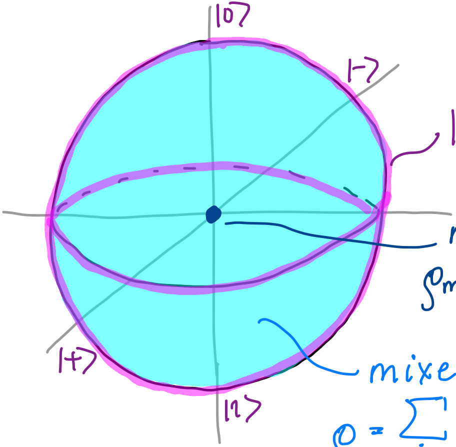
CPEN 400Q Lecture 19
Noisy quantum systems; introducing the oracle

Monday 20 March 2023

Announcements

- Quiz 8 beginning of class today
- Literacy assignment 3 due 29 March (Wednesday) at 23:59
- Assignment 3 (last assignment) posted later this week
- Project presentations start 31 March (**next Friday**)
  - will post a randomized schedule on Piazza tomorrow
  - 12 mins + 3 mins for questions; max 10 slides incl. title page
  - rubric and grading details posted tomorrow with schedule

We introduced *density matrices* and *mixed states*.



pure states

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

maximally mixed state

$\rho_{max} = \frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1| = \frac{1}{2}\mathbb{1}$

mixed states

$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$

Last time

We saw how states, operations, and measurements look like on mixed vs. pure states.

	Pure state	Pure state ρ	Mixed state ρ
States	$\vert\psi\rangle$	$\rho = \vert\psi\rangle\langle\psi\vert$	$\rho = \sum_i p_i \vert\psi_i\rangle\langle\psi_i\vert$
Ops.	$\vert\psi'\rangle = U\vert\psi\rangle$	$\rho' = U\rho U^\dagger$ $\vert\psi'\rangle\langle\psi'\vert = U\vert\psi\rangle\langle\psi\vert U^\dagger$	$\rho' = U\rho U^\dagger$
Meas.*	$\vert\langle\varphi_i\vert\psi\rangle\vert^2$ $\langle\psi\vert B\vert\psi\rangle$	$\mathrm{Tr}(\vert\varphi_i\rangle\langle\varphi_i\vert\,\vert\psi\rangle\langle\psi\vert)$ $\mathrm{Tr}(B\rho)$	$\mathrm{Tr}(P_i\rho)$ $\mathrm{Tr}(B\rho)$

↳ Born rule

* where $\{\varphi_i\}$ form an orthonormal basis, and $\{P_i\}$ may be the associated set of projectors $P_i = |\varphi_i\rangle\langle\varphi_i|$ or more generally a POVM (in either case, $\sum_i P_i = I$).

Learning outcomes

- Describe the effects of common noise channels on qubit states
- Add noise to quantum circuits in PennyLane
- Define an oracle, and query complexity

Quantum channels

Noise occurring in quantum systems is represented by **quantum channels**.

A quantum channel Φ *maps* states to other states.

$$\rho \mapsto \rho' = \Phi(\rho)$$

More formally, quantum channels are linear CPTP (Completely Positive, Trace-Preserving) maps.

1

Example: applying a unitary U is a channel, $\mathcal{U}$.

$$\rho \mapsto \rho' = \mathcal{U}(\rho) = U\rho U^\dagger$$

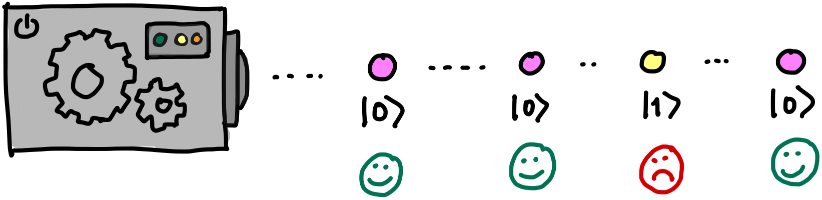
$RX(\theta)$ vs. $RX(\theta + \varepsilon)$

