Lecture 9: Phase estimation (continued); the quantum Fourier transform

February 16, 2006

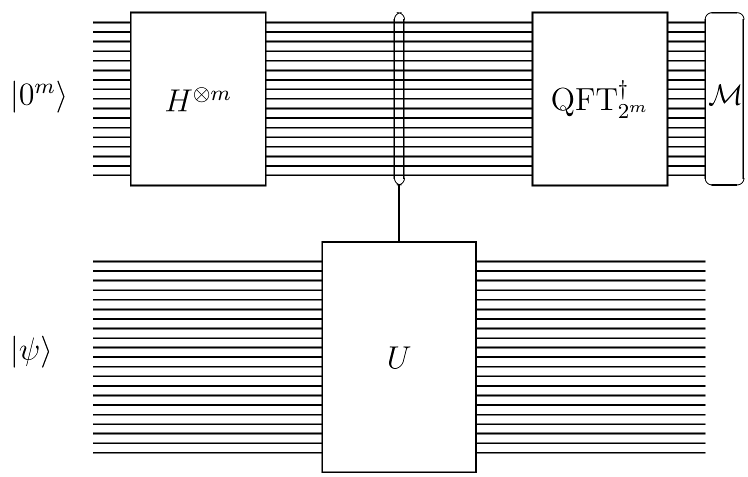
Phase estimation (continued)

In the previous lecture we discussed phase estimation. Recall that the set-up was as follows. We have a quantum circuit for performing the transformation $\Lambda_m(U)$, defined by

$$\Lambda_m(U)\,|k\rangle\,|\phi\rangle = |k\rangle\,U^k\,|\phi\rangle\,,$$

for some unitary transformation U and positive integer m, along with a quantum state $|\psi\rangle$ that is an eigenvector of U. The eigenvalue associated with $|\psi\rangle$ is $e^{2\pi i\theta}$ for $\theta \in [0, 1)$, and the goal is to approximate θ.

We had devised the following procedure, which works perfectly when $\theta = j/2^m$ for some integer $j \in \{0, \ldots, 2^m - 1\}$:



Specifically, in the case $\theta = j/2^m$, the measurement results in outcome j with probability 1.

We were in the process of analyzing the (more typical) case when θ does not have the form $j/2^m$ for some integer j. We had determined that the probability associated with each possible outcome $j \in \{0, \ldots, 2^m - 1\}$ of the measurement was

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

Because we already dealt with the case that $\theta = j/2^m$ for some choice of $j \in \{0, \ldots, 2^m - 1\}$, we may now assume this is not the case, and so

$$e^{2\pi i(\theta - j/2^m)} \neq 1$$

for every integer j. Using the same formula for the sum of a geometric series from last time,

$$\sum_{k=0}^{n-1} x^k = \frac{x^n - 1}{x - 1}$$

for $x \neq 1$, we may then simplify:

$$p_j = \frac{1}{2^{2m}} \left| \frac{e^{2\pi i (2^m \theta - j)} - 1}{e^{2\pi i (\theta - j/2^m)} - 1} \right|^2$$

Let us first consider the probability of obtaining the **best possible** j, meaning that

$$e^{2\pi i \theta} = e^{2\pi i (j/2^m + \varepsilon)}$$

for some real number ε with $|\varepsilon| \leq 2^{-(m+1)}$. This is equivalent to saying

$$\theta = \frac{j}{2^m} + \varepsilon \pmod 1$$

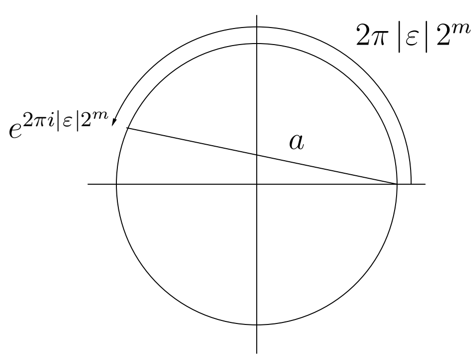
for $|\varepsilon| \leq 2^{-(m+1)}$, where "equality mod 1" means that the fractional parts of the two sides of the equation agree. Assuming that j satisfies this equation we may prove a lower bound on p_j as follows. Let

$$\begin{aligned} a &= \left| e^{2\pi i (2^m \theta - j)} - 1 \right| = \left| e^{2\pi i \varepsilon 2^m} - 1 \right|, \\ b &= \left| e^{2\pi i (\theta - j/2^m)} - 1 \right| = \left| e^{2\pi i \varepsilon} - 1 \right|, \end{aligned}$$

so that

$$p_j = \frac{1}{2^{2m}} \frac{a^2}{b^2}.$$

To get a lower bound on p_j we need a lower bound on a and an upper bound on b. To get a lower bound on a, consider the following picture:



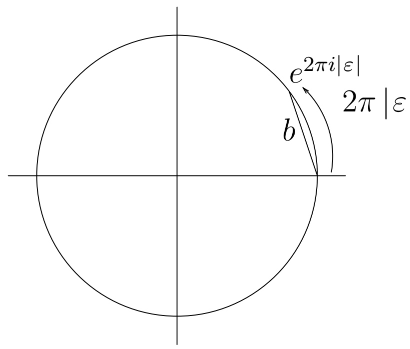
The ratio of the minor arc length to the chord length is at most $\pi/2$, so

$$\frac{2\pi\,|\varepsilon|\,2^m}{a} \le \frac{\pi}{2},$$

which implies

$$a \ge 4\,|\varepsilon|\,2^m.$$

Along similar lines, we may consider b along with the fact that the ratio of arc length to chord length is at least 1:



We obtain

$$\frac{2\pi\,|\varepsilon|}{b} \ge 1$$

so

$$b \le 2\pi\,|\varepsilon|\,.$$

Putting the two bounds together, we obtain

$$p_j \ge \frac{1}{2^{2m}}\frac{16\,|\varepsilon|^2\,2^{2m}}{4\pi^2\,|\varepsilon|^2} = \frac{4}{\pi^2} > 0.4.$$

Although you might not think that 40% is very good, in fact it is amazing in a way—this is the probability that every single one of the bits you measure is correct, so that your approximation to θ is good to m bits of precision.

We can use basically the same methods to put upper bounds on the probability of obtaining inaccurate results. Suppose now that for a given value of j we have

$$e^{2\pi i\theta} = e^{2\pi i(j/2^m+\varepsilon)}$$

for some real number ε with $\frac{\alpha}{2^m} \le |\varepsilon| < 1/2$. Here α is an arbitrary positive number that we can choose later. As before we have

$$p_j = \frac{1}{2^{2m}}\frac{a^2}{b^2}$$

for

$$a = \left|e^{2\pi i\varepsilon 2^m} - 1\right|,$$
$$b = \left|e^{2\pi i\varepsilon} - 1\right|.$$

This time we will simply use the fact that $a \leq 2$. The bound $b \geq 4|\varepsilon|$ follows by similar reasoning to the bound on a from before. Now we have

$$p_j \leq \frac{4}{2^{2m}(4|\varepsilon|)^2} = \frac{1}{4\alpha^2}.$$

This implies that highly inaccurate results are very unlikely. For example, if we consider $\alpha = 1$, meaning that our assumption is only that $|\varepsilon| \geq 2^{-m}$, the probability of obtaining the corresponding value of j is at most $1/4$. For worse approximations, implying a larger bound on $|\varepsilon|$, the probability of obtaining the corresponding value of j quickly becomes very small.

So, what should you do if you want better than a $4/\pi^2$ probability of obtaining an approximation of θ that is good to, say, k bits of precision? One way to do this is to set $m = k+2$, say, run the phase estimation procedure several times, and to look for the most commonly appearing outcome. At least one outcome, which is accurate to $k+2$ bits of precision, occurs with probability at least $4/\pi^2$. Outcomes with fewer than k bits of precision are much less likely as argued above. If you now take the most commonly occurring outcome and round it to k bits of precision, the probability of correctness approaches 1 exponentially fast in the number of times the procedure is repeated. Notice also that you do not need multiple copies of the state $|\psi\rangle$ to perform this process, because the state $|\psi\rangle$ remains on the lower collection of qubits each time the procedure is performed and can simply be fed into the next iteration.

