Last class we discussed one generalization of Shor's algorithm to the Hidden Subgroup Problem. Today we discuss another generalization due to Alexei Kitaev (1995) of the Solovay-Kitaev theorem fame (his name will pop up again later with the similarly seminal Kitaev toric code). This generalization is so pervasive the KLM textbook we roughly follow presents Shor's algorithm in this framework. The generalization I'm talking about is

Eigenvalue (or phase) estimation
The idea behind the eigenvalue estimation problem is given a unitary operator \( U \) and eigenvector \( |u\rangle \) such that

\[
U|u\rangle = \lambda |u\rangle
\]

Compute or otherwise estimate \( \lambda \). This is often called phase estimation because of the following fact:

(Unitary eigenvalues have norm 1)

Let \( U \) be a unitary operator with eigenvalues \( \lambda_1, \ldots, \lambda_k \).

Then \( |\lambda_i| = 1 \) for all \( i \). In polar coordinates,

\[
\lambda_i = e^{2\pi i w_i}
\]

**Proof**

Let \( U|u_i\rangle = \lambda_i |u_i\rangle \). Then

\[
<u_i|u_i> = <u_i|U^\dagger U|u_i> = (U|u_i\rangle)^\dagger (U|u_i\rangle) = (\lambda_i |u_i\rangle)^\dagger (\lambda_i |u_i\rangle) = |\lambda_i|^2 <u_i|u_i>
\]

Which implies that \( |\lambda_i|^2 = 1 \)

---

**Eigenvalue estimation problem**

**Input**: A unitary \( U \) & eigenvector \( |u\rangle \) of \( U \).

**Goal**: Estimate \( \lambda \) such that \( U|u\rangle = e^{2\pi i w} |u\rangle \)

Before we discuss why we might want to solve the eigenvalue estimation problem (spoiler alert—quantum chemistry) let's think about how to do it.
Eigenvalue estimation is based on phase kick-back:

\[
\begin{align*}
\frac{10\gamma + 11\gamma}{\sqrt{2}} & \quad \frac{10\gamma + e^{2\pi i/11}\gamma}{\sqrt{2}} \\
\downarrow & \quad \text{U} \\
\downarrow & \quad 1\gamma
\end{align*}
\]

we previously used phase kick-back in Deutsch's algorithm. In particular, suppose \(U\upsilon\gamma = e^{2\pi i/11}\upsilon\gamma\). Then the above circuit implements the transformation

\[
c - U\left(\frac{10\gamma + 11\gamma}{\sqrt{2}} \otimes \upsilon\gamma\right) = c - U\left(\frac{10\gamma \upsilon\gamma + 11\gamma \upsilon\gamma}{\sqrt{2}}\right)
\]

\[
= \frac{10\gamma \upsilon\gamma + 11\gamma \upsilon\gamma}{\sqrt{2}}
\]

\[
= \frac{10\gamma \upsilon\gamma + e^{2\pi i/11} \upsilon\gamma}{\sqrt{2}}
\]

\[
= \left(\frac{10\gamma + e^{2\pi i/11}\gamma}{\sqrt{2}}\right) \otimes \upsilon\gamma
\]

In effect the phase becomes associated with the control bit as a relative phase, hence the phase imparted on \(\upsilon\gamma\) by \(U\) is "kicked" into the control.

**Ex.**

Suppose \(U\) has eigenvalues \(\pm 1\). Given a vector \(\upsilon\gamma\) promised to be an eigenvector of \(U\), how can we determine its eigenvalue?

\[
\begin{align*}
\text{\text{10\gamma}} & \quad \text{H} \\
\text{\text{11\gamma}} & \quad \text{H} \\
\downarrow & \quad \text{U} \\
\downarrow & \quad \upsilon\gamma
\end{align*}
\]

\[
\begin{align*}
\{10\gamma\} & \text{ if } U\upsilon\gamma = 1\upsilon\gamma \\
\{11\gamma\} & \text{ if } U\upsilon\gamma = -1\upsilon\gamma
\end{align*}
\]

The above circuit uses phase kickback to produce the state \(\frac{10\gamma \upsilon\gamma + 11\gamma \upsilon\gamma}{\sqrt{2}}\) = \(\{1+1\gamma\} \text{ if } U\upsilon\gamma = 1\upsilon\gamma \text{ and then } \{1-1\gamma\} \text{ if } U\upsilon\gamma = -1\upsilon\gamma\)

measures in the hadamard basis. This is exactly Deutsch's algorithm! Recall that \(1\gamma = \frac{10\gamma - 11\gamma}{\sqrt{2}}\) in Deutsch's algorithm which is a \(-1\) eigenvector of \(X\) and a \(+1\) eigenvector of \(I\).

