CPEN 400Q / EECE 571Q Lecture 11
The quantum Fourier transform and quantum phase estimation

Tuesday 15 February 2022

Announcements

- Project group / topic selection due today
- Please upgrade to PennyLane v0.21; new `requirements.txt` file will be included later with Quiz 5 and with Assignment 3.

Quiz 5 after class today.

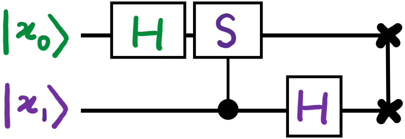
We introduced the quantum Fourier transform, and saw how it is the analog of the classical inverse discrete Fourier transform.

$$QFT|x\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{xk} |k\rangle$$

$$QFT = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{pmatrix}$$

where for n qubits, $N = 2^n$, and $\omega = e^{2\pi i/N}$

We saw the circuits for some special cases. For 1 qubit, it is just the Hadamard. For 2 and 3 qubits:



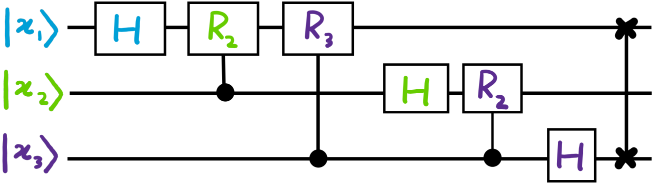


Image credit: Xanadu Quantum Codebook node F.2, F.3

Quantum Fourier transform

I showed you what the general form of the circuit looked like:

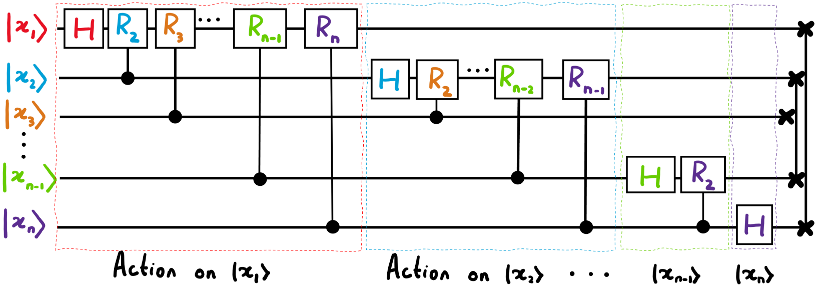


Image credit: Xanadu Quantum Codebook node F.3

Learning outcomes

- Derive the QFT circuit and implement it in PennyLane
- Describe the steps of the quantum phase estimation (QPE) subroutine
- Use the QFT to implement QPE

Review: fractional binary notation

Example

Let $k = k_1 k_2 k_3 k_4 = 0.1001$. The numerical value of this is:

$$
\begin{aligned}
0.1001 &= \frac{1}{2} + \frac{0}{2^2} + \frac{0}{2^3} + \frac{1}{2^4} \\
&= \frac{1}{2} + \frac{1}{16} \\
&= 0.5625
\end{aligned}
$$

We need this for the QFT because in the exponent, we have

$$\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{xk} |k\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i x (k/N)} |k\rangle$$

and k/N is a fractional value.

What we are going to show is that

$$\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{xk} |k\rangle$$

can be rewritten in the following factorized form:

$$\frac{\left(|0\rangle + e^{2\pi i 0.x_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.x_{n-1}x_n}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi i 0.x_1\cdots x_n}|1\rangle\right)}{\sqrt{N}}$$

Then, we will see how this form reveals to us the circuit that creates this state!

A circuit for the QFT

Start by rewriting k/N using fractional binary.

