

COMSW4281 Lecture Notes
Professor Anargyros Papageorgiou
Spring 2007

Lecture 1 (17 January)

Course Information

Course email: cs4281@columbia.edu (to contact TA, submit homeworks)
Professor email: ap@cs
Textbook: Nielsen & Chuang
Courseworks
Grading: Homework 30%, Midterm 30%, Final 40%
Matlab recommended for programming
Quizzes might be given, averaged into midterm/final grades
Office Hours: Tues 1-2, Wed 12-1

Why Quantum Computing

1. Moore's Law - Density of transistors increases every 18 months, which means within 10-20 years quantum effects will become a barrier
2. Quantum Algorithms - Factoring, Search Algorithms

Best Classical Factoring Algorithm: Number Field

Performance: $\log 2^{C(\log N)^{1/2}} (\log \log N)^{2/3}$

Quantum Algorithm Discovered 1994 by Peter Shor

Performance: $(\log N)^3$ w/ probability $O(1)$ (or $> \frac{1}{2}$)

In general, quantum algorithms do not give results with certainty, but if probability is greater than one half, they can be repeated to get sufficient certainty

Discrete Logarithm Problem: given numbers a and $b = a^s$, find s .

Quantum solution: 1 query $O((\log s)^2)$

When looking at quantum performance, have to look at the number of queries as well as number of operations.

Search Algorithm

Given binary string length N (for some huge N), find position of substring

Classical Lower Bound $O(N)$, even for randomized algorithms, no success guarantee

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

Quantum Algorithm only has polynomial advantage over classical, not exponential as in factoring

Grover's Algorithm

Boolean mean, Numerical integration - related to search, same polynomial speedup

3. Quantum cryptography for key distribution

Can replace PKI, which assumes that factoring is hard. Private key distribution over a quantum channel can't be eavesdropped upon without detection. Earliest successes and applications of quantum computing are in this area.

4. Study of Quantum Mechanics as a model of computation

Church-Turing Thesis - any algorithmic computation can be simulated (efficiently) by a Turing machine (which is probabilistic).

(efficiently) - part of strong version of thesis, weaker one leaves it out
 (Thesis) - means conjecture or belief, not a provable theorem.
 (which is probabilistic) - fix for thesis added in 1970s. Solovay Strassev came up with a randomized algorithm for primality testing which can give arbitrary levels of certainty at vastly improved efficiency over deterministic algorithm

Deutsch 1985

Came up with quantum model of computation in order to produce more solid version of CT thesis. Conjectured a Universal Quantum Computer that can simulate any physical process. It isn't known if this is really possible. (His paper is in courseworks, first 3-4 pages are philosophical and accessible, rest is more technical)

5. Study of Entanglement. 2 qubit system that can't be separated... EPR state... Quantum Teleportation... More later in the course.

Quantum Systems

Size of quantum system is C^N where N is number of particles, C is number of coordinates, set of complex numbers used to represent a state.

Quantum states can be expressed as vectors

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{C}^N$$

Dirac notation is used in many cases instead of standard linear algebra notation. Dirac "captures the transpose of a matrix" (X^H - hermetian transpose of X)

For quantum states, $\|x\| = 1$. ($\|x\|$ - euclidean norm of X, square root of sum of squares)

States are transformed by unitary matrices U_x . Unitary means preserves length, that $UU^H = I$, or $U^{-1} = U^H$. Examples of unitary matrices: identity matrix, any rotation matrix, and the Hausholder matrix ($P = I - 2UU^H$, $\|U\| = 1$). Hausholder matrix has something to do with mirrors/reflection.

Quantum Computing Nutshell

X_0 - initial states

$Y = V_T = U_3 U_2 U_1 X_0$ - result is unitary matrices applied to input

Complexity is $n \cdot T$, where T is number of matrices, $n = \log N$ = number of qubits, and N = length of quantum register

Lecture 2 (22 January)

Quantum Computation

Start with X_0 , initial quantum state which is a vector. Apply unitary operations, $U_1 \dots U_t$ (Unitary meaning $U_j^H U_j = I$). Cost can be measured in terms of number of qubits n , and number of operations t .

Qubits

Classical computation uses bits with states 0 and 1

Quantum computation uses qubits whose states are superpositions of states $|0\rangle$ and $|1\rangle$. (pronounced "ket zero" and "ket one.")

Superposition states of qubits occur in a variety of physical systems. Simplest example is an electron that can be in a ground state 0 and excited state 1. You can shine light to put electron in excited state and reduce light to put it into ground state, or you can increase and decrease light repeatedly to put electron into superposition state. Other examples of superposition states occur in polarizations of photons, and intermediate angle spins of electrons traveling through magnetic fields

Mathematical description of qubits. Formally, a qubit is an element of a two dimensional hilbert space \mathcal{H} .

A hilbert space is _____. (couldn't make out word)

Qubit states can be represented as two complex numbers a and b, in a vector like

$$\begin{pmatrix} a \\ b \end{pmatrix} \in \mathbb{C}^2$$

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

Euclidean norms are 1 ($\| |0\rangle \| = 1, \| |1\rangle \| = 1$)

$|0\rangle$ and $|1\rangle$ are orthonormal ($|0\rangle \cdot |1\rangle = 0$)

Dirac Notation

$|x\rangle$ called "ket", denotes coordinate vector $\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$

$\langle x|$ called "bra" denotes $(\bar{x}_1 \ \bar{x}_2 \ \dots \ \bar{x}_n)$
(where \bar{n} is complex conjugate of n)

$\langle x|y\rangle$ called "braket" denotes inner product, magnitude of projection of y on to x.

$$\langle x|y\rangle = \langle x| \cdot |y\rangle = (\bar{x}_1 \ \bar{x}_2 \ \dots \ \bar{x}_n) \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \sum_{j=1}^n \bar{x}_j y_j$$

$$\langle x|x\rangle = \| |x\rangle \|^2$$

A qubit is a unitary vector in hilbert space.

$$|y\rangle = a|0\rangle + b|1\rangle \text{ where } a, b \in \mathbb{C} \text{ and } |a|^2 + |b|^2 = 1$$

A qubit is a linear combination of ket 0 and ket 1.

$$|y\rangle = a \begin{pmatrix} 1 \\ 0 \end{pmatrix} + b \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix}$$

Showing qubit y is unitary.

$$\| |y\rangle \|^2 = (\bar{a} \ \bar{b}) \begin{pmatrix} a \\ b \end{pmatrix} = \bar{a}a + \bar{b}b = |a|^2 + |b|^2 = 1$$

Showing qubit y is unitary in dirac notation.

$$\begin{aligned} \| |y\rangle \|^2 &= \langle y|y\rangle = \langle y| \cdot |y\rangle \\ &= (\bar{a} \langle 0| + \bar{b} \langle 1|) \cdot (a|0\rangle + b|1\rangle) \\ &= \bar{a}a \langle 0|0\rangle + \bar{a}b \langle 0|1\rangle + \bar{b}a \langle 1|0\rangle + \bar{b}b \langle 1|1\rangle \\ &= \bar{a}a + \bar{b}b = |a|^2 + |b|^2 = 1 \end{aligned}$$

One feature of dirac notation demonstrated here is that you can treat matrix multiplication as regular multiplication

Bloch Sphere representation of Qubits

Bloch sphere lets you represent qubits in a picture

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

Substitute $\alpha = |\alpha| e^{i\varphi_\alpha}$, $\beta = |\beta| e^{i\varphi_\beta}$

$$|\psi\rangle = |\alpha| e^{i\varphi_\alpha} |0\rangle + |\beta| e^{i\varphi_\beta} |1\rangle$$

$$|\psi\rangle = e^{i\varphi_\alpha} (|\alpha| |0\rangle + |\beta| e^{i(\varphi_\beta - \varphi_\alpha)} |1\rangle)$$

Substitute $\varphi = \varphi_\beta - \varphi_\alpha$, $\gamma = \varphi_\alpha$ for $\varphi \in [0, 2\pi]$

$$|\psi\rangle = e^{i\gamma} (|\alpha| |0\rangle + |\beta| e^{i\varphi} |1\rangle)$$

Substitute $|\alpha| = \cos \vartheta$, $|\beta| = \sin \vartheta$, for $\vartheta \in [0, \frac{\pi}{2}]$

$$|\psi\rangle = e^{i\gamma} (\cos \vartheta |0\rangle + \sin \vartheta e^{i\varphi} |1\rangle)$$

Substitute $|\alpha| = \cos \frac{\vartheta}{2}$, $|\beta| = \sin \frac{\vartheta}{2}$, for $\vartheta \in [0, \pi]$

$$|\psi\rangle = e^{i\gamma} (\cos \frac{\vartheta}{2} |0\rangle + \sin \frac{\vartheta}{2} e^{i\varphi} |1\rangle)$$

Any qubit can be expressed this way, in terms of γ , ϑ , and φ . And if you disregard γ , (global phase which it turns out to be impossible to physically measure anyway), you can take angles ϑ and φ and use them to plot points on a unit sphere.

(Drawing bloch spheres

- z axis up, y axis right, x axis out

- ϑ = angle between point and z axis (latitude, but starting at north pole and extending down)

- φ = angle between projection of point at z=0 and x axis (longitude)

Textbook Figure 1.3, page 15)

Measuring Qubits

Attempting to measure a qubit in a superposition state, $|y\rangle = a|0\rangle + b|1\rangle$, will give you state $|0\rangle$ with probability $|a|^2$, and $|1\rangle$ with probability $|b|^2$. After measuring the state, the qubit collapses, it ceases to be a superposition, and it changes to whatever state the measurement gave, either $|0\rangle$ or $|1\rangle$.

Collapse is a non-unitary transformation. It can be considered a projection and renormalization.

Using standard linear algebra notation, you can express measurement of some state u (where $u \in \mathbb{C}^N$) from a superposition state x as being a projection of x onto u , followed by the normalization. To compute the result of the projection, you can multiply x by a matrix M where $M = u \cdot u^H$. M is called a projection matrix and Mx will be the result of projecting x onto u . You don't actually need to form the matrix, however, since $Mx = uu^Hx$, and u^Hx can be computed as a simple dot product.

Using dirac notation, and taking a measurement of $|0\rangle$ as an example:

$$\begin{aligned} M|x\rangle &= |0\rangle \langle 0|x\rangle \\ &= |0\rangle \langle 0|(a|0\rangle + b|1\rangle) \\ &= a|0\rangle \langle 0|0\rangle + b|0\rangle \langle 0|1\rangle \\ &= a|0\rangle \end{aligned}$$

After normalizing, final state is $\frac{a}{|a|} |0\rangle$. Same process can be used for a measurement of $|1\rangle$ to express final state as $\frac{b}{|b|} |1\rangle$

Lecture 3 (24 January)

Course Information

Midterm - Wednesday, 7 March
Midterm Review - Monday before

Lecture 2 Review

$$|y\rangle = a|0\rangle + b|1\rangle = e^{i\gamma} \left(\cos \frac{\vartheta}{2} |0\rangle + e^{i\varphi} \sin \frac{\vartheta}{2} |1\rangle \right)$$

Collapse probability

$$|0\rangle, p = |a|^2$$

$$|1\rangle, p = |b|^2$$

Alternate Basis

$$|+\rangle = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) = \cos\left(\frac{\pi}{4}\right) |0\rangle + e^{i0} \sin\left(\frac{\pi}{4}\right) |1\rangle$$

$$|-\rangle = \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) = \cos\left(\frac{\pi}{4}\right) |0\rangle + e^{i\pi} \sin\left(\frac{\pi}{4}\right) |1\rangle$$

Bloch sphere representations

$|+\rangle$, $\vartheta = \pi/2$, $\varphi = 0$, on equator at front of sphere

$|-\rangle$, $\vartheta = \pi/2$, $\varphi = \pi$, on equator at back of sphere

Proving orthogonality of $|+\rangle$ and $|-\rangle$:

$$\begin{aligned} \langle + | - \rangle &= \left(\frac{1}{\sqrt{2}} (\langle 0| + \langle 1|) \right) \cdot \left(\frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \right) \\ &= \frac{1}{2} (\langle 0|0\rangle - \langle 0|1\rangle + \langle 1|0\rangle - \langle 1|1\rangle) \\ &= \frac{1}{2} (1 - 0 + 0 - 1) = 0 \end{aligned}$$

Proving $|+\rangle$ is unit vector (proof is similar for $|-\rangle$):

$$\begin{aligned} \langle + | + \rangle &= \left(\frac{1}{\sqrt{2}} (\langle 0| + \langle 1|) \right) \cdot \left(\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \right) \\ &= \frac{1}{2} (\langle 0|0\rangle + \langle 0|1\rangle + \langle 1|0\rangle + \langle 1|1\rangle) \\ &= \frac{1}{2} (1 + 0 + 0 + 1) = 1 \end{aligned}$$

Equivalences, change of base

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

$$|1\rangle = \frac{1}{\sqrt{2}} (|+\rangle - |-\rangle)$$

Hadamard Gate

Hadamard matrix maps $|0\rangle$ to $|+\rangle$, $|1\rangle$ to $|-\rangle$

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

You can see it by looking at columns. In general when looking at a transformation matrix, the first column shows what first basis vector ($|0\rangle$) is transformed to, second column shows second basis vector ($|1\rangle$), and so on.

Hadamard mapping for general state $|y\rangle = a|0\rangle + b|1\rangle$

$$\begin{aligned} H|y\rangle &= \frac{1}{\sqrt{2}} ((a+b)|0\rangle + (a-b)|1\rangle) \\ &= \frac{1}{\sqrt{2}} (a(|0\rangle + |1\rangle) + b(|0\rangle - |1\rangle)) \\ &= a|+\rangle + b|-\rangle \end{aligned}$$

Hadamard transform is unitary and is its own inverse.

Example:

$$\begin{aligned} H|0\rangle &= \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \\ HH|0\rangle &= \frac{1}{\sqrt{2}} (H|0\rangle + H|1\rangle) = \frac{1}{2} (|0\rangle + |1\rangle + |0\rangle - |1\rangle) = |0\rangle \end{aligned}$$

The two half- $|0\rangle$ kets adding up is called constructive interference. The two half- $|1\rangle$ kets cancelling out is called destructive interference.

Other Gates

Pauli Matrix:

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

X is analogous to NOT gate:

$$\begin{aligned} X|0\rangle &= |1\rangle \\ X|1\rangle &= |0\rangle \\ X|y\rangle &= a|1\rangle + b|0\rangle \end{aligned}$$

in Dirac notation:

$$\begin{aligned} X &= |1\rangle\langle 0| + |0\rangle\langle 1| \\ &= \begin{pmatrix} 0 \\ 1 \end{pmatrix} (1 \ 0) + \begin{pmatrix} 1 \\ 0 \end{pmatrix} (0 \ 1) \\ &= \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{aligned}$$

In general, when you have a unitary operation U and you know how it transforms basis states: $U|0\rangle = |s\rangle$, $U|1\rangle = |t\rangle$, you can express U as:

$$U = |s\rangle\langle 0| + |t\rangle\langle 1|$$

Verification:

$$\begin{aligned} U|0\rangle &= |s\rangle\langle 0|0\rangle + |t\rangle\langle 1|0\rangle = |s\rangle \\ U|1\rangle &= |s\rangle\langle 0|1\rangle + |t\rangle\langle 1|1\rangle = |t\rangle \end{aligned}$$

Linear Algebra Review

Given U is unitary matrix, $\langle y|y\rangle = 1, U|y\rangle = \lambda|y\rangle$. Eigenvalues $\lambda \in \mathbb{C}$ can be expressed as $\lambda = e^{it}$, $t \in \mathbb{R}$.

Show unitary matrix has eigenvalues of unit length:

$$\begin{aligned}\langle y|U^H &= \bar{\lambda}\langle y| \\ \langle y|U^H U|y\rangle &= \bar{\lambda}\lambda\langle y|y\rangle \\ 1 &= |\lambda|^2\end{aligned}$$

Because they have unit length, eigenvalues can be written in the form $\lambda = e^{it}$.

Show a hermitian matrix $A^H = A$ has eigenvalues that are real numbers:

$$\begin{aligned}A|y\rangle &= \lambda|y\rangle \\ \langle y|A|y\rangle &= \lambda\langle y|y\rangle = \lambda \\ \langle y|A^H &= \bar{\lambda}\langle y| \\ \langle y|A^H|y\rangle &= \bar{\lambda}\langle y|y\rangle = \bar{\lambda}\end{aligned}$$

$A = A^H$ therefore $\lambda = \bar{\lambda}$ therefore $\lambda \in \mathbb{R}$

Spectral Theorem

Unitary matrix U can be diagonalized as $U = V\Lambda V^H$ where

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_1 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \lambda_n \end{pmatrix}$$

and $V = (Y_1 \ Y_2 \ \dots \ Y_n)$, columns Y_i are eigenvectors.

The spectral theorem applies more generally to any Normal matrix. Normal matrices commute with their transpose, $UU^H = U^H U$.

(Matrix types

Normal - $U^H U = U U^H$

Unitary - $U^H U = U U^H = I$, type of normal matrix

Hermetian - $U^H = U$, type of normal matrix)

Pauli Matrices

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

Eigenvalues are $\lambda_1 = 1$, $\lambda_2 = -1$. Eigenvectors:

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

Called X matrix because on Bloch Sphere, eigenvectors are aligned with X axis.

$$\begin{aligned}
Y &= \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \\
Y|0\rangle &= i|1\rangle \\
Y|1\rangle &= -i|0\rangle \\
Y|y\rangle &= ia|1\rangle - ib|0\rangle
\end{aligned}$$

Eigenvectors are

$$|\otimes\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ i \end{pmatrix} = \frac{1}{\sqrt{2}} (|0\rangle + i|1\rangle), \vartheta = \frac{\pi}{2}, \varphi = \frac{\pi}{2}$$

$$|\oplus\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -i \end{pmatrix} = \frac{1}{\sqrt{2}} (|0\rangle - i|1\rangle), \vartheta = \frac{\pi}{2}, \varphi = \frac{3\pi}{2}$$

$$\begin{aligned}
Z &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \\
Z|y\rangle &= a|0\rangle - b|1\rangle = a|0\rangle - e^{i\pi}b|1\rangle
\end{aligned}$$

Lecture 4 (29 January)

Lecture 3 Review

Hadamard matrix

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Pauli Matrices

$$\begin{aligned}
X &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\
Y &= \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \\
Z &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi} \end{pmatrix}
\end{aligned}$$

First column of each matrix tells you where $|0\rangle$ maps, second column tells you where $|1\rangle$ maps.

New Gates

Phase gate

$$\begin{aligned}
S &= \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{pmatrix} \\
S^2 &= X
\end{aligned}$$

Pi over 8 gate

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix} = e^{i\frac{\pi}{8}} \begin{pmatrix} e^{-i\frac{\pi}{8}} & 0 \\ 0 & e^{i\frac{\pi}{8}} \end{pmatrix}$$

Homework tip: We are allowed to cite eigenvalues from class. We can also guess and verify other eigenvalues without going through characteristic equations. Also, for diagonal matrices, eigenvalues can be read off the diagonal and eigenvectors are $|0\rangle$ and $|1\rangle$.

Multiple Qubit Systems

Starting with 2 qubits. 2 qubits mean 4 base states: $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ or, equivalently $|0\rangle|0\rangle, |0\rangle|1\rangle, |1\rangle|0\rangle, |1\rangle|1\rangle$.

An arbitrary state is a superposition

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

constrained by $\sum_{j,k} |a_{jk}|^2 = 1, a_{jk} \in \mathbb{C}$

Measurement of any outcome j occurs with probability $|a_j|^2$. Following measurement $|y\rangle$ collapses to $|j\rangle$. It is also possible to make partial measurements, measuring some qubits but not others, taking sums as you would expect. (The probability of making the partial measurement is just the sum of probabilities of the full measurements containing the partial measurement. Following measurement, base states that aren't compatible with the measurement have their coefficients set to 0 and the rest are renormalized.)

Combining qubit states

$$\begin{aligned} |y_1\rangle &= a_1|0\rangle + b_1|1\rangle \\ |y_2\rangle &= a_2|0\rangle + b_2|1\rangle \end{aligned}$$

You can combine two independent qubit states to get a state describing the probability of each joint outcome

$$\begin{aligned} |y_1y_2\rangle &= |y_1\rangle|y_2\rangle = (a_1|0\rangle + b_1|1\rangle)(a_2|0\rangle + b_2|1\rangle) \\ &= a_1|0\rangle a_2|0\rangle + a_1|0\rangle b_2|1\rangle + b_1|1\rangle a_2|0\rangle + b_1|1\rangle b_2|1\rangle \\ &= a_1a_2|00\rangle + a_1b_2|01\rangle + b_1a_2|10\rangle + b_1b_2|11\rangle \end{aligned}$$

(Note: all the products in the above equations are really tensor products, which are defined below. In previous lecture notes, products inside dirac expressions implied standard matrix multiplication, which makes no sense here.)

You cannot generally break a multiple qubit state up into separate independent states. Example: EPR state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. You can see visually that there are no values for a_1, b_1, a_2, b_2 that will let you write the EPR state in above form.

For an n -qubit system, each basis state can be expressed as a bitstring $|j_{n-1}j_{n-2}\dots j_0\rangle$ for $j_m \in \{0, 1\}$. Or, for convenience, it can just be written as a normal number $|j\rangle$ where $j = \sum_{m=0}^{n-1} 2^m j_m$

Now that we have a notion for multiple qubit states, have to answer 3 questions:

Q1: What is the coordinate representation of a state $|j\rangle$?

Q2: How can you decompose and recompose states? For example, how do you compute $|y\rangle = |y_1\rangle|y_2\rangle$ where $|y_1\rangle$ and $|y_2\rangle$ are multiple qubit states.

Q3: How can you compose operations. Given $|y_1\rangle$ and $|y_2\rangle$ which are multiple qubit states and unitary operators U_1 and U_2 which apply to $|y_1\rangle$ and $|y_2\rangle$, respectively, how can you find a U matrix that satisfies $U|y\rangle = U_1|y_1\rangle U_2|y_2\rangle$

Once you answer these questions you can start looking at algorithms.

Tensor Products

Tensor products are a new concept.

Given two matrices, A which has size $m \times n$, and B which has size $p \times q$, the tensor product $A \otimes B$ will be a huge matrix of size $mp \times nq$:

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & \vdots & \ddots & \\ a_{m1}B & a_{m2}B & & a_{mn}B \end{pmatrix}$$

Example:

$$\begin{aligned} \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \otimes \begin{pmatrix} 10 & 0 \\ 0 & 20 \end{pmatrix} &= \begin{matrix} 1 \begin{pmatrix} 10 & 0 \\ 0 & 20 \end{pmatrix} & 2 \begin{pmatrix} 10 & 0 \\ 0 & 20 \end{pmatrix} & 3 \begin{pmatrix} 10 & 0 \\ 0 & 20 \end{pmatrix} \\ 4 \begin{pmatrix} 10 & 0 \\ 0 & 20 \end{pmatrix} & 5 \begin{pmatrix} 10 & 0 \\ 0 & 20 \end{pmatrix} & 6 \begin{pmatrix} 10 & 0 \\ 0 & 20 \end{pmatrix} \end{matrix} \\ &= \begin{pmatrix} 10 & 0 & 20 & 0 & 30 & 0 \\ 0 & 20 & 0 & 40 & 0 & 60 \\ 40 & 0 & 50 & 0 & 60 & 0 \\ 0 & 80 & 0 & 100 & 0 & 120 \end{pmatrix} \end{aligned}$$

Example: (using column vectors):

$$\begin{aligned} X &= \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, Y = \begin{pmatrix} 10 \\ 100 \end{pmatrix} \\ X \otimes Y &= \begin{pmatrix} 10 \\ 100 \\ 20 \\ 200 \\ 30 \\ 300 \end{pmatrix}, Y \otimes X = \begin{pmatrix} 10 \\ 20 \\ 30 \\ 100 \\ 200 \\ 300 \end{pmatrix} \end{aligned}$$

Example: (using qubits)

$$\begin{aligned} X &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle, Y = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle \\ X \otimes Y &= |0\rangle \otimes |0\rangle = |0\rangle |0\rangle = |00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \end{aligned}$$

$$\text{Similarly, } |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

And more generally, $|j\rangle = |j_1 j_0\rangle$ will be a column vector which is all zeros except for single 1 entry at position $j + 1$ (see “Multiple Qubit Systems” above, $j = \sum_{m=0}^{n-1} 2^m j_m$).

Example: (using qubits again):

$$\begin{aligned} H|0\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ (H|0\rangle) \otimes (H|0\rangle) &= \left(\frac{1}{\sqrt{2}} \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right) \right) \otimes \left(\frac{1}{\sqrt{2}} \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right) \right) \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \otimes \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \end{aligned}$$

Repeat example using dirac notation:

$$\begin{aligned}
 (H|0\rangle) \otimes (H|0\rangle) &= \frac{1}{2} (|0\rangle + |1\rangle) (|0\rangle + |1\rangle) \\
 &= \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle + |11\rangle) \\
 &= \frac{1}{2} \left(\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \right) = \frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}
 \end{aligned}$$

Associative property of tensor product:

$$|x\rangle \otimes |y\rangle \otimes |z\rangle = |x\rangle \otimes |yz\rangle = |xy\rangle \otimes |z\rangle = |xyz\rangle$$

Notation for tensor exponentiation:

$$|\psi\rangle^{\otimes k} = |\psi\rangle \otimes |\psi\rangle \cdots \otimes |\psi\rangle$$

Powers of $H|0\rangle$

$$(H|0\rangle)^{\otimes k} = \frac{1}{2^{k/2}} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

where length of column vector is 2^k .