If \(f\) is constant then \(C - U = C - I\) and if \(f\) is balanced then \(C - U = C - X\) (both up to a local \(X\) on the eigenvector).
(Estimating a binary phase)

Suppose we want to estimate the eigenphase

$$e^{2\pi i \frac{x}{2^n}} = e^{2\pi i \lambda x}$$

of \(U_1|u\rangle\) where \(x \in [0, 2^n-1]\) is a binary integer.

We know that the QFT\(_{2^n}\) maps \(|x\rangle \rightarrow \sum_{y \in \mathbb{Z}_{2^n}} e^{2\pi i \frac{xy}{2^n}} |y\rangle\)

so the inverse should send

$$\sum_{y \in \mathbb{Z}_{2^n}} e^{2\pi i \frac{xy}{2^n}} |y\rangle \rightarrow |x\rangle$$

Our goal then is to prepare the state \(\sum_{y \in \mathbb{Z}_{2^n}} e^{2\pi i \frac{xy}{2^n}} |y\rangle\).

Letting \(w = \frac{x}{2^n}\) we have

$$yw = y_n w + y_{n-1} w + \ldots + y_1 w + y_0 w$$

$$= y_n w + y_{n-1}(2w) + \ldots + y_1(2^{n-1} w)$$

So if we had unitaries \(U_1 \ldots U_n\) and eigenvectors \(|u_1\rangle, \ldots, |u_n\rangle\) such that \(U_i |u_i\rangle = e^{2\pi i \frac{x_i}{2^n}} |u_i\rangle\) then we could prepare a register \(\frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{Z}_{2^n}} |y\rangle\) and apply phase kick-back to apply each term.

Moreover, we can take \(|u_i\rangle = |u\rangle\) and \(U_i = U_i^{2^i}\) since

$$U_i^{2^i} |u\rangle = \left(e^{2\pi i \frac{x_i}{2^n}}\right)^{2^i} |u\rangle = e^{2\pi i \frac{x_i 2^i}{2^n}} |u\rangle$$
Hence the following circuit solves the eigenvalue estimation problem in this case.

\[ QFT^{-1}, \quad \cdots, \quad 10^\uparrow, \quad 10^\uparrow, \quad \ldots, \quad 10^\uparrow, \quad 10^\uparrow \]

(The general case)

Suppose now that the eigenphase \( w \) has an infinite (or unknown) size representation in binary. That is,

\[ w = 0.x_1x_2 \ldots \]

We can round \( w \) to \( \tilde{w} \) with \( n \) bits of precision and say \( \tilde{w} = 0.x_1x_2 \ldots x_n = \frac{w}{2^n} \). It turns out that the above circuit gives us the \( n \)-digit approximation \( \tilde{w} \) with high probability for the same reason that Shor's algorithm worked with high probability.

(Error analysis)

First note that \( |w - \tilde{w}| \leq \frac{1}{2^n} \), and \( w = \tilde{w} \pm \varepsilon \), \( \varepsilon \leq \frac{1}{2^n} \).

Now as in Shor's algorithm,

\[ QFT^{-1}_{2^n} \left( \frac{1}{\sqrt{2^n}} \sum_y e^{2\pi i wy} |y\rangle \right) = \frac{1}{2^n} \sum_y e^{2\pi i (wy - \frac{\varepsilon}{2^n})} |y\rangle \]

Consider the case when \( z = x = 2^n \tilde{w} \). Then

\[ \frac{1}{2^n} \sum_y e^{2\pi i (wy - y \tilde{w})} = \frac{1}{2^n} \sum_y e^{2\pi i y(w - \tilde{w})} = \frac{1}{2^n} \sum_y e^{2\pi i y\varepsilon} \]

This is the picture from the Shor analysis that we did—summing up the first \( 2^n \) \( 2^{n+1} \)th roots of unity and taking the average. It's a bit easier this time however since everything is a power of \( 2 \tilde{w} \).
Theorem

The phase estimation algorithm produces an estimate

\[ x = 2^n \tilde{\omega} \]

which is within \( \frac{1}{2^n} \) of \( \omega \) with probability \( \frac{4}{\pi^2} \). With probability \( \frac{8}{\pi^2} \approx 0.81 \) it will be within \( \frac{1}{2^n} \).

