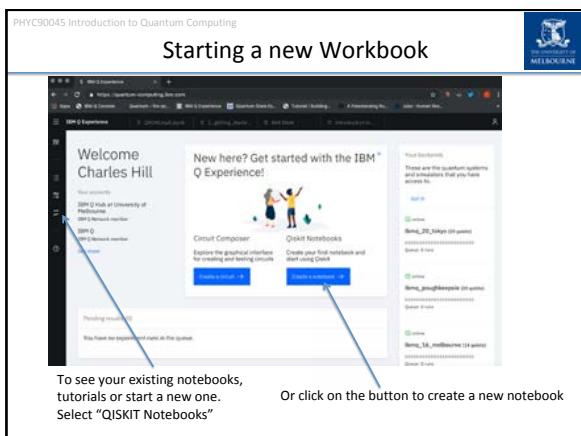
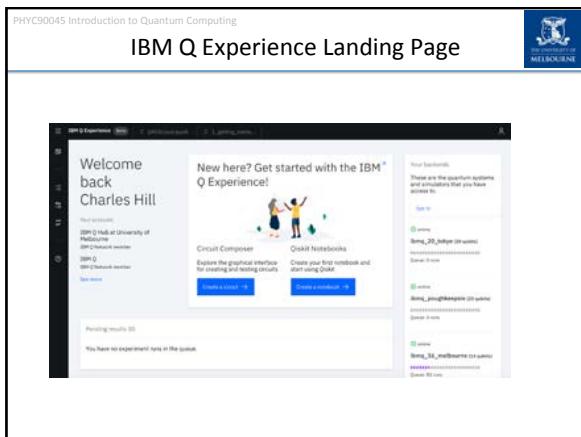
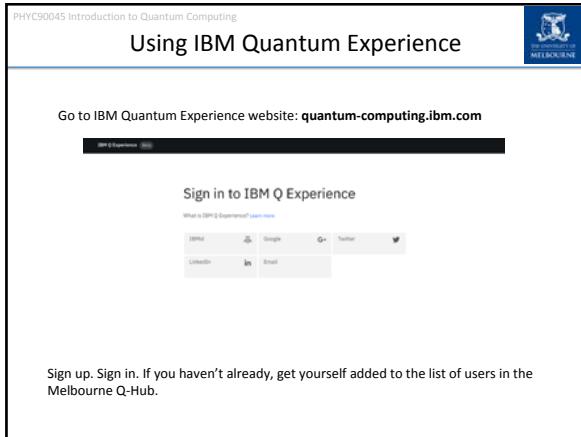
PHYC90045 Introduction to Quantum Computing

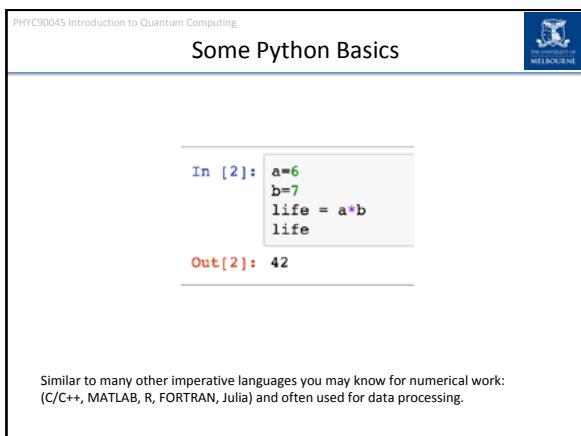
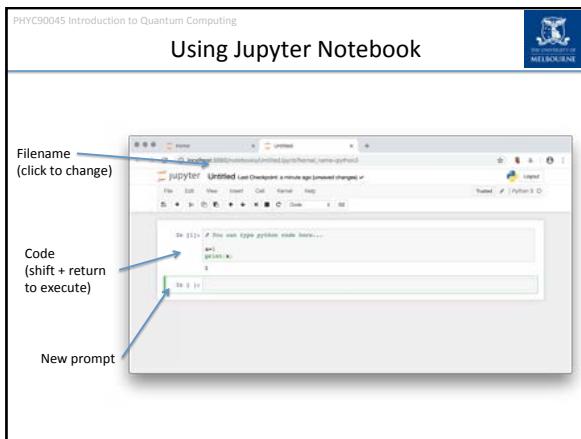
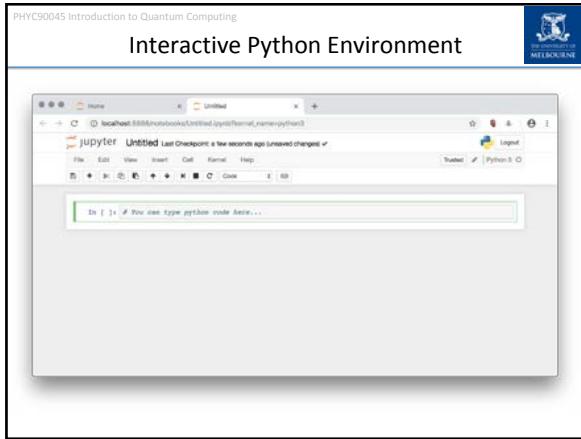
Jupyter notebook



```
cdhil ~ -bash ~ 100x24
Last login: Fri Oct 11 09:21:55 on ttys003
-> jupyter notebook
```







PHYC90045 Introduction to Quantum Computing

Defining and calling Functions

The diagram illustrates Python code examples with annotations:

- Defining a Function:** The code `def square(x):
 # This is a comment
 return x*x` is shown. Annotations explain:
 - def keyword indicates a new function
 - No types on parameters
 - Colon
 - Comment
- Function Call Examples:** Two examples are shown: `square(4)` and `square(x=4)`. Annotations explain:
 - Whitespace is significant in python.
Indentation indicates a new block.
 - No semicolons.
 - Newline is the end of a statement
- Calling a function:** The code `square(4)` is shown with the annotation "Calling a function:".
- Named parameters:** The code `square(x=4)` is shown with the annotation "Named parameters".

PHYC90045 Introduction to Quantum Computing

Lists and for loops

Lists store a sequence of values. Square brackets indicate a list:

```
[ "This", "is", "a", "list"]
primes = [2, 3, 5, 7, 11]
```

Eg. For loops often use lists:

```
for p in primes:
    print(p)
```

2
3
5
7
11

Accessing an individual element.
0-based!

```
primes[2]
```

5

PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

Dictionaries

Dictionaries store key-value pairs.

Curly braces indicate a dictionary

```
me = {"name": "Charles", "height":1.79, "favourite_food": "pizza"}  
me["favourite_food"]  
  
'pizza'
```

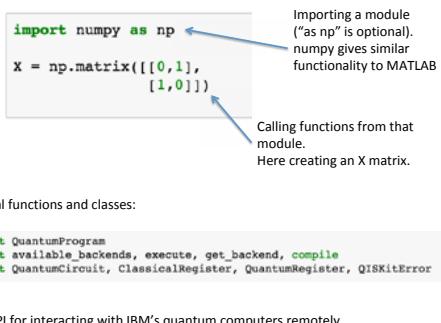
key

value

```
me["favourite_food"] = "sweet and sour pork"  
me["favourite_food"]  
  
'sweet and sour pork'
```

PHYC90045 Introduction to Quantum Computing

Importing other libraries



```
import numpy as np
X = np.matrix([[0,1],
[1,0]])
```

Importing a module ("as np" is optional). numpy gives similar functionality to MATLAB.

Calling functions from that module. Here creating an X matrix.

Or import individual functions and classes:

```
from qiskit import QuantumProgram
from qiskit import available_backends, execute, get_backend, compile
from qiskit import QuantumCircuit, ClassicalRegister, QuantumRegister, QISKitError
```

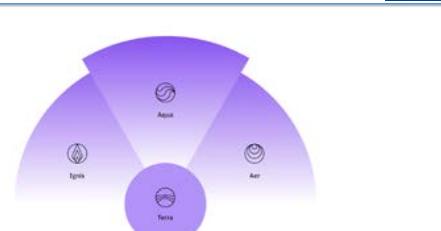
qiskit is an API for interacting with IBM's quantum computers remotely.

PHYC90045 Introduction to Quantum Computing

IBM's Qiskit

PHYC90045 Introduction to Quantum Computing

Terra, Aer, Ignis, Aqua



Tera (Earth): Access to IBM Q Devices through python interface
Aer (Air): Classical simulation of quantum algorithms/circuits
Ignis (Fire): Characterisation of errors, tomography
Aqua (Water): Large selection of quantum algorithms

PHYC90045 Introduction to Quantum Computing

QISKit Introduction



Introduction to Qiskit

PHYC90045 Introduction to Quantum Computing



Constructing a circuit in Qiskit

In this tutorial we will construct a circuit to create a Bell-state using Qiskit.

PHYC90045 Introduction to Quantum Computing



First we will load the required libraries from qiskit.

```
In [1]: %matplotlib inline
# Importing standard Qiskit libraries and configuring account
import numpy as np
from qiskit import QuantumCircuit, transpile
from qiskit.compiler import transpile, assemble
from qiskit.tools.jupyter import *
from qiskit.visualization import *
```

PHYC90045 Introduction to Quantum Computing

The University of Melbourne

Now, we will construct a quantum circuit with two qubits.

```
In [3]: # Create a Quantum Register with 2 qubits.
q = QuantumRegister(2, 'q')
c = ClassicalRegister(2, 'c')

# Create a Quantum Circuit acting on the q register
circ = QuantumCircuit(q, c)
```

PHYC90045 Introduction to Quantum Computing

The University of Melbourne

Adding gates to our circuit.

```
In [4]: # Add a H gate on qubit 0, putting this qubit in superposition.
circ.h(q[0])
# Add a CX (CNOT) gate on control qubit 0 and target qubit 1, putting
# them in a Bell state.
circ.cx(q[0], q[1])

# Measure the results
circ.measure(q, c)
```

```
Out[4]: <qiskit.circuit.instructionset.InstructionSet at 0x1244de690>
```

```
In [5]: circ.draw()
```

```
Out[5]:
```

q_0: (0)> ── H ──■── M ──
q_1: (0)> ── X ──■── M ──
c_0: 0 ───────────────────
c_1: 0 ───────────────────

PHYC90045 Introduction to Quantum Computing

The University of Melbourne

Simulating the circuit

First, let's run the circuit on a simulator.

PHYC90045 Introduction to Quantum Computing

IBM's simulators are in a package called Aer. Let's get a backend which can simulate our circuit.