The formula is obvious for $k = 1$. Proof for rest of cases is by induction, (base case $k = 2$ was shown in previous section.) Inductive case is:

$$\begin{aligned}
 \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right)^{\otimes k} &= \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right)^{\otimes k-1} \otimes \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) \\
 &= \frac{1}{2^{(k-1)/2}} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \otimes \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{2^{k/2}} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}
 \end{aligned}$$

Tensor Products Continued

[DUNNO] I'm not exactly sure what the following equations are supposed to show. I think I was lost at the time I was taking these notes.

$$|j\rangle = |j_{n-1}j_{n-2} \cdots j_0\rangle$$

Now make an arbitrary split at qubit k .

$$\begin{aligned}
 |j\rangle &= |j_{n-1} \cdots j_k\rangle |j_{k-1} \cdots j_0\rangle \\
 &= |j^{(1)}\rangle |j^{(2)}\rangle
 \end{aligned}$$

$$\begin{aligned}
j &= \sum_{m=0}^{n-1} 2^m j_m \\
j^{(1)} &= \sum_{m=k}^{n-1} 2^{m-k} j_m \\
j^{(2)} &= \sum_{m=0}^{k-1} 2^m j_m
\end{aligned}$$

Known $n=1, n=2$.

$$\begin{aligned}
|j\rangle &= |j^{(1)}\rangle |j^{(2)}\rangle \\
&= \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix}
\end{aligned}$$

$|j^{(1)}\rangle$ has 1 at position $j^{(1)} + 1$
 $|j^{(2)}\rangle$ has 1 at position $j^{(2)} + 1$
 $|j\rangle$ has 1 at position $j + 1 = j^{(1)} \cdot 2^k + j^{(2)} + 1$

New Course Information

The class now has a TA: James Li (sp?)

Lecture 5 (31 January)

Lecture 4 Review

In n -dimensional qubit systems, basis vectors are $\begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} = |j\rangle$ with the 1 at position $j + 1$.

Any state $|j\rangle$ is a linear combination of j vectors.

Any outcome $|j\rangle$ has probability $|c_j|^2$ for $|y\rangle = \sum_{j=0}^{2^n-1} c_j |j\rangle$

First homework assignment is out today, 3 problems, due 14 Feb.

Showed last time that

$$(H|0\rangle)^{\otimes k} = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right)^{\otimes k} = \frac{1}{2^{k/2}} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

producing column vector of 2^k rows. Will be generalizing today for inputs other than $|0\rangle$.

Properties of Tensor Products

1. Associative property (see lecture 4)
2. Distributing scalar multiple over tensor product
 $a(|x\rangle \otimes |y\rangle) = (a|x\rangle) \otimes |y\rangle = |x\rangle \otimes (a|y\rangle)$
3. Distributing tensor product over addition
 $|y\rangle \otimes (|x_1\rangle + |x_2\rangle) = |y\rangle \otimes |x_1\rangle + |y\rangle \otimes |x_2\rangle$
4. Applying tensor products of operations individually to vectors.
 $(A \otimes B)(|x\rangle \otimes |y\rangle) = (A|x\rangle) \otimes (B|y\rangle)$

Example:

$$\begin{aligned} \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right)^{\otimes k} &= H|0\rangle \otimes \dots \otimes H|0\rangle \\ &= (H \otimes \dots \otimes H)(|0\rangle \otimes \dots \otimes |0\rangle) \\ &= H^{\otimes k} |0\rangle^{\otimes k} \end{aligned}$$

Example:

$$(X \otimes S)|00\rangle = X|0\rangle \otimes S|0\rangle$$

(X is NOT gate from lecture 3, S is phase gate from lecture 4)

Lets you apply transformations to individual inputs instead of applying huge combination operations with big matrices.

Example:

$$(A \otimes B) \left(\sum_j a_j |x_j\rangle |y_j\rangle \right) = \sum_j a_j (A|x_j\rangle) \otimes (B|y_j\rangle)$$

6. Distributing Hermitian transpose over tensor product
 $(A \otimes B)^H = A^H \otimes B^H$
7. Applying tensor product of operations to other operations.
 $(A \otimes B)(C \otimes D) = AC \otimes BD$

Proof can be stated in terms of property 4, just break down C and D into individual column vectors $|x\rangle$ and $|y\rangle$. This is a useful technique in general, substituting vectors to figure out how matrices work.

8. If A and B are unitary operations, then $A \otimes B$ is unitary.

Proof: $A^H A = I, B^H B = I$

$$(A \otimes B)^H (A \otimes B) = (A^H \otimes B^H) (A \otimes B) = (A^H A) \otimes (B^H B) = I \otimes I = I$$

Powers of Hadamard Gate

Result of applying hadamard transforms to each qubit in some base (non-superposition) state j , made of k qubits.

$$\begin{aligned} H^{\otimes k} |j_{k-1} \dots j_0\rangle &= H|j_{k-1}\rangle \dots H|j_0\rangle \\ &= \frac{|0\rangle + (-1)^{j_{k-1}} |1\rangle}{\sqrt{2}} \otimes \dots \otimes \frac{|0\rangle + (-1)^{j_0} |1\rangle}{\sqrt{2}} \\ &= \bigotimes_{s=k-1}^0 \frac{|0\rangle + (-1)^{j_s} |1\rangle}{\sqrt{2}} \\ &= \frac{1}{2^{k/2}} \sum_{m=0}^{2^k-1} (-1)^{j \cdot m} |m\rangle \end{aligned}$$

Where $j \cdot m = j_{k-1}m_{k-1} + j_{k-2}m_{k-2} + \dots + j_0m_0$. $j_i, m_i \in \{0, 1\}$ are digits of j and m expressed as binary strings.

(Observe that for some single qubit state, j_s , $H|j_s\rangle = \frac{|0\rangle + (-1)^{j_s}|1\rangle}{\sqrt{2}}$ to see the intuition in the last step. At $k = 2$,

$$\begin{aligned} & \frac{|0\rangle + (-1)^{j_1}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + (-1)^{j_0}|1\rangle}{\sqrt{2}} \\ &= |00\rangle + (-1)^{j_0}|01\rangle + (-1)^{j_1}|10\rangle + (-1)^{j_0+j_1}|11\rangle \end{aligned}$$

At $k = 3$,

$$\begin{aligned} & \frac{|0\rangle + (-1)^{j_2}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + (-1)^{j_1}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + (-1)^{j_0}|1\rangle}{\sqrt{2}} \\ &= |000\rangle + (-1)^{j_0}|001\rangle + (-1)^{j_1}|010\rangle + (-1)^{j_0+j_1}|011\rangle \\ &+ (-1)^{j_2}|100\rangle + (-1)^{j_0+j_2}|101\rangle + (-1)^{j_1+j_2}|110\rangle + (-1)^{j_0+j_1+j_2}|111\rangle \end{aligned}$$

You can see that $(-1)^{j_i}$ coefficients follow the $|1\rangle$'s and get included in the expanded terms where m_i is 1 instead of 0.)

For two qubit system, inputs $|+\rangle, |+\rangle$:

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} \rightarrow \begin{array}{l} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{array}$$

For $n + 1$ qubit system, inputs: $|+\rangle, |y\rangle$: $\frac{|0\rangle + |1\rangle}{\sqrt{2}} |y\rangle = \frac{|0y\rangle + |1y\rangle}{\sqrt{2}}$ (has $n + 1$ qubits if $|y\rangle$ has n qubits)

Solution is then made up of terms like $|m_{k-1}m_{k-2}\dots m_0\rangle$

if $m_{k-2} = 1$ then term has $|1\rangle$ at second cubit, but may be + or -

if $j_{k-2} = 1$ then it's $-|1\rangle$ else if $j_{k-2} = 0$ then $|1\rangle$

if $m_{k-2} = 0$ or $j_{k-2} = 0$ then no problem with + or - since we have 0 at location k-2

Summary: $(-1)^{j_{k-2}m_{k-2}} |?(0 \text{ or } 1)???\rangle$

Repeat for all qubits to get $(-1)^{j_{k-1}m_{k-1} + \dots + j_0m_0} |m_{k-1}\dots m_0\rangle$

Upshot: $H^{\otimes k} |j\rangle = \frac{1}{2^{k/2}} \sum_{m=0}^{2^k-1} (-1)^{m \cdot j} |m\rangle$

Homework hint: This is half of work needed to solve one of the problems. It tells you columns of the matrix. Homework asks you to find the whole matrix.

Properties of Tensor Products (continued)

9. If you have two states $|x\rangle$ and $|y\rangle$ which you can decompose with same dimensions.

$$|y\rangle = |y_1\rangle |y_2\rangle$$

$$|x\rangle = |x_1\rangle |x_2\rangle$$

$$\text{Then } \langle x|y\rangle = \langle x_1x_2|y_1y_2\rangle = \langle x_1|y_1\rangle \langle x_2|y_2\rangle$$

Example:

$$|d\rangle = |011000\rangle$$

$$|k\rangle = |011010\rangle$$

You can tell $\langle d|k\rangle = 0$ based solely on the fact that the fifth qubit's inner product ($\langle 0|1\rangle$) is zero.

Example:

$$\langle z_1|\langle z_2|X \otimes Y |\psi_1\rangle |\psi_2\rangle = \langle z_1|X |\psi_1\rangle \otimes \langle z_2|Y |\psi_2\rangle$$

Completeness Relation

$$|j\rangle\langle i| = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \begin{pmatrix} 0 & \cdots & 1 & \cdots & 0 \end{pmatrix} = \begin{pmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & & \ddots & \vdots \\ \vdots & & 1 & & \vdots \\ \vdots & \ddots & & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 \end{pmatrix}$$

$|j\rangle$ is column vector with 1 at position $j+1$

$\langle i|$ is row vector with 1 at position $i+1$

$|j\rangle\langle i|$ is projection matrix with 1 at row $j+1$, col $i+1$

Completeness relation

$$\sum_{i=0}^{2^n-1} |i\rangle\langle i| = I$$

Next lecture will show completeness relation holds on any orthonormal basis, not just i .

Lecture 6 (5 February)

Lecture 5 Review

- $H^{\otimes k} |j\rangle = \frac{1}{2^{k/2}} \sum_{m=0}^{2^k-1} (-1)^{j \cdot m} |m\rangle$
- $\langle x_1| \otimes \langle x_2| \cdot |y_1\rangle \otimes |y_2\rangle = \langle x_1|y_1\rangle \otimes \langle x_2|y_2\rangle$

Example:

$$\langle 01| X \otimes A |01\rangle = \langle 0| X |0\rangle \otimes \langle 1| A |1\rangle$$

- $I = \sum_{j=0}^{2^n-1} |j\rangle\langle j|$

Completeness Relation

A proof of the completeness relation (3 above) was shown last lecture based on adding up projections to get the diagonal matrix. This is another proof:

An arbitrary state is a linear combination of basis states:

$$|y\rangle = \sum_j c_j |x_j\rangle$$

Each constant is just:

$$c_j = \langle x_j|y\rangle \in \mathbb{C}$$

Substituting c_j above gives:

$$|y\rangle = \sum_j |x_j\rangle \langle x_j|y\rangle$$

In general

$$|y\rangle = A |y\rangle \rightarrow A = I$$

So

$$\sum_j |x_j\rangle \langle x_j| = I$$

$$A = AI = A \sum_j |x_j\rangle \langle x_j| = \sum_j (A |x_j\rangle) \langle x_j|$$

More Linear Algebra

An $n \times n$ matrix has n eigenvalues ($\lambda_i \in \mathbb{C}$) and orthonormal eigenvectors ($|x_j\rangle$ $j = \{1 \dots n\}$) iff A is normal ($A^H A = A A^H$).

$V = (|x_1\rangle \ \dots \ |x_n\rangle)$, eigenvector matrix, $V^H V = I$

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \lambda_n \end{pmatrix}, \text{ eigenvalue matrix}$$

$$\begin{aligned} A &= V \Lambda V^H = (\lambda_1 |x_1\rangle \ \dots \ \lambda_n |x_n\rangle) \begin{pmatrix} \langle x_1| \\ \vdots \\ \langle x_n| \end{pmatrix} \\ &= \sum_{j=1}^n \lambda_j |x_j\rangle \langle x_j| \end{aligned}$$

Eigenvalues and Tensor Products:

If A eigenvalues are λ_j , eigenvectors are $|x_j\rangle$, for $j=1..n$,
and B eigenvalues are λ_k , eigenvectors are $|x_k\rangle$, for $k=1..n$
then $A \otimes B$ eigenvalues are $\lambda_j \lambda_k$, eigenvectors are $|x_j\rangle |x_k\rangle$

For any matrix A , you can compute A^2, A^3, A^4

If A is nonsingular, you can compute A^{-1}, A^{-2}

What about a general $f: \mathbb{D} \rightarrow \mathbb{C}$, like $\sin(A), e^A, f(A)$

If A is normal and $f(\lambda_j)$ is well defined for all eigenvalues

$$\begin{aligned} f(A) &= \sum_j f(\lambda_j) |x_j\rangle \langle x_j| \\ &= V \begin{pmatrix} f(\lambda_1) & 0 & 0 & 0 \\ 0 & f(\lambda_2) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & f(\lambda_n) \end{pmatrix} V^H \end{aligned}$$

Example:

$$f(z) = z^k$$

$$f(A) = A^k = (V \Lambda V^H)(V \Lambda V^H) \dots (V \Lambda V^H) \text{ [multiplied } k \text{ times]}$$

$$= V \Lambda^k V^H$$

If $A \in \mathbb{R}^{n,n}$ and $A = A^T$ then e^{iA} is unitary. Proof:

$$\begin{aligned} (e^{iA})^H (e^{iA}) &= \left(\sum_j e^{i\lambda_j} |x_j\rangle \langle x_j| \right)^H \left(\sum_j e^{i\lambda_j} |x_j\rangle \langle x_j| \right) \\ &= \left(\sum_j e^{-i\lambda_j} |x_j\rangle \langle x_j| \right) \left(\sum_j e^{i\lambda_j} |x_j\rangle \langle x_j| \right) \\ &= \sum_{j,k} e^{-i\lambda_j} e^{i\lambda_k} |x_j\rangle \langle x_j| |x_k\rangle \langle x_k| \\ &= \sum_j 1 |x_j\rangle \langle x_j| = I \end{aligned}$$

Controlled Gates

Controlled gates have a control input and a normal input. Control output is always the same as the control input. If control input is $|0\rangle$, normal output is the same as the normal input. If control input $|1\rangle$, normal output is some function of the normal input, where the function depends on the type of controlled gate.

Controlled Not Gate

$$|00\rangle \rightarrow |00\rangle$$

$$|01\rangle \rightarrow |01\rangle$$

$$|10\rangle \rightarrow |11\rangle$$

$$|11\rangle \rightarrow |10\rangle$$

$$Q_{\text{CNOT}} |i\rangle |j\rangle = |i\rangle |i \oplus j\rangle$$

$$Q_{\text{CNOT}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix}$$

Each column of matrix is derived directly by writing CNOT mappings listed above for $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$ inputs. Matrix can be shown to be unitary by multiplying with its conjugate transpose and getting the identity matrix.

CNOT for Hadamard inputs:

$$|+\rangle |+\rangle \rightarrow |+\rangle |+\rangle$$

$$|-\rangle |+\rangle \rightarrow |-\rangle |+\rangle$$

$$|+\rangle |-\rangle \rightarrow |-\rangle |-\rangle$$

$$|-\rangle |-\rangle \rightarrow |+\rangle |-\rangle$$

When dealing with inputs that aren't 0 and 1, calling the same input the "control input" doesn't necessarily make sense. When superposition states are fed to the CNOT, as above, the state of the second qubit controls a NOT operation on the first.

Formally:

$$Q_{\text{CNOT}} H |i\rangle H |j\rangle = H |i \oplus j\rangle H |j\rangle$$

for $i, j \in 0, 1$

Proof:

$$\begin{aligned} Q_{\text{CNOT}} (H|i\rangle H|j\rangle) &= Q_{\text{CNOT}} \left(\left(\frac{|0\rangle + (-1)^i |1\rangle}{\sqrt{2}} \right) \left(\frac{|0\rangle + (-1)^j |1\rangle}{\sqrt{2}} \right) \right) \\ &= Q_{\text{CNOT}} \left(\frac{1}{2} \left(|00\rangle + (-1)^j |01\rangle + (-1)^i |10\rangle + (-1)^{i+j} |11\rangle \right) \right) \\ &= \frac{1}{2} \left(|00\rangle + (-1)^j |01\rangle + (-1)^i |11\rangle + (-1)^{i+j} |10\rangle \right) \\ &= \frac{1}{2} \left(|00\rangle + (-1)^j |01\rangle + (-1)^{i+j} |10\rangle + (-1)^i |11\rangle \right) \\ &= \frac{1}{2} \left(|00\rangle + (-1)^j |01\rangle + (-1)^{i+j} |10\rangle + (-1)^{i+j+j} |11\rangle \right) \\ &= \left(\frac{|0\rangle + (-1)^{i+j} |1\rangle}{\sqrt{2}} \right) \left(\frac{|0\rangle + (-1)^j |1\rangle}{\sqrt{2}} \right) \\ &= H |i \oplus j\rangle H |j\rangle \end{aligned}$$

Example circuit:

$$(H \otimes H) (Q_{\text{CNOT}}) (H \otimes H) |i\rangle |j\rangle$$

$$|i\rangle |j\rangle \xrightarrow{H \otimes H} H |i\rangle H |j\rangle \xrightarrow{Q_{\text{CNOT}}} H |i \oplus j\rangle H |j\rangle \xrightarrow{H \otimes H} H^{\otimes 2} |i \oplus j\rangle H^{\otimes 2} |j\rangle = |i \oplus j\rangle |j\rangle$$

Controlled-U Gate

Notation looks like $|i\rangle |j\rangle \xrightarrow{Q_{C-U}} |i\rangle U^i |j\rangle$

$$\begin{aligned} |00\rangle &\rightarrow |00\rangle \\ |01\rangle &\rightarrow |01\rangle \\ |10\rangle &\rightarrow |1\rangle Q |0\rangle \\ |11\rangle &\rightarrow |1\rangle Q |1\rangle \end{aligned}$$

$$Q_{C-U} = \begin{pmatrix} I & 0 \\ 0 & U \end{pmatrix}$$

Lecture 7 (7 February)

Lecture 6 Review

Circuit: $(|A\rangle |B\rangle) = Q_{\text{CNOT}} |i\rangle |j\rangle$

If i and j are defined on the computational basis, output is $|i\rangle |i \oplus j\rangle$, but output in terms of some other set of basis states will be expressed differently.

Controlled Z Gate

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

$$Q_Z |i\rangle |j\rangle = |i\rangle Z^i |j\rangle$$

$$\begin{aligned} Z |0\rangle &= |0\rangle \\ Z |1\rangle &= -|1\rangle \end{aligned} \Rightarrow Z |j\rangle = (-1)^j |j\rangle$$

$$Q_Z |i\rangle |j\rangle = (-1)^{ij} |i\rangle |j\rangle$$

The gate with the control reversed, $Z^j |i\rangle |j\rangle$, going through the steps above, has the exact same definition, $(-1)^{ij} |i\rangle |j\rangle$. So for the controlled Z gate, it is reasonable to think of either input as being the controlling input on the computational basis.

Swap Gate

Circuit: $(Q_{\text{CNOT}}) (Q'_{\text{CNOT}}) (Q_{\text{CNOT}}) |i\rangle |j\rangle$

where $Q_{\text{CNOT}} |i\rangle |j\rangle = |i\rangle X^i |j\rangle$ and $Q'_{\text{CNOT}} |i\rangle |j\rangle = X^j |i\rangle |j\rangle$

$$|i\rangle |j\rangle \xrightarrow{Q_{\text{CNOT}}} |i\rangle |i \oplus j\rangle \xrightarrow{Q'_{\text{CNOT}}} |i \oplus i \oplus j = j\rangle |i \oplus j\rangle \xrightarrow{Q_{\text{CNOT}}} |j\rangle |i\rangle$$

For arbitrary states:

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

1-Qubit Gates as Rotations

(See also section 4.2 of the textbook)

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

Implementation of all 1-qubit gates can be seen as counterclockwise rotations by an angle ϑ , of points plotted on the Bloch sphere around the X, Y, and Z axes.

$$\begin{aligned}
 e^{-i\vartheta X/2} &= R_X(\vartheta) = \cos\left(\frac{\vartheta}{2}\right)I - i\sin\left(\frac{\vartheta}{2}\right)X = \begin{pmatrix} \cos\left(\frac{\vartheta}{2}\right) & -i\sin\left(\frac{\vartheta}{2}\right) \\ -i\sin\left(\frac{\vartheta}{2}\right) & \cos\left(\frac{\vartheta}{2}\right) \end{pmatrix} \\
 e^{-i\vartheta Y/2} &= R_Y(\vartheta) = \cos\left(\frac{\vartheta}{2}\right)I - i\sin\left(\frac{\vartheta}{2}\right)Y = \begin{pmatrix} \cos\left(\frac{\vartheta}{2}\right) & -\sin\left(\frac{\vartheta}{2}\right) \\ \sin\left(\frac{\vartheta}{2}\right) & \cos\left(\frac{\vartheta}{2}\right) \end{pmatrix} \\
 e^{-i\vartheta Z/2} &= R_Z(\vartheta) = \cos\left(\frac{\vartheta}{2}\right)I - i\sin\left(\frac{\vartheta}{2}\right)Z = \begin{pmatrix} e^{-i\vartheta/2} & 0 \\ 0 & e^{i\vartheta/2} \end{pmatrix}
 \end{aligned}$$

Proof of the equations above will be done in two parts. First, by deriving the matrices from rotations around the axes, and then by showing that the exponential forms are equivalent.

