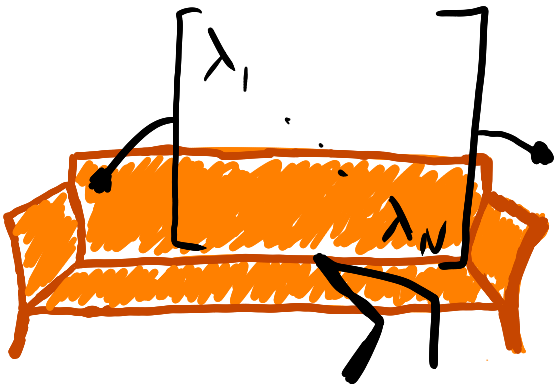
CMPT Lecture 23

Eigenvalue estimation



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 λ. 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

$$\langle u_i|u_i\rangle = \langle u_i|U^\dagger U|u_i\rangle = (U|u_i\rangle)^\dagger (U|u_i\rangle)$$
$$= (\lambda_i|u_i\rangle)^\dagger (\lambda_i|u_i\rangle)$$
$$= |\lambda_i|^2 \langle u_i|u_i\rangle$$

Which implies that $|\lambda_i|^2 = 1$ □

Each eigenvalue is a kth root of unity

If a unitary matrix has an order of r, then it means that it has rth roots of unity.

Eigenvalue estimation problem

input: A unitary U & eigenvector $|u\rangle$ of U.
goal: Estimate w such that $U|u\rangle = e^{2\pi i w}$

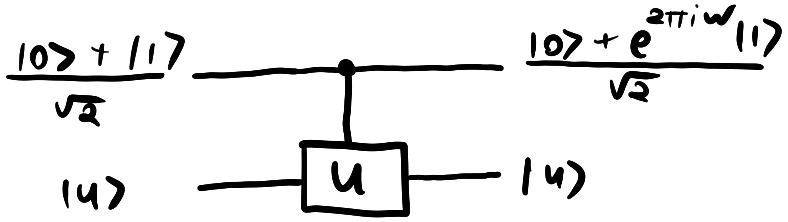
Before we discuss why we might want to solve the eigenvalue estimation problem (spoiler alert - quantum chemistry) let's think about how to do it

If we use shor's algorithm for this, we will get k, which tells us the order? Ask the professor about this.

If we have a unitary operator, then the eigenvectors of the operator forms the basis for the Hilbert space we are approximating over.

(Phase kick-back)

Eigenvalue estimation is based on **phase kick-back**:



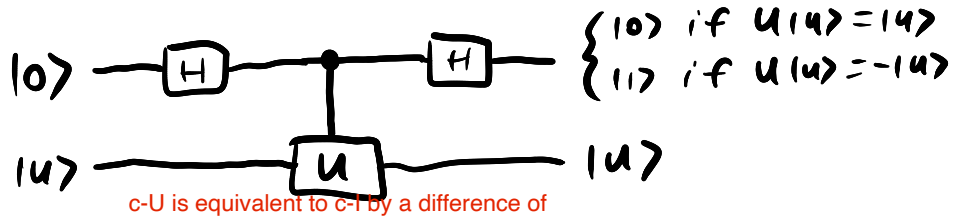
we previously used phase kick-back in Deutsch's algorithm. In particular, suppose $U|u\rangle = e^{2\pi i w}|u\rangle$. Then the above circuit implements the transformation

$$c\text{-}U\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}} \otimes |u\rangle\right) = c\text{-}U\left(\frac{|0\rangle|u\rangle + |1\rangle|u\rangle}{\sqrt{2}}\right)$$
$$= \frac{|0\rangle|u\rangle + |1\rangle(U|u\rangle)}{\sqrt{2}}$$
$$= \frac{|0\rangle|u\rangle + e^{2\pi i w}|1\rangle|u\rangle}{\sqrt{2}}$$
$$= \left(\frac{|0\rangle + e^{2\pi i w}|1\rangle}{\sqrt{2}}\right) \otimes |u\rangle$$

In effect the phase becomes associated with the control bit as a **relative phase**, hence the phase imparted on $|u\rangle$ by U is "kicked" into the control.

