Intro to Quantum Computing

Lecture 2 bonus slides: the Quantum Fourier transform
May 10, 2019

Olivia Di Matteo

Quantum Information Science Associate, TRIUMF

Quantum Fourier transform

Physicists are familiar with the Fourier transform (FT) that re-expresses a wavefunction over position as one over momentum:

$$\phi(p) = \frac{1}{\sqrt{2\pi\hbar}} \int_{-\infty}^{\infty} \psi(x) e^{-ipx/\hbar} dx \tag{1}$$

There is also a *discrete* version of the FT that acts on finite-dimensional vectors rather than continuous spaces:

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \omega_N^{-jk} \tag{2}$$

where $\omega_N = e^{2\pi i/N}$ is an N^{th} root of unity.
It has many uses in e.g. signal processing, data compression.

The quantum Fourier transform (QFT) is the quantum analog of the (inverse) discrete FT. It is the underlying subroutine in Shor's factoring algorithm, as well as eigenvalue estimation algorithms.

Let $|x\rangle$ be an n-qubit computational basis state, $N = 2^n$. The QFT sends

$$|x\rangle \to \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{xk} |k\rangle \tag{3}$$

We are sending individual computational basis states to linear combinations of computational basis states.

The QFT is a unitary operation:

$$QFT = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} \omega_N^{-jk} |k\rangle \langle j| \tag{4}$$

It looks something like this...

$$QFT = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_N & \omega_N^2 & \cdots & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & \cdots & \omega_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \cdots & \omega_N^{(N-1)(N-1)} \end{pmatrix} \tag{5}$$

But... can we implement this unitary efficiently? It looks like it would be very messy.

A circuit for the QFT

Consider the expression

$$|x\rangle \to \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{xk} |k\rangle \tag{6}$$

Here x and k are represented as integers.

They are n-qubit computational basis states so they also have binary equivalents $|x\rangle = |x_1 \cdots x_n\rangle$, $|k\rangle = |k_1 \cdots k_n\rangle$:

$$x = 2^{n-1}x_1 + 2^{n-2}x_2 + \cdots + 2x_{n-1} + x_n$$

and similarly for k.

Recall that $\omega_N = e^{2\pi i/N}$.

We are working with

$$\omega_N^{xk} = e^{2\pi i x(k/N)} \tag{7}$$

with $N = 2^n$.

We can write a fraction $k/2^n$ in a 'decimal version' of binary:

$$k/2^n = 0.k_1 k_2 \cdots k_n = 2^{-1}k_1 + 2^{-2}k_2 + \cdots 2^{-n}k_n \tag{8}$$

Binary notation for decimal numbers

Example: let $k = 0.11010$.

The numerical value of this is:

$$
\begin{aligned}
0.11010 &= \frac{1}{2} + \frac{1}{2^2} + \frac{0}{2^3} + \frac{1}{2^4} + \frac{0}{2^5} \\
&= \frac{1}{2} + \frac{1}{4} + \frac{1}{16} \\
&= 0.8125
\end{aligned}
$$

This seems like a very convoluted way to write decimal numbers; this is going to become **very** hideous but I promise it is going somewhere.

A circuit for the QFT

Using the decimal version of k/N, we will work through the expression of the Fourier transform and see how we can reshuffle and *factor* the output state to get something that will make clear a circuit.

Let's start:

$$
\begin{aligned}
|x\rangle \quad &\rightarrow \quad \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{xk} |k\rangle \\
&= \quad \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i x (k/N)} |k\rangle \\
&= \quad \frac{1}{\sqrt{N}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} e^{2\pi i x \left(\sum_{\ell=1}^{n} k_\ell 2^{-\ell}\right)} |k_1 \cdots k_n\rangle
\end{aligned}
$$