The bit flip channel

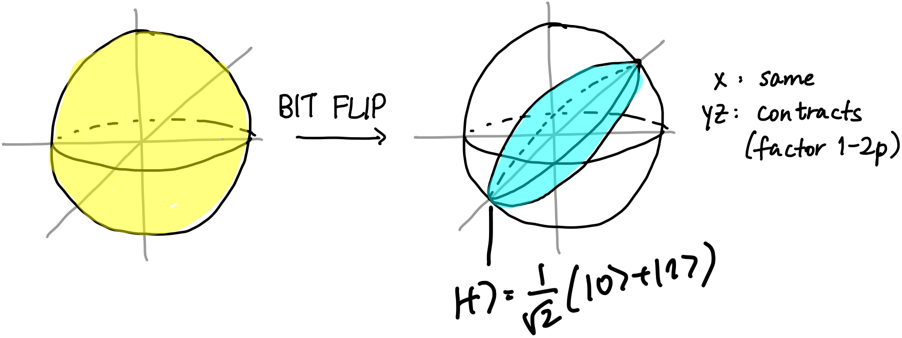
$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

Suppose a “bit flip” (Pauli X) error occurs with probability p.

$$\mathcal{E}(\rho) = (1-p)\cdot\rho + p\cdot X\rho X^\dagger$$



We can visualize the effects of such a channel by observing how it deforms the Bloch sphere.

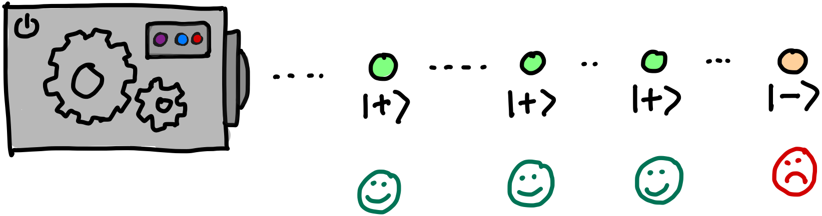


The phase flip channel

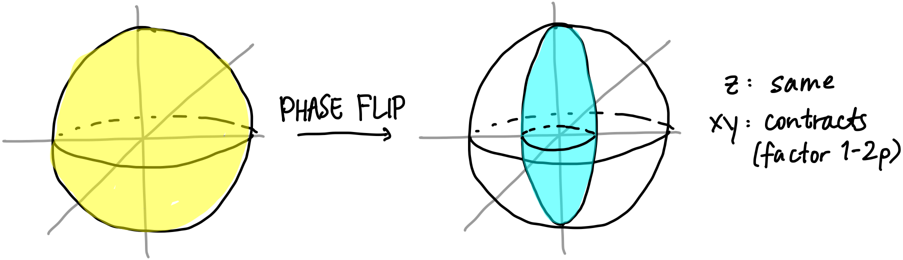
$\alpha|0\rangle + \beta|1\rangle$

Suppose a "phase flip" (Pauli Z) error occurs with probability p.

$$\mathcal{E}(\rho) = (1-p)\cdot\rho + p\cdot Z\rho Z$$



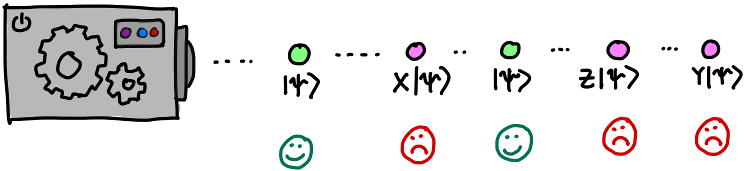
The phase flip channel



The depolarizing channel

Suppose each Pauli error occurs with probability $p/3$. This is called the *depolarizing channel*.

$$\mathcal{E}(\rho) = (1-p)\cdot\rho + \frac{p}{3}\cdot X\rho X + \frac{p}{3} Y\rho Y + \frac{p}{3} Z\rho Z$$



The depolarizing channel

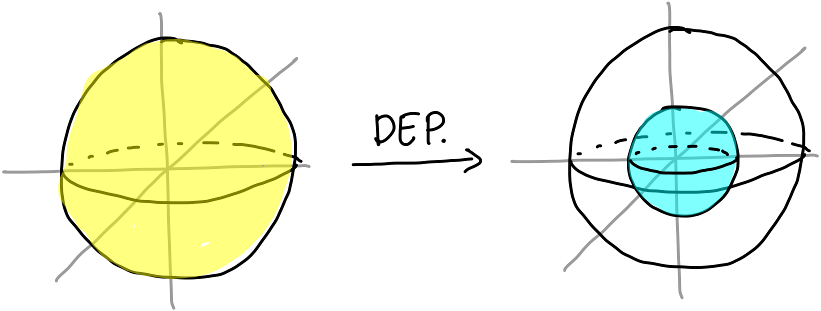
$$\mathcal{E}(\rho) = (1-p)\rho + \frac{p}{3}X\rho X + \frac{p}{3}Y\rho Y + \frac{p}{3}Z\rho Z$$

can also be written as

$$\mathcal{E}(\rho) = (1-p)\rho + p\frac{I}{2}$$

Think of this as outputting ρ w/probability $1-p$, and maximally mixed state with probability p.