Ex.

Suppose U has eigenvalues ± 1. Given a vector $|u\rangle$ promised to be an eigenvector of U, how can we determine its eigenvalue?



c-U is equivalent to c-I by a difference of a global phase?

The above circuit uses phase kickback to produce the state $\frac{|0\rangle|u\rangle + |1\rangle(U|u\rangle)}{\sqrt{2}} = \begin{cases} |+\rangle|u\rangle & \text{if } U|u\rangle = |u\rangle \\ |-\rangle|u\rangle & \text{if } U|u\rangle = -|u\rangle \end{cases}$ and then measures in the hadamard basis. This is exactly Deutsch's algorithm! Recall that $|u\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$ in Deutsch's algorithm which is a -1 eigenvector of X and a $+1$ eigenvector of I. If f is constant then $c\text{-}U = c\text{-}I$ and if f is balanced then $c\text{-}U = c\text{-}X$ (both up to a local X on the eigenvector).

Ask professor about this: How can I have this up to a local X on the eigenvector? Should I not apply a local X and change the eigenvector?

(Estimating a binary phase)

Suppose we want to estimate the eigenphase

$$e^{2\pi i \frac{x}{2^n}} = e^{2\pi i 0.x_1 x_2 \cdots x_n}$$

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

We know that the QFT_{2^n} maps $|x\rangle \to \sum_{y \in \mathbb{Z}_{2^n}} e^{2\pi i \frac{xy}{2^n}} |y\rangle$

so the inverse should send

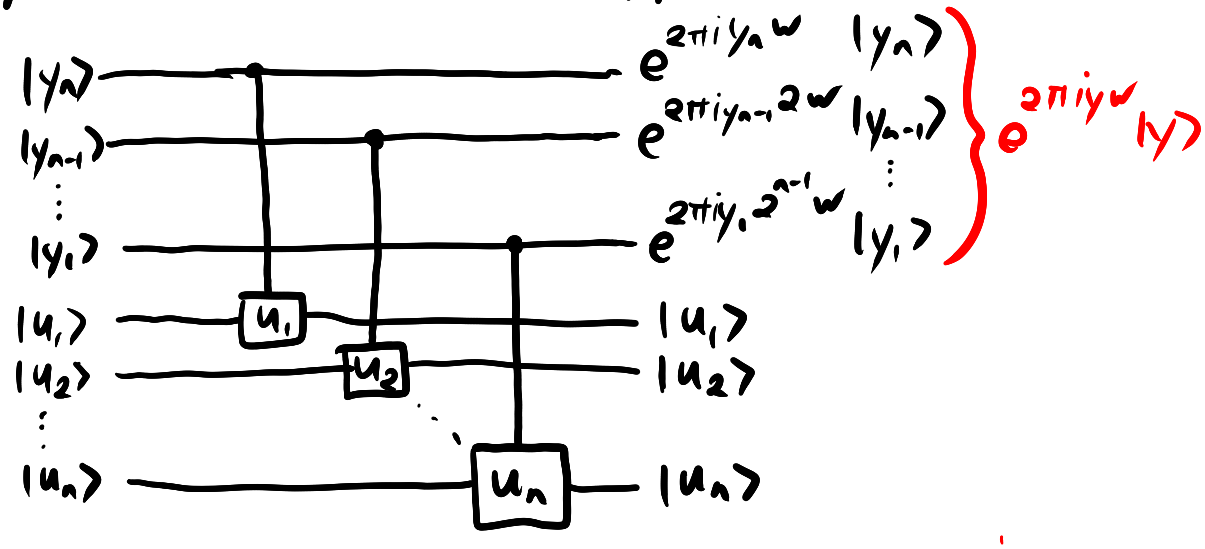
$$\sum_{y \in \mathbb{Z}_{2^n}} e^{2\pi i \frac{x}{2^n} \cdot y} |y\rangle \to |x\rangle$$