Efficient implementation of the quantum Fourier transform

Now let us consider how the quantum Fourier transform may be implemented by quantum circuits. Recall that

$$\mathrm{QFT}_{2^m} |j\rangle = \frac{1}{\sqrt{2^m}} \sum_{k=0}^{2^m-1} e^{2\pi i jk/2^m} |k\rangle .$$

Let us generalize some notation we used last time and let

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

for any positive integer N. Let us also define a unitary mapping $\widetilde{\mathrm{QFT}}_{2^m}$ to be the same as QFT_{2^m} except with the output qubits in reverse order. Specifically, if an integer $k \in \{0, \ldots, 2^m - 1\}$ is written in binary notation as $k_{m-1}k_{m-2}\cdots k_0$ then we define

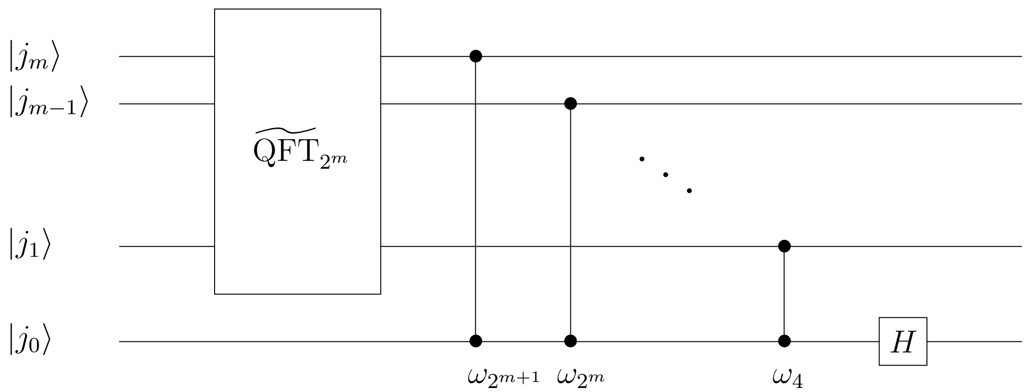
$$\widetilde{\mathrm{QFT}}_{2^m} |j_{m-1}j_{m-2}\cdots j_0\rangle = \frac{1}{\sqrt{2^m}} \sum_{k=0}^{2^m-1} \omega_{2^m}^{jk} |k_0k_1\cdots k_{m-1}\rangle .$$

Certainly if we can come up with an efficient implementation of $\widetilde{\mathrm{QFT}}_{2^m}$, then an efficient implementation of QFT_{2^m} follows—just reverse the order of the output qubits after performing $\widetilde{\mathrm{QFT}}_{2^m}$. The reason why we consider $\widetilde{\mathrm{QFT}}_{2^m}$ rather than QFT_{2^m} is simply for convenience.

Our description of quantum circuits for performing $\widetilde{\mathrm{QFT}}_{2^m}$ for any given value of m is essentially recursive. Let us start with the base case, which is $m = 1$. The transformation $\widetilde{\mathrm{QFT}}_2$ is just a fancy name for a Hadamard transform:

$$\widetilde{\mathrm{QFT}}_2 \left|j\right\rangle = \frac{1}{\sqrt{2}} \sum_{k=0}^{1} \omega_2^{jk} \left|k\right\rangle = \frac{1}{\sqrt{2}} \left|0\right\rangle + \frac{1}{\sqrt{2}} (-1)^j \left|1\right\rangle = H \left|j\right\rangle .$$

For general $m \geq 2$, the following circuit computes $\widetilde{\mathrm{QFT}}_{2^{m+1}}$:



Of course the diagram assumes you know how to implement the transformation $\widetilde{\mathrm{QFT}}_{2^m}$, but using the fact that $\widetilde{\mathrm{QFT}}_2$ is the same as a Hadamard transform we can easily unwind the recursion if we want an explicit description of a circuit.

Now let us show that the circuit works correctly. It suffices as usual to show that it works correctly on classical states. We wish to show that

$$\widetilde{\mathrm{QFT}}_{2^{m+1}} \left|j_m\, j_{m-1}\, \cdots\, j_0\right\rangle = \frac{1}{\sqrt{2^{m+1}}} \sum_{k=0}^{2^{m+1}-1} \omega_{2^{m+1}}^{jk} \left|k_0\, k_1\, \cdots\, k_m\right\rangle$$

for each $j \in \{0, \ldots, 2^{m+1} - 1\}$.