(Connection to Shor's algorithm)

You may have noticed that phase estimation looks a lot like Shor's period finding algorithm—in fact exactly the same except for the initial state of the target register. The connection is in the eigenvalues of the modular multiplication

\[ U_q |x\rangle = |x \cdot a \mod M\rangle \]

Remember that in Shor's algorithm, we needed to find \( r \) such that \( a^r \equiv 1 \mod M \), and hence

\[ U_q^r |x\rangle = |x \cdot a^r \mod M\rangle = |x\rangle \]

The following fact establishes that the eigenvalues of \( U_q \) are \( r \)-th roots of unity.

(Eigenvalues of order \( r \) unitaries)

Let \( U \) be a unitary operator such that \( U^r = I \). Then the eigenvalues of \( U \) are of the form

\[ e^{\frac{2\pi i k}{r}} = \omega_r^k \]

Proof

Let \( \lambda \) be an eigenvalue of \( U \) with eigenvector \( |u\rangle \). Then

\[ U |u\rangle = \lambda |u\rangle = |u\rangle, \quad \text{so} \quad \lambda^r = 1 \]

and by definition must be an \( r \)-th root of unity.
As a consequence of the above, if we prepared the target register in some state \( |u> \) which is an \( e^{2\pi i \frac{1}{n}} \)-eigenvector of \( U_0 \), the circuit which is identical to the phase estimation circuit (modulating the irrelevant choice of \( QFT_n \) or \( QFT_n^{-1} \)) would produce an estimate of \( \frac{1}{\lambda} \) as in our original analysis.

However, we don't know (and can't prepare) \( |u> \) a priori. A key insight is that the input state \( |u> \) doesn't matter if we don't care which eigenvalue we estimate.

**(Eigenbasis)**

By the Spectral theorem, the eigenvectors of a unitary \( U \) form a basis of the Hilbert space on which it acts. This is called the eigenbasis of \( U \).

**(Phase estimation without a known eigenvector)**

Let \( U \) be a unitary which we wish to estimate an eigenvalue of. The following circuit produces an estimate of some eigenvalue of \( U \).
The key is to write \( |0 \rangle = \sum_j \alpha_j |u_j \rangle \) where \( \{ |u_j \rangle \} \) is the eigenbasis of \( U \). So, before the final QFT we have a linear combination of eigenphased states

\[
\sum_j \alpha_j (\sum_i \lambda_i x_i |x \rangle U^* |u_i \rangle) = \sum_j \alpha_j (\sum_i \lambda_i^x x_i |x \rangle |u_i \rangle)
\]

where \( U |u_i \rangle = \lambda_i |u_i \rangle \).

In the case of Shor's algorithm, since all eigenvalues of \( U_a \) take the form of \( \frac{2^k}{r} \) where \( r \) is the period, performing this phase estimation amounts to sampling from

\[
\frac{2^k}{r}, \frac{2^k}{2r}, \frac{2^k}{3r}, \frac{2^k}{4r}, \ldots
\]

And so is equivalent to the Shor-style analysis of period finding.

**Shor picture:**

\[ | \begin{array}{c} \frac{\frac{2^k}{r}}{x} \ \frac{\frac{2^k}{2r}}{x} \ \frac{\frac{2^k}{3r}}{x} \ \frac{\frac{2^k}{4r}}{x} \end{array} \rangle \]

**Kitaev picture:**

\[ | \begin{array}{c} \frac{\frac{2^k}{r}}{x} \end{array} \rangle + | \begin{array}{c} \frac{\frac{2^k}{2r}}{x} \end{array} \rangle + | \begin{array}{c} \frac{\frac{2^k}{3r}}{x} \end{array} \rangle \]
Many, many practical problems ultimately reduce
to finding eigenvalues or vectors. Google’s seminal
\textbf{PageRank} algorithm is one such example, as are
some problems in machine learning. Theoretically,
phase estimation allows the sampling of eigenvalues
of an \(N \times N\) matrix \(A\) in time \(O(\log(N))\). Does
this mean quantum computers can perform eigenvalue
calculations exponentially faster for large \(O(2^n)\)
matsices?

\textbf{In general, NO}.

There are a few roadblocks preventing speed-up in
general:

\begin{itemize}
  \item The matrix \(A\) must be unitary
        (rules out \textbf{PageRank})
  \item An arbitrary \(2^n \times 2^n\) unitary takes exponentially
        many gates to approximate
        (full \textbf{ST} theorem is exponential in \(n\))
  \item We can’t get an \textbf{eigenvector} which is the
        more useful thing to have in many cases
        (we can do measurements on the eigenvector though)
\end{itemize}

\textbf{(The HHL algorithm)}
\textbf{Harrow, Hassidim, Lloyd}

The HHL algorithm uses phase estimation (along
with other techniques we haven’t discussed yet)
to solve a linear system
\[
A \tilde{x} = \tilde{b}
\]
in roughly \(O(\log(1/\epsilon))\) time.
This algorithm is one of the reasons some believe that Quantum machine learning will lead to speed-ups, but it’s important to remember the above points—while unitarity turns out to not be very important, to implement $A$ efficiently it needs to be

1. Sparse
2. Low condition

Even worse is the fact that the vector $\hat{b}$ needs to be encoded in a quantum state $|\hat{b}\rangle$, called the QRAM assumption, for which there is no known efficient method in general. On top of that, even if $A$ & $\hat{b}$ are efficient to implement, the last point above still stands and we can only measure $\hat{x}$.