A circuit for the QFT

(keeping the last equation from the previous slide)

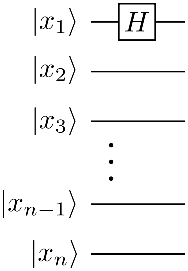
A circuit for the QFT

(keeping the last equation from the previous slide)

Starting with the state

$$|x\rangle = |x_1 \cdots x_n\rangle,$$

apply a Hadamard to qubit 1:

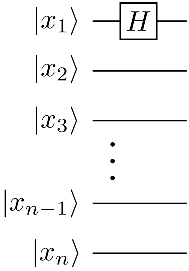


A circuit for the QFT

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.x_1}|1\rangle\right)|x_2 \cdots x_n\rangle$$

If $x_1 = 0$, $e^0 = 1$ and we get the $|+\rangle$ state.

If $x_1 = 1$, $e^{2\pi i(1/2)} = e^{\pi i} = -1$ and we get the $|-\rangle$ state.



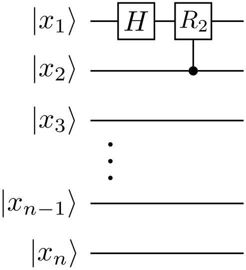
We are trying to make a state that looks like this:

$$|x\rangle \rightarrow \frac{\left(|0\rangle + e^{2\pi i 0.x_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.x_{n-1}x_n}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi i 0.x_1\cdots x_n}|1\rangle\right)}{\sqrt{N}}$$

Every qubit has a different *phase* on the $|1\rangle$ state. We are going to need some way of creating this.

We define the gate:

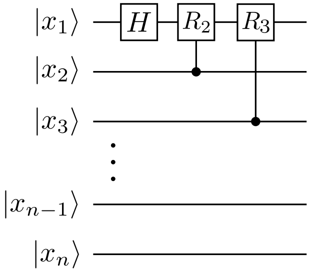
Now let's apply a controlled R_2 gate from qubit 2 to qubit 1



The first qubit picks up a phase:

A circuit for the QFT

Now let's apply a controlled R_3 gate from qubit 3 to qubit 1

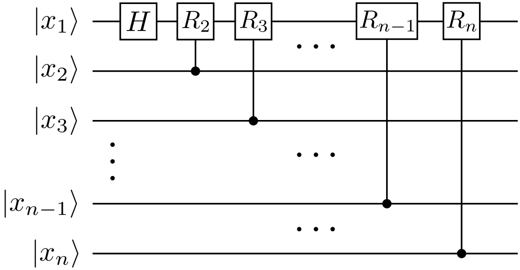


The first qubit picks up another phase:

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.x_1x_2}|1\rangle\right)|x_2\cdots x_n\rangle \to \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.x_1x_2x_3}|1\rangle\right)|x_2\cdots x_n\rangle$$

We can apply a controlled R_4 from the fourth qubit, etc. up to the n-th qubit to get

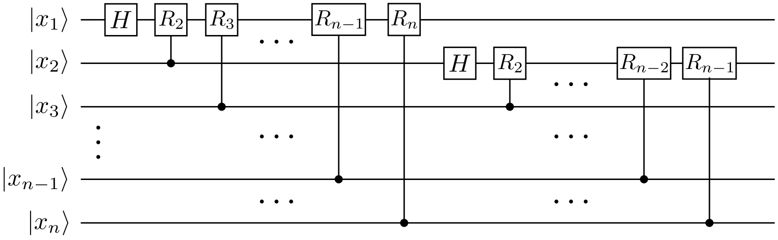
$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.x_1 x_2 \cdots x_n}|1\rangle\right)|x_2 \cdots x_n\rangle$$



A circuit for the QFT

Next, ignore the first qubit and do the same thing with the second qubit: apply H, and then controlled rotations from every qubit from 3 to n to get

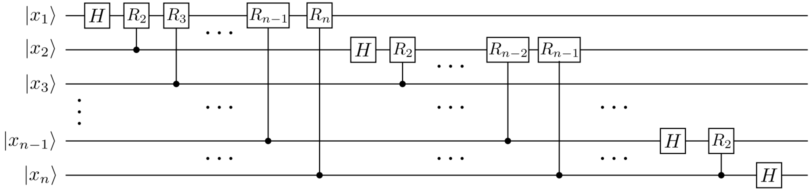
$$\frac{1}{\sqrt{2}^2}\left(|0\rangle + e^{2\pi i 0.x_1 x_2 \cdots x_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.x_2 \cdots x_n}|1\rangle\right)|x_3 \cdots x_n\rangle$$