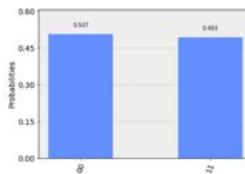
```
In [6]: backend = Aer.get_backend('qasm_simulator')
job = execute(circ, backend)
```

PHYC90045 Introduction to Quantum Computing

We can obtain samples from this circuit (from the simulator). To do this:

```
In [7]: result_sim = job.result()
counts_sim = result_sim.get_counts(circ)
plot_histogram(counts_sim)
```

Out[7]:



String	Probability
00	0.50
11	0.49

PHYC90045 Introduction to Quantum Computing

Using a real IBM Q Device

Now, let us submit the circuit to a real quantum computer, and see how it performs.

A number of different machines are available for use. Note the machines available to you might be different.

```
In [27]: print("Available backends:")
provider.backends()

Out[27]: {'ibmq_qasm_simulator': IBMQBackend('ibmq_qasm_simulator') from IBMQ(hub='ibm-q', group='open', project='main'),
          'ibmqx2': IBMQBackend('ibmqx2') from IBMQ(hub='ibm-q', group='open', project='main'),
          '<IBMQBackend("ibmq_16_melbourne") from IBMQ(hub='ibm-q', group='open', project='main')',
          '<IBMQBackend("ibmq_vigo") from IBMQ(hub='ibm-q', group='open', project='main'),
          '<IBMQBackend("ibmq_ourense") from IBMQ(hub='ibm-q', group='open', project='main')}
```

PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

PHYC90045 Introduction to Quantum Computing



 The University of
MELBOURNE

PHYC90045 Introduction to Quantum Computing

The University of Melbourne

Now, let us use the Qiskit API to submit job to IBM remotely.

```
In [11]: from qiskit.tools.monitor import job_monitor
shots = 1024 # Number of shots to run the program (experiment); maximum is 8192 shots.

job_exp = execute(qc, backend=backend, shots=shots)
job_monitor(job_exp)

Job Status: job has successfully run
```

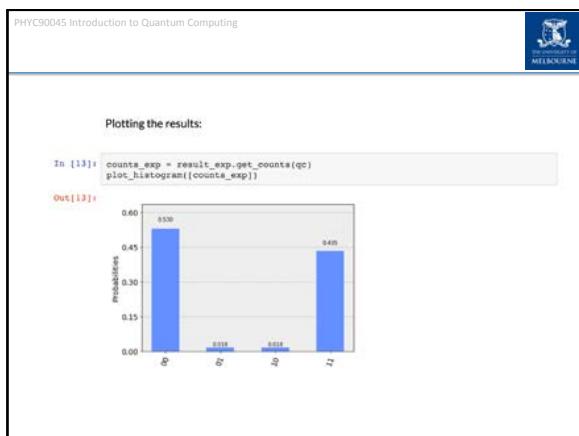
PHYC90045 Introduction to Quantum Computing

The University of Melbourne

Once the job has run (we may need to wait), we can examine the results:

```
In [12]: result_exp = job_exp.result()
result_exp
```

```
Out[12]: Result(backend_name='ibmq_vigo', backend_version='1.0.1', date=datetime.datetime
(2019, 10, 2, 12, 34, 26, tzinfo=tzutc()), execution_id='f4831262-e510-11e9-9f14-ac1efb47c31b', header=Obj('backend', name='ibmq_vigo', backend_version='1.0.1'), job_id='a0c1993a-499a-4887-9001-197d4449', qobj_id='621', label='qasm', memory='400c-075-043444', results=ExperimentResults(), data=qobj_to_dict(resultData={'c': 0, 'q0': 0, 'q1': 0, 'q2': 0, 'q3': 0, 'q4': 0, 'q5': 0, 'q6': 0, 'q7': 0}, QobjData={'c': 0, 'q0': 0, 'q1': 0, 'q2': 0, 'q3': 0, 'q4': 0, 'q5': 0, 'q6': 0, 'q7': 0}, header=Obj('clbit_label=[[\'c\', 0], [\'c\', 1]], creg_sizes=[[\'c\', 2]], memory_slots=2, n_qubits=5, name=\'circuit2\', qreg_sizes=[[\'q\', 5]], qubit_labels=[[\'q\', 0], [\'q\', 1], [\'q\', 2], [\'q\', 3], [\'q\', 4]]}, max_error_magnitude=0.0, success=True, status='Successful completion', success=True, time_taken=8.3299090440216064)
```



PHYC90045 Introduction to Quantum Computing

Now with more qubits

Let's go over that one more time with the five qubit GHZ state:

$$\frac{|00000\rangle + |11111\rangle}{\sqrt{2}}$$

```
In [14]: %matplotlib inline
# Importing standard Qiskit libraries and configuring account
import numpy as np
from qiskit import *
In [17]: # Loading your IBM Q account()
provider = IBMQ.load_account()
```

PHYC90045 Introduction to Quantum Computing

```
In [18]: # Create a Quantum Register with 5 qubits.
q = QuantumRegister(5, 'q')
c = ClassicalRegister(5, 'c')

# Create a Quantum Circuit acting on the q register
qc = QuantumCircuit(q, c)
```

PHYC90045 Introduction to Quantum Computing

```
qc.h(q[0])
qc.cx(q[0], q[1])
qc.cx(q[1], q[2])
qc.cx(q[2], q[3])
qc.cx(q[3], q[4])

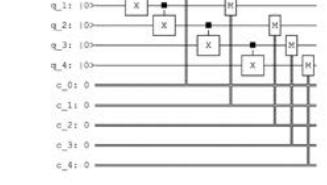
qc.measure(q, c)
```

PHYC90045 Introduction to Quantum Computing



Draw the circuit, and check everything is correct!

```
In [21]: qc.draw()
Out[21]:
```

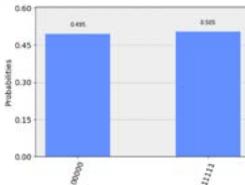


PHYC90045 Introduction to Quantum Computing



Simulate the circuit and ensure you get correct results.

```
In [23]: backend = Aer.get_backend('qasm_simulator')
job = execute(qc, backend)
result_sim = job.result()
counts_sim = result_sim.get_counts(qc)
plot_histogram([counts_sim])
Out[23]:
```



PHYC90045 Introduction to Quantum Computing

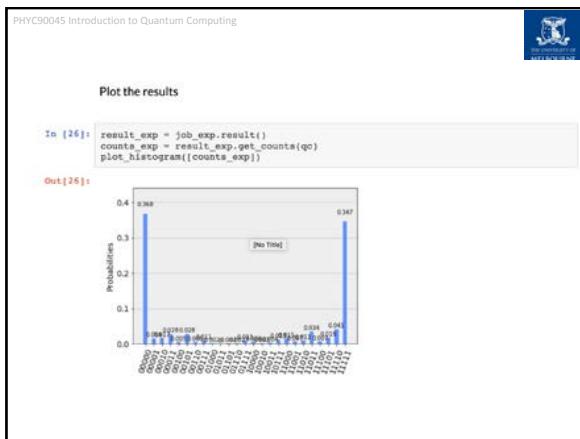


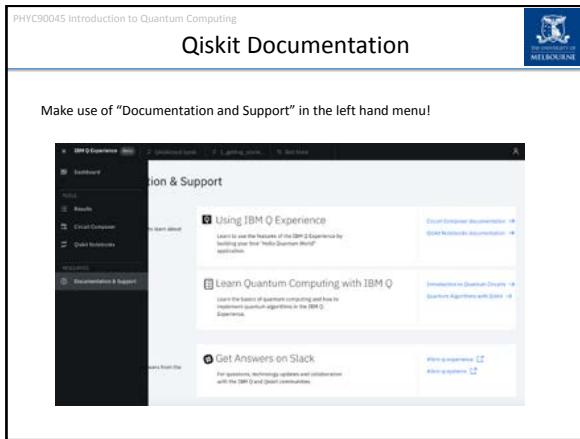
When everything is correct, submit the job to IBM Q Device to run:

```
In [24]: from qiskit.providers.ibmq import least_busy
large_enough_devices = provider.backends(filters=lambda x: x.configuration().n_qubits < 10 and
                                          not x.configuration().simulator)
backend = least_busy(large_enough_devices)
print("The best backend is " + backend.name())
The best backend is ibmq_vigo
```

PHYC90045 Introduction to Quantum Computing

The University of
MELBOURNE





PHYC90045 Introduction to Quantum Computing

Week 11



Lecture 21
Python, IBM's QISKit

Lecture 22
- Further quantum algorithms

Lab 11
Implementing small algorithms on IBM's 16 qubit machine

PHYC90045 Introduction to Quantum Computing

Week 11



Lecture 21
Python, IBM's QISKit

Lecture 22
- Further quantum algorithms – HHL algorithm

Lab 11
Implementing small algorithms on IBM's 16 qubit machine

PHYC90045 Introduction to Quantum Computing



Solving Linear Equations

Physics 90045
Lecture 23

PHYC90045 Introduction to Quantum Computing

Overview



This lecture we will introduce our final quantum computing algorithm for the course:

- Solving Linear Equations
- Mapping Linear Equations to a Quantum Computer
- Solving them: The HHL Algorithm

Introduction paper:
<https://arxiv.org/pdf/1802.08227.pdf>

PHYC90045 Introduction to Quantum Computing

Solving Linear Equations is Ubiquitous!

The University of Melbourne

So many areas of application:

- Finance
- Engineering (eg. Electronics, civil engineering)
- Defence (eg. Radar)
- Science (Least squares optimization, ODE solving...)
- Machine learning, control theory
- ... so many more

PHYC90045 Introduction to Quantum Computing

Linear Equations

The University of Melbourne

$$\begin{aligned} 2x + y &= 5 \\ x - 2y &= 0 \end{aligned}$$

What is the solution for x and y?

PHYC90045 Introduction to Quantum Computing

Solving Linear Equations

The University of Melbourne

Write as a matrix equation:

$$Ax = b$$

In our case:

$$\begin{bmatrix} 2 & 1 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$$

Can solve by row reduction, or by finding the inverse of A.
We will find the inverse of A.