One general note to make here is that any matrix of the form $\begin{pmatrix} e^{ia} & 0 \\ 0 & e^{ib} \end{pmatrix}$ can be expressed as a rotation around the Z axis by some angle. Just observe:

$$\begin{pmatrix} e^{ia} & 0 \\ 0 & e^{ib} \end{pmatrix} = e^{i\frac{a+b}{2}} \begin{pmatrix} e^{i\frac{a-b}{2}} & 0 \\ 0 & e^{i\frac{b-a}{2}} \end{pmatrix}$$

Rotation around X axis

Take an arbitrary qubit $|y\rangle = \cos\frac{\vartheta_1}{2}|0\rangle + e^{i\varphi}\sin\frac{\vartheta_1}{2}|1\rangle$. Trying to find an expression for this qubit after a rotation about the X axis is messy. But for the case where $\varphi = \frac{\pi}{2}$, finding the rotation is easy, and using that case is sufficient for finding the general rotation matrix. So let

$$|\tilde{y}\rangle = \cos\frac{\vartheta_1}{2}|0\rangle + e^{i\pi/2}\sin\frac{\vartheta_1}{2}|1\rangle = \cos\frac{\vartheta_1}{2}|0\rangle + i\sin\frac{\vartheta_1}{2}|1\rangle$$

This is a projection of $|y\rangle$ onto the YZ plane ($x=0$). Rotating this point ϑ radians counterclockwise around the X axis is the same thing as subtracting ϑ from ϑ_1 . The rotated point is

$$|\tilde{y}'\rangle = \cos\frac{\vartheta_1 - \vartheta}{2}|0\rangle + i\sin\frac{\vartheta_1 - \vartheta}{2}|1\rangle$$

In terms of a rotation matrix, the mapping from $|\tilde{y}\rangle$ to $|\tilde{y}'\rangle$ looks like:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \cos\frac{\vartheta_1}{2} \\ i\sin\frac{\vartheta_1}{2} \end{pmatrix} = \begin{pmatrix} \cos\frac{\vartheta_1 - \vartheta}{2} \\ i\sin\frac{\vartheta_1 - \vartheta}{2} \end{pmatrix}$$

Using the angle sum identities:

$$\begin{aligned}\sin(x+y) &= \sin x \cos y + \cos x \sin y \\ \cos(x+y) &= \cos x \cos y - \sin x \sin y\end{aligned}$$

on the right hand side, and matrix multiplication on the left hand side gives:

$$\begin{pmatrix} a \cos \frac{\vartheta_1}{2} + ib \sin \frac{\vartheta_1}{2} \\ c \cos \frac{\vartheta_1}{2} + id \sin \frac{\vartheta_1}{2} \end{pmatrix} = \begin{pmatrix} \cos \frac{\vartheta_1}{2} \cos \frac{\vartheta}{2} + \sin \frac{\vartheta_1}{2} \sin \frac{\vartheta}{2} \\ i \sin \frac{\vartheta_1}{2} \cos \frac{\vartheta}{2} - i \cos \frac{\vartheta_1}{2} \sin \frac{\vartheta}{2} \end{pmatrix}$$

Comparing the two sides of this equation gives you the following solution:

$$\begin{aligned}a &= \cos \frac{\vartheta}{2} \\ b &= -\sin \frac{\vartheta}{2} \\ c &= -i \sin \frac{\vartheta}{2} \\ d &= \cos \frac{\vartheta}{2}\end{aligned}$$

So

$$R_X(\vartheta) = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} \cos\left(\frac{\vartheta}{2}\right) & -i \sin\left(\frac{\vartheta}{2}\right) \\ -i \sin\left(\frac{\vartheta}{2}\right) & \cos\left(\frac{\vartheta}{2}\right) \end{pmatrix}$$

Rotation around Z axis

Again, start with an arbitrary qubit, $|y\rangle = \cos \frac{\vartheta}{2} |0\rangle + e^{i\varphi} \sin \frac{\vartheta}{2} |1\rangle$. Taking $\vartheta = \frac{\pi}{2}$ projects $|y\rangle$ onto the XY plane (on the equator at $z=0$).

$$|\tilde{y}\rangle = \frac{1}{\sqrt{2}} |0\rangle + e^{i\varphi} \frac{1}{\sqrt{2}} |1\rangle$$

Rotating that point by ϑ radians counterclockwise gives:

$$|\tilde{y}'\rangle = \frac{1}{\sqrt{2}} |0\rangle + e^{i(\varphi+\vartheta)} \frac{1}{\sqrt{2}} |1\rangle$$

Expressed as a rotation matrix:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ e^{i\varphi} \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ e^{i(\varphi+\vartheta)} \frac{1}{\sqrt{2}} \end{pmatrix}$$

Simplifying:

$$\begin{aligned} \begin{pmatrix} a + be^{i\varphi} \\ c + de^{i\varphi} \end{pmatrix} &= \begin{pmatrix} 1 \\ e^{i\varphi} e^{i\vartheta} \end{pmatrix} \\ R_Z(\vartheta) = \begin{pmatrix} a & b \\ c & d \end{pmatrix} &= \begin{pmatrix} 1 & 0 \\ 0 & e^{i\vartheta} \end{pmatrix} = e^{i\vartheta/2} \begin{pmatrix} e^{-i\vartheta/2} & 0 \\ 0 & e^{i\vartheta/2} \end{pmatrix} \end{aligned}$$

Rotation around Y axis

This was left as an exercise and not covered in the lecture. But, taking a qubit on the XZ plane so $\varphi = 0$ gives:

$$|\tilde{y}\rangle = \cos \frac{\vartheta_1}{2} |0\rangle + \sin \frac{\vartheta_1}{2} |1\rangle$$

Rotating that point by ϑ radians counterclockwise gives:

$$|\tilde{y}'\rangle = \cos \frac{\vartheta_1 + \vartheta}{2} |0\rangle + \sin \frac{\vartheta_1 + \vartheta}{2} |1\rangle$$

Expressed using a rotation matrix this is:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \cos \frac{\vartheta_1}{2} \\ \sin \frac{\vartheta_1}{2} \end{pmatrix} = \begin{pmatrix} \cos \frac{\vartheta_1 + \vartheta}{2} \\ \sin \frac{\vartheta_1 + \vartheta}{2} \end{pmatrix}$$

Which becomes:

$$\begin{pmatrix} a \cos \frac{\vartheta_1}{2} + b \sin \frac{\vartheta_1}{2} \\ c \cos \frac{\vartheta_1}{2} + d \sin \frac{\vartheta_1}{2} \end{pmatrix} = \begin{pmatrix} \cos \frac{\vartheta_1}{2} \cos \frac{\vartheta}{2} - \sin \frac{\vartheta_1}{2} \sin \frac{\vartheta}{2} \\ \sin \frac{\vartheta_1}{2} \cos \frac{\vartheta}{2} + \cos \frac{\vartheta_1}{2} \sin \frac{\vartheta}{2} \end{pmatrix}$$

So

$$R_Y(\vartheta) = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} \cos \frac{\vartheta}{2} & -\sin \frac{\vartheta}{2} \\ \sin \frac{\vartheta}{2} & \cos \frac{\vartheta}{2} \end{pmatrix}$$

Rotations as Exponentials of Pauli Matrices

Theorem 1 (exercise 4.2): If $A^2 = I$ then $e^{ixA} = \cos xI - i \sin xA$

Proof (using Taylor series expansions from calculus):

$$\begin{aligned} e^{-ixA} &= \sum_{k=0}^{\infty} \frac{i^k x^k A^k}{k!} = \sum_{k \text{ odd}} \frac{i^k x^k A^k}{k!} + \sum_{k \text{ even}} \frac{i^k x^k A^k}{k!} \\ e^{-ixA} &= \sum_{k=0}^{\infty} \frac{i^{(2k+1)} x^{(2k+1)} A^{(2k+1)}}{(2k+1)!} + \sum_{k=0}^{\infty} \frac{i^{(2k)} x^{(2k)} A^{(2k)}}{(2k)!} \\ &= iA \sum_{k=0}^{\infty} \frac{(-1)^k x^{(2k+1)}}{(2k+1)!} + I \sum_{k=0}^{\infty} \frac{(-1)^k x^{(2k)}}{(2k)!} \\ &= iA \sin x + I \cos x \end{aligned}$$

Lecture 8 (12 February)

Z-Y Decomposition

Theorem 2: Any 1-qubit gate can be decomposed as

$$U = e^{i\alpha} R_Z(\beta) R_Y(\gamma) R_Z(\delta)$$

(This is theorem 4.1 in textbook)

Proof:

Start by expressing U as:

$$U = \begin{pmatrix} X_{11} e^{i\varphi_{11}} & X_{12} e^{i\varphi_{12}} \\ X_{21} e^{i\varphi_{21}} & X_{22} e^{i\varphi_{22}} \end{pmatrix}$$

Because U is orthonormal, vector norm of rows and columns is 1, so:

$$\begin{aligned} X_{11}^2 + X_{12}^2 &= 1 \\ X_{21}^2 + X_{22}^2 &= 1 \\ X_{11}^2 + X_{21}^2 &= 1 \\ X_{12}^2 + X_{22}^2 &= 1 \end{aligned}$$

Begin decomposing U by pulling out Z rotations (note that in general, post-multiplying with a diagonal matrix gives you multiples of the original columns, pre-multiplying with diagonal matrix gives you multiples of the original rows):

$$\begin{aligned} U &= \begin{pmatrix} X_{11} & X_{12} \\ X_{21}e^{i(\varphi_{21}-\varphi_{11})} & X_{22}e^{i(\varphi_{22}-\varphi_{12})} \end{pmatrix} \begin{pmatrix} e^{i\varphi_{11}} & 0 \\ 0 & e^{i\varphi_{12}} \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ 0 & e^{i(\varphi_{21}-\varphi_{11})} \end{pmatrix} \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22}e^{i(\varphi_{22}-\varphi_{12}-(\varphi_{21}-\varphi_{11}))} \end{pmatrix} \begin{pmatrix} e^{i\varphi_{11}} & 0 \\ 0 & e^{i\varphi_{12}} \end{pmatrix} \end{aligned}$$

The two outer matrices are Z rotations, and middle matrix can also be expressed as rotations, but different kinds of rotations, depending on the values inside.

$$\text{Case 1: } X_{11} = 0 \implies X_{12}^2 = 1 \implies X_{22} = 0 \implies X_{21}^2 = 1$$

Case 1.1: X_{12} and X_{21} have the same sign. In this case, the matrix is the product of Y and Z rotations:

$$\pm \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \pm \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \pm R_Y(\pi) Z$$

Case 1.2: X_{12} and X_{21} have different signs. In this case, the matrix is a Y rotation:

$$\pm \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} = \pm R_Y(\pi)$$

$$\text{Case 2: } X_{12} = 0 \implies X_{22}^2 = 1 \implies X_{21} = 0 \implies X_{11}^2 = 1$$

Case 2.1: X_{11} and X_{22} have the same sign. In this case the matrix is just identity:

$$\pm \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \pm I$$

Case 2.2: X_{11} and X_{22} have different signs. In this case the matrix is Z :

$$\pm \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \pm Z$$

Case 3: All $X_{ij} \neq 0$. Letting $\varphi = \varphi_{22} - \varphi_{12} - (\varphi_{21} - \varphi_{11})$, because the matrix is unitary we know:

$$\begin{aligned} \begin{pmatrix} X_{11} & X_{21} \end{pmatrix} \begin{pmatrix} X_{12} \\ X_{22}e^{i\varphi} \end{pmatrix} &= 0 \\ X_{11}X_{12} + X_{21}X_{22}e^{i\varphi} &= 0 \end{aligned}$$

In order for that to be true, the expression $e^{i\varphi} = \cos \varphi + i \sin \varphi$ cannot have an imaginary component, so $\sin \varphi = 0$ and $\varphi = k\pi$.

[DUNNO: The fact that the matrix is unitary and all entries are real is enough to make it rotation about Y ?]

Theorem 3: U can be decomposed as $U = e^{i\alpha}AXBXC$ where $ABC = I$. Book has recipe for ABC which serves as proof. (Corollary 4.2). In summary,

$$\begin{aligned} A &= R_z R_Y \\ B &= R_Y R_Z \\ C &= R_Z \end{aligned}$$

Controlled U, C(U), From 1-Qubit U

Use the Theorem 3 decomposition above. Start with just the phase shift, $e^{i\alpha}$. A controlled circuit for the phase shift $C(e^{i\alpha}I)$ should have the following behavior:

Input	Output
$ 0\rangle j\rangle$	$ 0\rangle j\rangle$
$ 1\rangle j\rangle$	$ 1\rangle e^{i\alpha} j\rangle$

or

Input	Output
$ 0\rangle 0\rangle$	$ 0\rangle 0\rangle$
$ 0\rangle 1\rangle$	$ 0\rangle 1\rangle$
$ 1\rangle 0\rangle$	$ 1\rangle e^{i\alpha} 0\rangle$
$ 1\rangle 1\rangle$	$ 1\rangle e^{i\alpha} 1\rangle$

The outputs from this controlled gate can be produced by a simple unitary transformation applied to the first qubit:

$$\begin{aligned} & \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix} \\ \text{Circuit: } & \left(I \otimes \begin{pmatrix} e^{i\alpha} & 0 \\ 0 & e^{i\alpha} \end{pmatrix}^{\#1} \right) = \left(\begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix} \otimes I \right) \end{aligned}$$

(My weird circuit notation would look a lot better in picture form, but it just shows the gates applied to each qubit (from top to bottom) written as tensor products. I means no gate operating on a qubit. Variables or matrices stand for real gates. Controlled gates are written with numeric superscripts indicating which qubit they are controlled by. For example, $X^{\#1}$ is C_{NOT} controlled by first qubit.)

Using the circuit above, you can write the circuit for a general $C(U)$ where $U = e^{i\alpha}AXBXC$ and $ABC = I$:

$$\text{Circuit: } \left(\begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix} \otimes A \right) \left(I \otimes X^{\#1} \right) \left(I \otimes B \right) \left(I \otimes X^{\#1} \right) \left(I \otimes C \right)$$

Input	Output
$ 0\rangle j\rangle$	$ 0\rangle ABC j\rangle = 0\rangle I j\rangle$
$ 1\rangle j\rangle$	$e^{i\alpha} 1\rangle AXBXC j\rangle = 1\rangle U j\rangle$

The circuit works because when $|i\rangle$ is 0, the C_{NOT} gates have no effect and the output for the second qubit is ABC , exactly the same as the input because $ABC = I$

Controlled U, $C^n(U)$, with multiple control inputs

Using n control qubits to control unitary operation of k qubits:

$$\begin{aligned} C^n(U) |x\rangle |y\rangle &= C^n(U) |x_0 \dots x_{n-1}\rangle |y_0 \dots y_{k-1}\rangle \\ &= |x_0 \dots x_{n-1}\rangle U^{x_0 \cdot x_1 \dots x_{n-1}} |y_0 \dots y_{k-1}\rangle \\ &= |x\rangle U^{x_0 \cdot x_1 \dots x_{n-1}} |y\rangle \end{aligned}$$

U is controlled by product of bits of $|x\rangle$, it is only applied when every bit is 1.

In matrix form,

$$C^n(U) = \begin{pmatrix} I & 0 \\ 0 & U \end{pmatrix}$$

Size of $C^n(U)$ is 2^{n+k} , size of I is $(2^n - 1)(2^k)$, size of U is 2^k . Each column of the matrix is the output of a basis state. The columns from $|0 \dots 0\rangle |0 \dots 0\rangle$ to $|1 \dots 10\rangle |1 \dots 1\rangle$ for the first part of the matrix (the bulk of it) just give identify. Then, the last columns from $|1 \dots 1\rangle |0 \dots 0\rangle$ to $|1 \dots 1\rangle |1 \dots 1\rangle$ apply U to the last k qubits of the input.

Toffoli Gate

An example of $C^n(U)$, with $n = 2$ and $U = X$.

$$X^{x \cdot y} |x\rangle |y\rangle |z\rangle = |x\rangle |y\rangle |(xy) \oplus z\rangle$$

NAND gate can be implemented in terms of Toffoli:

$$X^{x \cdot y} |x\rangle |y\rangle |1\rangle = |x\rangle |y\rangle |(xy) \oplus 1\rangle = |x\rangle |y\rangle |\overline{xy}\rangle$$

(\overline{xy} is logical inverse of xy)

Fanout can be implemented with:

$$X^{1 \cdot x} |1\rangle |x\rangle |0\rangle = |1\rangle |x\rangle |(1 \cdot x) \oplus 0\rangle = |1\rangle |x\rangle |x\rangle$$

Lecture 9 (14 February)

Implementing Controlled 1-Qubit Gate $C^2(U)$ with $V^2 = U$

$$\text{Circuit: } (I \otimes I \otimes U^{\#1, \#2}) = (I \otimes I \otimes V^{\#1}) (I \otimes X^{\#1} \otimes I) (I \otimes I \otimes V^{H\#2}) (I \otimes X^{\#1} \otimes I) (I \otimes I \otimes V^{\#2})$$

(Figure 4.8 in book)

Evaluated:

$ x_1\rangle$	$ x_2\rangle$	$ x_3\rangle$
$ x_1\rangle$	$ x_2\rangle$	$V^{x_2} x_3\rangle$
$ x_1\rangle$	$ x_1 \oplus x_2\rangle$	$V^{x_2} x_3\rangle$
$ x_1\rangle$	$ x_1 \oplus x_2\rangle$	$V^{Hx_1 \oplus x_2} V^{x_2} x_3\rangle$
$ x_1\rangle$	$ x_2\rangle$	$V^{Hx_1 \oplus x_2} V^{x_2} x_3\rangle$
$ x_1\rangle$	$ x_2\rangle$	$V^{x_1} V^{Hx_1 \oplus x_2} V^{x_2} x_3\rangle$

Evaluate third qubit output case by case

Case 1: If $X_1 \oplus X_2 = 1$ then $V^{x_1} V^H V^{x_2} |x_3\rangle$

Case 1.1: If $X_1 = 0$ then $X_2 = 1$ and $V^H V |x_3\rangle = |x_3\rangle$

Case 1.2: If $X_2 = 0$ then $X_1 = 1$ and $V V^H |x_3\rangle = |x_3\rangle$

Case 2: If $X_1 \oplus X_2 = 0$ then $V^{x_1} V^{x_2} |x_3\rangle$

Case 2.1: If $X_1 = X_2 = 0$ then $|x_3\rangle$

Case 2.2: If $X_1 = X_2 = 1$ then $V V |x_3\rangle = U |x_3\rangle$

Toffoli Gate

Using $V = \frac{1-i}{2} (I + iX)$ above gives the toffoli gate. Verify:

$$\begin{aligned} V^2 &= \frac{(1-2i-1)^2}{4} (I^2 + 2iX - X^2) \\ &= \frac{-2i}{4} 2iX = X \end{aligned}$$

V can also be rewritten:

$$\begin{aligned} V &= \frac{1-i}{2} (I + iX) \\ &= \frac{1-i}{\sqrt{2}} \left(\frac{\sqrt{2}}{2} I + \frac{\sqrt{2}}{2} iX \right) \\ &= \frac{1-i}{\sqrt{2}} e^{i\frac{\pi}{4}X} = \frac{1-i}{\sqrt{2}} R_X \left(-\frac{\pi}{2} \right) \end{aligned}$$

No Cloning Theorem

(Box 12.1, Page 532 in book)

Is there an operator U that satisfies $U(|\psi\rangle|s\rangle) = |\psi\rangle|\psi\rangle$ for arbitrary states $|\psi\rangle$ with some standard input $|s\rangle$?

Assuming there is such an operator, the following will be true for two states $|\psi_1\rangle$ and $|\psi_2\rangle$

$$\begin{aligned} U(|\psi_1\rangle|s\rangle) &= |\psi_1\rangle|\psi_1\rangle \\ U(|\psi_2\rangle|s\rangle) &= |\psi_2\rangle|\psi_2\rangle \end{aligned}$$

Inner product of above equations, left hand side:

$$\begin{aligned} & (U(|\psi_1\rangle|s\rangle))^\dagger (U(|\psi_2\rangle|s\rangle)) \\ &= (|\psi_1\rangle|s\rangle)^\dagger U^\dagger U (|\psi_2\rangle|s\rangle) \\ &= (\langle\psi_1| \langle s|) (|\psi_2\rangle|s\rangle) \\ &= \langle\psi_1|\psi_2\rangle \langle s|s\rangle \\ &= \langle\psi_1|\psi_2\rangle \end{aligned}$$

Right hand side:

$$\begin{aligned} & (|\psi_1\rangle|\psi_1\rangle)^\dagger (|\psi_2\rangle|\psi_2\rangle) \\ &= (\langle\psi_1|\langle\psi_1|) (|\psi_2\rangle|\psi_2\rangle) \\ &= \langle\psi_1|\psi_2\rangle \langle\psi_1|\psi_2\rangle \\ &= \langle\psi_1|\psi_2\rangle^2 \end{aligned}$$

Both sides together:

$$\langle\psi_1|\psi_2\rangle = \langle\psi_1|\psi_2\rangle^2$$

which can only be true in two cases

Case 1: $\langle\psi_1|\psi_2\rangle = 0$ means $|\psi_1\rangle \perp |\psi_2\rangle$

Case 2: $\langle\psi_1|\psi_2\rangle = 1$ means $|\psi_1\rangle = |\psi_2\rangle$

So a cloning operator U will can work for a single state, or for two states that are orthogonal, there is no U that can clone states generally.

Implementing Controlled n-Qubit Gates $C^n(U)$ ($n > 2$)

Start off with simple examples and build in complexity

U=X, k=1, n=3

This just means taking a normal Toffoli gate

$$\text{Circuit: } \left(I \otimes I \otimes X^{\#1,\#2} \right) |x_1\rangle |x_2\rangle |x_3\rangle$$

and extending it to get an additional control line:

$$\text{Circuit: } \left(I \otimes I \otimes I \otimes I \otimes X^{\#3,\#4} \right) \left(I \otimes I \otimes X^{\#1,\#2} \otimes I \otimes I \right) |x_1\rangle |x_2\rangle |0\rangle |x_3\rangle |x_4\rangle$$

Output of circuit will be $|x_1\rangle |x_2\rangle |x_1x_2\rangle |x_3\rangle |(x_1x_2x_3) \oplus x_4\rangle$, and the last qubit has the output we are looking for.

U=any, k=1, n=any

In the general case, if you want to control a 1 qubit U with n inputs, you need to have $n - 1$ $|0\rangle$ inputs as well. Add $n - 1$ toffoli gates, taking the product of the first two qubit lines to the first $|0\rangle$ input, the product of the second two lines ($\#3$ and $\#4$) onto the second $|0\rangle$ input, and so on. Halfway down, you reach the first $|0\rangle$ lines, but you keep taking products in the same pattern, and at the end, the last $|0\rangle$ line will have the product of the first n qubits. Below that, the unitary operation U can be placed it's own line and can it be controlled by the last $|0\rangle$ line, right above it, which holds the product of all the control inputs. An additional $n - 1$ toffoli gates can be placed after the unitary operation, in the same pattern as before, to make the $|0\rangle$ lines have $|0\rangle$ outputs.

U= $U^{\otimes k}$, k=any, n=1

$$Q_U |c\rangle |t_1 t_2 \cdots t_k\rangle = |c\rangle U^C |t_1\rangle U^C |t_2\rangle \cdots U^C |t_k\rangle$$

A special case is $U = X$:

$$Q_X |c\rangle |t_1 t_2 \cdots t_k\rangle = |c\rangle |c \oplus t_1\rangle |c \oplus t_2\rangle \cdots |c \oplus t_k\rangle$$

which can be drawn with a single control node, k X nodes (\oplus) for each affected qubit, and a line connecting all the nodes.

In matrix form,

$$C(X^{\otimes k}) = \begin{pmatrix} I & 0 \\ 0 & X^{\otimes k} \end{pmatrix}$$

Size of $C(X^{\otimes k})$ is 2^{k+1} , size of I , $X^{\otimes k}$ and the 0 matrices is 2^k . Each column of the matrix is the output of a basis state. The columns from $|0\rangle |0 \cdots 0\rangle$ to $|0\rangle |1 \cdots 1\rangle$ for the first half of the matrix just give identify. Then, the last columns from $|1\rangle |0 \cdots 0\rangle$ to $|1\rangle |1 \cdots 1\rangle$ apply $X^{\otimes k}$ to the last k qubits of the input. $X^{\otimes k}$ looks like a reflected identify matrix, with 1s going from the bottom left corner to the top right, and 0s everywhere else.

Above can be generalized,

$$\begin{aligned} Q_U |c\rangle |t_1 t_2 \cdots t_k\rangle &= |c\rangle U^C |t_1\rangle U^C |t_2\rangle \cdots U^C |t_k\rangle \\ &= |c\rangle (U^C)^{\otimes k} |t_1 \cdots t_k\rangle \end{aligned}$$

And

$$Q_U = \begin{pmatrix} I & 0 \\ 0 & U^{\otimes k} \end{pmatrix}$$

2 Level Matrix Gate

Acts non-linearly on at most 2 components of a vector, example:

$$(U)_{d \times d} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_d \end{pmatrix} = \begin{pmatrix} x'_1 \\ x'_2 \\ x_3 \\ \vdots \\ x_d \end{pmatrix} \text{ or } = \begin{pmatrix} x'_1 \\ x_2 \\ x'_3 \\ \vdots \\ x_d \end{pmatrix} \text{ or } = \begin{pmatrix} x_1 \\ x'_2 \\ x'_3 \\ \vdots \\ x_d \end{pmatrix}$$

Mentioned: QR Decomposition, Hausholder matrix

Lecture 10 (19 February)

[DUNNO: Was late to class, no idea what this is]

Implementation Topic

$C^2(U)$, $C^n(U)$

2-Level Gates / Matrices

$$\begin{pmatrix} \frac{\alpha_1}{x} & \frac{\alpha_2}{x} & 0 \\ \frac{\alpha_2}{x} & \frac{\alpha_1}{x} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha_1 & \beta_1 & \gamma_1 \\ \alpha_2 & \beta_2 & \gamma_2 \\ \alpha_3 & \beta_3 & \gamma_3 \end{pmatrix} = \begin{pmatrix} \alpha_1 & \beta_1 & \gamma_1 \\ 0 & \beta_2 & \gamma_2 \\ \alpha_3 & \beta_3 & \gamma_3 \end{pmatrix}$$

$$x = \sqrt{|\alpha_1|^2 + |\alpha_2|^2}$$

$$\begin{pmatrix} \frac{\alpha_1}{x} & \frac{-\alpha_2}{x} \\ \frac{-\alpha_2}{x} & \frac{\alpha_1}{x} \end{pmatrix} \begin{pmatrix} \frac{-\alpha_2}{x} \\ \frac{\alpha_1}{x} \end{pmatrix} = 0$$

First matrix is U_1 , second matrix is U .

$$\begin{pmatrix} \bar{\alpha}'_1 & 0 & \bar{\gamma}_1 \\ 0 & 1 & 0 \\ -\alpha_3 & 0 & \gamma'_3 \end{pmatrix} \begin{pmatrix} \alpha''_1 & \beta''_1 & \gamma''_1 \\ 0 & \beta''_2 & \gamma''_2 \\ 0 & \beta''_3 & \gamma''_3 \end{pmatrix} = \begin{pmatrix} \alpha''_1 & 0 & 0 \\ 0 & \beta''_2 & \gamma''_2 \\ 0 & \beta''_3 & \gamma''_3 \end{pmatrix}$$

Repeat as submatrix $|\alpha''_1| = 1$

$$U_3 U_2 U_1 U = \begin{pmatrix} \alpha''_1 & 0 & 0 \\ 0 & \beta''_2 & 0 \\ 0 & 0 & \gamma''_3 \end{pmatrix} D$$

D is some diagonal matrix holding relative phases.

$$U = U_1^{-1} U_2^{-1} U_3^{-1} D = U_1^H U_2^H U_3^H D$$

Measurements

(Book 2.2.3) Measurements are made with collections of measurement operators, $\{M_j\}$, where operators are Hermitian matrices, not necessarily unitary. There is one measurement operator M_j for each possible outcome, j . The measurements satisfy the completeness equation:

$$\sum_{j=0}^{k-1} M_j^H M_j = I$$

Given a state $|\psi\rangle$, an outcome j occurs with probability

$$p(j) = \langle \psi | M_j^H M_j | \psi \rangle = \|M_j |\psi\rangle\|^2$$

causing the state to collapse to $\frac{M_j |\psi\rangle}{\sqrt{p(j)}}$.