Let us write

$$\begin{aligned} j' &= j_m\, j_{m-1}\, \cdots\, j_1 = \lfloor j/2 \rfloor, \\ k' &= k_{m-1}\, k_{m-2}\, \cdots\, k_0 = k - k_m 2^m. \end{aligned}$$

The initial state $|j\rangle$ may therefore be written $|j'\rangle\,|j_0\rangle$, and the operation $\widetilde{\mathrm{QFT}}_{2^m}$ maps this state to

$$\frac{1}{\sqrt{2^m}} \sum_{k'=0}^{2^m-1} \omega_{2^m}^{j'k'} \left|k'_0\, k'_1\, \cdots\, k'_{m-1}\right\rangle \left|j_0\right\rangle .$$

The controlled phase-shifts then transform this state to

$$\frac{1}{\sqrt{2^m}} \sum_{k'=0}^{2^m-1} \omega_{2^m}^{j'k'} \omega_{2^{m+1}}^{j_0 k'_0} \omega_{2^m}^{j_0 k'_1} \cdots \omega_4^{j_0 k'_{m-1}} \left|k'_0\, k'_1\, \cdots\, k'_{m-1}\right\rangle \left|j_0\right\rangle .$$

Using the fact that $\omega_N = \omega_{rN}^r$ for any choice of positive integers N and r, we may simplify the above expression and conclude that the state of the circuit after the controlled phase shifts is

$$\begin{aligned}\frac{1}{\sqrt{2^m}} \sum_{k'=0}^{2^m-1} \omega_{2^{m+1}}^{2j'k'+j_0k_0'+j_0(2k_1')+\cdots j_0(2^{m-1}k_{m-1}')} \left|k_0'\, k_1' \, \cdots \, k_{m-1}'\right\rangle \left|j_0\right\rangle \\ = \frac{1}{\sqrt{2^m}} \sum_{k'=0}^{2^m-1} \omega_{2^{m+1}}^{jk'} \left|k_0'\, k_1' \, \cdots \, k_{m-1}'\right\rangle \left|j_0\right\rangle .\end{aligned}$$

Finally, the Hadamard transform maps this state to

$$\frac{1}{\sqrt{2^{m+1}}} \sum_{k'=0}^{2^m-1} \sum_{k_m=0}^{1} \omega_{2^{m+1}}^{jk'} (-1)^{k_m j_0} \left|k_0'\, k_1' \, \cdots \, k_{m-1}'\right\rangle \left|k_m\right\rangle .$$

Notice that

$$(-1)^{k_m j_0} = (-1)^{k_m j} = \omega_{2^{m+1}}^{j(2^m k_m)},$$

which implies that the final state is

$$\frac{1}{\sqrt{2^{m+1}}} \sum_{k'=0}^{2^m-1} \sum_{k_m=0}^{1} \omega_{2^{m+1}}^{jk'+j(2^m k_m)} \left|k_0'\, k_1' \, \cdots \, k_{m-1}'\right\rangle \left|k_m\right\rangle = \frac{1}{\sqrt{2^{m+1}}} \sum_{k=0}^{2^{m+1}-1} \omega_{2^{m+1}}^{jk} \left|k_0\, k_1 \, \cdots \, k_m\right\rangle$$

as required.

How many gates are required in the above circuit? Letting $g(m)$ denote the number of gates needed to perform $\widetilde{QFT}_{2^m}$, we have the following recurrence:

$$\begin{aligned} g(1) &= 1 \\ g(m+1) &= g(m) + (m+1). \end{aligned}$$

The solution to this recurrence is

$$g(m) = \sum_{j=1}^{m} j = \binom{m+1}{2}.$$

Thus, we need only $O(m^2)$ gates to compute the quantum Fourier transform on m qubits. In fact there are better bounds known that are based on fast multiplication methods. However, these constructions are much more complicated and would probably not be practical (assuming we had a quantum computer) until m is quite large.