PHYC90045 Introduction to Quantum Computing

Calculating Matrix Inverse Using Eigenvalues

We would like to invert the matrix A:

$$A = \begin{bmatrix} 2 & 1 \\ 1 & -2 \end{bmatrix}$$

To find the eigenvalues, we will first write the characteristic equation:

$$|A - \lambda I| = \begin{vmatrix} 2 - \lambda & 1 \\ 1 & -2 - \lambda \end{vmatrix} = \lambda^2 - 5$$

And find the roots:

$$\lambda^2 - 5 = 0 \quad \therefore \lambda = \pm\sqrt{5}$$

PHYC90045 Introduction to Quantum Computing

Matrix and Matrix Inverse

$$A = VDV^\dagger$$

Change back Change to eigenbasis

$$D = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \quad D^{-1} = \begin{bmatrix} \frac{1}{\lambda_1} & 0 \\ 0 & \frac{1}{\lambda_2} \end{bmatrix}$$

Inverse of A is

$$A^{-1} = VD^{-1}V^\dagger$$

Checking:

$$AA^{-1} = VDV^\dagger VD^{-1}V^\dagger = VDD^{-1}V^\dagger = VV^\dagger = I$$

PHYC90045 Introduction to Quantum Computing

Eigenvalues to Inverse

We can find the corresponding eigenvectors (exercise for you!):

$$\lambda_1 = -\sqrt{5}, \quad u_1 = \begin{bmatrix} 2 - \sqrt{5} \\ 1 \end{bmatrix}$$

$$\lambda_2 = \sqrt{5}, \quad u_2 = \begin{bmatrix} 2 + \sqrt{5} \\ 1 \end{bmatrix}$$

Based on the previous slide we can find the inverse:

$$A^{-1} = VD^{-1}V^\dagger$$

$$A^{-1} = \frac{1}{5} \begin{bmatrix} 2 & 1 \\ 1 & -2 \end{bmatrix}$$

PHYC90045 Introduction to Quantum Computing

The University of
MELBOURNE

Invert the matrix

$$Ax = b$$

$$A^{-1}Ax = A^{-1}b$$

$$x = A^{-1}b$$

So in our case:

$$\begin{aligned} x &= \frac{1}{5} \begin{bmatrix} 2 & 1 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} 5 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 2 \\ 1 \end{bmatrix} \end{aligned}$$

And we've solved the linear equations. The HHL quantum algorithm works in a similar way on a quantum computer.

PHYC90045 Introduction to Quantum Computing

THE UNIVERSITY OF MELBOURNE

Steps for solving classically

- 1) Write as a matrix
- 2) Find eigenvalues of the matrix, A
- 3) Use eigenvalues to invert matrix, A^{-1}
- 4) Apply A^{-1} to b

PHYC90045 Introduction to Quantum Computing

HHL Algorithm

Quantum algorithm for linear systems of equations

Aram W. Harrow,¹ Avinatan Hassidim,² and Seth Lloyd¹

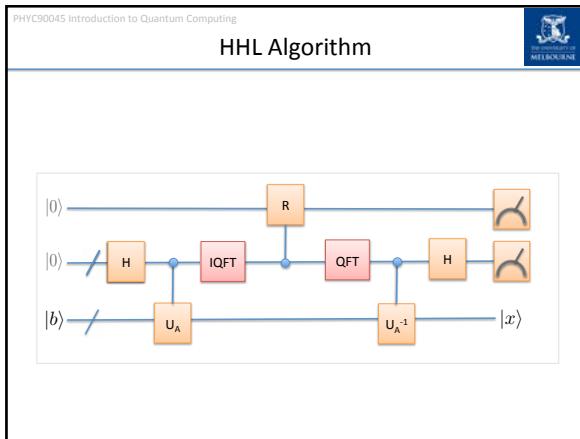
¹*Department of Mathematics, University of British Columbia, B651 ITFW, U.K.*
²*MIT Research Laboratory for Electronics, Cambridge, MA 02139, USA*

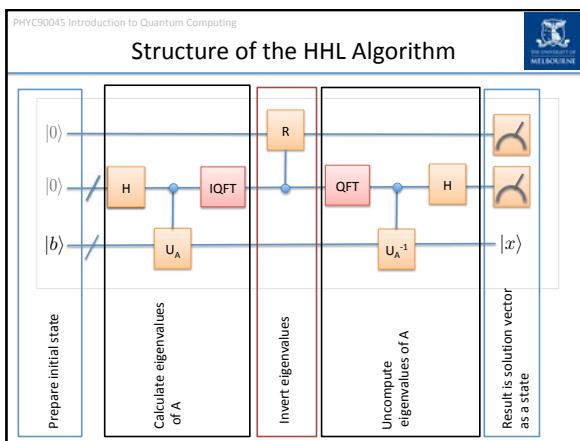
³*MIT - Research Laboratory for Electronics and Department of Mechanical Engineering, Cambridge, MA 02139, USA*

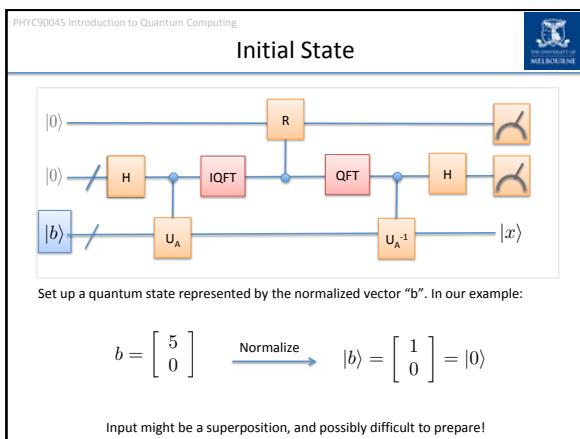
Solving linear systems of equations is a common problem that arises both on its own and as a subroutine in more complex problems, given a matrix A and a vector \vec{b} , find a vector \vec{x} such that $A\vec{x} = \vec{b}$. We consider the case where one doesn't need to know the solution \vec{x} itself, but rather an approximation \vec{x}' to within some tolerance ϵ . For a sparse system with condition number κ and unitary matrix M . In this case, when A is sparse, $N \times N$ and has condition number κ , classical algorithms can find \vec{x} and estimate $\|\vec{x}' - \vec{x}\|$ in $O(\kappa\sqrt{N})$ time. Here, we exhibit a quantum algorithm for this task that runs in $\text{poly}(\log N, \kappa)$ time, an exponential improvement over the best classical algorithm.

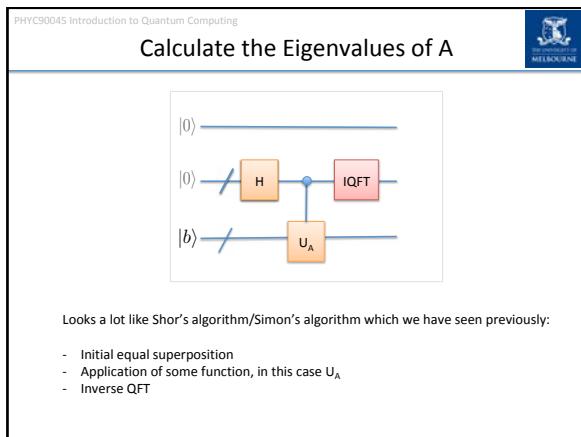
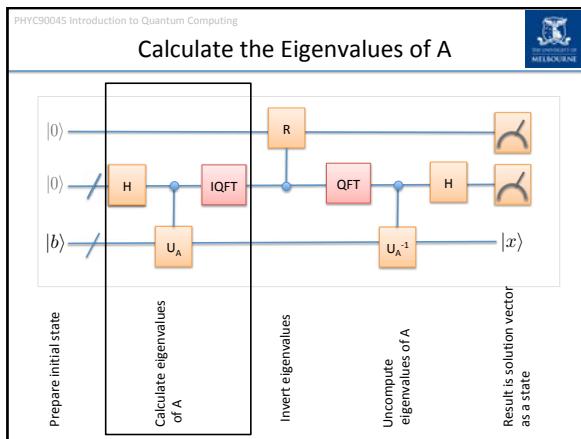
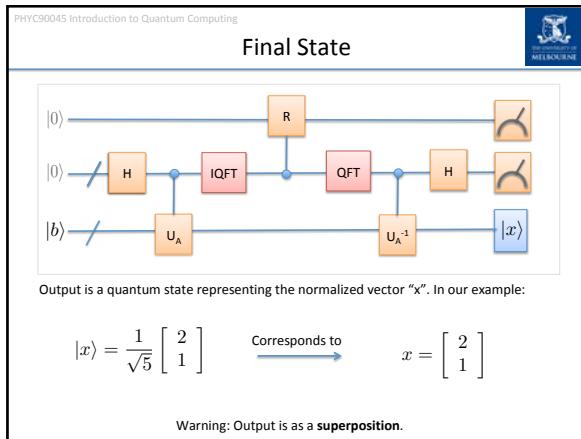
L. INTRODUCTION

Quantum computers are devices that harness quantum mechanics to perform computations in ways that classical computers cannot. For certain problems, quantum algorithms supply exponential speedups over their classical counterparts, the most famous example being Shor's factoring algorithm [1]. Few such exponential speedups are known, and those that are (such as the use of quantum computers to simulate other quantum systems [2]) have so far had limited use outside the domain of quantum mechanics. This paper presents a quantum algorithm to estimate features









PHYC90045 Introduction to Quantum Computing

U_A

$$U_A = \exp(iAt)$$

How to exponentiate a general A as a circuit? One way if A is Hermitian:

- Break up as Pauli matrices
- Use Trotter!

In our case:

$$A = \begin{bmatrix} 2 & 1 \\ 1 & -2 \end{bmatrix} = X + 2Z$$

t is the state of the control qubit!

PHYC90045 Introduction to Quantum Computing

Matrix as linear combination of Pauli operators

Always possible decompose a matrix as a sum of Paulis. If you have a matrix only:

$$E_i = \frac{\text{Tr}[\sigma_i H]}{d}$$

Where d is the dimension of the system, H is the Hamiltonian and σ_i is the Pauli. If the matrix is Hermitian, the co-efficients you find, E_i , should be real.

Express the Hamiltonian as linear combination of Pauli matrices:

$$H = \sum_i E_i \sigma_i$$

For example: $H = B_1 X_1 + B_2 X_2 + J_{12} Z_1 Z_2$

PHYC90045 Introduction to Quantum Computing

Trotter Approximation

But what if you do want to create the gate:

$$\exp(A + B)$$

We might try:

$$\exp(A + B) \approx \exp(A) \exp(B)$$

$$\exp(A + B) \approx \exp\left(\frac{A}{2}\right) \exp\left(\frac{B}{2}\right) \exp\left(\frac{A}{2}\right) \exp\left(\frac{B}{2}\right)$$

$$\exp(A + B) \approx \left(\exp\left(\frac{A}{n}\right) \exp\left(\frac{B}{n}\right) \right)^n$$

This is called the Trotter (sometimes Trotter-Suzuki) approximation – useful!