The probabilities of all possible outcomes j sum to one. Proof

$$\begin{aligned} 1 &= \langle \psi | \psi \rangle = \langle \psi | I | \psi \rangle = \langle \psi | \sum_j M_j^H M_j | \psi \rangle \\ &= \sum_j \langle \psi | M_j^H M_j | \psi \rangle = \sum_j p(j) \end{aligned}$$

Projective Measurements

Projective measurements are measurements on the computational basis.

Example 1:

$$M_j = |j\rangle \langle j| = \begin{pmatrix} & & 0 & & \\ & & \vdots & & \\ 0 & \dots & 1 & \dots & 0 \\ & & \vdots & & \\ & & 0 & & \end{pmatrix}$$

M_j is zero matrix with single one row and column $j + 1$.

Observe $M_j^H = M_j$ (matrix is Hermitian) and $M_j^H M_j = M_j$ (because it's a projection matrix and $|j\rangle \langle j| j\rangle \langle j| = |j\rangle \langle j|$).

$$p(j) = \|M_j |\psi\rangle\|^2 = \||j\rangle \langle j | \psi\rangle\|^2 = |\langle j | \psi\rangle|^2$$

$$M_j |\psi\rangle = |j\rangle \langle j | \psi\rangle = \langle j | \psi\rangle |j\rangle$$

$$|\psi\rangle = \sum_j |j\rangle \langle j | \psi\rangle = \sum_j c_j |j\rangle$$

$$\sum_{j=0}^{2^n-1} M_j^H M_j = \sum_j M_j = \sum_j |j\rangle \langle j| = I$$

Example 2:

$$\begin{aligned}
|\psi\rangle &= a|0\rangle + b|1\rangle \\
M_1 &= |+\rangle\langle +| \\
M_2 &= |-\rangle\langle -| \\
|+\rangle &= \frac{|0\rangle + |1\rangle}{\sqrt{2}} \\
|-\rangle &= \frac{|0\rangle - |1\rangle}{\sqrt{2}} \\
M_j^H &= M_j \\
M_j^H M_j &= M_j
\end{aligned}$$

Completeness relation

$$|+\rangle\langle +| + |-\rangle\langle -| = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$M_1 |\psi\rangle = |+\rangle\langle +|\psi\rangle = \dots = \left(\frac{a}{\sqrt{2}} + \frac{b}{\sqrt{2}} \right) |+\rangle$$

$$M_2 |\psi\rangle = |-\rangle\langle -|\psi\rangle = \frac{1}{\sqrt{2}} (a - b) |-\rangle$$

$$p(1) = \|M_1 |\psi\rangle\|^2 = \frac{|a+b|^2}{2}$$

$$p(2) = \|M_2 |\psi\rangle\|^2 = \frac{|a-b|^2}{2}$$

Example 3

Measuring some qubits and not others. Say measuring k qubits in a system of $k+n$ qubits. Then $M_j = |j\rangle\langle j| \otimes I$ where $|j\rangle\langle j|$ is a size 2^k matrix and I is size 2^n .

Properties

$$M_j^H = M_j$$

Means matrix is Hermetian and therefore that it has real eigenvalues. In other words it's observable because eigenvalues are what you observe.

$$M_j^H M_j = (|j\rangle\langle j| \otimes I) (|j\rangle\langle j| \otimes I) = |j\rangle\langle j| |j\rangle\langle j| \otimes II = |j\rangle\langle j| \otimes I = M_j$$

$$\begin{aligned}
M_j |\psi\rangle &= M_j |\psi_1\rangle |\psi_2\rangle \\
&= (|j\rangle\langle j| \otimes I) |\psi_1\rangle |\psi_2\rangle \\
&= |j\rangle\langle j| \psi_1 \otimes I |\psi_2\rangle \\
&= \langle j | \psi_1 \rangle (|j\rangle |\psi_2\rangle)
\end{aligned}$$

$$\begin{aligned}
p(j) &= \|M_j |\psi\rangle\|^2 \\
&= \|\langle j | \psi_1 \rangle (|j\rangle |\psi_2\rangle)\|^2 \\
&= |\langle j | \psi_1 \rangle|^2 \| |j\rangle |\psi_2\rangle \|^2 \\
&= |\langle j | \psi_1 \rangle|^2
\end{aligned}$$

Reason for last step is that $|j\rangle$ and $|\psi_2\rangle$ are both normal:

$$\begin{aligned} \||j\rangle|\psi_2\rangle\|^2 &= (|j\rangle|\psi_2\rangle)^H (|j\rangle|\psi_2\rangle) \\ &= (\langle j|\langle\psi_2|)(|j\rangle|\psi_2\rangle) \\ &= \langle j|j\rangle\langle\psi_2|\psi_2\rangle \\ &= 1 \cdot 1 = 1 \end{aligned}$$

Lecture 11 (21 February)

Distinguishing states with certainty

Can we distinguish between two states $|\psi_1\rangle$ and $|\psi_2\rangle$ with certainty (with probability 1)? (Similar proof page 87, box 2.3)

Assume it is possible, then there are measurement operators M_1, M_2, \dots, M_k so that

$$I = \sum_j M_j^H M_j$$

Given states $|\psi_1\rangle$ and $|\psi_2\rangle$, then probabilities of outcomes 1 and 2 (respectively) should be:

$$\begin{aligned} p_{\psi_1}(1) &= \langle\psi_1|M_1^H M_1|\psi_1\rangle = 1 \\ p_{\psi_2}(2) &= \langle\psi_2|M_2^H M_2|\psi_2\rangle = 1 \end{aligned}$$

This implies that $M_1|\psi_1\rangle$ and $M_1|\psi_2\rangle$ are unit vectors. Using the completeness equation, it also implies $M_1|\psi_2\rangle = 0$ and $M_2|\psi_1\rangle = 0$.

(Completeness equation

$$M_1^H M_1 + M_2^H M_2 = I$$

leads to

$$\begin{aligned} 1 &= \langle\psi_1|I|\psi_1\rangle = \langle\psi_1|M_1^H M_1|\psi_1\rangle + \langle\psi_1|M_2^H M_2|\psi_1\rangle \\ 1 &= \langle\psi_2|I|\psi_2\rangle = \langle\psi_2|M_1^H M_1|\psi_2\rangle + \langle\psi_2|M_2^H M_2|\psi_2\rangle \end{aligned}$$

and because $\langle\psi_1|M_2^H M_2|\psi_1\rangle = 1$ and $\langle\psi_2|M_2^H M_2|\psi_2\rangle = 1$ the other terms must be 0.)

State $|\psi_1\rangle$ can be written in using $|\psi_2\rangle$ and another vector $|Z\rangle$ as a basis, where $\|Z\| = 1$, $|Z\rangle \perp |\psi_2\rangle$:

$$\begin{aligned} |\psi_1\rangle &= \alpha|\psi_2\rangle + \beta|Z\rangle \\ M_1|\psi_1\rangle &= \alpha M_1|\psi_2\rangle + \beta M_1|Z\rangle \end{aligned}$$

Assume $|\psi_1\rangle$ and $|\psi_2\rangle$ are not orthogonal, then $\langle\psi_1|\psi_2\rangle \neq 0$ and $\alpha \neq 0$, so $|\beta| < 1$. Then below, you have a contradiction, proving they must be orthogonal.

$$1 = \|M_1|\psi_1\rangle\|^2 = |\beta|^2 \|M_1|Z\rangle\|^2 \leq |\beta|^2 < 1$$

Note that $\|M_1|Z\rangle\|^2$ just means $\langle Z|M_1^H M_1|Z\rangle$.

Some Properties of Operators and Completeness

Given:

$$\begin{aligned} & |\psi_1\rangle \perp |\psi_2\rangle \\ M_1 &= |\psi_1\rangle \langle \psi_1| \\ M_2 &= |\psi_2\rangle \langle \psi_2| \end{aligned}$$

M_1 and M_2 are symmetric non-negative definite, which means that for all $|x\rangle$, $\langle x|M|x\rangle \geq 0$. This can be verified as follows:

$$\langle x|M|x\rangle = \langle x|M^H M|x\rangle = \|M|x\rangle\|^2 \geq 0$$

Symmetric non-negative definite matrices have non-negative eigenvalues.

Define:

$$M = I - M_1 - M_2$$

Observe it's symmetric. This means eigenvalues are real. Check that it is positive semi-definite, that for all $|\psi\rangle$,

$$\begin{aligned} \langle \psi|M|\psi\rangle &= \langle \psi|\psi\rangle - \langle \psi|M_1|\psi\rangle - \langle \psi|M_2|\psi\rangle \geq 0 \\ M_1|\psi\rangle &= |\psi_1\rangle \langle \psi_1|\psi\rangle \\ M_2|\psi\rangle &= |\psi_2\rangle \langle \psi_2|\psi\rangle \\ \langle \psi|M_1|\psi\rangle &= |\langle \psi_1|\psi\rangle|^2 \\ \langle \psi|M_2|\psi\rangle &= |\langle \psi_2|\psi\rangle|^2 \\ \langle \psi|M|\psi\rangle &= 1 - |\langle \psi_1|\psi\rangle|^2 - |\langle \psi_2|\psi\rangle|^2 \stackrel{?}{=} 0 \end{aligned}$$

Need to find prove that above is ≥ 0 . $|\psi\rangle$ can be expressed as

$$|\psi\rangle = |\psi_1\rangle \langle \psi_1|\psi\rangle + |\psi_2\rangle \langle \psi_2|\psi\rangle + |z\rangle$$

For some $|z\rangle \perp |\psi_1\rangle, |\psi_2\rangle$. Taking inner product with $|\psi\rangle$ gives:

$$1 = \langle \psi|\psi\rangle = |\langle \psi_1|\psi\rangle|^2 + |\langle \psi_2|\psi\rangle|^2 + \| |z\rangle \|^2 \geq 0$$

Rearranging,

$$1 - |\langle \psi_1|\psi\rangle|^2 - |\langle \psi_2|\psi\rangle|^2 = \| |z\rangle \|^2 = \langle \psi|M|\psi\rangle \geq 0$$

Which tells us M is symmetric non-negative definite. This means we can do spectral decomposition:

$$\begin{aligned} M &= V \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \lambda_4 \end{pmatrix} V^H \\ &= V \begin{pmatrix} \sqrt{\lambda_1} & 0 & 0 & 0 \\ 0 & \sqrt{\lambda_2} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \sqrt{\lambda_4} \end{pmatrix} V^H V \begin{pmatrix} \sqrt{\lambda_1} & 0 & 0 & 0 \\ 0 & \sqrt{\lambda_2} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \sqrt{\lambda_4} \end{pmatrix} V^H \end{aligned}$$

since $V^H V = I$. Now, define $M_3 = \sqrt{M}$

Results of measuring state $|\psi_1\rangle$:

Outcome	Probability
1	$p_{\psi_1}(1) = 1 = \langle \psi_1 M_1 \psi_1 \rangle$
2	$p_{\psi_1}(2) = 0 = \langle \psi_1 M_2 \psi_1 \rangle$
3	$p_{\psi_1}(3) = 0$ (see below)

$$\begin{aligned}
p_{\psi_1}(3) = 0 &= \langle \psi_1 | M_3^H M_3 | \psi_1 \rangle \\
&= \langle \psi_1 | (I - M_1 - M_2) | \psi_1 \rangle \\
&= \langle \psi_1 | \psi_1 \rangle - \langle \psi_1 | M_1 | \psi_1 \rangle - \langle \psi_1 | M_2 | \psi_1 \rangle \\
&= 1 - 1 - 0
\end{aligned}$$

For $|\psi_2\rangle$

Outcome	Probability
1	$p_{\psi_2}(1) = 0$
2	$p_{\psi_2}(2) = 1$
3	$p_{\psi_2}(3) = 0$

(Review of matrix types:

Normal - $A^H A = A A^H$, matrix commutes with its transpose

Unitary - $A^H A = I$ or $A^H = A^{-1}$, type of normal matrix, eigenvalues all have absolute value of 1.

Hermitian - $A = A^H$, type of normal matrix, eigenvalues are real

Positive Definite - hermitian and $x^H A x > 0$ for all x . (if not hermitian, some values of $x^H A x$ would be complex)

Projection matrix - hermitian and $A^2 = A$, is positive semi-definite, eigenvalues are all 0 or all 1)

EPR States / Bell States

(page 25)

Circuit: $(I \otimes X^{\#1}) (H \otimes I)$

$$|0\rangle |0\rangle \xrightarrow{(H \otimes I)} \frac{|00\rangle + |10\rangle}{\sqrt{2}} \xrightarrow{(I \otimes X^{\#1})} \frac{|00\rangle + |11\rangle}{\sqrt{2}} = |\beta_{00}\rangle$$

$$|i\rangle |j\rangle \xrightarrow{(H \otimes I)} \frac{|0\rangle + (-1)^i |1\rangle}{\sqrt{2}} |j\rangle \xrightarrow{(I \otimes X^{\#1})} \frac{|0j\rangle + (-1)^i |1\bar{j}\rangle}{\sqrt{2}} = |\beta_{ij}\rangle$$

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

$$|\beta_{01}\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}}$$

$$|\beta_{10}\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}}$$

$$|\beta_{11}\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}$$

Property 1: if you measure first qubit, you know second qubit.

Property 2: Bell states are entangled, and therefore can't be written as the product of two entangled qubits.

$$(a_1 |0\rangle + b_1 |1\rangle)(a_2 |0\rangle + b_2 |1\rangle)$$

$$a_1 a_2 |00\rangle + a_1 b_2 |01\rangle + b_1 a_2 |10\rangle + b_1 b_2 |11\rangle$$

There are no values of a_1, b_1, a_2, b_2 that can give one of the bell states.

Property 3: Bell states are pairwise orthogonal. Proof:

$$\begin{aligned} \langle \beta_{i_1 j_1} | \beta_{i_2 j_1} \rangle &= \left(\frac{\langle 0j_1 | + (-1)^{i_1} \langle 1\bar{j}_1 |}{\sqrt{2}} \right) \left(\frac{|0j_2\rangle + (-1)^{i_2} |1\bar{j}_2\rangle}{\sqrt{2}} \right) \\ &= \frac{1}{2} \left(\langle 0j_1 | 0j_2 \rangle + 0 + 0 + (-1)^{i_1+i_2} \langle 1\bar{j}_1 | 1\bar{j}_2 \rangle \right) \\ &= \frac{1}{2} \left(\langle j_1 | j_2 \rangle + (-1)^{i_1+i_2} \langle \bar{j}_1 | \bar{j}_2 \rangle \right) \end{aligned}$$

Case 1: $j_1 \neq j_2$ then $\langle \beta_{i_1 j_1} | \beta_{i_2 j_1} \rangle = 0$

Case 2: $j_1 = j_2$

Case 2.1: $i_1 \neq i_2$ then $\langle \beta_{i_1 j_1} | \beta_{i_2 j_1} \rangle = 0$ (because exponent $i_1 + i_2$ is odd)

Case 2.2: $i_1 = i_2$ then $\langle \beta_{i_1 j_1} | \beta_{i_2 j_1} \rangle = 1$

So inner product is zero unless $i_1 = i_2$ and $j_1 = j_2$.

$$\langle \beta_{i_1 j_1} | \beta_{i_2 j_1} \rangle = \left(\frac{\langle 0j_2 | + (-1)^{i_2} \langle 1\bar{j}_2 |}{\sqrt{2}} \right) \left(\frac{|0j_2\rangle + (-1)^{i_2} |1\bar{j}_2\rangle}{\sqrt{2}} \right)$$

Quantum Teleportation

[Book 1.3.7 p27]

Circuit: $(I \otimes I \otimes Z^{M\#1} X^{M\#2}) (H \otimes I \otimes I) (I \otimes X^{\#1} \otimes I) |\psi\rangle |\beta_{00}\rangle$

$$\begin{aligned} |\psi\rangle |\beta_{00}\rangle &= |\psi\rangle \frac{|00\rangle + |11\rangle}{\sqrt{2}} \\ \xrightarrow{(I \otimes X^{\#1} \otimes I)} &a |0\rangle \frac{|00\rangle + |11\rangle}{\sqrt{2}} + b |1\rangle \frac{|10\rangle + |01\rangle}{\sqrt{2}} \\ \xrightarrow{(H \otimes I \otimes I)} &\frac{a}{2} (|0\rangle + |1\rangle) (|00\rangle + |11\rangle) + \frac{b}{2} (|0\rangle - |1\rangle) (|10\rangle + |01\rangle) \\ &= \frac{1}{2} |00\rangle (a |0\rangle + b |1\rangle) + \frac{1}{2} |01\rangle (a |1\rangle + b |0\rangle) \\ &\quad + \frac{1}{2} |10\rangle (a |0\rangle - b |1\rangle) + \frac{1}{2} |11\rangle (a |1\rangle - b |0\rangle) \\ &= \frac{1}{2} (|00\rangle |\psi\rangle + |01\rangle X |\psi\rangle + |10\rangle Z |\psi\rangle + |10\rangle XZ |\psi\rangle) \end{aligned}$$

By measuring first two qubits, you can know what X and Z filters to apply the third qubit in to make it equivalent original input value $|\psi\rangle$. This means that if you have two entangled qubits (of the bell state β_{00}) in separate locations, you can use them to transmit an arbitrary qubit over a classical channel.

Lecture 12 (26 February)

Lecture 11 Review

EPR States: $\beta_{ij} = |0j\rangle + -(-1)^i 1|1\bar{j}\rangle$

Superdense Coding

[Book 2.3, p97]

Just like in teleportation, Alice and Bob each have 1 qubit of a 2-qubit entangled state, β_{00} . Alice wants to send 2 classical bits of information by sending Bob a single quantum bit. She can do this by sending Bob her half of the entangled qubit after she applies one of four operations to it, depending on the classical bits she wants to send. The operations are shown below, and result in Bell states which Bob can distinguish to determine the original classical bits.

Bits	Qubit
00	$ \beta_{00}\rangle = \frac{ 00\rangle+ 11\rangle}{\sqrt{2}} \xrightarrow{I \otimes I} \frac{ 00\rangle+ 11\rangle}{\sqrt{2}} = \beta_{00}\rangle$
01	$ \beta_{00}\rangle = \frac{ 00\rangle+ 11\rangle}{\sqrt{2}} \xrightarrow{Z \otimes I} \frac{ 00\rangle- 11\rangle}{\sqrt{2}} = \beta_{10}\rangle$
10	$ \beta_{00}\rangle = \frac{ 00\rangle+ 11\rangle}{\sqrt{2}} \xrightarrow{X \otimes I} \frac{ 10\rangle+ 01\rangle}{\sqrt{2}} = \beta_{01}\rangle$
11	$ \beta_{00}\rangle = \frac{ 00\rangle+ 11\rangle}{\sqrt{2}} \xrightarrow{iY \otimes I} \frac{ 01\rangle- 10\rangle}{\sqrt{2}} = \beta_{11}\rangle$

Quantum Queries

Mechanism to insert data into a quantum computer.

Assume you have a boolean function $f = \{0, 1\} \rightarrow \{0, 1\}$, then this is a corresponding unitary operation U_f which is defined on the basis states as: $U_f |i\rangle |j\rangle = |i\rangle |j \oplus f(i)\rangle$.

Proving U_f is unitary:

$$|i\rangle |j\rangle \xrightarrow{U_f} |i\rangle |j \oplus f(i)\rangle \xrightarrow{U_f} |i\rangle |j \oplus f(i) \oplus f(i)\rangle = |i\rangle |j\rangle$$

$(U_f)^2 = I$ therefore $U_f = U_f^{-1}$ and U_f is unitary.

Quantum Parallelism

Inputting a superposition state to U_f computes a result of multiple inputs to the function f in just a single operation.

Circuit: $U_f (H \otimes I) |0\rangle |0\rangle$

$$\begin{aligned} |00\rangle &\xrightarrow{H \otimes I} \frac{|0\rangle + |1\rangle}{\sqrt{2}} |0\rangle \\ &\xrightarrow{U_f} \frac{|0\rangle}{\sqrt{2}} |0 \oplus f(0)\rangle + \frac{|1\rangle}{\sqrt{2}} |0 \oplus f(1)\rangle \\ &= \frac{|0f(0)\rangle + |1f(1)\rangle}{\sqrt{2}} \end{aligned}$$

Single output state contains both values of function f .

General Quantum Queries

Assume you have a boolean function $f = \{0, \dots, 2^n - 1\} \rightarrow \{0, 1\}$, then there is a corresponding unitary operation U_f which is defined on the basis states as: $U_f |i\rangle |j\rangle = |i\rangle |j \oplus f(i)\rangle$.

$$\text{Circuit: } U_f (H^{\otimes n} \otimes I) |0\rangle^{\otimes n} |0\rangle$$

$$\begin{aligned} |0 \dots 0\rangle |0\rangle &\xrightarrow{H^{\otimes n} \otimes I} H^{\otimes n} |0 \dots 0\rangle |0\rangle \\ &= (H|0\rangle \otimes \dots \otimes H|0\rangle) |0\rangle \\ &= \frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} |j\rangle |0\rangle \\ &\xrightarrow{U_f} \frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} |j\rangle |f(j)\rangle \end{aligned}$$

Quantum parallelism is not enough to exploit power of quantum computing, because even though the final state is a combination of all possible outputs of the function f , when measurement occurs it will only give a single, random output. Need to find ways to combine the values to be able to efficiently measure some global property of the function.

Matrix Representation of U_f

When f is boolean function of 1 bit:

$$U_f = \begin{pmatrix} X^{f(0)} & \\ & X^{f(1)} \end{pmatrix}$$

$$U_f |i\rangle |j\rangle = |i\rangle |j \oplus f(i)\rangle$$

$$\text{Case } i = 0, f(0) = 0, |0j\rangle \rightarrow |0j\rangle$$

$$\text{Case } i = 0, f(0) = 1, |0j\rangle \rightarrow |0\bar{j}\rangle$$

$$\text{Case } i = 1, f(1) = 0, |1j\rangle \rightarrow |1j\rangle$$

$$\text{Case } i = 1, f(1) = 1, |1j\rangle \rightarrow |1\bar{j}\rangle$$

When f is boolean function of m bits:

$$U_f = \begin{pmatrix} X^{f(0)} & & & \\ & X^{f(1)} & & \\ & & \ddots & \\ & & & X^{f(2^m-1)} \end{pmatrix}$$

When f is function of m bits returning n bits:

$$f = \{0, \dots, 2^m - 1\} \rightarrow \{0, \dots, 2^n - 1\}$$

$$U_f |i_1 \dots i_m\rangle |j_1 \dots j_n\rangle = U_f |i\rangle |j\rangle = |i\rangle |j \oplus f(i)\rangle$$

where \oplus is addition mod n . U_f is a permutation matrix (which makes it unitary) with a single 1 in each column.

U_f is made up of shifting matrixes, which work like: $A_p |j\rangle = |p \oplus j\rangle$ for $j, p = 0, \dots, 2^n - 1$

Therefore $|f(0)\rangle - |\overline{f(0)}\rangle = (-1)^{f(0)} (|0\rangle - |1\rangle)$

Continuing...

$$\begin{aligned}
 &= \frac{1}{2} \left((-1)^{f(0)} |0\rangle (|0\rangle - |1\rangle) + (-1)^{f(1)} |1\rangle (|0\rangle - |1\rangle) \right) \\
 &= \frac{1}{\sqrt{2}} \left((-1)^{f(0)} |0\rangle + (-1)^{f(1)} |1\rangle \right) \frac{|0\rangle - |1\rangle}{\sqrt{2}} \\
 &= \frac{1}{\sqrt{2}} (-1)^{f(0)} \left(|0\rangle + (-1)^{f(1)-f(0)} |1\rangle \right) \frac{|0\rangle - |1\rangle}{\sqrt{2}}
 \end{aligned}$$

Case $f(0) = f(1)$, then $\left(|0\rangle + (-1)^{f(1)-f(0)} |1\rangle \right) = |0\rangle + |1\rangle$

Case $f(0) \neq f(1)$, then $\left(|0\rangle + (-1)^{f(1)-f(0)} |1\rangle \right) = |0\rangle - |1\rangle$

Continuing...

$$\begin{aligned}
 \xrightarrow{H \otimes I} & \begin{cases} (-1)^{f(0)} |0\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} & \text{if } f(0) = f(1) \\ (-1)^{f(0)} |1\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} & \text{if } f(0) \neq f(1) \end{cases} \\
 = & \pm |f(0) \oplus f(1)\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}
 \end{aligned}$$

Deutsch-Jonsa Algorithm

Given $f : \{0, \dots, 2^n - 1\} \rightarrow \{0, 1\}$

where function f is either balanced or constant. Constant means $f(j) = f(0) \forall j$. Balanced means if $A_0 = \{j : f(j) = 0\}$ and $A_1 = \{j : f(j) = 1\}$, then $|A_0| = |A_1| = 2^{n-1}$. The algorithm determines whether a function is constant or balanced, assuming it won't be anything else.