The depolarizing channel



The depolarizing channel

Exercise: Suppose we prepare a system in the state

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

However, depolarization with strength $p = 0.02$ occurs. What is the probability of measuring (in the computational basis) and obtaining the $|0\rangle$ state as output?

Solution 1: solve by hand.

... too tedious, but you can evaluate

$$
\begin{aligned}
\rho_\psi &= |\psi\rangle\langle\psi| \\
\mathcal{E}(\rho_\psi) &= 0.98\rho_\psi + \frac{0.02}{3}\left(X\rho_\psi X + Y\rho_\psi Y + Z\rho_\psi Z\right) \\
\Pr(0) &= \mathrm{Tr}\left(|0\rangle\langle 0| \cdot \mathcal{E}(\rho_\psi)\right)
\end{aligned}
$$

Solution 2: solve with PennyLane's ``‘default.mixed’`` device!

Take another look at the depolarizing channel:

$$\mathcal{E}(\rho) = (1-p)\rho + \frac{p}{3}X\rho X + \frac{p}{3}Y\rho Y + \frac{p}{3}Z\rho Z$$

More generally, a CPTP quantum channel $\Phi(\rho)$ is characterized by a set of **Kraus operators** $\{K_i\}$ where

$$\Phi(\rho) = \sum_i K_i \rho K_i^\dagger$$

where

$$\sum_i K_i^\dagger K_i = I$$

Comparing density matrices

How can we quantify "how much" error occurs? How close is $\sigma = \mathcal{E}(\rho)$ to ρ?

One common metric is the **trace distance**:

$$T(\rho,\sigma) = \frac{1}{2}\|\rho-\sigma\|_1 = \frac{1}{2}\mathrm{Tr}\sqrt{(\rho-\sigma)^\dagger(\rho-\sigma)}$$

Value of trace distance is bounded by $0 \leq T(\rho,\sigma) \leq 1$, and *lower* trace distance is better.

qml.fidelity

Another is the **fidelity**:

$$F(\rho, \sigma) = \left(\mathrm{Tr}\sqrt{\sqrt{\rho}\,\sigma\sqrt{\rho}}\right)^2$$

Value of fidelity is bounded by $0 \leq F(\rho, \sigma) \leq 1$, and *higher* fidelity is better.

VQE on a noisy device

Let's solve the deuteron VQE problem on a simulated noisy device.

Image credit:

`https://quantum-computing.ibm.com/services/resources?tab=systems&skip=20&system=ibmq_manila`

(Captured 2023-03-18)

Comparing quantum computers

Clearly noise is a problem. How do we measure the quality of noisy quantum devices?

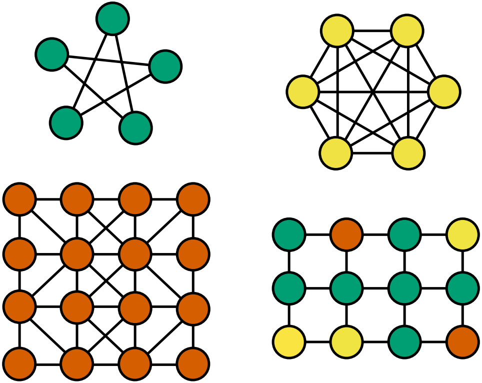


Image credit: quantum volume demo, `https://pennylane.ai/qml/demos/quantum_volume.html`

Comparing quantum computers

It is challenging to characterize, benchmark, and to compare quantum computers: it's more than just number of qubits.

- error rates
- qubit connectivity
- software/compiler quality
- gate operation times
- size of problem it can solve
- size of meaningful problem it can solve
- ...

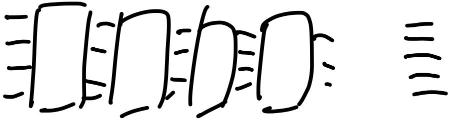
Comparing supercomputers

It's more than just number of cores: comparison based on LINPACK benchmark (FLOPS while solving dense linear system).