PHYC90045 Introduction to Quantum Computing

If A is not Hermitian

If A is not Hermitian, can make it Hermitian:

$$A' = \begin{bmatrix} 0 & A \\ A^\dagger & 0 \end{bmatrix}$$

PHYC90045 Introduction to Quantum Computing

Step by Step

Initially, the state is:
 $|\psi\rangle = |0\rangle|0\rangle|b\rangle$

PHYC90045 Introduction to Quantum Computing

Step by Step

After the Hadamard gates the “t” register is in an equal superposition:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{t=0}^{N-1} |0\rangle|t\rangle|b\rangle$$

PHYC90045 Introduction to Quantum Computing

Step by Step

We then apply the U_A gate:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{t=0}^{N-1} |0\rangle |t\rangle \exp(iAt) |b\rangle$$

Angle depends on the “t” register.

PHYC90045 Introduction to Quantum Computing

Step by Step

For now let us imagine that b is an eigenstate of a (we will remove this assumption later)

$$\begin{aligned} |\psi\rangle &= \frac{1}{\sqrt{N}} \sum_{t=0}^{N-1} |0\rangle |t\rangle \exp(iAt) |b\rangle \\ &= \frac{1}{\sqrt{N}} \sum_{t=0}^{N-1} |0\rangle \exp(i\lambda_b t) |t\rangle |b\rangle \end{aligned}$$

Phase kickback on the “t” register. Periodic according to the eigenvalue!

PHYC90045 Introduction to Quantum Computing

Step by Step

Applying IQFT extracts the eigenvalue:

$$\begin{aligned} |\psi\rangle &= U_{IQFT} \frac{1}{\sqrt{N}} \sum_{t=0}^{N-1} |0\rangle \exp(i\lambda_b t) |t\rangle |b\rangle \\ &= |0\rangle |\tilde{\lambda}_b\rangle |b\rangle \end{aligned}$$

Giving an approximation to the Eigenvalue in the “t” register.

PHYC90045 Introduction to Quantum Computing

Superposition of Eigenstates

If, instead of a single Eigenstate, b was made up of a superposition of Eigenstates of A:

$$|b\rangle = \sum_i b_i |u_i\rangle$$

Then so too, after performing the algorithm, would we be in a superposition of states:

$$|\psi\rangle = \sum_i b_i |0\rangle |\tilde{\lambda}_i\rangle |u_i\rangle$$

What we want is the solution, which depends on quantities we have calculated:

$$|x\rangle = \sum_i \frac{b_i}{\lambda_i} |u_i\rangle$$

How can we reduce each amplitude b_i by a factor of λ ? This is *not even unitary!*

PHYC90045 Introduction to Quantum Computing

Calculate the Eigenvalues of A

Prepare initial state

Calculate eigenvalues of A

Invert eigenvalues

Uncompute eigenvalues of A

Result is solution vector as a state

PHYC90045 Introduction to Quantum Computing

Inverting the Eigenvalues

The t register contains the eigenvalue. Based on this register we can make a controlled rotation on the ancilla qubit to obtain the state (of the ancilla register):

$$\sqrt{1 - \frac{C^2}{\lambda^2}} |0\rangle + \frac{C}{\lambda} |1\rangle$$

$$R_y \text{ by } \theta = -2 \cos^{-1} \left(\frac{C}{\lambda} \right)$$

We now measure. If we obtain the state 0, we redo the algorithm ("post-select") until we have measured the 1 state.

Initially the state is: $|\psi\rangle = \sum_i b_i |0\rangle |\tilde{\lambda}_i\rangle |u_i\rangle$

Becomes: $\sum_i b_i \left(\sqrt{1 - \frac{C^2}{\lambda^2}} |0\rangle + \frac{C}{\lambda} |1\rangle \right) |\tilde{\lambda}_i\rangle |u_i\rangle$

PHYC90045 Introduction to Quantum Computing

Applied to our state

$|\psi\rangle = \sum_i b_i |0\rangle |\tilde{\lambda}_i\rangle |u_i\rangle$

After rotation, R:

$$|\psi\rangle = \sum_i b_i \left(\sqrt{1 - \frac{C^2}{\lambda^2}} |0\rangle + \frac{C}{\lambda} |1\rangle \right) |\tilde{\lambda}_i\rangle |u_i\rangle$$

And after post-selecting based on measuring the state 1,

Badly normalized:

$$|\psi\rangle = \sum_i b_i \frac{C}{\lambda} |1\rangle |\tilde{\lambda}_i\rangle |u_i\rangle$$

$$|\psi\rangle = \sum_i \frac{b_i}{\lambda} |1\rangle |\tilde{\lambda}_i\rangle |u_i\rangle$$

PHYC90045 Introduction to Quantum Computing

“Uncompute” the Eigenvalues

Prepare initial state

Calculate eigenvalues of A

Invert eigenvalues

Uncompute eigenvalues of A

Result is solution vector as a state

PHYC90045 Introduction to Quantum Computing

Invert the eigenvalues

From our state:

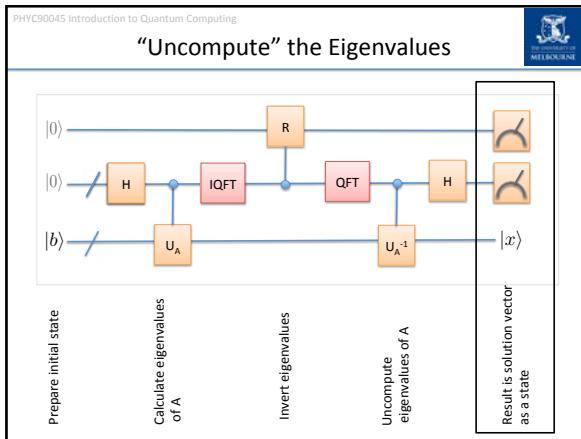
$$|\psi\rangle = \sum_i \frac{b_i}{\lambda} |1\rangle |\tilde{\lambda}_i\rangle |u_i\rangle$$

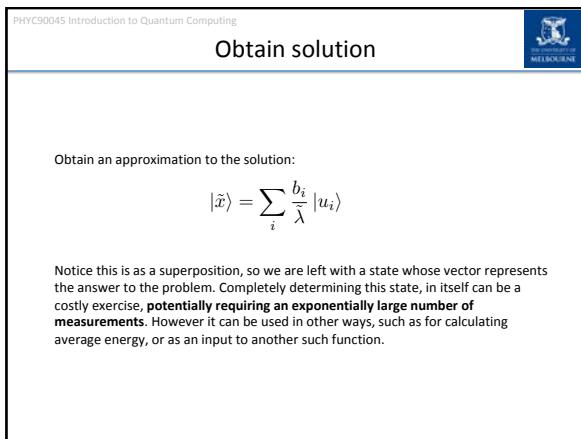
We apply the opposite procedure which we used to calculate the eigenvalues. This resets them to the zero state:

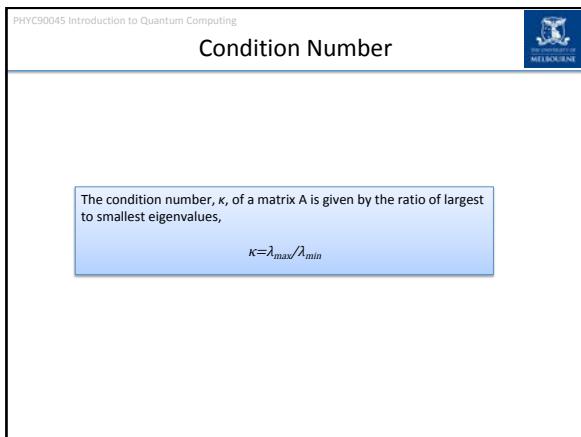
$$|\psi\rangle = \sum_i \frac{b_i}{\lambda} |1\rangle |0\rangle |u_i\rangle$$

Compare to the solution:

$$|x\rangle = \sum_i \frac{b_i}{\lambda_i} |u_i\rangle$$







PHYC90045 Introduction to Quantum Computing

Running time

The best general classical algorithm is the conjugate gradient method, with runtime:

$$O(N s \kappa \log(1/\epsilon))$$

HHL algorithm has runtime:

$$O(\log(N) s^2 \kappa^2 / \epsilon)$$

Where N is the dimension of the matrix
 s is the sparsity, κ is the condition number of A, and ϵ is the error

PHYC90045 Introduction to Quantum Computing

Summary

- HHL algorithm solves systems of linear equations
- Need a method for emulating $\exp(iAt)$
- Can be used as a quantum subroutine
- Has a runtime, exponentially faster than classical versions.

PHYC90045 Introduction to Quantum Computing

Week 11

Lecture 21
Python, IBM's QISKit

Lecture 22
- Further quantum algorithms – HHL algorithm

Lab 11
Implementing small algorithms on IBM's 16 qubit machine

Lecture 23

Quantum Computing architectures and quantum complexity classes

Lecture 24

Quantum Computing Review

Lab 12

HHL algorithm using the QUI

Quantum Computing Implementations and Complexity Classes

Physics 90045

Lecture 23

Implementations

Different Quantum Computing Hardware

Different ways to implement a quantum computer:

- Ion Traps
- Superconducting qubits
- NV centres
- Quantum Optics
- Semiconductors: Donors and Dots

Ion Traps

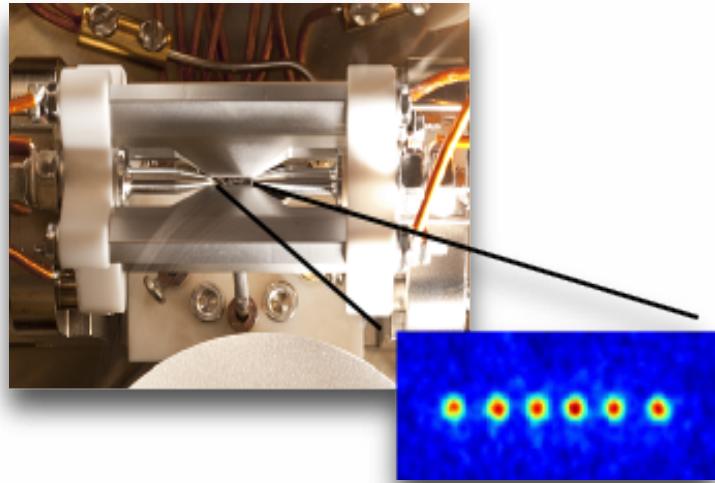


Image: Rainer Blatt (Innsbruck). Electrodes to trap six ^{40}Ca ions. Ions are laser cooled, and can remain in the trap for days once cooled. Qubit manipulation and readout via laser, CCD camera.

Ion Traps (2)

The Qubit: Quantum states of trapped ions (typically ^{40}Ca), interacting via coupling to collective degree of freedom.