Classical Solution

Requires $2^{n-1} + 1$ evaluations worst case. Algorithm is to loop through the inputs, checking to see if the function ever returns two different values. If it does return two different values, the function is not constant and the loop can be terminated. After the halfway point, if only one value has been returned, the function is not balanced and can be labeled constant.

Quantum Solution

Circuit: $[M \otimes I](H \otimes I)U_f(H^{\otimes n} \otimes H)|0\rangle^{\otimes n}|1\rangle$

$$\begin{aligned}
|0\rangle^{\otimes n} |1\rangle &\xrightarrow{H^{\otimes n} \otimes H} \left(\frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} |j\rangle \right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \\
&= \frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} \left(\frac{|j0\rangle - |j1\rangle}{\sqrt{2}} \right) \\
&\xrightarrow{U_f} \frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} \left(\frac{|jf(j)\rangle - |j\overline{f(j)}\rangle}{\sqrt{2}} \right) \\
&= \frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} |j\rangle \left(\frac{|f(j)\rangle - |\overline{f(j)}\rangle}{\sqrt{2}} \right) \\
&= \frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} |j\rangle (-1)^{f(j)} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \\
&\xrightarrow{H^{\otimes n} \otimes I} \frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} \left(\frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} (-1)^{j \cdot k} |k\rangle \right) (-1)^{f(j)} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \\
&= \sum_k a_k |k\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)
\end{aligned}$$

Setting $a_k = \frac{1}{2^n} \sum_j (-1)^{j \cdot k} (-1)^{f(j)}$ to represent the “amplitude” of each k .

Looking at $a_0 = \frac{1}{2^n} \sum_j (-1)^{f(j)}$:

If function f is constant then $a_0 = \pm 1$ which means $a_k = 0$ for all $k \neq 0$. This is because of completeness, if coefficient of one basis state 1, all others must be zero. If function f is balanced then $a_0 = 0$.

Homework: What do we need to do to detect 3/4 balanced instead of 1/2 balanced.

Randomized Classical Algorithm

Generate k random input to function f , if function evaluated at any two of the inputs is different, the function will be labeled balanced, otherwise it's considered constant. Algorithm can fail, outputting constant when the function is actually balanced.

$$A_0 = \{j : f(j) = 0\}, A_1 = \{j : f(j) = 1\}$$

Balanced when $|A_0| = |A_1|$

$$\text{Probability of picking 1 sample which is 0: } p(1, 0) = \frac{1}{2}$$

$$\text{Probability of picking } k \text{ samples which are 0: } p(k, 0) = \frac{1}{2^k}$$

$$\text{Probability of picking 1 sample which is 1: } p(1, 1) = \frac{1}{2}$$

$$\text{Probability of picking } k \text{ samples which are 1: } p(k, 1) = \frac{1}{2^k}$$

$$\text{Probability of failure given } f \text{ is balanced: } p(k, 0) + p(k, 1) = \frac{2}{2^k}$$

Set $p > \frac{2}{2^k}$ to succeed with arbitrary probability.

Lectures 14/15 (5/7 March)

Midterm and Midterm Review

Lecture 16 (19 March)

[Book Chapter 5]

Discrete Fourier Transform

Maps vectors $x_0, \dots, x_{N-1} \rightarrow y_0, \dots, y_{N-1}$ like:

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi i k j / N}$$

Cost $O(N^2)$

1965 – Cooley-Tukey Fast Fourier Transform (FFT) cost $O(N \log N)$

Quantum Fourier Transform

Input state $|j\rangle$ for $j = 0, \dots, N-1$ where $N = 2^n$ (N is power of 2):

$$F|j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k / N} |k\rangle$$

Equivalence to Discrete Transform

$$\begin{aligned} F \left(\sum_{j=0}^{N-1} x_j |j\rangle \right) &= \sum_{j=0}^{N-1} x_j F|j\rangle \\ &= \sum_{j=0}^{N-1} x_j \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k / N} |k\rangle \\ &= \sum_{k=0}^{N-1} \left(\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi i j k / N} \right) |k\rangle \end{aligned}$$

Applied to $|0\rangle$

$$F|0\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i 0 k / N} |k\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle = H^{\otimes n} |0\rangle^{\otimes n}$$

Matrix representation

$$F|j\rangle = \frac{1}{\sqrt{N}} \begin{pmatrix} \vdots \\ e^{2\pi i j k / N} \\ e^{2\pi i j (k+1) / N} \\ \vdots \end{pmatrix}$$

$$F = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & e^{2\pi i 1/N} & & e^{2\pi i (N-1)/N} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{2\pi i (N-1)/N} & & e^{2\pi i (N-1)^2/N} \end{pmatrix}$$

$$F = \frac{1}{\sqrt{N}} (e^{2\pi i jk/N})_{k,j=0,\dots,N-1}$$

Proof F is Unitary

$$\begin{aligned} F^H &= \frac{1}{\sqrt{N}} (e^{-2\pi i jk/N})_{j,k=0,\dots,N-1} \\ F^H F &= \frac{1}{N} \left(\sum_{k=0}^{N-1} e^{-2\pi i pk/N} e^{2\pi i kq/N} \right)_{p,q=0,\dots,N-1} \\ &= \frac{1}{N} \left(\sum_{k=0}^{N-1} e^{2\pi i k(q-p)/N} \right)_{p,q} \end{aligned}$$

If $p = q$, $(F^H F)_{p,q} = 1$

If $p \neq q$,

$$\begin{aligned} (F^H F)_{p,q} &= \sum_{k=0}^{N-1} e^{2\pi i k(q-p)/N} \\ &= \frac{e^{2\pi i (q-p) \frac{N-1}{N}} e^{2\pi i (q-p) \frac{1}{N}} - 1}{e^{2\pi i (q-p) \frac{1}{N}} - 1} \\ &= \frac{e^{2\pi i (q-p)} - 1}{e^{2\pi i (q-p) \frac{1}{N}} - 1} \\ &= 1 - 1 = 0 \end{aligned}$$

Therefore, $F^H F = I$, and F is unitary.

(Formula for sum of a geometric progression used above: $\sum_{k=0}^n r^k = \frac{r^{n+1}-1}{r-1}$)

Tensor Product Representation

Notation:

$$j = j_1 j_2 \dots j_n = j_1 2^{n-1} + j_2 2^{n-2} + \dots + j_n$$

$$\frac{j}{2^n} = 0.j_1 j_2 \dots j_n = j_1 \frac{1}{2} + j_2 \frac{1}{2^2} + j_n \frac{1}{2^n}$$

$$\frac{j}{2^\ell} = j_1 \dots j_{n-\ell} . j_{n-\ell+1} \dots j_n$$

$$e^{2\pi i \frac{j}{2^\ell}} = e^{2\pi i (j_1 \dots j_{n-\ell} . j_{n-\ell+1} \dots j_n)}$$

Lemma:

$$F |j\rangle = F |j_1 \dots j_n\rangle = \frac{|0\rangle + e^{2\pi i 0.j_n} |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i 0.j_{n-1} j_n} |1\rangle}{\sqrt{2}} \otimes \dots \otimes \frac{|0\rangle + e^{2\pi i 0.j_1 j_2 \dots j_n} |1\rangle}{\sqrt{2}}$$

Proof, start with:

$$F |j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i jk/N} |k\rangle$$

Decompose $k = (k_1 \dots k_n)$

$$\begin{aligned}
F |j\rangle &= \frac{1}{\sqrt{N}} \sum_{k_1=0}^1 \dots \sum_{k_n=0}^1 e^{2\pi i j 0.k_1 \dots k_n} |k_1\rangle \dots |k_n\rangle \\
&= \frac{1}{\sqrt{N}} \sum_{k_1=0}^1 \dots \sum_{k_n=0}^1 e^{2\pi i j (k_1 \frac{1}{2} + k_2 \frac{1}{2^2} + \dots + k_n \frac{1}{2^n})} |k_1\rangle \dots |k_n\rangle \\
&= \frac{1}{\sqrt{N}} \sum_{k_1=0}^1 \dots \sum_{k_n=0}^1 \bigotimes_{\ell=1}^n e^{2\pi i j k_\ell \frac{1}{2^\ell}} |k_\ell\rangle \\
&= \frac{1}{\sqrt{N}} \bigotimes_{\ell=1}^n \sum_{k_\ell=0}^1 e^{2\pi i j k_\ell \frac{1}{2^\ell}} |k_\ell\rangle
\end{aligned}$$

Simple example illustrating last step:

$$\begin{aligned}
&\sum_{k_1=0}^1 \sum_{k_2=0}^1 e^{2\pi i j k_1 \frac{1}{2}/N} e^{2\pi i j k_2 \frac{1}{4}} |k_1\rangle |k_2\rangle \\
&= \left(\sum_{k_1=0}^1 e^{2\pi i j k_1 \frac{1}{2}} |k_1\rangle \right) \left(\sum_{k_2=0}^1 e^{2\pi i j k_2 \frac{1}{4}} |k_2\rangle \right)
\end{aligned}$$

Continuing:

$$F |j\rangle = \bigotimes_{\ell=1}^n \left(\frac{|0\rangle + e^{2\pi i j / 2^\ell} |1\rangle}{\sqrt{2}} \right)$$

Dividing j by 2^ℓ is equivalent to moving the decimal point in the binary representation of j by ℓ places to the left. Additionally, after the shift, any digits to the left of the dot can be discarded. This is because the function e^{xi} has a period of 2π , so the only the fractional part of $\frac{j}{2^\ell}$ matters, and the whole number part can be set to 0.

$$F |j\rangle = F |j_1 \dots j_n\rangle = \frac{|0\rangle + e^{2\pi i 0.j_n} |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i 0.j_{n-1}j_n} |1\rangle}{\sqrt{2}} \otimes \dots \otimes \frac{|0\rangle + e^{2\pi i 0.j_1j_2 \dots j_n} |1\rangle}{\sqrt{2}}$$

At $\ell = 1$, term is: $\left(\frac{|0\rangle + e^{2\pi i 0.j_n} |1\rangle}{\sqrt{2}} \right)$

At $\ell = 2$, term is: $\left(\frac{|0\rangle + e^{2\pi i 0.j_{n-1}j_n} |1\rangle}{\sqrt{2}} \right)$

At $\ell = 3$, term is: $\left(\frac{|0\rangle + e^{2\pi i 0.j_{n-2}j_{n-1}j_n} |1\rangle}{\sqrt{2}} \right)$

Tip: Remember this lemma for final

Fourier Transform as Circuit

Define $R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i / 2^k} \end{pmatrix}$

$$R_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I$$

$$R_1 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = Z$$

$$R_2 = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} = S$$

$$R_3 = \begin{pmatrix} 1 & 0 \\ 0 & e^{\pi i/4} \end{pmatrix} = T$$

When evaluating cost of implementing fourier transform, assume implementing each of one these R gates has unit cost. The assumption may not necessarily be true in an actual implementation.

$$\text{Circuit: } \bigotimes_{\ell=1}^n \left(R_{n-\ell+1}^{\#n} \dots R_2^{\#\ell+1} H \right) |j_\ell\rangle$$

First Qubit:

$$|j_1\rangle \xrightarrow{H} \frac{|0\rangle + (-1)^{j_1} |1\rangle}{\sqrt{2}} = \frac{|0\rangle + e^{2i\pi \cdot 0 \cdot j_1} |1\rangle}{\sqrt{2}}$$

$$\text{Trick used: } e^{2i\pi \cdot 0 \cdot j_1} = \begin{cases} 1 & j_1 = 0 \\ e^{2\pi i/2} & j_1 = 1 \end{cases} = (-1)^{j_1}$$

$$\begin{aligned} & \xrightarrow[w/j_2]{R_2} \frac{R_2^{j_2} |0\rangle + e^{2i\pi \cdot 0 \cdot j_1} R_2^{j_2} |1\rangle}{\sqrt{2}} \\ & = \frac{|0\rangle + e^{2i\pi \cdot 0 \cdot j_1} e^{2\pi i j_2/2^2} |1\rangle}{\sqrt{2}} \\ & = \frac{|0\rangle + e^{2i\pi \cdot 0 \cdot j_1 j_2} |1\rangle}{\sqrt{2}} \\ & \xrightarrow[w/j_3]{R_3} \frac{R_3^{j_3} |0\rangle + e^{2i\pi \cdot 0 \cdot j_1} R_3^{j_3} |1\rangle}{\sqrt{2}} \\ & = \frac{|0\rangle + e^{2i\pi \cdot 0 \cdot j_1 j_2 j_3} |1\rangle}{\sqrt{2}} \\ & \vdots \\ & \xrightarrow[w/j_n]{R_n} \frac{|0\rangle + e^{2i\pi \cdot 0 \cdot j_1 j_2 \dots j_n} |1\rangle}{\sqrt{2}} \end{aligned}$$

Second qubit:

$$\begin{aligned}
|j_2\rangle &\xrightarrow{H} \frac{|0\rangle + e^{2i\pi 0 \cdot j_2} |1\rangle}{\sqrt{2}} \\
&\xrightarrow[\frac{w}{j_3}]{R_2} \frac{|0\rangle + e^{2i\pi 0 \cdot j_2} e^{2\pi i j_3 / 2^2} |1\rangle}{\sqrt{2}} \\
&= \frac{|0\rangle + e^{2i\pi 0 \cdot j_2 j_3} |1\rangle}{\sqrt{2}} \\
&\xrightarrow[\frac{w}{j_4}]{R_3} \frac{|0\rangle + e^{2i\pi 0 \cdot j_2 j_3} e^{2\pi i j_4 / 2^3} |1\rangle}{\sqrt{2}} \\
&= \frac{|0\rangle + e^{2i\pi 0 \cdot j_2 j_3 j_4} |1\rangle}{\sqrt{2}} \\
&\vdots \\
&\xrightarrow[\frac{w}{j_n}]{R_{n-1}} \frac{|0\rangle + e^{2i\pi 0 \cdot j_2 \dots j_n} |1\rangle}{\sqrt{2}}
\end{aligned}$$

Last qubit:

$$|j_n\rangle \xrightarrow{H} \frac{|0\rangle + e^{2i\pi 0 \cdot j_n} |1\rangle}{\sqrt{2}}$$

Circuit output is

$$\frac{|0\rangle + e^{2i\pi 0 \cdot j_1 \dots j_n} |1\rangle}{\sqrt{2}} \otimes \dots \otimes \frac{|0\rangle + e^{2i\pi 0 \cdot j_n} |1\rangle}{\sqrt{2}}$$

which is the Lemma 1 expression for the Fourier transform with qubits in reverse order. To correct this, $\lfloor \frac{n}{2} \rfloor$ swap gates can be used to swap the top and bottom bits, second to top and second to bottom bits, and so on. Each swap gate is made of 3 CNOT gates.

Cost: Each qubit requires $O(n)$ gates to transform, total cost is $O(n^2)$ for transforming all qubits and $O(n)$ for swaps, which is $O(n^2)$ total. Best classical cost is $O(N \log N) = O(2^n n)$, so quantum implementation represents an exponential speedup.

Next Lecture: Phase Estimation

Homework: Implement *reverse* fourier transform, finding algorithm and cost.

Lecture 17 (21 March)

Lecture 16 Review

$$F|j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k / N} |k\rangle$$

$$F|j_1 \dots j_n\rangle = \left(\frac{|0\rangle + e^{2\pi i 0 \cdot j_n} |k\rangle}{\sqrt{0}} \right) \otimes \dots \otimes \left(\frac{|0\rangle + e^{2\pi i 0 \cdot j_1 \dots j_n} |k\rangle}{\sqrt{0}} \right)$$

Cost: quantum implementation $O(n^2)$ beats classical $O(2^n n) = O(N \log N)$

Hint: Finding F^H , problem 1 next homework. Two approaches. One is to look at the circuit and determine meaning of the conjugate transpose as an operator. Other is to look at the definition of F^H which differs only by a minus sign.

$$F^H|j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{-2\pi i j k / N} |k\rangle$$

Cost should be the same.

Phase Estimation

Overview

Heart of many quantum algorithms. Related to solution to Schrodinger's equation, important for quantum simulation.

Problem: Given unitary matrix U , size $N \times N$ where $N = 2^k$ (using k instead of n as in previous lecture because n is used for something else here). Also given $|u\rangle$, eigenvector of U , so

$$U |u\rangle = \lambda |u\rangle = e^{2\pi i \varphi} |u\rangle$$

for $\varphi \in [0, 1]$. Goal is to find approximation of φ with accuracy 2^{-n} . Algorithm is covered in this lecture, the correctness is shown next lecture. φ can be represented as:

$$\varphi = 0.\varphi_1\varphi_2 \dots \varphi_n\varphi_{n+1}$$

If only n digits are given, precision is lost but bounded by a maximum error. Maximum error can be computed by assuming every digit after the n th is 1 when it should be zero:

$$\sum_{j=n+1}^{\infty} \frac{1}{2^j} = \frac{1}{2^{n+1}} \sum_{j=0}^{\infty} \frac{1}{2^j} = \frac{1}{2^{n+1}} \left(\frac{1}{1 - \frac{1}{2}} \right) = 2^{-n}$$

Givens

1. Given $|u\rangle$, eigenvector as superposition state k qubits long.
 2. Given controlled operators U^{2^j} for $j = 0, 2, \dots$ implemented as black boxes.
- [See also paper: Quantum Algorithms Revisited]

Algorithm

To see understand the algorithm, it helps to look at how a U^{2^j} gate controlled by an $H|0\rangle$ qubit acts:

$$\begin{aligned} \frac{|0\rangle |u\rangle + |1\rangle |u\rangle}{\sqrt{2}} &\xrightarrow{C(U^{2^j})} \frac{|0\rangle |u\rangle + |1\rangle U^{2^j} |u\rangle}{\sqrt{2}} \\ &= \frac{|0\rangle |u\rangle + |1\rangle e^{2\pi i \varphi 2^j} |u\rangle}{\sqrt{2}} \\ &= \frac{|0\rangle + |1\rangle e^{2\pi i \varphi 2^j}}{\sqrt{2}} |u\rangle \end{aligned}$$

Using different values of j results in qubits that can be expressed in terms of different portions of the bitwise representation of φ :

$$U^{2^{t-1}} \rightarrow |0\rangle + e^{2\pi i \varphi 2^{t-1}} |1\rangle = |0\rangle + e^{2\pi i 0.\varphi_t \varphi_{t+1} \dots} |1\rangle$$

$$U^{2^{t-2}} \rightarrow |0\rangle + e^{2\pi i \varphi 2^{t-2}} |1\rangle = |0\rangle + e^{2\pi i 0.\varphi_{t-1} \varphi_t \varphi_{t+1} \dots} |1\rangle$$

\vdots

$$U^{2^1} \rightarrow |0\rangle + e^{2\pi i \varphi 2} |1\rangle = |0\rangle + e^{2\pi i 0.\varphi_2 \dots \varphi_t \varphi_{t+1} \dots} |1\rangle$$

$$U^{2^0} \rightarrow |0\rangle + e^{2\pi i\varphi} |1\rangle = |0\rangle + e^{2\pi i0.\varphi_1\dots\varphi_t\varphi_{t+1}\dots} |1\rangle$$

As can be seen above, U^{2^j} essentially means shift φ by j bits to the left. The whole number portion of the resulting numbers is discarded because the exponential function is periodic.

The states shown above look like the states that would result from $F|\varphi_1\dots\varphi_t\rangle$, where F is the quantum Fourier transform.

Circuit

Circuit is made of two registers, top register is t $|0\rangle$ qubits, bottom register is $|u\rangle$ (which is k qubits long). Hadamard gates are applied to each $|0\rangle$ qubit in the first register, and output of those is used to control a sequence of U^{2^j} gates on the second register. The first gate on the second register, U^{2^0} , is controlled by the $H|0\rangle$ output on the first qubit, then there is a U^{2^1} gate controlled by $H|0\rangle$ from the second qubit, followed by a U^{2^2} gate controlled by the third qubit, and so on. [Textbook figure 5.2 page 222]

After these gates, an inverse Fourier transform is applied to the first register, and measurement of that register on the computational basis yields the binary digits of the representation of φ .

We need to show that before the inverse Fourier transform is applied, the top register has value $\frac{1}{2^{t/2}} \sum_{k=0}^{2^t-1} e^{2\pi ik\varphi} |k\rangle$. This can be proved with induction. Start with the last two qubits:

$$\begin{aligned} & \frac{1}{2} (|0\rangle + e^{2\pi i2\varphi} |1\rangle) (|0\rangle + e^{2\pi i\varphi} |1\rangle) \\ &= |00\rangle + e^{2\pi i\varphi} |01\rangle + e^{2\pi i2\varphi} |10\rangle + e^{2\pi i3\varphi} |11\rangle \end{aligned}$$

Then the inductive step is:

$$\begin{aligned} & \frac{1}{\sqrt{2}} (|0\rangle + e^{2\pi i2^{t-1}\varphi} |1\rangle) \left(\frac{1}{2^{(t-1)/2}} \sum_{k=0}^{2^{t-1}-1} e^{2\pi ik\varphi} |k\rangle \right) \\ &= \frac{1}{2^{t/2}} \sum_{k=0}^{2^{t-1}-1} (e^{2\pi ik\varphi} |0k\rangle + e^{2\pi i(k+2^{t-1})\varphi} |1k\rangle) \\ &= \frac{1}{2^{t/2}} \sum_{j=0}^{2^t-1} e^{2\pi ij\varphi} |j\rangle \end{aligned}$$

Circuit and Expression

A second interpretation of phase estimation can be seen by looking at the overall circuit diagram [Textbook figure 5.3 page 223].

$$\begin{aligned} |0\rangle^{\otimes t} |u\rangle & \xrightarrow{H^{\otimes t} \otimes I} \frac{1}{2^{t/2}} \sum_{j=0}^{2^t-1} |j\rangle |u\rangle \\ & \xrightarrow{U^j} \frac{1}{2^{t/2}} \sum_{j=0}^{2^t-1} |j\rangle U^j |u\rangle \\ & = \frac{1}{2^{t/2}} \sum_{j=0}^{2^t-1} |j\rangle e^{2\pi ij\varphi} |u\rangle \\ & \xrightarrow{F^H} \sum_{\ell=0}^{2^t-1} g(\ell, \varphi) |\ell\rangle \end{aligned}$$

We aren't solving for the coefficients of possible output basis states right now, we just refer to them here as $g(\ell, \varphi)$ or α_ℓ . (The next lecture solves for α_ℓ). Now when φ can be expressed as $0.\varphi_1 \dots \varphi_t$ exactly, there is unique ℓ so that

$$|a_\ell| = |g(\ell_1 \varphi)| = 1$$

and all the other α_ℓ values are 0. The algorithm succeeds in this case, but it also succeeds more generally, and this is shown in the next lecture.

(Above depends on the fact that the sequence of controlled-U operations in the circuit transform a basis state $|j\rangle |u\rangle$ to $|j\rangle U^j |u\rangle$. This is exercise 5.7 in the book, and can be seen from the fact that if j has a t bit representation:

$$\begin{aligned} |j\rangle U^j |u\rangle &= |j\rangle U^{j_1 2^0 + j_2 2^1 + \dots + j_t 2^{t-1}} |u\rangle \\ &= |j\rangle U^{2^0 j_1} \times U^{2^1 j_2} \times \dots \times U^{2^{t-1} j_t} |u\rangle \end{aligned}$$

)

Cost of circuit in gates is t H gates, t controlled U^{2^j} gates (assuming the exponents don't affect cost), and an F^H gate which has cost t^2 . $O(t + t + t^2) = O(t^2)$. Cost in qubits is $t + k$, $O(t + k)$

Application: If you have real matrix, A , so that $A = A^T$, e^{iA} is unitary, $i\hbar \frac{g\psi(x,t)}{gt} = H\psi(x,t)$ where h is Planck's constant, H is Hamiltonian.

Lecture 18 (26 March)

Missed Class, filling in blanks from Textbook section 5.2.1.

Output of the first register of the phase estimation circuit before inverse fourier transform is:

$$\begin{aligned} &\frac{1}{2^{t/2}} \left(|0\rangle + e^{2\pi i 2^{t-1} \varphi} |1\rangle \right) \left(|0\rangle + e^{2\pi i 2^{t-2} \varphi} |1\rangle \right) \dots \left(|0\rangle + e^{2\pi i 2^0 \varphi} |1\rangle \right) \\ &= \frac{1}{2^{t/2}} \sum_{k=0}^{2^t-1} e^{2\pi i \varphi k} |k\rangle \end{aligned}$$

If $\varphi = 0.\varphi_1 \dots \varphi_t$ exactly, applying inverse fourier transform to this state gives state $|\varphi_1 \dots \varphi_t\rangle$. When φ cannot be represented with t bits, the analysis below applies.