Our goal then is to prepare the state $\sum_{y \in \mathbb{Z}_{2^n}} e^{2\pi i \frac{x}{2^n} \cdot y} |y\rangle$

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

$$\begin{aligned} yw &= y_n w + 2y_{n-1} w + \cdots + 2^{n-1} y_1 w \\ &= y_n w + y_{n-1}(2w) + \cdots + y_1 (2^{n-1} w) \end{aligned}$$

So if we had unitaries $U_1 \cdots U_n$ and eigenvectors $|u_1\rangle \cdots |u_n\rangle$ such that $U_i |u_i\rangle = e^{2\pi i 2^i w} |u_i\rangle$ 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.

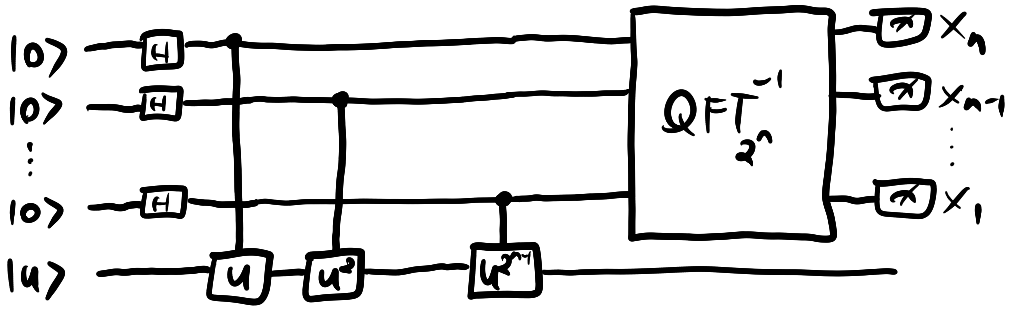


Moreover, we can take $|u_i\rangle = |u\rangle$ and $U_i = U^{2^i}$ since

This u is a gate and the other u is a ket?

$$U^{2^i} |u\rangle = \left(e^{2\pi i w}\right)^{2^i} |u\rangle = e^{2\pi i 2^i w} |u\rangle$$

Hence the following circuit solves the eigenvalue estimation problem in this case



QFT|k> gives you sum

QFT|k> gives us the exponent of the eigenvector thought of as an nth root of unity.

(The general case)

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

$$w = 0.x_1x_2\cdots$$

We can round w to $\tilde{w}$ with n bits of precision and say $\tilde{w} = 0.x_1x_2\cdots x_n = \frac{x}{2^n}$. 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.

Should this not be bounded by 1/(2^n)?

Could I not have that every nth + i bit, where i is greater than 0, is flipped on?

(Error analysis)

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

Now as in Shor's algorithm,

$$QFT_{2^n}^{-1}\left(\frac{1}{\sqrt{2^n}}\sum_y e^{2\pi i w y}|y\rangle\right) = \frac{1}{2^n}\sum_{y,z} e^{2\pi i\left(wy - \frac{yz}{2^n}\right)}|z\rangle$$

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 ☺.

Theorem

The phase estimation algorithm produces an estimate

$$x = 2^n \tilde{w}$$

which is within $\frac{1}{2^{n+1}}$ of w with probability $\frac{4}{\pi^2}$.
With probability $8/\pi^2 \approx 0.81$ 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

$$U_a |x\rangle = |x \cdot a \bmod M\rangle$$

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

$$U_a^r |x\rangle = |x \cdot a^r \bmod M\rangle = |x\rangle$$

The following fact establishes that the eigenvalues of U_a are r-th roots of unity.