Number of Qubits Demonstrated: ~14

Fidelity reported: 99.9%

Pros:

- Largest number of high fidelity qubits
- Extremely high fidelity
- Demonstrated error correction code (7 qubit) and largest “genuine” demonstration of Shor’s algorithm.

Challenges:

- Transport of qubits
- Comparatively large
- Heating

Corporate support

IonQ

Paper on measured error rates for ion traps

PHYSICAL REVIEW LETTERS 123, 110503 (2019)

Probing Qubit Memory Errors at the Part-per-Million Level

M. A. Sepiol, A. C. Hughes, J. E. Tarlton, D. P. Nadlinger, T. G. Ballance, C. J. Ballance,
T. P. Harty, A. M. Steane, J. F. Goodwin,^{*} and D. M. Lucas

*Department of Physics, University of Oxford, Clarendon Laboratory, Parks Road,
Oxford OX1 3PU, United Kingdom*



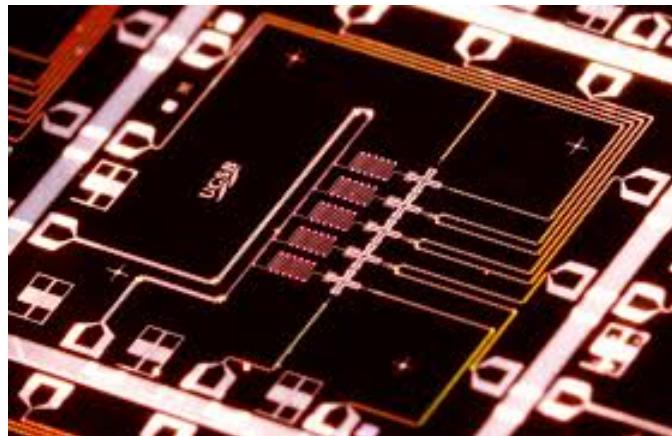
(Received 10 June 2019; published 13 September 2019)

Robust qubit memory is essential for quantum computing, both for near-term devices operating without error correction, and for the long-term goal of a fault-tolerant processor. We directly measure the memory error ϵ_m for a $^{43}\text{Ca}^+$ trapped-ion qubit in the small-error regime and find $\epsilon_m < 10^{-4}$ for storage times $t \lesssim 50$ ms. This exceeds gate or measurement times by three orders of magnitude. Using randomized benchmarking, at $t = 1$ ms we measure $\epsilon_m = 1.2(7) \times 10^{-6}$, around ten times smaller than that extrapolated from the T_2^* time, and limited by instability of the atomic clock reference used to benchmark the qubit.

DOI: [10.1103/PhysRevLett.123.110503](https://doi.org/10.1103/PhysRevLett.123.110503)

Sepiol et al, PRL, 123, 110503, (2019)

Superconducting Qubits



John Martinis (UCSB) Showing five (transmon or Xmon) superconducting qubits placed in a row, each coupled to their neighbour. Information is stored in the charge/phase degrees of freedom.

Superconducting Qubits (2)

The Qubit: The charge or phase degrees of freedom of superconducting circuits.
Modern designs (transmon qubits) minimize noise.

Number of Qubits Demonstrated: ~53

Fidelity reported: 99.4%

Pros:

- Large number of qubits
- Demonstrated transport
- Demonstrated genuine error suppression in five and nine qubit error correction codes

Challenges:

- Comparatively fast decoherence times
- Currently comparatively large

Corporate support

Google, IBM, Rigetti

Quantum supremacy



Quantum supremacy using a programmable superconducting processor

Google AI Quantum and collaborators[†]

The tantalizing promise of quantum computers is that certain computational tasks might be executed exponentially faster on a quantum processor than on a classical processor. A fundamental challenge is to build a high-fidelity processor capable of running quantum algorithms in an exponentially large computational space. Here, we report using a processor with programmable superconducting qubits to create quantum states on 53 qubits, occupying a state space $2^{53} \sim 10^{16}$. Measurements from repeated experiments sample the corresponding probability distribution, which we verify using classical simulations. While our processor takes about 200 seconds to sample one instance of the quantum circuit 1 million times, a state-of-the-art supercomputer would require approximately 10,000 years to perform the equivalent task. This dramatic speedup relative to all known classical algorithms provides an experimental realization of quantum supremacy on a computational task and heralds the advent of a much-anticipated computing paradigm.

In the early 1980s, Richard Feynman proposed that a quantum computer would be an effective tool to solve problems in physics and chemistry, as it is exponentially costly to simulate large quantum systems with classical computers [1]. Realizing Feynman's vision poses signifi-

A COMPUTATIONAL TASK TO DEMONSTRATE QUANTUM SUPREMACY

To demonstrate quantum supremacy, we compare our quantum processor against state-of-the-art classical computers in the task of sampling the output of a pseudo-random quantum circuit [24–26]. Random circuits are a

Google's leaked paper, Financial Times, September 20, 2019

Quantum Optics

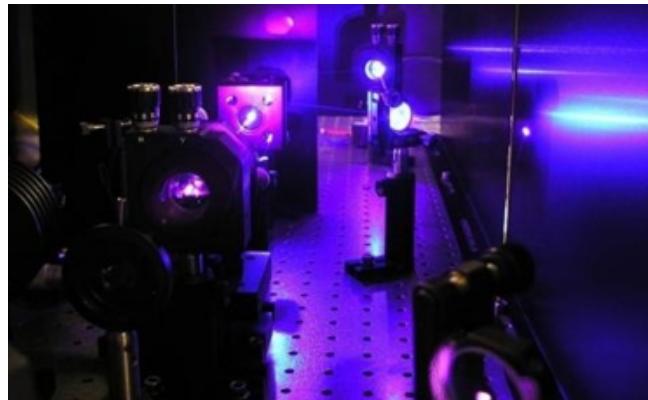


Image from Pryde (Griffiths University). Quantum states of light are used to represent quantum information, and manipulated using linear optics. First proposed by KLM, using entanglement generated by non-linear crystals, and subsequent linear optical operation.

Linear Optics (2)



The Qubit: The optical quantum states of light, including continuous variables, squeezed states, polarization and presence or absence of photons.

Pros:

- Communications and telecommunication applications
- Qubits impervious to their environment
- May demonstrate quantum advantage with Boson sampling

Challenges:

- Interacting large numbers of qubits
- Creating large quantum states
- Single photon sources

Generation of multi-qubit cluster states

**SHARE****REPORT**

Generation of time-domain-multiplexed two-dimensional cluster state

Warit Asavanant¹, Yu Shiozawa¹, Shota Yokoyama², Baramee Charoensombutamon¹, Hiroki Emura¹, Rafael N. Alexander³, S...

+ See all authors and affiliations

Science 18 Oct 2019;
Vol. 366, Issue 6463, pp. 373-376
DOI: 10.1126/science.aay2645

Article**Figures & Data****Info & Metrics****eLetters****PDF**

You are currently viewing the abstract.

[View Full Text](#)

Generating large-scale cluster states

The development of a practical quantum computer requires universality, scalability, and fault tolerance. Although much progress is being made in circuit platforms in which arrays of qubits are addressed and manipulated individually, scale-up of such systems is experimentally challenging. Asavanant *et al.* and Larsen *et al.* explore an alternative route: measurement-based quantum computation, which is a platform based on the generation of large-scale cluster states. As these are optically prepared and easier to handle (one simply performs local measurements on each individual component of the cluster state), such a platform is readily scalable and fault tolerant. The topology of the cluster state ensures that the approach meets the requirements for quantum computation.

Science, this issue p. 373, p. 369

**Science**Vol 366, Issue 6463
18 October 2019
[Table of Contents](#)
[Print Table of Contents](#)
[Advertising \(PDF\)](#)
[Classified \(PDF\)](#)
[Masthead \(PDF\)](#)
ARTICLE TOOLS

Email

Print

Request Permissions

Citation tools

Download Powerpoint

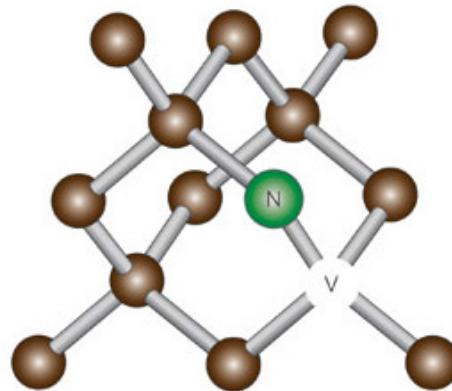
Save to my folders

Alerts

Share

**Asavanant et al, Science,
366, 6463, p373-376, 2019**

NV Centres



Nitrogen Vacancy centre in diamond forms a *room temperature*, electronic spin-1 system which can be manipulated with microwave fields, and read out optically.

Image: Ahanovic et al Nature Photonics, 2011

NV Centres (2)

The Qubit: The electronic spin-1 system of an NV defect in diamond. Can be coupled to nearby nuclear spins.

Number of Qubits Demonstrated: 10

Fidelity reported: 99.2%

Pros:

- Room temperature
- Biologically compatible
- First loophole-free violation of Bell's inequalities (Delft)

Challenges:

- Coupling two NVs is difficult
- Diamond difficult to work with, scale

NV quantum memory

[Featured in Physics](#)[Open Access](#)

A Ten-Qubit Solid-State Spin Register with Quantum Memory up to One Minute

C. E. Bradley, J. Randall, M. H. Abobeih, R. C. Berrevoets, M. J. Degen, M. A. Bakker, M. Markham, D. J. Twitchen, and T. H. Taminiau

Phys. Rev. X 9, 031045 – Published 11 September 2019

 See Synopsis: [Diamond Qubits Take the Stage](#)

[Article](#)[References](#)[No Citing Articles](#)[Supplemental Material](#)[PDF](#)[HTML](#)[Export Citation](#)

ABSTRACT

Spins associated with single defects in solids provide promising qubits for quantum-information processing and quantum networks. Recent experiments have demonstrated long coherence times, high-fidelity operations, and long-range entanglement. However, control has so far been limited to a few qubits, with entangled states of three spins demonstrated. Realizing larger multiquantum registers is challenging due to the need for quantum gates that avoid cross talk and protect the coherence of the complete register. In this paper, we present novel decoherence-protected gates that combine dynamical decoupling of an electron spin with selective phase-controlled driving of nuclear spins. We use these gates to realize a ten-qubit quantum register consisting of the electron spin of a nitrogen-vacancy center and nine nuclear spins in diamond. We show that the register is fully connected by generating entanglement between all 45 possible qubit pairs and realize genuine multipartite entangled states with up to seven qubits. Finally, we investigate the register as a multiquantum memory. We demonstrate the protection of an arbitrary single-qubit state for over 75 s—the longest reported for a single solid-state qubit—and show that two-qubit entanglement can be preserved for over 10 s.

