Notes 8.370/18.435 Fall 2022

Lecture 20 Prof. Peter Shor

Last time, I explained how to construct the circuit for the quantum Fourier transform, but I left one piece out. Today, we finish that piece.

Today, we will also look at one of the applications of the quantum Fourier transform: phase estimation. Later, we will use this to show how to factor and take discrete logs on a quantum computer, rather than using the quantum Fourier transform directly (which was the way these algorithms were originally discovered).

Recall that the quantum Fourier transform takes

$$|j\rangle \longrightarrow \frac{1}{2^{L/2}} \sum_{k=0}^{2^L-1} e^{2\pi i jk/2^L} |k\rangle \tag{1}$$

To derive the circut for the QFT, we needed to rewrite this as

$$|j_1\rangle |j_2\rangle \ldots |j_L\rangle \to \frac{(|0\rangle + e^{2\pi i 0.j_n} |1\rangle)(|0\rangle + e^{2\pi i 0.j_{n-1}n} |1\rangle) \ldots (|0\rangle + e^{2\pi i 0.j_1 j_2 \ldots j_n} |1\rangle)}{\sqrt{2^L}},$$

where j_1 is the most significant bit of j and j_n is the 1's bit of j, where

$$0.j_1 j_2 \ldots j_n = \tfrac{1}{2} j_1 + \tfrac{1}{4} j_2 + \tfrac{1}{8} j_3 + \ldots + \tfrac{1}{2^n} j_n,$$

and where the qubits on the righthand side are $|k_1\rangle$, $|k_2\rangle$, ..., $|k_n\rangle$.

How do we derive this expression? We start by plugging in the binary expressions for j and k into Eq. 1:

$$\begin{aligned} |j_1\rangle |j_2\rangle \ldots |j_L\rangle \longrightarrow & \frac{1}{2^{L/2}} \sum_{k=0}^{2^L-1} e^{2\pi i 2^L \left(\sum_{s=1}^{L} \frac{1}{2^s} j_s\right) 2^L \left(\sum_{t=1}^{L} \frac{1}{2^t} k_t\right)/2^L} |k_1\rangle |k_2\rangle \ldots |k_L\rangle \\ = & \frac{1}{2^{L/2}} \sum_{k=0}^{2^L-1} e^{2\pi i \sum_{t=1}^{L} \left(k_t 2^{L-t} \left(\sum_{s=1}^{L} \frac{1}{2^s} j_s\right)\right)} |k_1\rangle |k_2\rangle \ldots |k_L\rangle \end{aligned}$$

We next express the sum on the righthand side as a tensor product:

$$|j_1\rangle |j_2\rangle \ldots |j_L\rangle \longrightarrow \frac{1}{2^{L/2}} \sum_{k_1=0}^{1} \sum_{k_2=0}^{1} \cdots \sum_{k_L=0}^{1} \bigotimes_{t=1}^{L} \left(e^{2\pi i k_t 2^{L-t} \sum_{s=1}^{L} \frac{1}{2^s} j_s} |k_t\rangle \right)$$

Now, because the exponential of an integer times $2\pi i$ is 1, we can start the sum in the exponent, $2^{L-t} \sum_{s=1}^{L} \frac{1}{2^s} j_s$, with $s = t$ rather than $s = 1$. But

$$e^{2\pi i 2^{L-t} \sum_{s=t}^{L} \frac{1}{2^s} j_s} = e^{2\pi i 0.j_t j_{t+1} \ldots j_L}$$

So this gives

$$|j_1\rangle |j_2\rangle \ldots |j_L\rangle \longrightarrow \frac{1}{2^{L/2}} \sum_{k_1=0}^{1} \sum_{k_2=0}^{1} \cdots \sum_{k_L=0}^{1} \bigotimes_{t=1}^{L} \left(e^{2\pi i k_t 0.j_t j_{t+1} \ldots j_L} |k_t\rangle \right)$$

however, the value of the term for $|k_t\rangle$ in this expression depends only on whether $|k_t\rangle = |0\rangle$ or $|k_t\rangle = |1\rangle$. We can thus use the distributive law of tensor products over sums to rewrite it as

$$|j_1\rangle |j_2\rangle \dots |j_L\rangle \longrightarrow \frac{1}{2^{L/2}} \bigotimes_{t=1}^{L} \left(|0\rangle + e^{2\pi i 0.j_t j_{t+1} \dots j_L} |1\rangle\right),$$

which is what we were trying to derive.

What is quantum phase estimation? Suppose we have a unitary operator U and an eigenvector $|v_\theta\rangle$ of this operator. The phase estimation problem is to give an approximation to the eigenvalue associated with $|v_\theta\rangle$, that is, to approximate the θ such that

$$U|v_\theta\rangle = e^{i\theta} |v_\theta\rangle.$$

The phase estimation algorithm does not give very accurate estimates of θ unless you can take high powers of U. Thus, we will assume that we have some kind of circuit that will compute U^{2^ℓ} in polynomial time in ℓ. Is this a reasonable assumption? There are cases for which it is. Suppose the unitary takes a number $s \pmod p$ and multiplies it by a number $g \pmod p$, so

$$U|s \,(\mathrm{mod}\ p)\rangle = |gs \,(\mathrm{mod}\ p)\rangle$$

Then U^k simply multiplies a number s by $g^k \pmod p$,

$$U^k|s \,(\mathrm{mod}\ p)\rangle = \left|g^k s \,(\mathrm{mod}\ p)\right\rangle,$$

and if we know $g^k \pmod p$, U^k is essentially no harder to implement than U. We will use this unitary transformation in the factoring algorithm, and we will discuss it in more detail then.