For a unitary with order 2^m, its eigenvalues are 2^mth 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

$$e^{2\pi i \frac{k}{r}} = w_r^k$$

Proof

Let λ be an eigenvalue of U with eigenvector $|u\rangle$.
Then $U^r|u\rangle = \lambda^r |u\rangle = |u\rangle$, so $\lambda^r = 1$ 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 $|u\rangle$ which is an $e^{2\pi i \frac{k}{r}}$-eigenvector of U_a, the circuit which is identical to the phase estimation circuit (modulo the irrelevant choice of QFT_{2^n} or $QFT_{2^n}^{-1}$) would produce an estimate of k/r as in our original analysis.

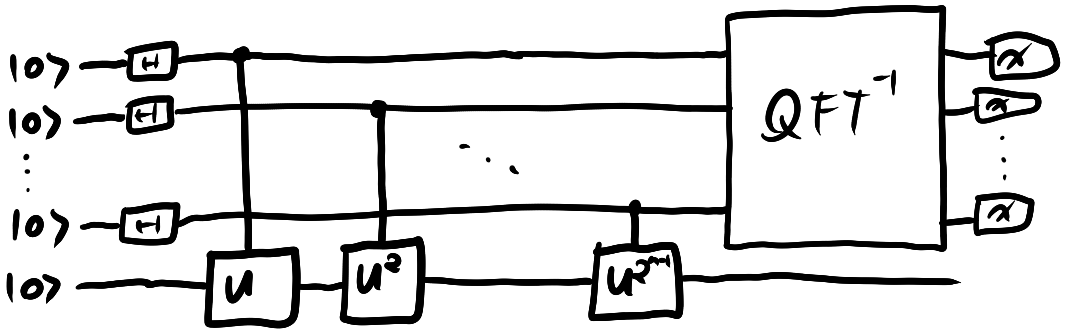
<u>However</u>, we don't know (and can't prepare) $|u\rangle$ a priori. A key insight is that the input state $|u\rangle$ doesn't matter if we don't care which eigenvalue we estimate.

(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 $|0\rangle = \sum_i \alpha_i |u_i\rangle$ where $\{|u_i\rangle\}$ is the eigenbasis of U. So, before the final QFT we have a linear combination of eigenphased states

Each of u_i is an operator

$$\sum_i \alpha_i \left(\sum_x |x\rangle U^x |u_i\rangle \right) = \sum_i \alpha_i \left(\sum_x \lambda_i^x |x\rangle |u_i\rangle \right)$$

where $U|u_i\rangle = \lambda_i |u_i\rangle$.

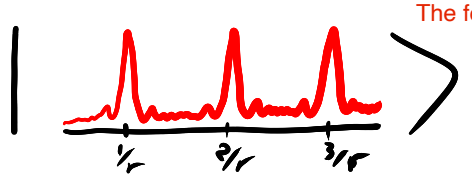
In the case of Shor's algorithm, since all eigenvalues of U_a take the form of $\frac{k}{r}$ where r is the period, performing this phase estimation amounts to sampling from

$$\frac{k_1}{r}, \frac{k_2}{r}, \frac{k_3}{r}, \frac{k_4}{r}, \ldots$$

and so is equivalent to the Shor-style analysis of period finding ☺

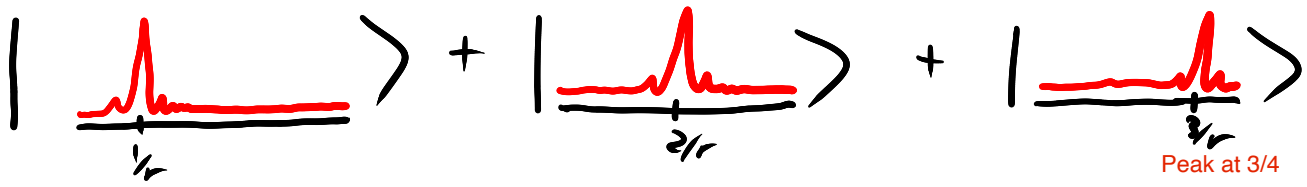
Shor picture:

The fourier coefficients are at 1/r, 2/r, and 3/4



Kitaev picture:

What does the ket mean?
Is this representing the sum of periodic functions that represent the signal?