Issue

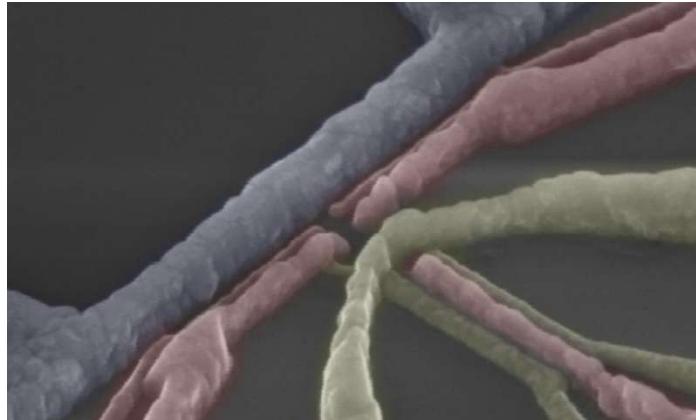
Vol. 9, Iss. 3 — July - September 2019

Subject Areas

Condensed Matter Physics
Quantum Physics
Quantum Information

[Check for updates](#)

Semiconductor Donors and Dots



UNSW. Qubits are the states of individual donors (either electron or nuclear spins) or the electronic levels of quantum dots (act as artificial atoms).

Donors and Dots (2)

The Qubit: The electrons spin or nuclear spin degrees of freedom of a donor or a dot.

Number of Qubits Demonstrated: 2+

Fidelity reported: > 99%

Pros:

- Intrinsically extremely long decoherence times (3s/minutes)
- Existing semiconductor industry at scale
- Comparatively small

Challenges:

- Need more qubits and coupling

Company support

Intel, Commonwealth Bank, Telstra

Donors: A two qubit swap gate



Letter | Published: 17 July 2019

A two-qubit gate between phosphorus donor electrons in silicon

Y. He, S. K. Gorman, D. Keith, L. Kranz, J. G. Keizer & M. Y. Simmons 

Nature 571, 371–375 (2019) | Download Citation 

9857 Accesses | 1 Citations | 136 Altmetric | Metrics 

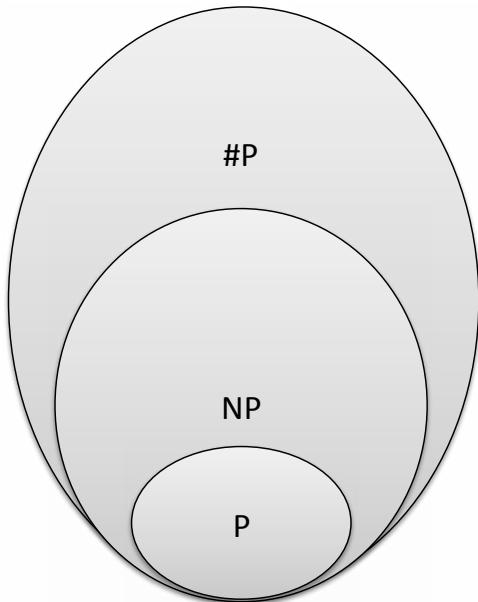
He et al,
Nature, 571,
371–375
(2019)

Abstract

Electron spin qubits formed by atoms in silicon have large (tens of millielectronvolts) orbital energies and weak spin-orbit coupling, giving rise to isolated electron spin ground states with coherence times of seconds^{1,2}. High-fidelity (more than 99.9 per cent) coherent control of such qubits has been demonstrated³, promising an attractive platform for quantum computing. However, inter-qubit coupling—which is essential for realizing large-scale circuits in atom-based qubits—has not yet been achieved. Exchange interactions between electron spins^{4,5} promise fast (gigahertz) gate operations with two-qubit gates, as recently demonstrated in gate-

Quantum Complexity Classes

Some classical complexity classes



P: Problems which can be solved in polynomial time

NP: Problems which can be checked in polynomial time
(ie. they have an efficiently verifiable proof)

#P: Problems which count the number of solutions in
NP

What are the equivalent for quantum computers?

Complexity classes with error

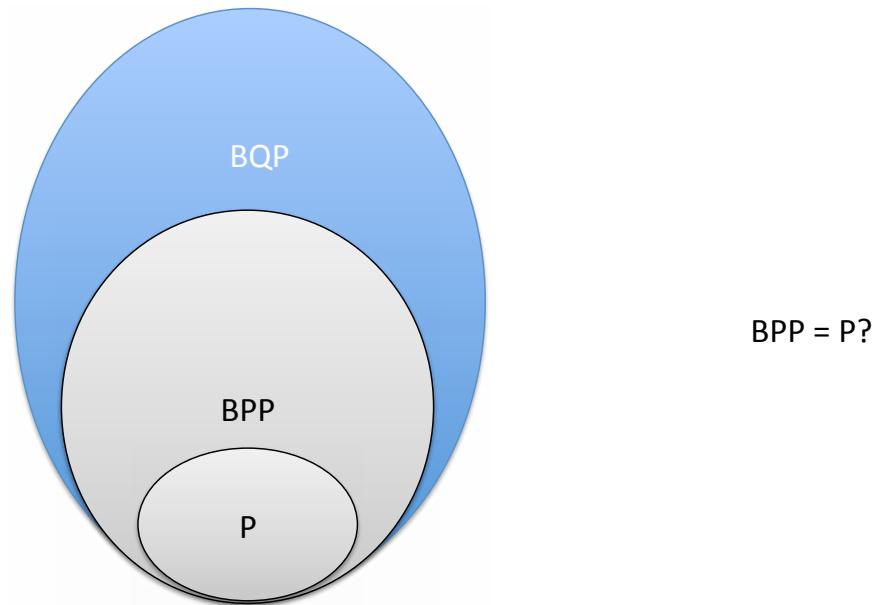
BPP is Bounded error Probabilistic Polynomial time. Polynomial time, but on a probabilistic computer allowing for an error of as much as $1/3$.

- Can flip coin and make random decisions
- Guaranteed to run in polynomial time
- On any given run of the algorithm has a probability $<1/3$ of the wrong answer, whether the answer is TRUE or FALSE.

Bounded error Quantum Polynomial Time

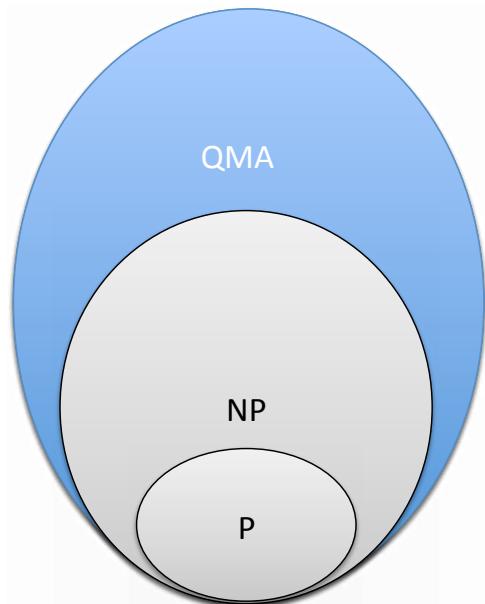
BQP is the set of decision problems solvable by a quantum computer in polynomial time, with an error of at most 1/3.

Polynomial Time Algorithms



So BQP is (roughly, including error) the quantum equivalent of P/BPP. What's the analogy to NP?

QMA and QMA Complete



Quantum Merlin-Arthur (QMA) is the analog of NP.

Informally:

NP is the set of problems you can verify in polynomial time.

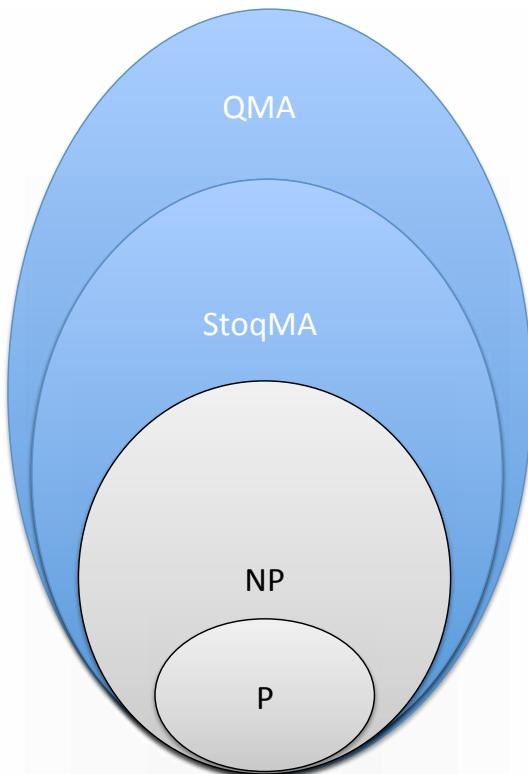
QMA is the set of problems you can verify in polynomial time with a quantum “proof”.

Example: Is does the lowest energy eigenstate of local Hamiltonian H have an energy less than E_a or are all the energies greater than E_b ?

Verification uses a quantum state as a “proof”, and measure the energy!

QMA Complete: The hardest problems in QMA. Can map any problem in QMA onto these.