A circuit for the QFT

(keeping the last equation from the previous slide)

$$
\begin{aligned}
&= \frac{1}{\sqrt{N}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} e^{2\pi i x \left(\sum_{\ell=1}^{n} k_\ell 2^{-\ell}\right)} |k_1 \cdots k_n\rangle \\
&= \frac{1}{\sqrt{N}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \bigotimes_{\ell=1}^{n} e^{2\pi i x k_\ell 2^{-\ell}} |k_\ell\rangle \\
&= \frac{1}{\sqrt{N}} \bigotimes_{\ell=1}^{n} \left(\sum_{k_\ell=0}^{1} e^{2\pi i x k_\ell 2^{-\ell}} |k_\ell\rangle \right) \\
&= \frac{1}{\sqrt{N}} \bigotimes_{\ell=1}^{n} \left(|0\rangle + e^{2\pi i x 2^{-\ell}} |1\rangle \right) \\
&= \frac{\left(|0\rangle + e^{2\pi i 0.x_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.x_{n-1}x_n}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi i 0.x_1\cdots x_n}|1\rangle\right)}{\sqrt{N}}
\end{aligned}
$$

So...

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

I trust this is as hideous as you hoped for, maybe even moreso.

But this form shows us a very nice way to create this state!

Starting with the state

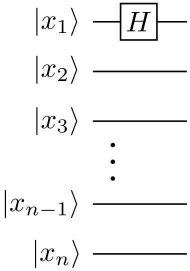
$$|x\rangle = |x_1 \cdots x_n\rangle,$$

apply a Hadamard to qubit 1:

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

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.

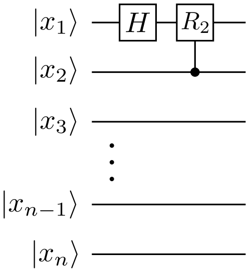


A circuit for the QFT

Define the phase gate

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix} \qquad (9)$$

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



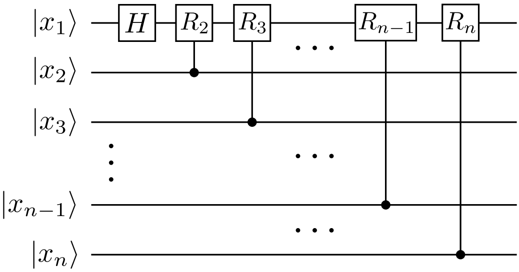
$$R_2 = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^2} \end{pmatrix} \qquad (10)$$

The first qubit picks up a phase:

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

We can apply a controlled R_3 from the third qubit, etc. up to the n-th qubit to get

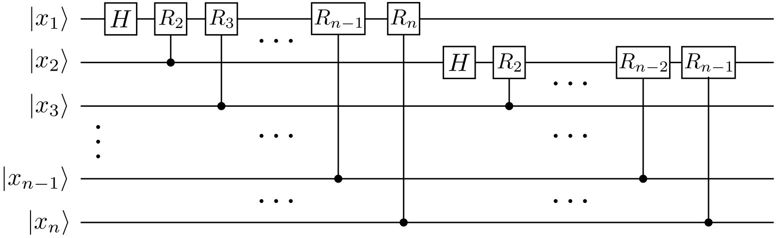
$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.x_1 x_2 \cdots x_n}|1\rangle\right)|x_2 \cdots x_n\rangle$$



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

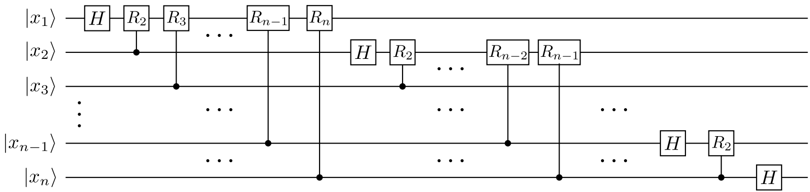
$$\frac{1}{\sqrt{2}^2}\left(|0\rangle + e^{2\pi i 0.x_1 x_2 \cdots x_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.x_2 \cdots x_n}|1\rangle\right)|x_3 \cdots x_n\rangle$$