Peak at 3/4

Peak at 2/r

We have a sum of states which contain only one peak.

(Applications of eigenvalue estimation)

Where does O(log(N)) come from?

If we're applying shors algorithm, it runs in O(log(N)) time.

Due to the complexity of implementing the gates, the complexity could be different.

If the size of the matrix is exponential, then we can solve the phase estimation problem in sub-exponential time.

Many, many practical problems ultimately reduce to finding eigenvalues or vectors. Google's seminal 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 ($O(2^n)$) matrices?

In general, NO :(

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

If we have these conditions, than we can compute this algorithm in sub exponential time.

- The matrix A must be unitary
  (rules out Pagerank)

Since the matrices are not usually unitary?

- An arbitrary $2^n \times 2^n$ unitary takes exponentially many gates to approximate
  (full Sk theorem is exponential in n)
- We can't get an eigenvector which is the more useful thing to have in many cases
  (we can do measurements on the eigenvector though)

Why can we not obtain an eigenvector?

(The HHL-algorithm)

Harrow Hassidim Lloyd

The HHL algorithm uses phase estimation (along with other techniques we haven't discussed yet) to solve a linear system

Contentious algorithm

$$A\vec{x} = \vec{b}$$

in roughly $O(\log(|A|))$ 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

What is suitably sparse?

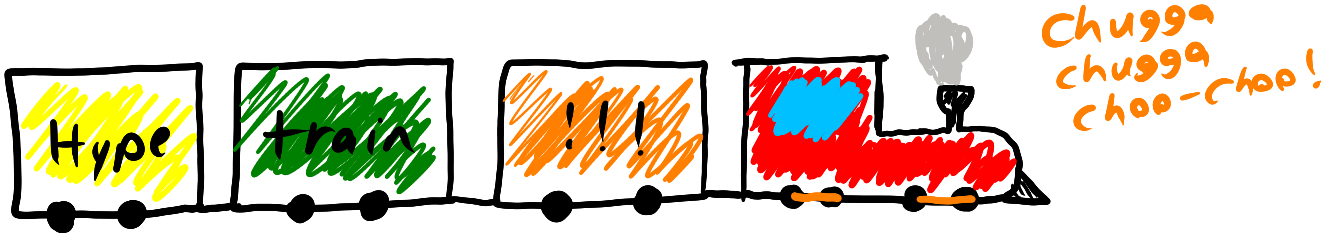
Condition number tells us how much a small change in the input affects the output.

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

QRAM is a model where a quantum processor and a classical processor communicate (ask professor).

In general, moving a classical vector into a quantum register takes exponential time.

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...

(Hamiltonians)

quantum phase estimation

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

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

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

$$i\frac{d}{dt}|\psi(t)\rangle = \hat{H}|\psi(t)\rangle$$

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

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

The operator $e^{-i\hat{H}t}$ turns out to be a unitary operator

How could we have guessed this?

$U(t)$, as we could have guessed, and so

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

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

$$U(t) = Pe^{-i\Lambda t}P^\dagger = P\begin{bmatrix} e^{-i\lambda_1 t} & & \\ & \ddots & \\ & & e^{-i\lambda_N t} \end{bmatrix}P^\dagger$$

(Hamiltonians & energy)

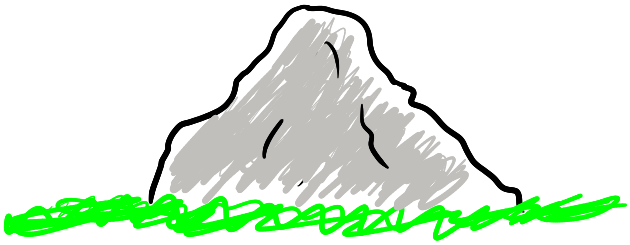
The Hamiltonian represents the **total energy** of a system, for instance the sum of the **kinetic** and **potential** energy from high school physics. The eigenvalues of $\hat{H}$ are the possible energies of the system, owing to the fact that energy is quantized.