“Stoquastic” Hamiltonians



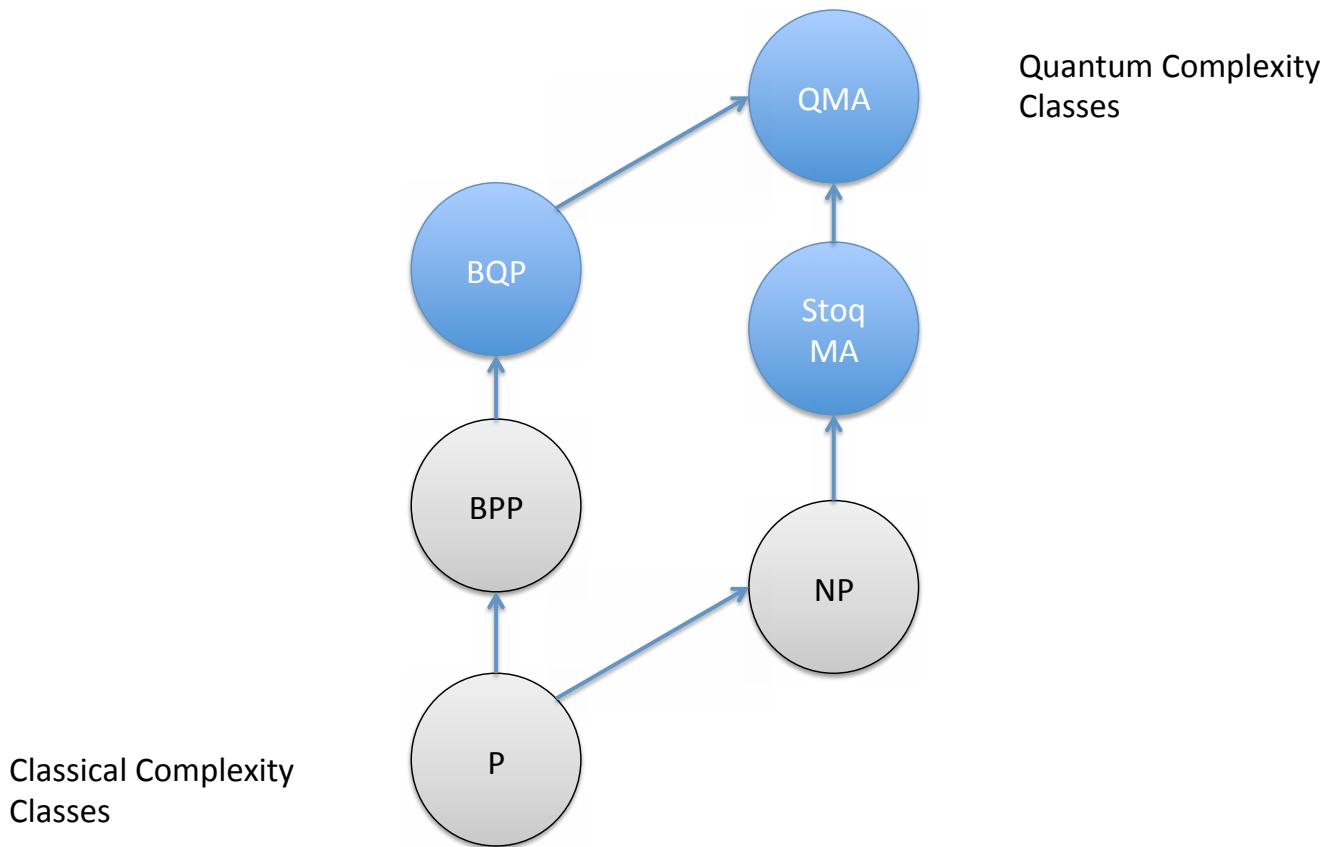
Not all Hamiltonians are equally hard to find the ground state of. In fact, those we have using for quantum annealing are easier in comparison to a general local Hamiltonian. These types of Hamiltonians are known as stoquastic Hamiltonians, and have their own complexity class:

StoqMA

Technically, a **Stoquastic matrix** is a Hermitian matrix, where all off-diagonal elements are real and non-positive.

Clearly, since we've been encoding NP-complete problems like MAX-CUT StoqMA includes NP.

Fitting it all together



Lecture 23

Quantum Computing architectures and quantum complexity classes

Lecture 24

Quantum Computing Review

Lab 12

HHL algorithm using the QUI

PHYC90045 Introduction to Quantum Computing

Week 12



Lecture 23
Quantum Computing architectures and quantum complexity classes

Lecture 24
Quantum Computing Review

Lab 12
HHL algorithm using the QUI

PHYC90045 Introduction to Quantum Computing



Quantum Computing Review

Physics 90045
Lecture 24

PHYC90045 Introduction to Quantum Computing



Review
(Selected Highlights)

PHYC90045 Introduction to Quantum Computing

Linear Algebra and Dirac notation

$|\psi\rangle = a|0\rangle + b|1\rangle$

For qubits we can use column vectors to represent a convenient basis for kets:

$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
 $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
 Computational basis states

$a|0\rangle + b|1\rangle = \begin{bmatrix} a \\ b \end{bmatrix}$
 $a, b \in \mathbb{C}$
 General qubit state
 a, b are "amplitudes"

PHYC90045 Introduction to Quantum Computing

The Bloch Sphere

A convenient geometric representation of single qubit states is the Bloch sphere:

PHYC90045 Introduction to Quantum Computing

Single Qubit Gates

Circuit symbol:

Matrix representation: $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

Action on ket states: $\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|1\rangle + \beta|0\rangle$

QUI example:

$\frac{\sqrt{3}}{2}|0\rangle + \frac{-i}{2}|1\rangle$
 complex amplitudes

X GATE
Rotate around the X axis by π radians.

PHYC90045 Introduction to Quantum Computing

Multiple Qubit States

$$\begin{bmatrix} a \\ b \end{bmatrix} \otimes \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} ac & ad \\ bc & bd \end{bmatrix}$$

00 amplitude
01 amplitude
10 amplitude
11 amplitude

$$|\psi\rangle = a|0\rangle + b|1\rangle$$

$$|\phi\rangle = c|0\rangle + d|1\rangle$$

PHYC90045 Introduction to Quantum Computing

Two qubit operations

$$\alpha|0\rangle + \beta|1\rangle$$

$$|0\rangle$$

$$|\psi\rangle \quad |\psi'\rangle$$

Before the CNOT, the state is:
 $|\psi\rangle = (\alpha|0\rangle + \beta|1\rangle) \otimes |0\rangle = \alpha|00\rangle + \beta|10\rangle$

After the CNOT, the state is:
 $|\psi'\rangle = \alpha|00\rangle + \beta|11\rangle$

PHYC90045 Introduction to Quantum Computing

Using the QUI

In labs you will learn to use the “Quantum User Interface” (QUI) to construct circuits. Your first lab will be all about single qubit rotations.

PHYC90045 Introduction to Quantum Computing

Entanglement

We can never find a, b, c, d, s.t.

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}} \neq (a|0\rangle + b|1\rangle) \otimes (c|0\rangle + d|1\rangle)$$

A state which is not separable is called an **entangled** state.

Entanglement is a uniquely quantum mechanical property, with no direct classical analogue.

PHYC90045 Introduction to Quantum Computing

Dense Coding Circuit

Bell state preparation: $|0\rangle \xrightarrow{\text{H}} |0\rangle$

Alice encoding: $|0\rangle \xrightarrow{\text{CNOT}} |0\rangle$

Bob's measurement: $|0\rangle \xrightarrow{\text{H}} |0\rangle$

$|00\rangle \xrightarrow{\sqrt{2}} \frac{|00\rangle + |11\rangle}{\sqrt{2}}$

$|11\rangle \xrightarrow{\sqrt{2}} \frac{|00\rangle - |11\rangle}{\sqrt{2}}$

0, 0 $\frac{|00\rangle + |11\rangle}{\sqrt{2}} \xrightarrow{\text{H}} |00\rangle + |11\rangle$

1, 0 $X_2 \frac{|00\rangle + |11\rangle}{\sqrt{2}} \xrightarrow{\text{H}} |01\rangle + |10\rangle$

0, 1 $Z_2 \frac{|00\rangle + |11\rangle}{\sqrt{2}} \xrightarrow{\text{H}} |00\rangle - |11\rangle$

1, 1 $X_2 Z_2 \frac{|00\rangle + |11\rangle}{\sqrt{2}} \xrightarrow{\text{H}} |01\rangle - |10\rangle$

Alice applies one of four different operations to her qubit, based on the classical information she would like to send.

PHYC90045 Introduction to Quantum Computing

Teleportation

Alice measures: $\alpha|0\rangle + \beta|1\rangle$

Bob's qubit: $|0\rangle$

i.e. after correction Bob has successfully reconstructed Alice's original state.

Alice measures	Bob's qubit	
0, 0	$\alpha 0\rangle + \beta 1\rangle$	$\alpha 0\rangle + \beta 1\rangle \rightarrow \alpha 0\rangle + \beta 1\rangle$
0, 1	$\alpha 1\rangle + \beta 0\rangle$	$X(\alpha 1\rangle + \beta 0\rangle) \rightarrow \alpha 0\rangle + \beta 1\rangle$
1, 0	$\alpha 0\rangle - \beta 1\rangle$	$Z(\alpha 0\rangle - \beta 1\rangle) \rightarrow \alpha 0\rangle + \beta 1\rangle$
1, 1	$\alpha 1\rangle - \beta 0\rangle$	$ZX(\alpha 1\rangle - \beta 0\rangle) \rightarrow \alpha 0\rangle + \beta 1\rangle$

PHYC90045 Introduction to Quantum Computing



 The University of
 MELBOURNE

Deutsch-Josza algorithm

- Given a boolean function, f , determine if:
 f is constant (always gives the same result), or
 f is balanced (gives equal numbers of 0s and 1s)
- Classical algorithm (worst case) needs $2^n/2+1$ queries**
- Quantum algorithm needs just 1 query.**



The diagram shows a quantum circuit with two input states, $|0\rangle$ and $|1\rangle$. The circuit consists of the following sequence of operations:

- Input $|0\rangle$ passes through a Hadamard gate (H) and then a $H^{\otimes n}$ gate.
- Input $|1\rangle$ passes through a Hadamard gate (H).
- The outputs from both paths enter a box labeled U_f .
- The output from the U_f box passes through another $H^{\otimes n}$ gate.
- The final measurement is performed using an $\text{A}^\dagger \text{A}$ meter.

PHYC90045 Introduction to Quantum Computing



Bernstein-Vazirani Algorithm

Given a Boolean function, f :

$$f(x) = x \cdot s \mod 2$$

find s .

$$x \cdot s = \sum_i x_i s_i$$

- **Classical algorithm** needs n queries
- **Quantum algorithm** needs just 1 query.

PHYC90045 Introduction to Quantum Computing



Simon's Algorithm

Given a 2-to-1 function, f , such that

$$f(x) = f(x \oplus a)$$

Find a .

Classical algorithm: $O(\sqrt{N})$ Queries to the oracle (probabilistically)

Quantum algorithm: $O(n)$ Queries to the oracle

PHYC90045 Introduction to Quantum Computing



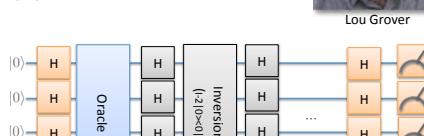
 THE UNIVERSITY OF MELBOURNE

Grover's Algorithm

- Unordered search, find one marked item among many
- Classically, this requires $N/2$ uses of the oracle
- Quantum mechanically, requires only $O(\sqrt{N})$.