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,730,112	1,102.00	1,685.65	21,100
2	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
3	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,220,288	309.10	428.70	6,016
4	**Leonardo** - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, Atos EuroHPC/CINECA Italy	1,463,616	174.70	255.75	5,610
5	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148.60	200.79	10,096

Screencap: `https://www.top500.org/lists/top500/2022/11/`

Comparing quantum computers



Many competing metrics proposed by competing companies:

- quantum volume (IBM)
- CLOPS: circuit layer operations per second (layer = quantum volume circuit layer; IBM)
- algorithmic qubits (IonQ)
- Q-score (Atos)

(I will post a note on Piazza with links to some articles / tutorials)

Comparing quantum computers

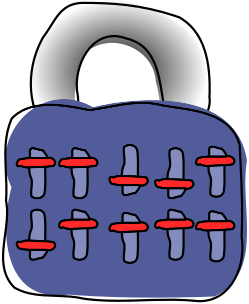
All this goes to say: current quantum hardware is noisy. Noise comes from a variety of sources, and depends on the qubit technology and architecture.

We need to do a combination of:

- Better-characterizing the behaviour of devices to learn how to improve their operation
- Processing the results to mitigate the effects of noise as much as possible

Oracles: motivating problem

Suppose we would like to find the combination for a "binary" lock:



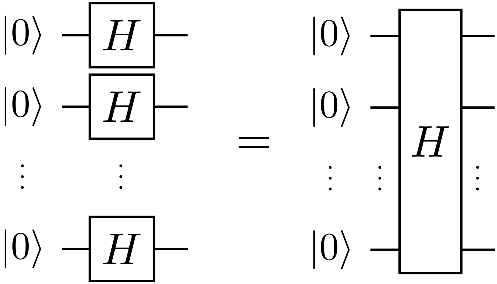
How do we solve this classically?

Image credit: Codebook node A.1

Idea: use superposition

Can we do better with a quantum computer?

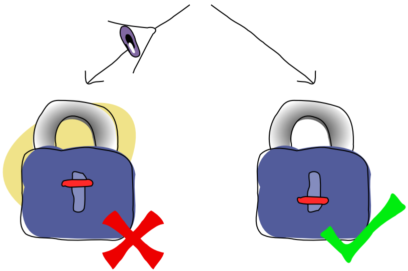
What if we take *n* qubits and put them in a superposition with all possible combinations?



Often called the *Hadamard transform*.

Idea: use superposition

Measurements are probabilistic - just because we put things into a uniform superposition of states, and our solution is "in" there, doesn't mean we are any closer to solving our problem.



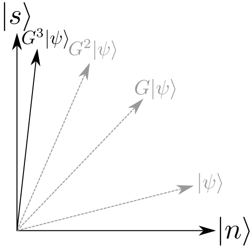
Quantum computers are **NOT** faster because they can "compute everything at the same time."

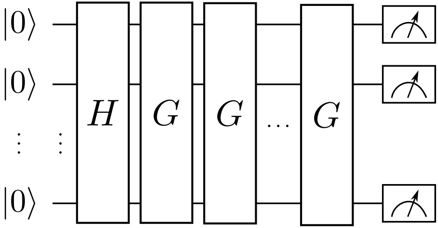
Image credit: Codebook node A.1

Solving problems with quantum computers

Can we solve this problem better with a quantum computer?

Yes: **amplitude amplification**, and **Grover's algorithm**



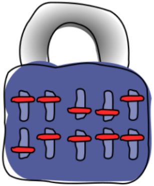


We will explore the algorithmic primitives that are involved, and some other cases where we can do better with quantum computing.

Oracles

Motivating problem

Suppose we would like to find combination for a "binary" lock:



Classically, we would have to try every possible combination. If there are n bits, that's 2^n possible tries. Can we do better with a quantum computer?

Image credit: Codebook node A.1

6 / 26

What is a "try"?

Oracles

We often express these tries as **evaluations** of a function that tells us whether we have found the correct answer.