A circuit for the QFT

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

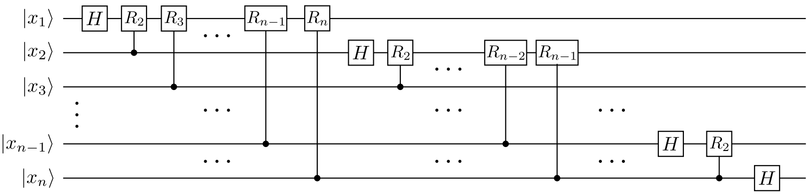
$$|x\rangle \to \frac{\left(|0\rangle + e^{2\pi i 0.x_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.x_{n-1}x_n}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi i 0.x_1\cdots x_n}|1\rangle\right)}{\sqrt{N}}$$



(though note that 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!

One of the key uses for the QFT is estimating the eigenvalues of unitary matrices. This has applications, for example, in finding the different energy states of molecules.

Eigenvalue (phase) estimation

Given a unitary U and one of its eigenvectors $|u\rangle$, find the eigenvalue φ such that

$$U|u\rangle = e^{2\pi i\varphi}|u\rangle \tag{11}$$

We will do this using controlled-U gates and the QFT.

Eigenvalue estimation

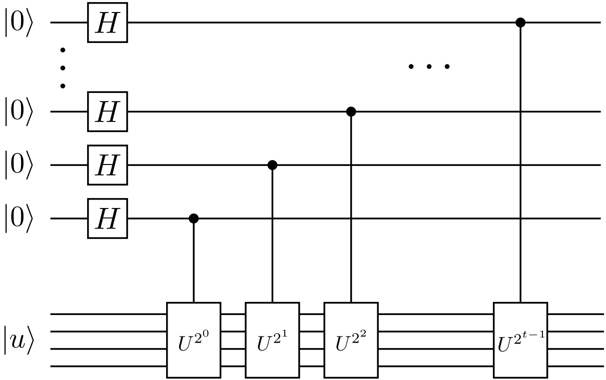
To do this, we are going to have to choose a finite precision to which we learn our eigenvalue. Suppose we would like to know its value up to t bits:

$$\varphi = 0.\varphi_1 \cdots \varphi_t \tag{12}$$

using the same binary expansion as last time.
We will make a circuit with two registers of qubits:

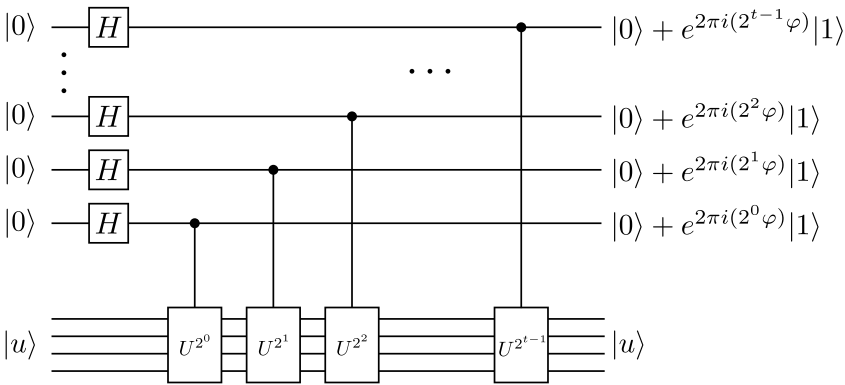
- t qubits in state $|0\rangle$, which at the end will hold the state $|\varphi_1 \cdots \varphi_t\rangle$
- n qubits in state $|u\rangle$, where n is the dimension of the system.

Now consider the following circuit:



We apply controlled-U gates a different number of times for each qubit. As $|u\rangle$ is an eigenvector, we'll pick up multiple copies of φ.

Eigenvalue estimation



This looks very similar to the state that comes out of the QFT! In fact if you work out the exponents, you'll find that it is the same.

So if we apply the inverse QFT, we can get the state $|\varphi_1 \cdots \varphi_t\rangle$ which we can then measure to learn the numerical value of φ.