These conditions have led many (rightfully so) to be skeptical of the impacts of HHL and other provably superpolynomial speedups to Machine Learning. That isn’t to say it’s impossible and there are many QML proposals being explored currently, but we should approach the QML hype with a healthy skepticism until we have provable theoretical or practical results.

So, if eigenvalue estimation is maybe not that useful for classical linear algebra, what is it useful for?

Quantum chemistry has entered the chat...
To understand applications of the QPE algorithm to quantum chemistry, we need to go back to the physics for a moment.

In modern quantum theory, the time evolution of a physical system is governed by its Hamiltonian $\hat{H}$. The Hamiltonian is (for our purposes) a Hermitian operator

$$\hat{H}^+ = \hat{H}$$

The evolution of a system with Hamiltonian $\hat{H}$ over time $t$ is governed by the famous Schrödinger equation

$$i \frac{\hat{\psi}}{\hat{t}} (\psi(t)) = \hat{H} \psi(t)$$

In the case where $\hat{H}$ is independent of time, we can solve Schrödinger's equation as

$$\psi(t) = e^{-i \hat{H} t} \psi(0)$$

The operator $e^{-i \hat{H} t}$ turns out to be a unitary operator $U(t)$, as we could have guessed, and so

$$\psi(t) = U(t) \psi(0)$$

Recall that by the spectral theorem, $\hat{H}$ is diagonalizable as $\hat{H} = P \Lambda P^+$ where $P$ is unitary and $\Lambda$ is a diagonal matrix encoding $\hat{H}$'s eigenvalues. Hence

$$U(t) = P e^{-i \Lambda t} P^+ = P \left[ e^{-i \lambda_1 t} \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{pmatrix} \right] P^+$$
Hamiltonians & Energy

The Hamiltonian represents the total energy of a system, for instance the sum of the kinetic and potential energy from high school physics. The eigenvalues of \( \hat{H} \) are the possible energies of the system, owing to the fact that energy is quantized.

Ground State

The smallest eigenvalue of \( \hat{H} \) is called the system’s ground state energy, and an eigenvector for this \( \lambda \) is called a ground state. Higher-energy eigenvectors are known as excited states. As an eigenstate, systems stay in a particular energy state until energy enters or dissipates through interaction with the environment.

A simple example is a stationary rock:

![Image of a stationary rock]

It will stay stationary until someone comes along and expends some of their energy to lift it, importing some of that energy to the rock as potential energy.

![Image of a person pushing a rock]

If the person lets go, the rock prefers the low energy state, so it will fall, dissipating its potential energy into the environment.
Atoms also prefer low-energy states, like the ground state where every electron is in its lowest orbital.

An electron jumps to a higher orbital when it absorbs a photon, increasing the energy and pushing the atom into an excited state. As the atom prefers to be in a low energy state, the electron may spontaneously fall back to the unexcited state, releasing the extra energy in the form of a photon.

(Quantum chemistry)

(ground state energy estimation)

An important step in understanding chemical processes like Nitrogen fixation, which is used in fertilizer production and currently accounts for 3-5% of the worldwide natural gas usage, is to understand the ground state energies of the molecules involved. This is computationally intractable to do exactly as the Hamiltonians scale exponentially with the number of electrons. Approximation methods exist but are generally not accurate enough for much of quantum chemistry and materials science.
If however we could simulate the molecule on a quantum computer by way of its unitary evolution operator

\[ e^{-i\hat{H}t} = U(t) \]

Then we could apply the phase estimation algorithm to sample an eigenvalue \( e^{2\pi i w t} \) of \( U(t) \). Now \( w \) is an eigenvalue of \( \hat{H} \) so we can efficiently estimate eigenvalues of \( \hat{H} \). Moreover, if we can prepare an initial state which is close to a ground state, we can estimate the ground state energy. While challenging, we have methods that generally work, which has led many in the field to believe that quantum chemistry will be the first practical application of large-scale quantum computers.

**Next class**

How do we implement \( U(t) = e^{-i\hat{H}t} \)?