What is the basic idea of the algorithm? What we will do is create the state

$$\frac{1}{2^{L/2}} \sum_{k=0}^{2^L-1} e^{ik\theta} |k\rangle \tag{2}$$

Now, recall that the quantum Fourier transform maps

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

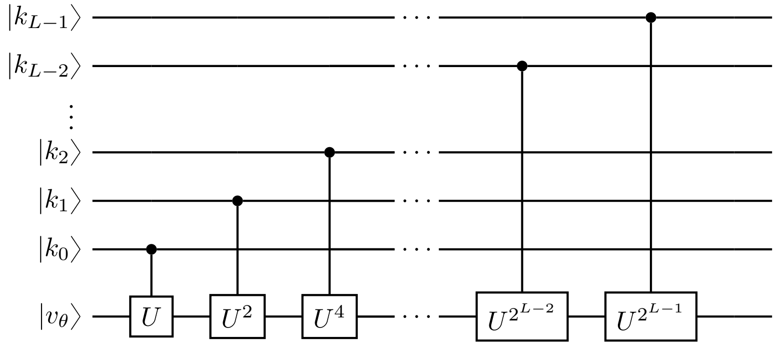
And since on the left-hand side, the states $|j\rangle$ form a basis, so do the states on the rigith-hand-side. But if $\theta = 2\pi m/2^L$ for some integer m, it's one of these basis states, and thus to identify θ, all we need to do is measure in that basis. How do we do that? We take the inverse Fourier transform, which takes

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

to $|j\rangle$, and then measure in the standard basis. Thus, if we get j, we know that $\theta = 2\pi j/2^L$.

To complete this sketch of the algorithm, we need to do two things. First, we need to show how to create the state (2). Second, we need to show that the algorithm works even if θ is not an integer multiple of $2\pi/2^L$.

We first explain how to create the state (2). Let's assume that we know how to implement the transformation U^{2^k} efficiently (in polynomial time, if we want the entire algorithm to be polynomial-time). Now, consider the following quantum circuit.



If the input is $k = k_{L-1}k_{L-2}\ldots k_1k_0$ in binary, this circuit applies $U^{k_0+2k_1+4k_2+\ldots+2^{L-1}k_{L-1}}$ to $|v_\theta\rangle$, which is the same as applying $U^k\,|v_\theta\rangle = e^{ik\theta}$, we have the output of $|k\rangle\,e^{i\theta k}$. Thus, to get our desired state

$$\frac{1}{2^{L/2}}\sum_{k=0}^{2^L-1} e^{ik\theta}\,|k\rangle\,|v_\theta\rangle\,, \tag{3}$$

we simply need to input

$$\frac{1}{2^{L/2}}\sum_{k=0}^{2^L-1}|k\rangle\,|v_\theta\rangle$$

into the circuit above. We can do this by putting a $|+\rangle$ state into each of the first L quantum wires.

Next, we will computer what happens when we take the above state and apply the inverse Fourier transform to it. even in the case where θ is not an integer multiple of $2\pi/2^L$. The inverse Fourier transform is

$$|k\rangle \longrightarrow \frac{1}{2^{L/2}}\sum_{j=0}^{2^L-1} e^{-2\pi ijk/L}\,|j\rangle\,,$$

so when we plug it into (3), we get

$$\frac{1}{2^L}\sum_{k=0}^{2^L-1} e^{ik\theta}\sum_{j=0}^{2^L-1} e^{-2\pi ijk/2^L}\,|j\rangle = \frac{1}{2^L}\sum_{j=0}^{2^L-1}|j\rangle\left(\sum_{k=0}^{2^L-1} e^{ik(\theta-2\pi j/2^L)}\right)$$

The last piece is a geometric sum, so we can use the formula for geometric sums, and show that the probability of seeing $|j\rangle$ is

$$\left| \frac{1}{2^L} \sum_{k=0}^{2^L-1} e^{ik(\theta - 2\pi j/2^L)} \right|^2 = \frac{1}{4^L} \left| \frac{1 - e^{i(2^L\theta - 2\pi j)}}{1 - e^{i(\theta - 2\pi j/2^L)}} \right|^2 .$$

We now show that the value of j we obtain will give us a very good approximation of θ. There are more accurate ways of doing this than the one we use, but this one is fairly simple. will bound the numerator by 2, and approximate the denominator by $i(\theta - 2\pi j/2^L)$. Let j' be the value (not necessarily an integer) which would give the right value of θ, that is the j' that makes $2\pi j'/2^L = \theta$. We see that the probability of seeing some specific j with $j > j' + \alpha$ or $j < j' - \alpha$ is at most around

$$\frac{1}{4^L} \left| \frac{2}{2\pi(j' - j)/2^L} \right|^2 \leqslant \frac{1}{|\pi\alpha|^2} .$$

This shows that j' is very tightly concentrated around $2^L\theta/(2\pi)$, and thus we can get a good estimate of the phase θ.