Let

- $\mathbf{x}$ be an n-bit string that represents an input to the lock
- $\mathbf{s}$ be the solution to the problem (i.e., the correct combination)

We can represent trying a lock combination as a function:

$$f(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} = \mathbf{s} \\ 0 & \text{otherwise.} \end{cases}$$

Oracles

We don't necessarily care *how* this function gets evaluated, only that it gives us an answer (more specifically, a yes/no answer).

$$f(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} = \mathbf{s} \\ 0 & \text{otherwise.} \end{cases}$$

We consider this function as a black box, or an **oracle**.

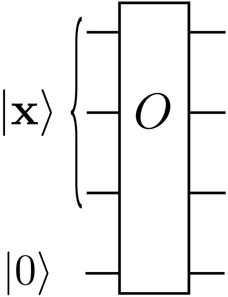
Every time we try a lock combination, we are **querying the oracle**. The amount of queries we make is the **query complexity**.

Quantum oracles

To solve this problem using quantum computing, we need some circuit that plays the role of the oracle.

Idea 1: encode the result in the state of an additional qubit.

$$O|\mathbf{x}\rangle|y\rangle = |\mathbf{x}\rangle|y \oplus f(\mathbf{x})\rangle$$

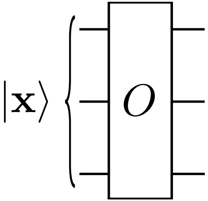


$$O|000\rangle|0\rangle = |000\rangle|0\rangle$$
$$O|001\rangle|0\rangle = |001\rangle|0\rangle$$
$$O|010\rangle|0\rangle = |010\rangle|0\rangle$$
$$O|011\rangle|0\rangle = |011\rangle|0\rangle$$
$$O|100\rangle|0\rangle = |100\rangle|0\rangle$$
$$O|101\rangle|0\rangle = |101\rangle|0\rangle$$
$$O|110\rangle|0\rangle = |110\rangle|1\rangle$$
$$O|111\rangle|0\rangle = |111\rangle|0\rangle$$

Quantum oracles

Idea 2: encode the result in the phase of a qubit.

$$O|\mathbf{x}\rangle = (-1)^{f(\mathbf{x})}|\mathbf{x}\rangle$$



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

Motivation: You are given access to an oracle and are promised that it implements one of the following 4 functions:

Name	Action	Name	Action
f_1	$f_1(0) = 0$ $f_1(1) = 0$	f_2	$f_2(0) = 1$ $f_2(1) = 1$
f_3	$f_3(0) = 0$ $f_3(1) = 1$	f_4	$f_4(0) = 1$ $f_4(1) = 0$

Functions f_1 and f_2 are *constant* (same output no matter what the input), and f_3 and f_4 are balanced.

How many queries do you need to make to the oracle to determine if the function is constant or balanced? (i.e., either one of f_1/f_2, or one of f_3/f_4).

Name	Action	Name	Action
f_1	$f_1(0) = 0$ $f_1(1) = 0$	f_2	$f_2(0) = 1$ $f_2(1) = 1$
f_3	$f_3(0) = 0$ $f_3(1) = 1$	f_4	$f_4(0) = 1$ $f_4(1) = 0$

How many queries do you need to make to the oracle to determine if the function is constant or balanced? (i.e., either one of f_1/f_2, or one of f_3/f_4).

Name	Action	Name	Action
f_1	$f_1(0) = 0$ $f_1(1) = 0$	f_2	$f_2(0) = 1$ $f_2(1) = 1$
f_3	$f_3(0) = 0$ $f_3(1) = 1$	f_4	$f_4(0) = 1$ $f_4(1) = 0$

Classical solution: 2
We always need to query both inputs 0 and 1 to find out the nature of the function.

How many queries do you need to make to the oracle to determine if the function is constant or balanced? (i.e., either one of f_1/f_2, or one of f_3/f_4).

Name	Action	Name	Action
f_1	$f_1(0) = 0$ $f_1(1) = 0$	f_2	$f_2(0) = 1$ $f_2(1) = 1$
f_3	$f_3(0) = 0$ $f_3(1) = 1$	f_4	$f_4(0) = 1$ $f_4(1) = 0$

Quantum solution: 1
How???

Next time

Last few classes:

- Deutsch's algorithm, amplitude amplification, Grover's algorithm

Action items:

1. Literacy assignment 3
2. Project code and presentation

Recommended reading:

- Quantum volume demo `https://pennylane.ai/qml/demos/quantum_volume.html`
- Check Piazza tonight/tomorrow for more links about metrics and characterization
- Codebook modules A and G