Let b be integer in the range 0 to $2^t - 1$ such that $b/2^t = 0.b_1 \dots b_t$ is the best t bit approximation to φ which is less than φ . Error is $\delta \equiv \varphi - b/2^t$ and $0 \leq \delta \leq 2^{t-1}$. Applying the inverse fourier transform to the first register gives:

$$\begin{aligned} &\frac{1}{2^t} \sum_{\ell=0}^{2^t-1} \sum_{k=0}^{2^t-1} e^{-2\pi i k \ell / 2^t} e^{2\pi i \varphi k} |\ell\rangle \\ &= \sum_{\ell=0}^{2^t-1} \frac{1}{2^t} \sum_{k=0}^{2^t-1} \left(e^{2\pi i (\varphi - \ell/2^t)} \right)^k |\ell\rangle \\ &= \sum_{\ell=0}^{2^t-1} \alpha_{\ell-b} |\ell\rangle \end{aligned}$$

Let α_ℓ be the amplitude of $|b + \ell\rangle$ (taking addition inside the state and subtraction in the subscript of α to

be modulo 2^t):

$$\begin{aligned}
\alpha_\ell &\equiv \frac{1}{2^t} \sum_{k=0}^{2^t-1} \left(e^{2\pi i(\varphi-(b+\ell)/2^t)} \right)^k \\
&= \frac{1}{2^t} \frac{1 - e^{2\pi i(2^t\varphi-(b+\ell))}}{1 - e^{2\pi i(\varphi-(b+\ell)/2^t)}} \\
&= \frac{1}{2^t} \frac{1 - e^{2\pi i(2^t\delta-\ell)}}{1 - e^{2\pi i(\delta-\ell/2^t)}}
\end{aligned}$$

The second step follows from the formula for sum of a geometric series, the third from substituting $\delta = \varphi - b/2^t$.

Introduce new variables. Take the output of the final measurement to be m , and chose an error tolerance, e which is a positive integer such that if $|m - b| > e$, the algorithm is considered to have failed. The probability of that failure condition is:

$$p(|m - b| > e) = \sum_{\ell=-2^{t-1}+1}^{-(e+1)} |\alpha_\ell|^2 + \sum_{\ell=e+1}^{2^t-1} |\alpha_\ell|^2$$

For any real θ , $|1 - e^{i\theta}| \leq 2$, so

$$|\alpha_\ell| \leq \frac{1}{2^t |1 - e^{2\pi i(\delta-\ell/2^t)}|}$$

Whenever $-\pi \leq \theta \leq \pi$, then $|1 - e^{i\theta}| \geq 2|\theta|/\pi$. And when $-2^{t-1} < \ell \leq 2^{t-1}$, then $-\pi \leq 2\pi(\delta - \ell/2^t) \leq \pi$, therefore:

$$|\alpha_\ell| \leq \frac{1}{2^t \cdot 2(\delta - \ell/2^t)} = \frac{1}{-\frac{1}{2}(\ell - 2^t\delta)}$$

And

$$\begin{aligned}
p(|m - b| > e) &\leq \frac{1}{4} \left(\sum_{\ell=-2^{t-1}+1}^{-(e+1)} \frac{1}{(\ell - 2^t\delta)^2} + \sum_{\ell=e+1}^{2^t-1} \frac{1}{(\ell - 2^t\delta)^2} \right) \\
&\leq \frac{1}{4} \left(\sum_{\ell=-2^{t-1}+1}^{-(e+1)} \frac{1}{\ell^2} + \sum_{\ell=e+1}^{2^t-1} \frac{1}{(\ell - 1)^2} \right) \\
&\leq \frac{1}{2} \sum_{\ell=e}^{2^{t-1}+1} \frac{1}{\ell^2} \\
&\leq \frac{1}{2} \int_{e-1}^{2^{t-1}-1} \frac{1}{\ell^2} d\ell \\
&= \frac{1}{2(e-1)}
\end{aligned}$$

Second step follows because $0 \leq 2^t\delta \leq 1$.

When approximating φ to an accuracy of 2^{-n} , $e = 2^{t-n} - 1$. When using $t = n + p$ qubits in phase estimation, then the probability of success is $1 - \frac{1}{2(2^p-2)}$. Let ϵ be the probability of failure, then you can find minimum t that won't exceed that failure rate:

$$\begin{aligned}
\epsilon &= \frac{1}{2(2^p - 2)} \\
2^p &= \frac{1}{2\epsilon} + 2 \\
p &= \log_2 \left(\frac{1}{2\epsilon} + 2 \right)
\end{aligned}$$

So for success probability of at least $1 - \epsilon$, choose $t = n + \lceil \log_2(2 + \frac{1}{2\epsilon}) \rceil$:

Lecture 19 (28 March)

Homework Hint

Homework 4, Problem 2: Convolution Theorem and Fourier Transform

Given coefficients of basis states α_0, α_{N-1} and β_0, β_{N-1} which transform into γ_0, γ_{N-1} and δ_0, δ_{N-1}

Discrete FT:

$$\gamma_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \alpha_k e^{2\pi i j k / N}, \quad \delta_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \beta_k e^{2\pi i j k / N}$$

Inverse FT:

$$\alpha_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \gamma_k e^{-2\pi i j k / N}, \quad \beta_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \delta_k e^{-2\pi i j k / N}$$

Convolution:

$$\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \sum_{\ell=0}^{N-1} \alpha_\ell \beta_{j-\ell} |j\rangle$$

Use FT formulas to substitute α_ℓ and $\beta_{j-\ell}$ above:

$$\beta_{j-\ell} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \delta_k e^{-2\pi i (j-\ell)k / N}$$

Result is messy, you end up with four sums, but it simplifies.

Performance Analysis of Phase Estimation

We can compute bounded probability of failure and use that to determine how many qubits to use in top register to get desired accuracy. Last lecture proved:

$$Pr \{ |m - b| > e = 2^{t-n} - 1 \} \leq \frac{1}{2(e-1)} < \epsilon$$

where ϵ is highest allowed probability of failure, m is the measurement of the first register on the computational basis, b is the representation of φ expressed as a measurement, e is our error tolerance, expressed as the highest allowed absolute difference between m and b . t is the number of qubits in the top register and n is the desired accuracy in bits, which is just an alternate expression of error tolerance.

This expression of probability of failure and accuracy is unwieldy and not what we originally set out to determine, which was finding:

$$Pr \{ |\varphi - \hat{\varphi}| \leq 2^{-n} \}$$

where $\hat{\varphi} = \frac{m}{2^t}$. To find this, we switch to finding probability of failure instead of probability of success because that it is easier to bound that from above.

$$Pr \{ |\varphi - \hat{\varphi}| > 2^{-n} \} = Pr \left\{ \left| \varphi - \frac{b}{2^t} + \frac{b}{2^t} - \hat{\varphi} \right| > 2^{-n} \right\}$$

Using the triangle inequality ($|a + b| \leq |a| + |b|$) :

$$\begin{aligned}
&\leq Pr \left\{ \left| \varphi - \frac{b}{2^t} \right| + \left| \frac{b}{2^t} - \hat{\varphi} \right| > 2^{-n} \right\} \\
&\leq Pr \left\{ 2^{-t} + \left| \frac{b}{2^t} - \hat{\varphi} \right| > 2^{-n} \right\} \\
&= Pr \left\{ \left| \frac{b}{2^t} - \hat{\varphi} \right| > 2^{-n} - 2^{-t} \right\} \\
&= Pr \{ |b - m| > 2^{t-n} - 1 \} \\
&= Pr \{ |b - m| > e \} \\
&\leq \frac{1}{2(e-1)}
\end{aligned}$$

P.E. w/ approx Eigenvector

Phase estimation algorithm requires an eigenvector of the matrix U , but it is also possible to use approximations of an eigenvector and still get meaningful results. [Paper: Abrams + Lloyd]

Given $|u\rangle$, an approximate eigenvector which can be expressed in terms of real eigenvectors $|u_k\rangle$ as:

$$|u\rangle = \sum_{k=0}^{N-1} d_k |u_k\rangle$$

We would like to estimate the phase φ_0 corresponding to $\lambda_0 = e^{2\pi i \varphi_0} |u_0\rangle$. The of the P.E. on this input is:

$$\begin{aligned}
|0\rangle^{\otimes t} |u\rangle &\xrightarrow{H^{\otimes t} \otimes I} \frac{1}{2^{t/2}} \sum_{j=0}^{2^t-1} |j\rangle |u\rangle \\
&= \frac{1}{2^{t/2}} \sum_{j=0}^{2^t-1} |j\rangle \sum_{k=0}^{2^t-1} d_k |u_k\rangle \\
&= \sum_{k=0}^{2^t-1} d_k \frac{1}{2^{t/2}} \sum_{j=0}^{2^t-1} |j\rangle |u_k\rangle \\
&\xrightarrow{U^j} \sum_{k=0}^{2^t-1} d_k \frac{1}{2^{t/2}} \sum_{j=0}^{2^t-1} |j\rangle U^j |u_k\rangle \\
&= \sum_{k=0}^{2^t-1} d_k \frac{1}{2^{t/2}} \sum_{j=0}^{2^t-1} e^{2\pi i \varphi_k j} |j\rangle |u_k\rangle \\
&\xrightarrow{F^H \otimes I} \sum_{k=0}^{2^t-1} d_k \sum_{j=0}^{2^t-1} g(\varphi_k, j) |j\rangle |u_k\rangle
\end{aligned}$$

$g(\varphi_k, j)$ in the previous lecture was α_j , the amplitude of output state j in the top register. The last lecture showed that α_j amplitudes were bounded from above, and that if j was far from b , then α_j would be low.

Next, measure top register to find $Pr \{ |m - b_0| < e \}$ where $0 < \varphi_0 - \frac{b_0}{2^t} < 2^{-t}$. Probability of any measurement m is:

$$P_m = \sum_{k=0}^{N-1} |d_k g(\varphi_k, m)|^2$$

Let G be set of measurements which satisfy $|m - b_o| < e$, then

$$\begin{aligned} Pr \{G\} &= \sum_{k=0}^{N-1} \sum_{m \in G} |d_k g(\varphi_k, m)|^2 \\ &\geq |d_0|^2 \sum_{m \in G} |g(\varphi_0, m)|^2 \\ &= |d_0|^2 (1 - \epsilon) \end{aligned}$$

where ϵ is probability of failure given real eigenvector $|u_0\rangle$ ($\frac{1}{2(e-1)} < \epsilon$) and $d_0 = \langle u_0 | u \rangle$

Applications of P.E.: Order Finding

Given x , N positive integers, $x < N$, and $\gcd(x, N) = 1$.

Definition: The order of x modulo N is the least positive integer r so $x^r = 1 \pmod{N}$

Example:

$$x = 1, r = 1$$

$$x = 2, N = 5, r = 4$$

$$x = 5, N = 21$$

Lemma

Lemma: The order of x modulo N is $\leq N$

Proof:

For $k = 1, 2, 3, \dots, N$, and $r_k \in \{0, 1, \dots, N - 1\}$

$$x^k = \ell_k N + r_k$$

Assume none of the remainders r_k in the range of k is one, $r_k \neq 1 \forall k = 1, \dots, N$

If that's the case, then two of the remainders in the range have be the same, $\exists k, p : r_k = r_p$

$$\begin{aligned} x^k &= \ell_k N + r_k \\ x^p &= \ell_p N + r_p \end{aligned}$$

Subtracting these:

$$x^k - x^p = x^p (x^{k-p} - 1) = (\ell_k - \ell_p) N$$

Case $\ell_p = \ell_k$, then $x^{k-p} = 1$

Case $\ell_p \neq \ell_k$, since N cannot divide x^p because $\gcd(x, N) = 1$, N divides $x^{k-p} - 1$:

$$\begin{aligned} x^{k-p} - 1 &= \ell N \\ x^{k-p} &= 1 \pmod{N} \end{aligned}$$

which means $r \leq k - p \leq N$.

Both of these cases contain contradictions, which means the assumption that none of the remainders r_k is 1 for $k = 1, 2, 3, \dots, N$ is false, and the order must be $\leq N$.

Algorithm

Classically: No known algorithm solving order finding in polylog N

Quantum: P.E. solves in poly log N operations (gates)

Given $L = \lceil \log_2 N \rceil$, the unitary operator to use for phase estimation transforms like:

$$U |y\rangle = \begin{cases} |xy \bmod N\rangle & y = 0, 1, 2, \dots, N-1 \\ |y\rangle & y = N, N+1, \dots, 2^L-1 \end{cases}$$

Example: $x = 2$, $N = 5$, $L = \lceil \log_2 5 \rceil = 3$

$ y\rangle$	$U y\rangle$
$ 0\rangle$	$ 0\rangle$
$ 1\rangle$	$ 2\rangle$
$ 2\rangle$	$ 4\rangle$
$ 3\rangle$	$ 1\rangle$
$ 4\rangle$	$ 3\rangle$
$ 5\rangle$	$ 5\rangle$
$ 6\rangle$	$ 6\rangle$
$ 7\rangle$	$ 7\rangle$

Lecture 20 (2 April)

Lecture 19 Review

Item 1: With initial state $|u\rangle$, eigenvector for phase estimation satisfies $\varphi - \hat{\varphi} \leq 2^{-n}$ where n is the number of bits of desired accuracy, and $\hat{\varphi} = \frac{m}{2^L}$ is the measured value approximating φ . This condition has to hold with probability $\geq (1 - \epsilon)$, ϵ being the allowed probability of failure.

Item 2: If the initial state is some arbitrary initial vector, $|\tilde{u}\rangle$, instead of an eigenvector, the output will still satisfy $\varphi - \hat{\varphi} \leq 2^{-n}$ with probability $\geq |\langle u|\tilde{u}\rangle|^2 (1 - \epsilon)$. [There is another proof of this fact in 2 lines in a paper online about constructing initial states]

Example: Take unitary matrix U , which has two phases φ_1, φ_2 and two eigenvectors u_1, u_2 . The eigenvalues are related to the phases like $\lambda_1 = e^{2\pi i \varphi_1}$, $\lambda_2 = e^{2\pi i \varphi_2}$. If the initial state is $|\tilde{u}\rangle = \frac{1}{2}|u_1\rangle + \frac{\sqrt{3}}{2}|u_2\rangle$, the phase estimation algorithm will give close approximation of φ_1 with probability $(\frac{1}{2})^2 (1 - \epsilon)$, and a close approximation of φ_2 with probability $(\frac{\sqrt{3}}{2})^2 (1 - \epsilon)$.

Order Finding

Given x, N : $x < N$ and $\gcd(x, N) = 1$ (meaning x , and N are coprime), find the least positive integer r so $x^r = 1 \pmod{N}$.

U Matrix

The phase estimation algorithm can solve this problem using the matrix U that transforms like:

$$U |y\rangle = \begin{cases} |xy \bmod N\rangle & y = 0, 1, 2, \dots, N-1 \\ |y\rangle & y = N, N+1, \dots, 2^L-1 \end{cases}$$

where $L = \lceil \log_2 N \rceil$.

U is a permutation matrix, meaning that if it is input with a basis state, it's output is another just basis state, and each different input maps to a different output, so it outputs all possible basis states. And because states from N to 2^L are mapped to themselves, the bottom right corner of the matrix is actually the identity matrix, so

$$U = \begin{pmatrix} P & 0 \\ 0 & I \end{pmatrix}$$

To show U is unitary it suffices to show that it is 1 : 1, and it suffices to show that by showing P is 1 : 1 since

$$U^H U = \begin{pmatrix} P^H & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} P & 0 \\ 0 & I \end{pmatrix} = \begin{pmatrix} P^H P & 0 \\ 0 & I \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix}$$

Number Theory Aside

Given x, N then there might be a multiplicative inverse b so $bx = 1 \pmod{N}$

Examples:

$$x = 2, N = 5, b = 3$$

$x = 2, N = 4$, there is no b because $\gcd(2, 4) = 2 > 1$:

$$\begin{aligned} 2b &= 1 \pmod{4} \\ 2b - 1 &= 4\ell \\ \text{odd} &= \text{even} \end{aligned}$$

x has multiplicative inverse $x^{-1} \pmod{N}$ iff $\gcd(x, N) = 1$.

Showing U is Unitary

To show that the mapping $xy \pmod{N}$ is 1:1 for different values of y , we need to show no two values of y give the same output.

Take $z = xy_1 \pmod{N}$ and $z = xy_2 \pmod{N}$. The order finding problem assumes x and N are coprime, so we don't have to worry about multiplicative inverses not existing and:

$$\begin{aligned} x^{-1}xy_1 \pmod{N} &= x^{-1}xy_2 \pmod{N} \\ (1 + \ell N)y_1 \pmod{N} &= (1 + \ell N)y_2 \pmod{N} \\ y_1 \pmod{N} &= y_2 \pmod{N} \\ y_1 &= y_2 \end{aligned}$$

Therefore P is 1:1 $\Rightarrow U$ is 1:1 $\Rightarrow U$ is unitary .

If P is a permutation matrix then $P^{-1} = P^T$

Eigenvector of U

Definition: For $S = 0, \dots, r - 1$ define $|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i sk/r} |x^k \pmod{N}\rangle$

Then

$$\begin{aligned}
U |u_s\rangle &= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i s k / r} U |x^k \pmod{N}\rangle \\
&= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i s k / r} |x^{k+1} \pmod{N}\rangle \\
&= \frac{1}{\sqrt{r}} \sum_{k=1}^r e^{-2\pi i s (k-1) / r} |x^k \pmod{N}\rangle \\
&= \frac{1}{\sqrt{r}} e^{2\pi i s / r} \sum_{k=1}^r e^{-2\pi i s k / r} |x^k \pmod{N}\rangle
\end{aligned}$$

Case $k = r$, then $e^{-2\pi i s r / r} = 1$, $|x^r \pmod{N}\rangle = |1\rangle$

Case $k = 0$, then $e^{-2\pi i s 0 / r} = 1$, $|x^0 \pmod{N}\rangle = |1\rangle$

This means we can sum from 0 to $r - 1$ instead of 1 to r :

$$\begin{aligned}
U |u_s\rangle &= \frac{1}{\sqrt{r}} e^{2\pi i s / r} \sum_{k=0}^{r-1} e^{-2\pi i s k / r} |x^k \pmod{N}\rangle \\
&= e^{-2\pi i s / r} |u_s\rangle
\end{aligned}$$

Phase Estimation for Order Finding

The phase of matrix U given eigenvector $|u_s\rangle$ will be $\frac{s}{r}$ and the output of phase estimation $\hat{\varphi}$, will approximate this.

Possible problems with using this approach to find r :

1. We do not know how to construct initial state $|u_s\rangle$.
2. We do not know how to get r from $\varphi = \frac{s}{r}$.
3. We do not know how to compute U^j .

Initial State for Phase Estimation

Regarding problem 1, it turns out there is a trivial way to construct a suitable approximate initial state. Take the combination of all eigenvectors:

$$\begin{aligned}
\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle &= \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i s k / r} |x^k \pmod{N}\rangle \\
&= \frac{1}{r} \sum_{k=0}^{r-1} \sum_{s=0}^{r-1} e^{-2\pi i s k / r} |x^k \pmod{N}\rangle
\end{aligned}$$

Case $k = 0$, this simplifies to $|1\rangle$

Case $k > 0$, the coefficients for each output state are given by geometric series:

$$\frac{e^{-2\pi i k / r \cdot (r-1+1)} - 1}{e^{-2\pi i k / r} - 1} = \frac{1 - 1}{e^{-2\pi i k / r} - 1} = 0$$

(Geometric series

$$\begin{aligned}
\sum_{k=0}^n r^k &= r^0 + r^1 + \dots + r^n \\
(1-r) \sum_{k=0}^n r^k &= (r^0 + r^1 + \dots + r^n) - (r^1 + r^2 + \dots + r^{n+1}) \\
&= r^0 - r^{n+1} \\
\sum_{k=0}^n r^k &= \frac{1 - r^{n+1}}{1 - r}
\end{aligned}$$

So,

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = \frac{1}{r} |x^0 \pmod{N}\rangle = |1\rangle$$

The state $|1\rangle$ is equal to a combination of all the eigenvectors of U , and also happens to be extremely easy to implement, making it a good initial input for phase estimation. Note that the state $|1\rangle$ is L qubits long, expressed as $|0\dots 01\rangle$ in binary.

Showing P.E. with this initial state (following circuit diagram 5.3 again, page 223):

$$\begin{aligned}
|0\rangle^{\otimes t} |1\rangle &= |0\rangle^{\otimes t} \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle \\
&= \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |0\rangle^{\otimes t} |u_s\rangle \\
&\xrightarrow{H^{\otimes t} \otimes I} \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \frac{1}{2^{t/2}} \sum_{j=0}^{2^t-1} |j\rangle |u_s\rangle \\
&\xrightarrow{U^j} \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \frac{1}{2^{t/2}} \sum_{j=0}^{2^t-1} |j\rangle U^j |u_s\rangle \\
&= \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \frac{1}{2^{t/2}} \sum_{j=0}^{2^t-1} |j\rangle e^{2\pi i s j / r} |u_s\rangle \\
&= \sum_{s=0}^{r-1} \frac{1}{\sqrt{r}} \left(\frac{1}{2^{t/2}} \sum_{j=0}^{2^t-1} e^{2\pi i s j / r} |j\rangle \right) |u_s\rangle = |\psi\rangle \\
&\xrightarrow{F^H \otimes I} \sum_{s=0}^{r-1} \frac{1}{\sqrt{r}} \left(\sum_{\ell=0}^{2^t-1} g(\varphi_s, \ell) |\ell\rangle \right) |u_s\rangle
\end{aligned}$$

φ_s is just $\frac{s}{r}$. If $\varphi_s 2^t$ is an exact whole number, then $g(\varphi_s, \ell)$ is 1 when $\ell = \varphi_s 2^t$ and 0 otherwise.

In the general case, assuming initial state is $|u_s\rangle$, phase estimation produces an approximation with n bits accuracy satisfying $|\varphi - \hat{\varphi}| \leq 2^{-n}$ with probability $(1 - \epsilon)$. Since we are starting with state $|1\rangle$ instead of $|u_s\rangle$, the success probability is $\geq \left(\frac{1}{\sqrt{r}}\right)^2 (1 - \epsilon)$

This success probability, which is the probability of $\hat{\varphi}$ being close to $\frac{s}{r}$ for some specific value of s , is very small. In reality though, we don't care which value of s (which eigenvector) phase estimation returns the phase for, because we only care about the ratio $\frac{s}{r}$ which we can recover from any s . The success probability is high enough to allow this.

Lecture 21 (4 April)

Lecture 20 Review

Phase estimation for order finding

Initial state $|1\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle$ (L qubits long)

U matrix given by:

$$U|y\rangle = \begin{cases} |xy \bmod N\rangle & y = 0, 1, 2, \dots, N-1 \\ |y\rangle & y = N, N+1, \dots, 2^L-1 \end{cases}$$

where $L = \lceil \log_2 N \rceil$.

Accuracy needed is $n = 2L + 1$.

$$|\varphi_s - \hat{\varphi}_s| = \left| \frac{s}{r} - \hat{\varphi}_s \right| \leq 2^{-(2L+1)}$$
$$\lambda_s = e^{2\pi i s/r}$$

Success probability is $\frac{1}{r} (1 - \epsilon)$ for each *individual* s using $t = 2L + 1 + \lceil \log_2 (\frac{1}{2\epsilon} + 2) \rceil$

Recovering denominator of $\frac{s}{r}$ is impossible for individual numerator because success probability is too low. But overall, if we don't care about individual values of s , then we can get a high enough probability ratio.

Deriving order from estimated phase

Question 2 unanswered from last time: How to get r from $\hat{\varphi}_s$, assuming phase estimation succeeded, or

$$\left| \frac{s}{r} - \hat{\varphi}_s \right| \leq 2^{-(2L+1)}$$

for some s .

Theorem: If $\left| \frac{s}{r} - \hat{\varphi}_s \right| \leq \frac{1}{2r^2}$, then $\frac{s}{r}$ is a convergent of a continued fraction for φ , and can be computed from φ in $O(L^3)$ operations, using continued fraction algorithm. [Theorem 5.1, and A.4.16 p637]

It is not hard to satisfy condition needed to apply this theorem:

$$\begin{aligned} L = \lceil \log_2 N \rceil &\geq \log_2 N \\ 2L + 1 &\geq 2 \log_2 N + 1 = 2 \log_2 N + \log_2 2 = \log_2 (2N^2) \\ -(2L + 1) &\leq -\log_2 (2N^2) \\ 2^{-(2L+1)} &\leq 2^{-\log_2 (2N^2)} = 2^{\log_2 (\frac{1}{2N^2})} = \frac{1}{2N^2} \leq \frac{1}{2r^2} \\ |\varphi - \hat{\varphi}_s| &\leq \frac{1}{2r^2} \end{aligned}$$

Continued Fractions

Any real number x be represented as a sequence of integers $[a_0, a_1, \dots, a_n]$ which are terms in a continued fraction:

$$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\dots + \frac{1}{a_{n-1} + \frac{1}{a_n}}}}}$$

Example:

$$\begin{aligned}
 \frac{43}{18} &= 2 + \frac{7}{18} \\
 &= 2 + \frac{1}{\frac{18}{7}} \\
 &= 2 + \frac{1}{2 + \frac{4}{7}} \\
 &= 2 + \frac{1}{2 + \frac{1}{1 + \frac{3}{4}}} \\
 &= 2 + \frac{1}{2 + \frac{1}{1 + \frac{1}{1 + \frac{1}{3}}}}
 \end{aligned}$$

The sequence of a_i values can continue forever when x is an arbitrary real, but will end when x is a rational number because the sequence of remainders will be strictly decreasing, and the procedure for determining a_i terminates in \log (numerator or denominator). In the case of order finding, it converges in $O(t)$ or $O(L)$ steps.