A circuit for the QFT

If we do this for all qubits, we get something similar to that big ugly state from earlier:

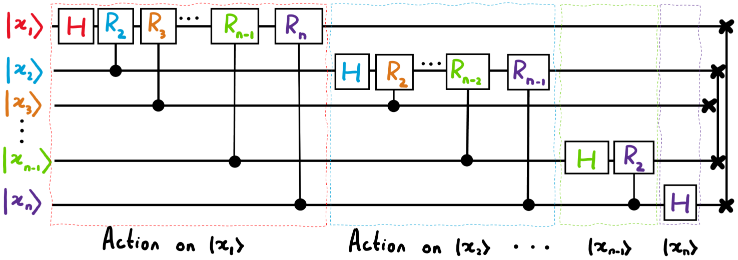
$$|x\rangle \to \frac{\left(|0\rangle + e^{2\pi i 0.x_1\cdots x_n}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi i 0.x_{n-1}x_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.x_n}|1\rangle\right)}{\sqrt{N}}$$



This is almost what we want: the order of the qubits is backwards. This is easily fixed with some SWAP gates.

So the QFT can be implemented using:

- n Hadamard gates
- $n(n-1)/2$ controlled rotations
- $\lfloor n/2 \rfloor$ SWAP gates if you care about the order



The number of gates is *polynomial in n*, so this can be implemented efficiently on a quantum computer! Let's try it...

Image credit: Xanadu Quantum Codebook node F.3

Quantum phase estimation

Fun fact: eigenvalues of unitary matrices are complex numbers with magnitude 1.

Proof:

So we can write

where θ_k is some phase angle such that $|\theta_k| \leq 1$.

What if we want to *learn* an unknown θ_k?

Eigenvalues of unitary matrices

Idea: apply U to the relevant eigenvector, because that's "what makes the phase come out".

...but this is an unobservable *global* phase!

We have to do something different: eigenvalue estimation, or quantum phase estimation (QPE).

Quantum phase estimation

Given a unitary U and one of its eigenvectors $|k\rangle$, estimate the value of θ_k such that

$$U|k\rangle = e^{2\pi i\theta_k}|k\rangle$$

Must determine:

- How to design a circuit that extracts the θ_k
- To what precision can we estimate it
- What to do if we don't know a $|k\rangle$ in advance

(*You will explore the last two in your homework!*)

Quantum phase estimation

Let U be an n-qubit unitary; therefore $|k\rangle$ is an n-qubit state.

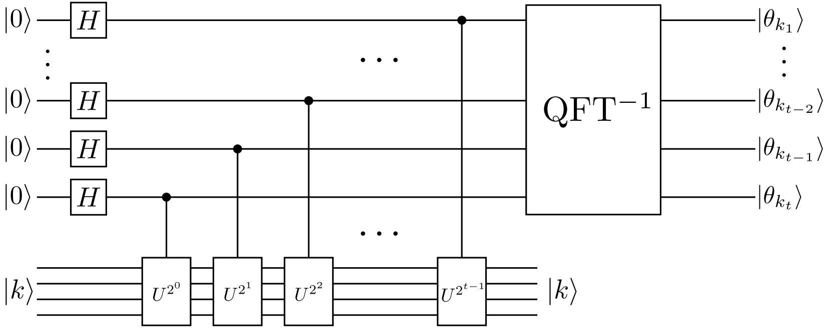
Assume for now that θ_k can be represented *exactly* using t bits in *fractional binary*:

$$\theta_k = 0.\theta_{k_1} \cdots \theta_{k_t}$$

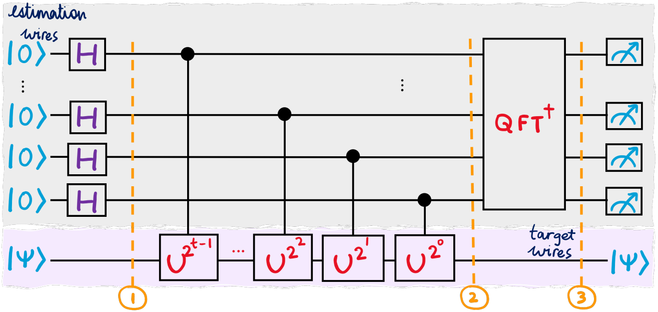
Fact: We can construct a circuit with $n + t$ qubits that recover the value of θ_k exactly by:

1. Preparing n qubits in state $|k\rangle$
2. Applying controlled applications of U to those qubits in a special way
3. Applying the inverse QFT

This is one version of the circuit:



The order of the controlled operations is irrelevant though, so you may see this too:



Why does this work? Let's analyze the state at points 1, 2, and 3 above.

Image credit: Xanadu Quantum Codebook node P.2

Quantum phase estimation: step 1

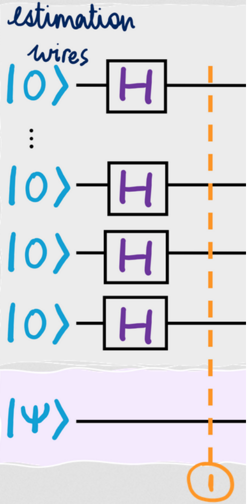


Image credit: Xanadu Quantum Codebook node P.2

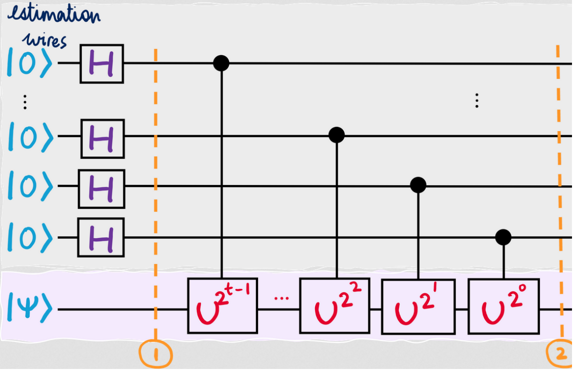
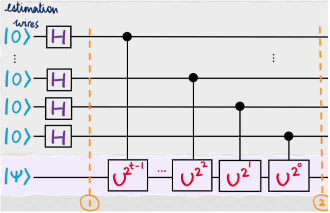


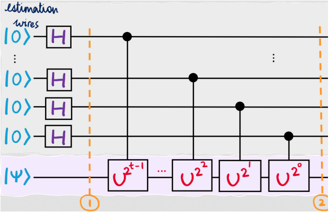
Image credit: Xanadu Quantum Codebook node P.2

Quantum phase estimation: step 2



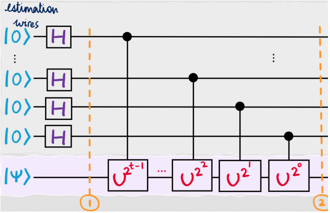
Consider the top-most qubit:

Quantum phase estimation: step 2



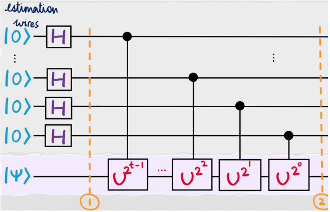
Use phase kickback

Quantum phase estimation: step 2



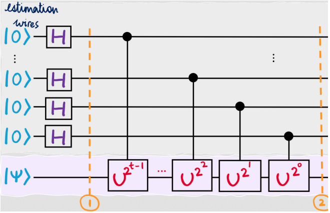
What is happening in the exponent?