This is why the hamiltonian can only contain positive values for the eigenvalues, since negative energy has no meaning here.

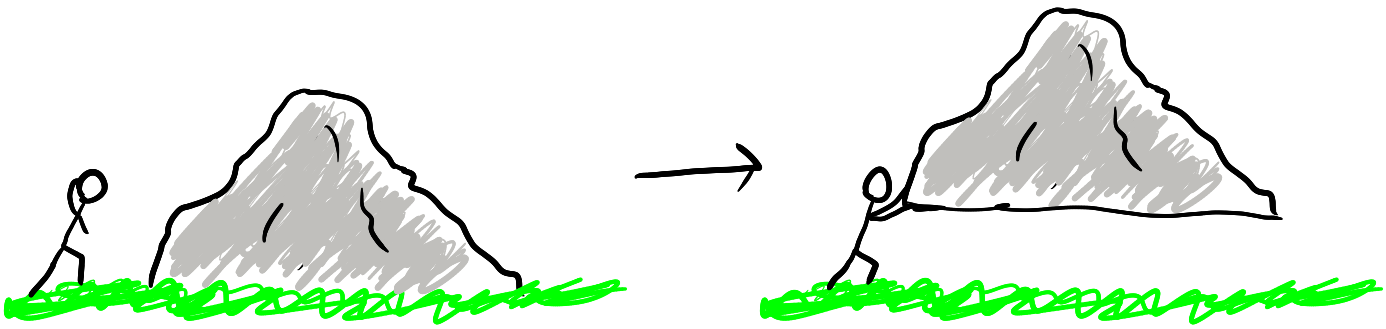
(Ground state)

The smallest eigenvalue of $\hat{H}$ is called the system's **ground state energy**, and an eigenvector for this λ 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:



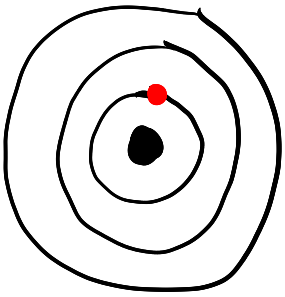
It will stay stationary until someone comes along and **expends** some of their energy to lift it, imparting some of that energy to the rock as **potential energy**.



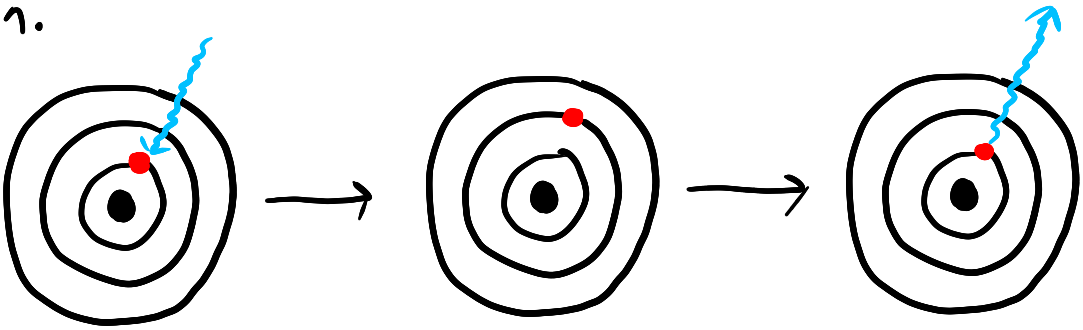
If the person lets go, the rock **prefers** the low energy state, so it will fall, dissipating its potential energy into the environment.

(Quantum chemistry)

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.



(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.

Why does the hamiltonian scale exponentially with the number of electrons?
Is it the number of hamiltonians or some aspect of the complexity that is scaling?
Is it scaling in the number of possible states that the combination of electrons can take on?

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 $U(t) = e^{-i\hat{H}t}$?