Reason: The procedure divides when remainder is ≥ 2 and stops when remainder is 1 or 0. Since are dividing repeatedly by numbers which are ≥ 2 , it only takes $\log N$ iterations before reaching 0 or 1.

Continued fraction cost per step is t^2 for the division of a t bit number. This is $O(L^2)$. Total cost of the algorithm is the cost per steps times number of steps, or $O(L^3)$.

[More information on Continued Fraction Algorithm in Appendix p635-6, theorem A.4.15).

Continued Fractions as Simple Fractions

Given $[a_0, a_1, \dots, a_N]$ then $[a_0, a_1, \dots, a_n] = \frac{P_n}{Q_n}$ for $n < N$. Running continued fraction algorithm on $\hat{\varphi}$ at some point in the middle should give the fraction $\frac{s}{r}$.

(Examples of continued fractions made into simple fractions:

$$a_0 = \frac{a_0}{1}$$

$$a_0 + \frac{1}{a_1} = \frac{a_0 a_1 + 1}{a_1}$$

$$a_0 + \frac{1}{a_1 + \frac{1}{a_2}} = a_0 + \frac{a_2}{a_1 a_2 + 1} = \frac{a_0 a_1 a_2 + a_0 + a_2}{a_1 a_2 + 1}$$

$$a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3}}} = a_0 + \frac{1}{a_1 + \frac{1}{\frac{a_2 a_3 + 1}{a_2}}} = a_0 + \frac{a_2 a_3 + 1}{a_1 a_2 a_3 + a_1 + a_3} = \frac{a_0 a_1 a_2 a_3 + a_0 a_1 + a_0 a_3 + a_2 a_3 + 1}{a_1 a_2 a_3 + a_1 + a_3}$$

)

The following formulas give numerators and denominators without the need for all the manipulation above:

$$p_0 = a_0, q_0 = 1$$

$$p_1 = a_0 a_1 + 1, q_1 = a_1$$

$$p_n = a_n p_{n-1} + p_{n-2}, q_n = a_n q_{n-1} + q_{n-2}$$

Examples:

$[a_0]$	$a_0 = \frac{a_0}{1} = \frac{p_0}{q_0}$
$[a_0, a_1]$	$a_0 + \frac{1}{a_1} = \frac{a_0 a_1 + 1}{a_1} = \frac{p_1}{q_1}$
$[a_0, a_1, a_2]$	$a_0 + \frac{1}{a_1 + \frac{1}{a_2}} = \frac{a_0(a_1 + \frac{1}{a_2}) + 1}{a_1 + \frac{1}{a_2}} = \frac{p_1 + \frac{p_0}{a_2}}{\frac{a_1 a_2 + 1}{a_2}} = \frac{p_1 a_2 + p_0}{a_2 q_1 + q_0} = \frac{p_2}{q_2}$

Continued Fraction Examples

Ex 1: $\frac{9}{15} = 0 + \frac{1}{\frac{15}{9}}$, continue running with $\frac{15}{9}$

Ex 2: $0.333 = \frac{333}{1000} = 0 + \frac{1}{\frac{1000}{333}} = 0 + \frac{1}{3 + \frac{1}{333}}$

$$p_0 = a_0 = 0, q_0 = 1, \frac{p_0}{q_0} = 0$$

$$p_1 = a_0 a_1 + 1 = 1, q_1 = a_1 = 3, \frac{p_1}{q_1} = \frac{1}{3}$$

$$p_2 = a_2 p_1 + p_0 = 333 \cdot 1 + 0 = 333$$

$$q_2 = a_2 q_1 + q_0 = 333 \cdot 3 + 1 = 1000$$

$$\frac{p_2}{q_2} = \frac{333}{1000}$$

If we would have started over with $\frac{1000}{333} = 3 + \frac{1}{333}$, then:

$$p_0 = a_0 = 3, q_0 = 1, \frac{p_0}{q_0} = \frac{3}{1}$$

$$p_1 = 333 \cdot 3 + 1 = 1000, q_1 = 333 \cdot 1 = 333, \frac{p_1}{q_1} = \frac{1000}{333}$$

Order Finding Algorithm

Algorithm after phase estimation is classical. Get $\hat{\varphi} = [a_0, \dots, a_m]$ determine $\frac{p_n}{q_n}$, for $n = 0, \dots, m$. For each q_n , check if $x^{q_n} \stackrel{?}{=} 1 \pmod{N}$. If the equation is satisfied, then $r = q_n$.

The algorithm may fail in two separate cases:

1. If phase estimation fails
2. If $\gcd(s, r) > 1$. If this is the case, recovering the denominator of $\frac{s}{r}$ will give r divided by the gcd, instead of just r . $\frac{s}{r} = \frac{s'}{r'}$, $r' < r$, $x^{r'} \neq 1 \pmod{N}$

Can overcome these types of failures by repeating algorithm, but need to compute success probability to know how many times to repeat. Note that # of primes $< r$ is $\frac{r}{2 \log r}$

So the overall success probability is: $\frac{r}{2 \log r} \frac{1}{r} (1 - \epsilon) = \frac{1 - \epsilon}{2 \log r} \geq \frac{1 - \epsilon}{2 \log N}$ since $r < N$

($\frac{r}{2 \log r} \frac{1}{r}$ is the probability that s is prime, since s is less than r)

If we repeat $2 \log N$ times, then success probability of having at least 1 success is $\geq \left(1 - \left(1 - \frac{1 - \epsilon}{2 \log N}\right)^{2 \log N}\right)$

$$\approx 1 - e^{-\frac{1 - \epsilon}{2 \log N} (2 \log N)} = 1 - e^{-(1 - \epsilon)}$$

Lecture 22 (9 April)

Review: Order Finding

Used P.E. to get $\left|\frac{s}{r} - \hat{\varphi}\right| \leq 2^{-(2L-1)}$, $L = \lceil \log_2 N \rceil$

Use $\hat{\varphi}$ as an approximation to $\frac{s}{r}$ in continued fraction algorithm to get s, r individually in $O(L^3)$ ops. Test $x^r \stackrel{?}{=} 1 \pmod{N}$ with result. If true, then finished, otherwise repeat.

The probability that phase estimation succeeds and that s is prime is

$$\frac{r}{2 \log r} \frac{1 - \epsilon}{r} = \frac{1 - \epsilon}{2 \log r}$$

(where $\frac{r}{2 \log r}$ is number of primes $\leq r$)

$$\geq \frac{1 - \epsilon}{2 \log N}$$

$2 \log N$ rep will yield # succ ≥ 1 w/prob $1 - e^{-(1-\epsilon)}$

Repeating overall means $O(L^4) = O((\log N)^4)$

Modular Exponentiation

(iii) How to implement U^j , or how to compute $X^i \pmod{N}$. Use modular multiplication:

$$0 \leq j \leq 2^t - 1, j = \sum_{k=0}^{t-1} a_k 2^k, a_k \in \{0, 1\}$$

$$x^j = x^{\sum_{k=0}^{t-1} a_k 2^k} = x^{a_0} x^{a_1 2} \dots x^{a_{(n-1)} 2^{t-1}}$$

$$x^j \pmod{N} = \left(x^{a_0} \dots x^{a_{(n-1)} 2^{t-1}} \right)$$

Ex: $x^5 \pmod{N} = x^{2^2+1} \pmod{N}$

$$= (x^2 \pmod{N} x^2 \pmod{N} x \pmod{N}) \pmod{N}$$

$$N = 11, x = 5$$

$$5^5 \pmod{N} = ((5^2 \pmod{N}) (5^2 \pmod{N})) (5 \pmod{N})$$

$$3 \cdot 3 \cdot 5 \pmod{11} = 1 \pmod{N}$$

P.E. $U, U^2, U^4, U^8, U^{2^t-1} O(1)$

$$x, x^2, x^4, x^8$$

$$x, x \cdot x, x^2 \cdot x^2, x^4 \cdot x^4$$

Why does modular multiplication work

$$x_1 = \ell_1 N + r_1$$

$$x_2 = \ell_2 N + r_2$$

$$x_1 x_2 \pmod{N} = (\ell_1 \ell_2 N^2 + \ell_1 r_2 N + \ell_2 r_1 N + r_1 r_2) \pmod{N} = r_1 r_2 \pmod{N}$$

If U has different forms, not always easy to compute powers. If it were, quantum could easily solve NP complete problems. Example of not easy U :

$$U = e^{iA}, U^j = e^{ijA}$$

Factoring

Shor's algorithm

Given composite number N , which we want to express as $m \cdot n$. Application: cryptography, where the problem is assumed to be hard. Traditional algorithm: number field sieve with time $2^{C(\log N)^{1/3}(\log \log N)^{2/3}}$, c is const > 0 .

Related to order finding.

Definitions:

N - composite number

x, N - coprime, $1 < x < N$

r - order $x^r = 1 \pmod{N}$

$L - \lceil \log_2 N \rceil$ - bits to represent N

Theorem 1: N composite, x is a nontrivial solution of $x^2 = 1 \pmod{N}$. Nontrivial means $1 < x < N - 1$.

$$\begin{aligned}x^2 - 1 &= (x - 1)(x + 1) \\x &\not\equiv 1 \pmod{N} = 1 \\x &\not\equiv -1 \pmod{N} = N - 1\end{aligned}$$

N should not divide $(x - 1)$ and $(x + 1)$. If true, at least one of $\gcd(x - 1, N)$ or $\gcd(x + 1, N)$ is a non-trivial factor, and can be computed in $O(L^3)$ operations using euclid's algorithm.

Proof

$$x^2 - 1 = \ell N, \text{ equivalently, } (x - 1)(x + 1) = \ell N$$

$$1 < x < N - 1$$

$$x - 1 < N - 2 < N - 1$$

N does not divide $(x + 1)$

Tells you N and $x - 1$ or $x + 1$ must have a common factor.

Take $\gcd(x - 1, N)$ and $\gcd(x + 1, N)$, done.

How to compute $\gcd(a, b)$ with a, b positive L -bit integers:

1. If $a < b$, $\gcd(a, b) = \gcd(b, a)$, find $\max(a, b)$ and make it a
2. $a = \ell b + r_1$

$$\gcd(a, b) = \gcd(b, r_1)$$

$$\text{if } r_1 = 0 \text{ } \gcd(a, b) = b$$

$$\text{if } r_1 = 1 \text{ } \gcd(a, b) = 1$$

else $r_1 \neq (0, 1)$ repeat

Gives a sequence of remainders $b = \ell_2 r_1 + r_2$

Example 1

$$\gcd(6825, 1430) = 65$$

$$6825 = 4 \cdot 1430 + 1105$$

$$1430 = 1 \cdot 1105 + 325$$

$$1105 = 3 \cdot 325 + 130$$

$$325 = 2 \cdot 130 + 65$$

$$130 = 2 \cdot 65 + 0$$

Example 2

$$\gcd(7, 4) = 1$$

$$7 = 1 \cdot 4 + 3$$

$$4 = 1 \cdot 3 + 1$$

$$3 = 3 \cdot 1 + 0$$

How many steps: decreasing sequence of r_k , dividing by numbers ≥ 2 , so $O(L)$ steps. Cost per step is cost per division is $O(L^2)$. Total cost: $O(L^3)$.

Theorem 2: (no proof given). N is composite and odd and $N = p_1^{a_1} p_2^{a_2} \dots p_m^{a_m}$, p_i are primes, of which there are m many. Choose $x \in \{1, \dots, N-1\}$ uniformly at random. If r is order of $x \pmod{N}$, so $x^r = 1 \pmod{N}$, then probability $\{r \text{ even and } x^{r/2} \neq -1 \pmod{N}\} \geq 1 - \frac{1}{2^m}$.

$$x^{r/2} + 1 + \ell N.$$

$$\text{if } n = 2 \text{ prob} > 1 - \frac{1}{2^2} = \frac{3}{4}$$

$$\text{even if } n = 1 \text{ prob} > \frac{1}{2}$$

If you can verify solution, even probabilities $< \frac{1}{2}$ are ok, just repeat. As long as probability isn't exponentially tiny, verification is all you need to get away with tiny probabilities.

Reduction of factoring to order finding

High level summary, steps later

Choose random x and find r .

$$x^r = 1 \pmod{N}$$

if r is even = $(x^{r/2})^2$ and you can use theorem 2

$$\text{return gcd}(x^{r/2} - 1, N) \text{ or } \text{gcd}(x^{r/2} + 1, N)$$

Wednesday: details, grover's algorithm

Lecture 23 (11 April)

Theorem 1: $x^2 = 1 \pmod{N}$

$$(x-1)(x+1) = \ell N \text{ gcd}(x-1, N) \text{ or } \text{gcd}(x+1, N)$$

$$x \neq \pm 1 \pmod{N}$$

Ex 1: $N = 15$, $x = 4$, $x^2 = 16 = 1 \pmod{15}$

$$(x-1)(x+1) = 15$$

$$\text{gcd}(3, 15) \times \text{gcd}(5, 15) = \ell N$$

Both gcd's are factors

Ex 2: $N = 12$, $x = 7$, $x^2 = 49 = 1 \pmod{12}$

$$(x-1)(x+1) = 6 \cdot 8 = 48 = 4 \cdot 12$$

$$\ell = 4, N = 12$$

$$\text{gcd}(6, 12) = 6, \text{gcd}(8, 12) = 4$$

$6 \cdot 4 \neq 12$, both gcd's are not factors

Therefore take one gcd, get another factor by division. Look at pages 15-16 of schor's papers.

Reduction to order finding

Choose $x \in \{1, \dots, N-1\}$

Find order r , $x^r = 1 \pmod{N}$, use theorem

If r is even $(x^{r/2} - 1)(x^{r/2} + 1) = \ell N$

then $\gcd(x^{r/2} - 1, N)$ or $\gcd(x^{r/2} + 1, N)$

there are constraints, we cover them later

assuming r is even, assuming not a trivial solution (as indicated by theorem). In even case, you know $x^{r/2} \not\equiv \pm 1 \pmod{N}$ because r is order, order is smallest r .

Theorem: If N is odd and composite and factors as $N = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_n^{\alpha_n}$ p_n primes and pick one $X = \{1, \dots, N-1\}$ uniformly at random. $\Pr\{r \text{ even and } x^{r/2} \not\equiv \pm 1 \pmod{N}\} \geq 1 - \frac{1}{2^n}$

Shor Factoring Algorithm

Input: Composite number N

Output: Factor of N

Runtime: $O(L^3) = O((\log N)^3)$

Probability success $\geq \frac{3}{4}$

Steps:

1. If N even output 2
2. Determine if $N = a^b$, for some $a \geq 1$, $b \geq 2$. If so, determine n and stop.
3. Uniformly chose $x \in \{1, \dots, N-1\}$. If $\gcd(x, N) > 1$ return $\gcd(x, N)$.
4. Only quantum step. Use order finding algorithm, obtain r order of $x \pmod{N}$.
5. If r is even and $x^{r/2} \not\equiv \pm 1 \pmod{N}$, then return $\gcd(x^{r/2} - 1, N)$ or $\gcd(x^{r/2} + 1, N)$ doesn't matter which. Otherwise algorithm fails.

Remarks

Step 1: Step 1 is easy

Step 2: How to check $N = a^b$ for $a \geq 3$, $b \geq 2$, we know $b \leq \log N \leq L$

$b = \{2, 3, 4, \dots, L\}$

Check if $a^b = N$ for each b and integer a . Book uses algorithm to find a , but you can use binary search. Bisection isn't so bad because only need sign information, don't need to compute whole powers. Cost of bisection in $\log N = L$ steps, error $\leq \frac{1}{2^L}$, cost per step is cost of repeated querying $O(\log \log N)$. Total cost $O(\log^2 N \log \log N)$

Alternative: Newton's method

$$x_{i+1} = x_i - \frac{f(x)}{f'(x)} = x_i - \frac{x_i^{-b} - N}{-bx_i^{b-1}}$$

$$|x_i - N^{1/b}| = O(|x_i - N^{1/b}|^2)$$

Quadratic convergence, taking as few steps as needed.

After steps 1 and 2 know x is odd and has more than one prime factor.

Combine w/ theorem 2 to see that $\text{prob} \geq \frac{3}{4} = 1 - \frac{1}{2^2}$. 2 is number of factors ($2 > 1$).

Step 3: Use euclid algorithm with cost $O(L^3) = O(\log^3 N)$ to get $\gcd(x, N)$. if > 1 stop, else continue knowing x, N are coprime.

Step 4: x, N coprime, so order finding. Cost $O(L^3)$ or $O(L^4)$, depending which way you do, not significant. Gives r , so $x^r = 1 \pmod{N}$.

Theorem 2: $Pr \{r \text{ even and } x^{r/2} \neq -1 \pmod{N}\} \geq 1 - \frac{1}{2^2} = \frac{3}{4}$.

Step 5: Check r even

$x^{r/2} \neq -1 \pmod{N}$?

If so, $\gcd(x^{r/2} - 1, N)$ or $\gcd(x^{r/2} + 1, N)$

Example:

$N = a(13 \times 7)$

1. Not even

2. Not $N \neq a^b$

3. Say $x = 4$, $\gcd(4, 9) = 1$ coprime

4. Order of $x = 4 \pmod{91} = 1$

$r = 6$, $4^6 = 2^{12} = 4096 = 1 \pmod{91}$

$4096 = 45 \cdot 91 + 1$

5. $r = 6$ even

$x^{r/2} = 4^3 = 2^6 = 64 = 64 \pmod{91} \neq -1 \pmod{91}$

$\gcd(63, 91) = \gcd(63, 28) = \gcd(28, 7) = 7$

$\gcd(65, 91) = \gcd(65, 26) = \gcd(26, 13) = 13$

Both of them are not factors, but they are both divisors.

Search, counting algorithms

Grover's algorithms

Boolean mean, Brassard et al (amplitude, amplification, estimations)

Applications many problems science / engineering, integration, approximation, path integrations, differential equations (Schrodinger eqn).

Lecture 24 (16 April)

Searching / Counting Algorithm

Given a function

$$f : \{0, 1, \dots, N - 1\} \rightarrow \{0, 1\}$$

as in Deutsch-Josza, find which inputs produce an output.

$N = 2^n$, N is huge

We use queries, or oracle calls

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

$|x\rangle$ is n qubits, $|y\rangle$ is 1 qubit.

Using $H|1\rangle$ as an input:

$$\begin{aligned} Q_f |x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} &= \frac{1}{\sqrt{2}} (|x\rangle |0 \oplus f(x)\rangle - |x\rangle |1 \oplus f(x)\rangle) \\ &= \frac{1}{\sqrt{2}} (|x\rangle |f(x)\rangle - |x\rangle |\overline{f(x)}\rangle) \end{aligned}$$

if $f(x) = 0$ then $|x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$.

if $f(x) = 1$ then $|x\rangle \frac{|1\rangle - |0\rangle}{\sqrt{2}}$.

$$Q_f |x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} = (-1)^{f(x)} |x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

With this input, the $|y\rangle$ qubit is unchanged, there is just an overall phase shift. This is more convenient for analysis than the general $Q_f |x\rangle |y\rangle$ function.

Search

$$\begin{aligned} M_f &= \{x : f(x) = 1\} \\ M &= |M_f| \geq 1 \end{aligned}$$

Problem is to find elements of M_f . This is a reverse lookup problem, like searching an unordered database. Unlike the quantum solution for the factoring problem, the quantum solution for this problem is not exponentially faster, just polylog. Another difference is that this quantum solution beats the upper bound of the equivalent classical solution, whereas the quantum factoring algorithm only beats known classical algorithms.

Related Problem: Boolean Mean

$$S(f) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) = \frac{M}{N}$$

Classical Search Algorithms

1. Deterministic Algorithm. Lower bound $O(N)$, because you may have to evaluate N times before search succeeds.
2. Randomized Algorithms. (See paper Beals et al). Lower bound is also $O(N)$. No proof, but basic idea follows.

Algorithm:

Choose x uniformly at random with replacement.

If $f(x) = 1$ stop with success.

If $f(x) = 0$ fail.

Repeat k times.

Probability of failure first time is $1 - \frac{M}{N}$. If $M = 1$, $1 - \frac{1}{N} > C$.

Probability to fail in k trials is $(1 - \frac{M}{N})^k \leq \delta$. Set desired tolerance to δ .

$$k \log \left(1 - \frac{M}{N} \right) = \log \delta$$

$$k = \frac{\log d}{\log\left(1 - \frac{M}{N}\right)} \approx \frac{\log d}{-\frac{M}{N}} = \frac{N}{M} \log \frac{1}{d}$$

Approximation holds when $M \ll N$. In this case, algorithm is $O(N)$, in general case algorithm is $O\left(\frac{N}{M}\right)$.

General hint: $\log(1+x) \approx x$ when x is tiny. This is used frequently in complexity analysis, and is based on the power series expansion.

When algorithm is run without replacement, probability of failure is

$$\frac{\binom{N-M}{k} \binom{M}{0}}{\binom{N}{k}} = \frac{\binom{N-1}{k}}{\binom{N}{k}} = \frac{(N-1)!}{k!(N-k-1)!} = \frac{N-k}{N} = 1 - \frac{k}{N}$$

this is bounded by a constant unless k is $O(N)$

Repeating

$$\left(1 - \frac{k}{N}\right)^S \leq \delta$$

$$S = \frac{N}{k} \log \frac{1}{\delta} \Rightarrow S_k = N \log \frac{1}{\delta} = O(N)$$

Quantum Search Algorithm

Grover's Algorithm, $O\left(\sqrt{\frac{N}{M}}\right)$ counting the number of queries of Q_f .

Algorithm: Given some initial state $|\psi_0\rangle$. Apply operator then query, then operator, repeating as necessary.

$$|\psi_1\rangle = Q_F U_T Q_F U_{T-1} \dots U_2 Q_F U_1 |\psi_0\rangle$$

using T queries.

Boolean Mean

$$S(f) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) = \frac{M}{N}$$

Classical Algorithms

1. Deterministic Algorithm. Find lower bound on number of evaluations given error tolerance ϵ . To do this we need to ensure

$$\max_f |S(f) - \hat{S}(f)| \leq \epsilon$$

For k evaluations, assume $f(x) = 0$ to get lower bound, we then know that $0 \leq S(f) \leq \frac{M-k}{N}$. Take an estimate at the middle of this range to get least possible error in worst case:

$$\hat{S}(f) = \frac{0 + \left(1 - \frac{k}{N}\right)}{2}$$

In this case

$$\max_f |S(f) - \hat{S}(f)|^2 = \frac{1}{2} \left(1 - \frac{k}{N}\right)$$

$$\begin{aligned} \frac{1}{2} \left(1 - \frac{k}{N} \right) &< \epsilon \\ N - k &\leq 2N\epsilon \\ k &\geq N(1 - 2\epsilon) \end{aligned}$$

Lecture 25 (18 April)

Searching + Counting

Classical Algorithms, Deterministic and Random $O(N)$, $1 \leq M \ll N$.

Quantum Algorithm $O\left(\sqrt{\frac{N}{M}}\right)$

Counting / Boolean Mean

Classical deterministic algorithm: $k \geq N(1 - 2\epsilon)$

Classical randomized algorithm: Choose k inputs at random computing

$$\hat{S}(f) = \frac{1}{k} \sum_{i=1}^k f(x_i)$$

$$k : x_i \in \{0, \dots, N-1\}$$

This is a Monte Carlo algorithm. Instead of finding the lower bound error, find the expected error

$$\left(E_{x_1 \dots x_k} [S(f) - \hat{S}(f)]^2 \right)^{1/2} \leq \frac{1}{\sqrt{k}} < \epsilon$$

$$k = \min \left(\frac{1}{\epsilon^2}, N \right)$$

Chebyshev inequality

$$Pr \left\{ |S(f) - \hat{S}(f)| > \tau \right\} \leq \frac{E \left((S(f) - \hat{S}(f))^2 \right)}{\tau^2}$$

$$\tau = 2\epsilon$$

$$Pr \left\{ |S(f) - \hat{S}(f)| > 2\epsilon \right\} \leq \frac{E \left((S(f) - \hat{S}(f))^2 \right)}{4\epsilon^2}$$

$$k = \min \left(\frac{1}{\epsilon^2}, N \right)$$

Quantum Algorithm

Provides provable polynomial speedup over classical randomized algorithm. This is unlike factoring where there is no known lower bound and we are beating known classical algorithms without any proof that there isn't an unknown classical algorithm which could be faster.