Quantum phase estimation: step 2



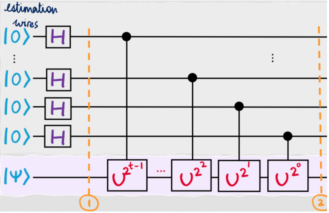
So we have the combined state:

Quantum phase estimation: step 2



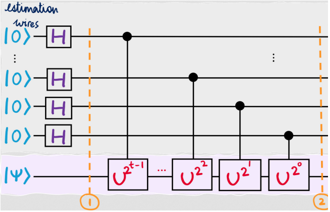
Let's do the second-last qubit (ignore what happens to others for now):

Quantum phase estimation: step 2



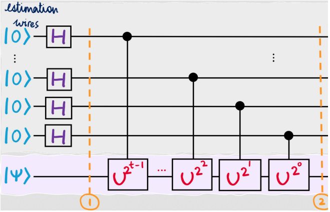
Again check the exponent...

Quantum phase estimation: step 2



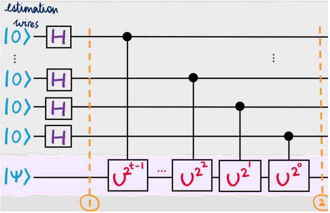
So we have the combined state:

Quantum phase estimation: step 2



Can show in the same way that for the last qubit

Quantum phase estimation: step 2

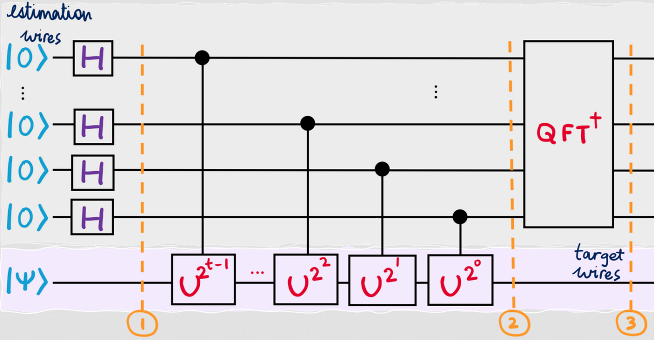


After step 2, we have the state

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0.\theta_{k_t}}|1\rangle) \cdots \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0.\theta_{k_2}\cdots\theta_{k_t}}|1\rangle)\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0.\theta_{k_1}\cdots\theta_{k_t}}|1\rangle)|k\rangle$$

Should look familiar!

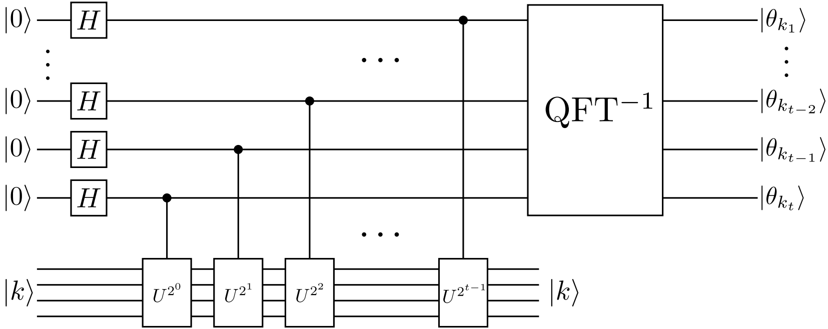
Quantum phase estimation: step 3



Last step is to apply the *inverse* QFT to recover the state...

Image credit: Xanadu Quantum Codebook node P.2

We can then measure to learn the numerical value of θ_k.



Let's implement it.

Next time

Content:

- Starting with Shor's algorithm

Action items:

1. E-mail me your project team and paper selection by end of day

Recommended reading:

- Codebook nodes F.1-F.3, P.1-P.4
- Nielsen & Chuang 5.1, 5.2