Lou Grover



The diagram illustrates a quantum circuit for Grover's algorithm. The circuit consists of several components arranged horizontally:

- Initial State:** Four qubits, each initialized to $|0\rangle$, represented by orange boxes.
- Hadamard Gates:** Each of the four qubits passes through a Hadamard gate (H), represented by grey boxes, before entering the Oracle block.
- Oracle Block:** A blue rectangular box labeled "Oracle".
- Inversion Block:** A grey rectangular box labeled " $(|2\rangle\langle 2| - |0\rangle\langle 0|)$ ".
- Hadamard Gates:** Each of the four qubits passes through a Hadamard gate (H) after exiting the Inversion block.
- Final Measurement:** Each of the four qubits is measured, represented by orange boxes with a meter icon.

PHYC90045 Introduction to Quantum Computing

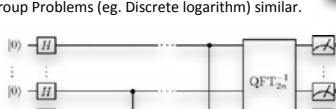
Shor's algorithm



- Efficient quantum algorithms for **factoring** semiprime numbers
- Best known classical algorithm is number field sieve (exponential in bit-length).
- Underpins the RSA cryptosystem
- Hidden Subgroup Problems (eg. Discrete logarithm) similar.



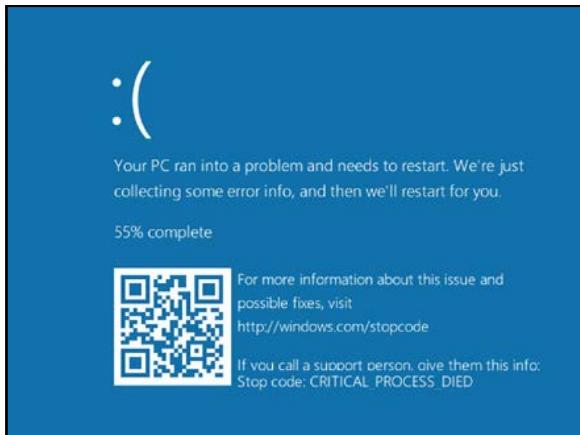
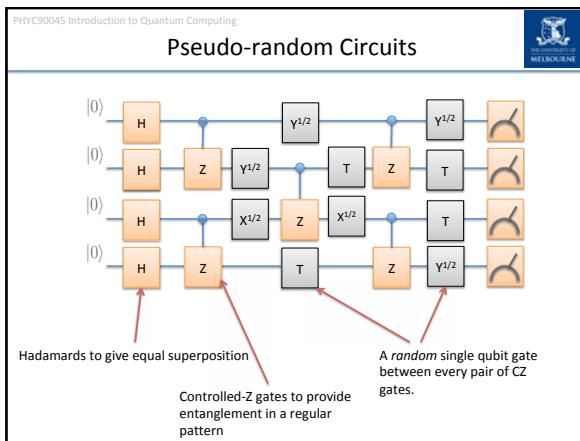
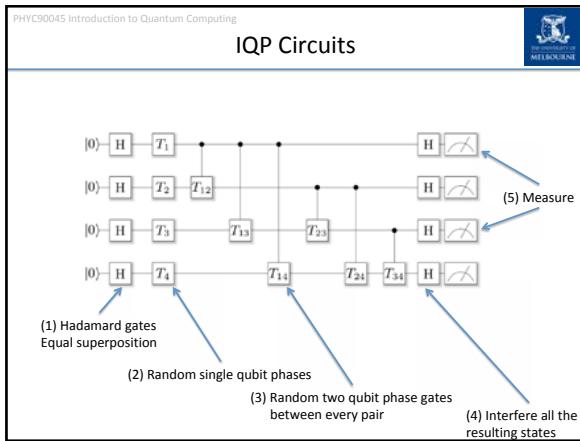
Peter Shor



The diagram shows a quantum circuit with four horizontal wires. The top wire starts with a Hadamard gate (H) followed by a sequence of control dots. The second wire starts with an H gate followed by control dots. The third wire starts with an H gate followed by control dots. The bottom wire starts with an H gate followed by control dots. Below the wires, the circuit consists of two main sections. The first section contains three controlled-U gates: $U_{a^{2^0}}$, $U_{a^{2^1}}$, and $U_{a^{2^{n-1}}}$. The second section is a large box labeled $QFT^{-1}_{2^n}$, which is connected to the wires via control dots. The circuit concludes with four measurement gates (represented by squares with diagonal lines) positioned at the end of each wire.

Shor, Proc 35th Ann Symp of Comp Sci, 26, (1995)

The diagram illustrates a quantum optics experiment setup. A vertical grey rectangle on the left is labeled "Photon in". A vertical grey rectangle on the right is labeled "Beamsplitter" above a blue arrow pointing to it and "Wall" below a blue arrow pointing to it. Between these two rectangles is a grid of grey diagonal lines representing beam splitters. A red line, labeled "Path of a photon" at its bottom end, traces the path of a photon through the grid, starting from the left and ending at the right. Blue vertical lines indicate other photons in the system.



PHYC90045 Introduction to Quantum Computing

Purity for one qubit

If the distance from the origin to the state is measured to be r , the purity is:

$$P = \frac{1 + r^2}{2}$$

Maximum purity of 1 for all pure states.

Minimum purity of $\frac{1}{2}$ for a completely mixed state.

Note: There's a more technical definition of purity in terms of density matrices, which we won't cover in this course.

PHYC90045 Introduction to Quantum Computing

Randomized Benchmarking

How good are our gates individual gate? We want a number for how much error doing each operation is. One way of determining this is to perform **randomized benchmarking**.

PHYC90045 Introduction to Quantum Computing

Quantum Error Correction

Similar to classical error correction codes, we can have a quantum repetition code:

$ 0\rangle \rightarrow 000\rangle$	"Logical 0"
$ 1\rangle \rightarrow 111\rangle$	"Logical 1"

In particular, a quantum superposition would be encoded as:

$$\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|000\rangle + \beta|111\rangle$$

Two key differences between quantum and classical error correction codes:

1. Cannot measure the codewords directly; would collapse the state
2. Phase errors

The figure consists of three parts. The top part is a grid of green hexagons representing a 2D lattice. The bottom-left part shows a quantum circuit for measuring an X-stabilizer. It starts with a register of four qubits labeled $| \Psi \rangle$. The first two qubits pass through a CNOT gate, while the third and fourth qubits pass through identity gates. The second qubit then passes through a w gate. The entire sequence is enclosed in a bracket labeled $\langle \Psi' |$. The bottom-right part shows a similar circuit for measuring a Z-stabilizer, starting with a register $| \Psi \rangle$, followed by a CNOT gate between the first two qubits, and the second qubit passing through a w gate. This sequence is enclosed in a bracket labeled $\langle \Psi' |$.

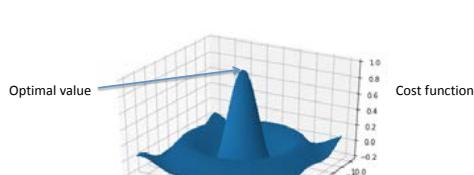
PHYC90045 Introduction to Quantum Computing



 THE UNIVERSITY OF
 MELBOURNE

Optimization Problems

Given some cost-function or “objective function” we would like to maximize/minimize. Often the inputs/parameters are constrained.



A 3D surface plot representing a cost function. The vertical axis is labeled "Cost function" and ranges from -0.2 to 1.0. The horizontal axes are labeled "Parameters" and range from -10 to 10. The surface shows a single peak. A blue arrow points to the peak, which is labeled "Optimal value".

PHYC90045 Introduction to Quantum Computing

Number partitioning as a QUBO problem

But if we square, we should get a positive solution (or zero). We want to find the assignment of spins which has the minimum energy (ie. closest to zero):

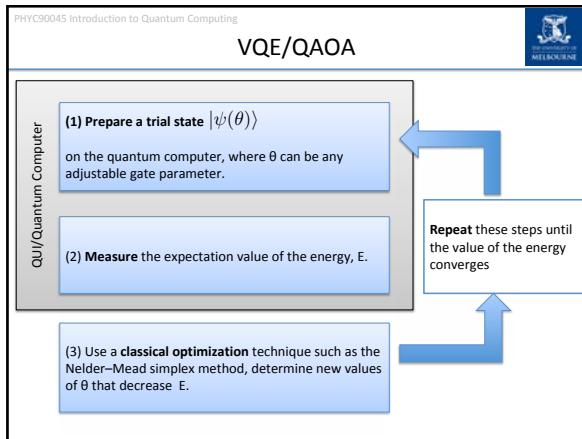
$$H = \left(\sum_i w_i Z_i \right)^2 = \sum_{i \neq j} 2w_i w_j Z_i Z_j + \sum_i w_i^2 I$$

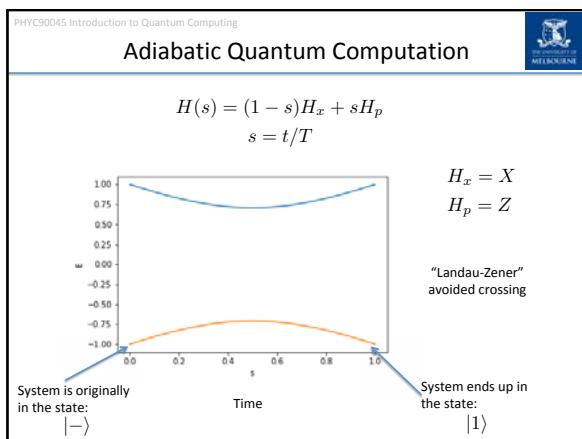
Coupling is the product of numbers

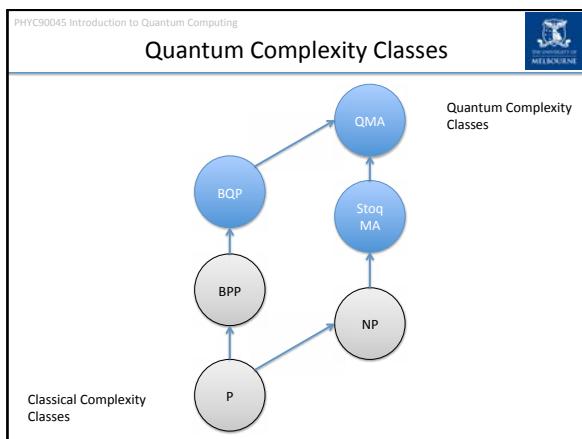
Eg. For the set {1, 2, 3}:

$$H = 4Z_1 Z_2 + 6Z_1 Z_3 + 12Z_2 Z_3 + 14I$$

Finding minimum energy state will solve the problem!







PHYC90045 Introduction to Quantum Computing



Questions?

PHYC90045 Introduction to Quantum Computing



PHYC90045 Introduction to Quantum Computing

Week 12



Lecture 23
Quantum Computing architectures and quantum complexity classes

Lecture 24
Quantum Computing Review

Lab 12
HHL algorithm using the QUI