Grover's Search

1. Apply oracle Q_f (from Deutsch-Jozsa) or the nice oracle O_f (from previous lecture), $O_f |x\rangle = (-1)^{f(x)} |x\rangle$.
2. Apply $H^{\otimes n}$
3. Apply phase shift Q_0 which is a reflection about $|0\rangle$

$$\begin{aligned} Q_0 &= 2|0\rangle\langle 0| - I \\ Q_0 |0\rangle &= 2|0\rangle\langle 0|0\rangle - I|0\rangle = |0\rangle \\ Q_0 |y\rangle &= 2|0\rangle\langle 0|y\rangle - I|y\rangle = -|y\rangle, y \neq 0 \end{aligned}$$

so for any basis state $|y\rangle$

$$Q_0 |y\rangle = (-1)^{\delta_{0y}} |y\rangle \quad \forall y = 0, \dots, N-1$$

4. Apply $H^{\otimes n}$

All 4 operations can be combined with one operator

$$\begin{aligned} G &= H^{\otimes n} (2|0\rangle\langle 0| - I) H^{\otimes n} O_f \\ &= (2H^{\otimes n}|0\rangle\langle 0|H^{\otimes n} - H^{\otimes n}H^{\otimes n}) O_f \\ &= (2|\psi\rangle\langle\psi| - I) O_f \end{aligned}$$

where

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle$$

$2|\psi\rangle\langle\psi| - I$ acts as a reflection about $|\psi\rangle$ because for $|z\rangle \perp |\psi\rangle$

$$\begin{aligned} (2|\psi\rangle\langle\psi| - I)|\psi\rangle &= |\psi\rangle \\ (2|\psi\rangle\langle\psi| - I)|z\rangle &= -|z\rangle \end{aligned}$$

Analysis

$$M_f = \{x : f(x) = 1\}, 1 \leq M = |M_f| \leq N-1$$

Define two states $|\alpha\rangle$ and $|\beta\rangle$ so

$$\begin{aligned} |\alpha\rangle &= \frac{1}{\sqrt{N-M}} \sum_{x \notin M_f} |x\rangle \\ |\beta\rangle &= \frac{1}{\sqrt{M}} \sum_{x \in M_f} |x\rangle \end{aligned}$$

$\| |\alpha\rangle \| = \| |\beta\rangle \| = 1$ and $\langle \alpha | \beta \rangle = 0$

$$\begin{aligned} |\psi\rangle &= \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle \\ &= \frac{1}{\sqrt{N}} \left(\sqrt{N-M} |\alpha\rangle + \sqrt{M} |\beta\rangle \right) \\ &= \frac{\sqrt{N-M}}{\sqrt{N}} |\alpha\rangle + \frac{\sqrt{M}}{\sqrt{N}} |\beta\rangle \end{aligned}$$

Idea of algorithm is to boost change initial state $|\psi\rangle$ boosting the amplitude of $|\beta\rangle$ so the state that is measured will likely be in M_f .

$$\begin{aligned}
|\psi\rangle &= \langle\alpha|\psi\rangle|\alpha\rangle + \langle\beta|\psi\rangle|\beta\rangle \\
\langle\alpha|\psi\rangle &= \frac{1}{\sqrt{N-M}} \frac{1}{\sqrt{N}} \sum_{x \notin M_f} \sum_{y=0}^{N-1} \langle x|y\rangle \\
&= \frac{1}{\sqrt{N-M}} \frac{1}{\sqrt{N}} \sum_{x \notin M_f} 1 \\
&= \frac{1}{\sqrt{N-M}} \frac{1}{\sqrt{N}} (N-M) \\
&= \sqrt{\frac{N-M}{N}}
\end{aligned}$$

Similarly for $|\beta\rangle$. Because $\| |\psi\rangle \|^2 = \frac{N-M}{N} \| |\alpha\rangle \|^2 + \frac{M}{N} \| |\beta\rangle \|^2 = 1$, $|\psi\rangle$ can be written as

$$|\psi\rangle = \cos \frac{\vartheta}{2} |\alpha\rangle + \sin \frac{\vartheta}{2} |\beta\rangle$$

where

$$\begin{aligned}
\cos \frac{\vartheta}{2} &= \sqrt{\frac{N-M}{N}} \\
\sin \frac{\vartheta}{2} &= \sqrt{\frac{M}{N}}
\end{aligned}$$

Powers of G

Want to determine $G|\psi\rangle, G^2|\psi\rangle, G^3|\psi\rangle, \dots, G^k|\psi\rangle$

$$\begin{aligned}
G &= (2|\psi\rangle\langle\psi| - I)O_f \\
O_f|\alpha\rangle &= \frac{1}{\sqrt{N-M}} \sum_{x \notin M_f} O_f|x\rangle \\
&= \frac{1}{\sqrt{N-M}} \sum_{x \notin M_f} (-1)^{f(x)=1} |x\rangle \\
&= |\alpha\rangle \\
O_f|\beta\rangle &= \frac{1}{\sqrt{M}} \sum_{x \in M_f} O_f|x\rangle \\
&= \frac{1}{\sqrt{M}} \sum_{x \in M_f} (-1)^{f(x)=1} |x\rangle \\
&= -|\beta\rangle
\end{aligned}$$

So when operator O_f is applied to $|\alpha\rangle$, it is unchanged. When the operator is applied to $|\beta\rangle$ it is reflected. Given this, we can see G applied to a state creates a rotation of that state in the $|\alpha\rangle, |\beta\rangle$ plane. Previously we showed expressed $|\psi\rangle$ in terms of an angle $\frac{\vartheta}{2}$

$$|\psi\rangle = \cos \frac{\vartheta}{2} |\alpha\rangle + \sin \frac{\vartheta}{2} |\beta\rangle$$

allowing the state $|\psi\rangle$ to be represented graphically as a 2 dimensional vector in the $|\alpha\rangle, |\beta\rangle$ plane oriented $\frac{\vartheta}{2}$ degrees above the $|\alpha\rangle$ axis. Applying O_f to this state negates $|\beta\rangle$ component reflecting it about the $|\alpha\rangle$ axis

resulting in a vector $\frac{\vartheta}{2}$ degrees below the $|\alpha\rangle$ axis. Applying $2|\psi\rangle\langle\psi| - I$ is the same as a reflection about $|\psi\rangle$. This results in a state oriented $\frac{\vartheta}{2} + (\frac{\vartheta}{2} - -\frac{\vartheta}{2}) = \frac{3\vartheta}{2}$ degrees above the $|\alpha\rangle$ axis. ($\frac{\vartheta}{2}$ being the angle of the $|\psi\rangle$ state relative to $|\alpha\rangle$, $(\frac{\vartheta}{2} - -\frac{\vartheta}{2})$ being the angular distance between $|\psi\rangle$ and $O_f|\psi\rangle$). The result of the two reflections is

$$G|\psi\rangle = \cos\frac{3\vartheta}{2}|\alpha\rangle + \sin\frac{3\vartheta}{2}|\beta\rangle$$

which taken together are equivalent to a rotation by 2ϑ . Applying the G operator repeatedly is then equivalent to

$$G^k|\psi\rangle = \cos\left(\frac{2k+1}{2}\vartheta\right)|\alpha\rangle + \sin\left(\frac{2k+1}{2}\vartheta\right)|\beta\rangle$$

when a good value for k is chosen, $\sin\left(\frac{2k+1}{2}\vartheta\right)$ will be close to 1 and measuring $G^k|\psi\rangle$ will yield a state from the $|\beta\rangle$ superposition with high probability. Recall that any state in $|\beta\rangle$ is a solution to the search problem. More formally, for a measurement $|x\rangle$ on the computational basis

$$Pr\{x \in M_f\} = \sin^2\left(\frac{2k+1}{2}\vartheta\right)$$

Lecture 26 (23 April)

Grover's Review

$$\begin{aligned} G &= H^{\otimes n} (2|0\rangle\langle 0| - I) H^{\otimes n} O_f \\ &= (2|\psi\rangle\langle\psi| - I) O_f \\ O_f|x\rangle &= (-1)^{f(x)}|x\rangle \\ |\alpha\rangle &= \frac{1}{\sqrt{N-M}} \sum_{x \notin M_f} |x\rangle \\ |\beta\rangle &= \frac{1}{\sqrt{M}} \sum_{x \in M_f} |x\rangle \\ |\psi\rangle &= \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle = H^{\otimes n} |0\rangle^{\otimes n} \\ G^k|\psi\rangle &= \cos\left(\frac{2k+1}{2}\vartheta\right)|\alpha\rangle + \sin\left(\frac{2k+1}{2}\vartheta\right)|\beta\rangle \\ \sin\left(\frac{\vartheta}{2}\right) &= \sqrt{\frac{M}{N}} \\ \cos\left(\frac{\vartheta}{2}\right) &= \sqrt{\frac{M-N}{N}} \end{aligned}$$

Grover's Analysis (continued)

Need to find value of k that gives $G^k |\psi\rangle$ close to $|\beta\rangle$. Start with success probability of algorithm assuming k is known. For $x \in M_f$

$$\begin{aligned}
 \langle x | G^k | \psi \rangle &= \left| \sin \left(\frac{2k+1}{2} \vartheta \right) \langle x | \beta \rangle \right|^2 \\
 &= \left| \sin \left(\frac{2k+1}{2} \vartheta \right) \frac{1}{\sqrt{M}} \right|^2 \\
 &= \sin^2 \left(\frac{2k+1}{2} \vartheta \right) \frac{1}{M} \\
 Pr \{ \text{measuring } x \in M_f \} &= M \sin^2 \left(\frac{2k+1}{2} \vartheta \right) \frac{1}{M} \\
 &= \sin^2 \left(\frac{2k+1}{2} \vartheta \right) \frac{1}{M}
 \end{aligned}$$

Success probability is therefore $\sin^2 \left(\frac{2k+1}{2} \vartheta \right)$ given a k . $\frac{2k+1}{2} \vartheta$ should be close to $\frac{\pi}{2}$, so

$$\frac{\pi}{2} - \frac{\vartheta}{2} \leq \frac{2k+1}{2} \vartheta < \frac{\pi}{2} + \frac{\vartheta}{2}$$

If $\frac{M}{N} < \frac{1}{2}$ then $\sqrt{\frac{M}{N}} = \sin \frac{\vartheta}{2} < \sin \frac{\pi}{4} = \sqrt{\frac{1}{2}}$ and $0 \leq \vartheta \leq \frac{\pi}{2}$

$$\begin{aligned}
 \pi - \vartheta &\leq (2k+1) \vartheta < \pi + \vartheta \\
 \frac{\pi}{\vartheta} - 1 &\leq 2k+1 < \frac{\pi}{\vartheta} + 1 \\
 \frac{\pi}{2\vartheta} - 1 &\leq k < \frac{\pi}{2\vartheta}
 \end{aligned}$$

$\sqrt{\frac{M}{N}} = \sin \frac{\vartheta}{2} \approx \frac{\vartheta}{2}$, a valid approximation when ϑ is tiny. In this case $\varphi \approx 2\sqrt{\frac{M}{N}}$, and $k = \frac{\pi}{2\vartheta} \approx \frac{\pi}{4} \sqrt{\frac{N}{M}}$, so a good k can be set $k = \left\lceil \frac{\pi}{4} \sqrt{\frac{N}{M}} \right\rceil$, making the complexity of the algorithm in terms of oracle calls $O(N)$.

When M is larger, you need a more precise analysis without the approximation. In this case use

$$\begin{aligned}
 \sin \frac{\vartheta}{2} &= \sqrt{\frac{M}{N}} \\
 \cos \frac{\vartheta}{2} &= \sqrt{\frac{N-M}{N}}
 \end{aligned}$$

Then use a trig identity

$$\begin{aligned}
 \sin \vartheta &= 2 \sin \frac{\vartheta}{2} \cos \frac{\vartheta}{2} = 2 \sqrt{\frac{M}{N}} \sqrt{\frac{N-M}{N}} \\
 \varphi &= \arcsin \left(2 \sqrt{\frac{M(N-M)}{N^2}} \right) \\
 k &= \frac{\pi}{2\vartheta}
 \end{aligned}$$

To see what happens when $\frac{M}{N} \geq \frac{1}{2}$ look at the relation

$$\sin^2 \vartheta = \frac{4M(N-M)}{N^2} = 4 \frac{M}{N} \left(1 - \frac{M}{N} \right)$$

and note that ϑ gets smaller as M increases from $\frac{N}{2}$ to N . Because ϑ gets smaller, k gets larger, and algorithm gets slower to the point where it is not better than a random classical algorithm.

Grover's Algorithm Circuit

$$\text{Circuit: } (G^k) (H^{\otimes n} \otimes I) \left(|0\rangle^{\otimes n} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)$$

(Diagram Figure 6.1, Page 251)

$$\begin{aligned} |0\rangle^{\otimes t} \frac{|0\rangle - |1\rangle}{\sqrt{2}} &\xrightarrow{H^{\otimes n} \otimes I} |\psi\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} \\ &\xrightarrow{G^k} \left(\cos\left(\frac{2k+1}{2}\vartheta\right) |\alpha\rangle + \sin\left(\frac{2k+1}{2}\vartheta\right) |\beta\rangle \right) \frac{|0\rangle - |1\rangle}{\sqrt{2}} \end{aligned}$$

Estimating M

Grover assumes M is known. If M is unknown, there are two approaches.

1. Brassard, et al. Apply G^j for random j , and show expected number of steps is $O\left(\sqrt{N/M}\right)$.
2. Find M using $\sin \frac{\vartheta}{2} = \sqrt{\frac{M}{N}}$ and estimating ϑ . Since this also gives you the boolean mean it lets you “hit 2 birds for the price of one.” We cover this second approach.

$$\begin{aligned} G &= (2|\psi\rangle\langle\psi| - I) O_f \\ G|\alpha\rangle &= \cos(\vartheta) |\alpha\rangle + \sin(\vartheta) |\beta\rangle \\ G|\beta\rangle &= \cos\left(\vartheta + \frac{\pi}{2}\right) |\alpha\rangle + \sin\left(\vartheta + \frac{\pi}{2}\right) |\beta\rangle \\ &= -\sin(\vartheta) |\alpha\rangle + \cos(\vartheta) |\beta\rangle \end{aligned}$$

The effect of G on the $|\alpha\rangle$ vector above is determined by looking at reflections and rotations on the $|\alpha\rangle, |\beta\rangle$ plane (Figure 6.3, page 253). Applying O_f to $|\alpha\rangle$ does not change anything because as shown earlier, O_f is a reflection about the $|\alpha\rangle$ axis. Applying $(2|\psi\rangle\langle\psi| - I)$ to $|\alpha\rangle$ reflects the state about $|\psi\rangle$. Since $|\psi\rangle$ is $\frac{\vartheta}{2}$ degrees above $|\alpha\rangle$, reflecting $|\alpha\rangle$ about it is equivalent to rotating the state by $2 \cdot \frac{\vartheta}{2} = \vartheta$ degrees. The resulting vector has magnitudes of $\cos(\vartheta)$ in the $|\alpha\rangle$ direction and $\sin(\vartheta)$ in the $|\beta\rangle$ direction. $G|\beta\rangle$ is derived similarly. The relation above can be used to express G as a transformation on the $|\alpha\rangle, |\beta\rangle$ basis:

$$G = \begin{pmatrix} \cos \vartheta & -\sin \vartheta \\ \sin \vartheta & \cos \vartheta \end{pmatrix}$$

The eigenvalues of G are given by

$$\begin{aligned} 0 &= \begin{vmatrix} \cos \vartheta - \lambda & -\sin \vartheta \\ \sin \vartheta & \cos \vartheta - \lambda \end{vmatrix} \\ &= (\cos(\vartheta) - \lambda)^2 + \sin^2(\vartheta) \\ (\cos(\vartheta) - \lambda)^2 &= -\sin^2(\vartheta) \\ \cos(\vartheta) - \lambda &= \pm i \sin(\vartheta) \\ \lambda &= \cos(\vartheta) \pm i \sin(\vartheta) \\ &= e^{\pm i\vartheta} \end{aligned}$$

Phase estimation is performed on G to find ϑ , which can in turn be used to find M and k .

Lecture 27 (25 April)

Review: Grover's Algorithm

$$G^k, k = \frac{\pi}{2\vartheta}, \sin \frac{\vartheta}{2} = \sqrt{\frac{M}{N}}$$

When M is small, the following approximation for k is valid: $k \approx \left\lceil \frac{\pi}{4} \sqrt{\frac{M}{N}} \right\rceil$

Otherwise k can be found using $\vartheta = \arcsin\left(2\sqrt{\frac{M(N-M)}{N^2}}\right)$.

If $M = 1$, $k = O(\sqrt{N})$

If M is unknown, can use estimate of ϑ to find it. Estimating ϑ happens with phase estimation on G , which expressed as a transformation on the $|\alpha\rangle, |\beta\rangle$ basis looks like

$$G = \begin{pmatrix} \cos \vartheta & -\sin \vartheta \\ \sin \vartheta & \cos \vartheta \end{pmatrix}$$

and has eigenvalues, $\lambda_{\pm} = e^{\pm i\vartheta}$.

Phase Estimation for ϑ

Phase estimation requires us to generate an initial state which approximates one or more eigenvectors of G .

Finding Eigenvectors of G

Denote unknown eigenvector as $(x|\alpha\rangle + y|\beta\rangle)$ so:

$$G(x|\alpha\rangle + y|\beta\rangle) = e^{\pm i\vartheta}(x|\alpha\rangle + y|\beta\rangle)$$

Expanding the left side gives:

$$xG|\alpha\rangle + yG|\beta\rangle = x(\cos \vartheta|\alpha\rangle + \sin \vartheta|\beta\rangle) + y(-\sin \vartheta|\alpha\rangle + \cos \vartheta|\beta\rangle)$$

Which is true when $x \cos \vartheta - y \sin \vartheta = xe^{\pm i\vartheta}$ and $x \sin \vartheta + y \cos \vartheta = ye^{\pm i\vartheta}$

$$\begin{aligned} x \cos \vartheta - y \sin \vartheta &= xe^{\pm i\vartheta} \\ x \cos \vartheta - y \sin \vartheta &= x(\cos \vartheta \pm i \sin \vartheta) \\ -y &= \pm ix \text{ when } \sin \vartheta \neq 0, \frac{\vartheta}{2} \neq k\frac{\pi}{2} \end{aligned}$$

Which gives eigenvector of the form $x|\alpha\rangle \pm ix|\beta\rangle$. Normalized, the eigenvectors are

$$\begin{aligned} |\psi_+\rangle &= \frac{|\alpha\rangle + i|\beta\rangle}{\sqrt{2}} \\ |\psi_-\rangle &= \frac{|\alpha\rangle - i|\beta\rangle}{\sqrt{2}} \end{aligned}$$

$|\alpha\rangle$ and $|\beta\rangle$ can be rewritten as

$$\begin{aligned} |\alpha\rangle &= \frac{1}{\sqrt{2}}(|\psi_+\rangle + |\psi_-\rangle) \\ |\beta\rangle &= \frac{i}{\sqrt{2}}(|\psi_+\rangle - |\psi_-\rangle) \end{aligned}$$

Generating Initial State

$|\psi\rangle$ can be used as an initial state because it is a combination of eigenvectors:

$$\begin{aligned}
 |\psi\rangle &= \frac{1}{\sqrt{N}} \sum_x |x\rangle \\
 &= \cos\left(\frac{\vartheta}{2}\right) |\alpha\rangle + \sin\left(\frac{\vartheta}{2}\right) |\beta\rangle \\
 &= \cos\left(\frac{\vartheta}{2}\right) \frac{1}{\sqrt{2}} (|\psi_+\rangle + |\psi_-\rangle) + \sin\left(\frac{\vartheta}{2}\right) \frac{i}{\sqrt{2}} (|\psi_+\rangle - |\psi_-\rangle) \\
 &= \frac{1}{\sqrt{2}} \left(\cos\left(\frac{\vartheta}{2}\right) + i \sin\left(\frac{\vartheta}{2}\right) \right) |\psi_+\rangle + \frac{1}{\sqrt{2}} \left(\cos\left(\frac{\vartheta}{2}\right) - i \sin\left(\frac{\vartheta}{2}\right) \right) |\psi_-\rangle \\
 &= \frac{1}{\sqrt{2}} e^{i\frac{\vartheta}{2}} |\psi_+\rangle + \frac{1}{\sqrt{2}} e^{-i\frac{\vartheta}{2}} |\psi_-\rangle
 \end{aligned}$$

Note: Coefficients to $|\psi_+\rangle$ and $|\psi_-\rangle$ above are not important. Any state which was a combination of the two eigenvectors would work for finding the boolean mean.

Phase Estimation Results

The result of phase estimation with G and initial state $|\psi\rangle$ can be used to find boolean mean

$$S(f) = \frac{1}{N} \sum_x f(x) = \frac{M}{N} = \sin^2\left(\frac{\vartheta}{2}\right)$$

Phase estimation gives $|\varphi - \hat{\varphi}| \leq 2^{-n_0}$, $\hat{\varphi} = \frac{d}{2^t}$ with probability $(1 - \epsilon)$ using $t = \eta_0 + \lceil \log(2 + \frac{1}{2\epsilon}) \rceil$ qubits in the top register. Phase estimation will approximate λ_+ with probability

$$(1 - \epsilon) \left| \frac{1}{\sqrt{2}} e^{i\frac{\vartheta}{2}} \right|^2$$

and approximate λ_- with probability

$$(1 - \epsilon) \left| \frac{1}{\sqrt{2}} e^{-i\frac{\vartheta}{2}} \right|^2$$

Eigenvalues again are

$$\begin{aligned}
 \lambda_+ &= e^{i\vartheta} = e^{2\pi i\vartheta/(2\pi)} \\
 \lambda_- &= e^{i(2\pi-\vartheta)} = e^{-2\pi i(2\pi-\vartheta)/(2\pi)} \\
 \varphi_+ &= \frac{\vartheta}{2\pi} \\
 \varphi_- &= \frac{2\pi - \vartheta}{2\pi}
 \end{aligned}$$

Error bounds look like

$$\begin{aligned}
 |\varphi_{\pm} - \hat{\varphi}| &\leq 2^{-\eta_0} \\
 |\pi\varphi_{\pm} - \pi\hat{\varphi}| &\leq \frac{\pi}{2^{\eta_0}} \\
 \pi\varphi_+ &= \frac{\vartheta}{2} \\
 \pi\varphi_- &= \pi - \frac{\vartheta}{2} \\
 \left| \frac{\vartheta}{2} - \pi\hat{\varphi} \right| &\leq \frac{\pi}{2^{\eta_0}} \text{ w/prob } \frac{1}{2}(1 - \epsilon) \\
 \left| \pi - \frac{\vartheta}{2} - \pi\hat{\varphi} \right| &\leq \frac{\pi}{2^{\eta_0}} \text{ w/prob } \frac{1}{2}(1 - \epsilon)
 \end{aligned}$$

But we aren't using phase estimation to compute phase, we are using it to compute M which is related to the sin, so we need a different bound:

$$\left| \sin^2\left(\frac{\vartheta}{2}\right) - \sin^2\left(\frac{\pi j}{2^t}\right) \right| \leq \frac{\pi}{2^{\eta_0}} \sqrt{S(f)(1 - S(f))} + \frac{\pi^2}{2^{2\eta_0}}$$

(from paper BHMT lemma 7, page 15)

j is measurement in computational basis

since $\sin^2\left(\frac{\varphi}{2}\right) = \sin^2\left(\pi - \frac{\varphi}{2}\right) = S(f)$

$$\left. \begin{aligned}
 \sin^2\left(\frac{\vartheta}{2}\right) &\approx \frac{\vartheta}{2} \\
 \sin^2\left(\frac{\pi j}{2}\right) &\approx \frac{\pi j}{2}
 \end{aligned} \right\} \rightarrow \left(\frac{\vartheta}{2}\right)^2 - \left(\frac{\pi j}{2}\right)^2 = \left(\frac{\vartheta}{2} + \frac{\pi j}{2}\right)\left(\frac{\vartheta}{2} - \frac{\pi j}{2}\right)$$

Phase Estimation Circuit

The circuit for using phase estimation to compute the boolean mean is the same as the circuit for normal phase estimation. (Figure 6.7 page 262)

The output of the circuit in the bottom register will be either $|\psi_+\rangle$ or $|\psi_-\rangle$. The top register needs just enough bits to be able to distinguish $\frac{1}{N}$ from 0.

In general, there is no efficient way to make G^{2^j} gates, just have to apply G repeatedly, so the number of queries is $\tau = 2^{t-1} = \Theta(2^t) = \Theta(2^{\eta_0})$. Error is $O\left(\frac{1}{\tau}\right)$. $t = \eta_0 + \lceil \log_2\left(2 + \frac{1}{2\epsilon}\right) \rceil$. This is the best possible, see paper [Nayak+Wu], algorithm is optimal.

Lecture 28 (30 April)

Review for Final