AQI: Advanced Quantum Information
Lecture 1 (Module 4): Quantum Fourier Transform and Phase Estimation

February 19, 2013

Lecturer: Dr. Mark Tame
(email: m.tame@imperial.ac.uk)

Introduction

Today and tomorrow I'll build up to Shor's algorithm and show how an exponential speedup (compared to all known classical methods) can be achieved in factoring numbers using a quantum computer. Here, the quantum Fourier transform is an operation that plays a vital role. In particular, the phase estimation algorithm uses it as part of the order-finding subroutine of Shor's algorithm. I'm sure you know, or have heard already, that Shor's algorithm can be used to break RSA encryption. To many people, including me, this is a surprising result and a pretty remarkable thing as the vast majority of security protocols used on the internet are based on this type of encryption. Armed with a quantum computer, a person (or nation) would have a great deal of power...

1 Quantum Fourier transform

1.1 Discrete Fourier transform (classical)

If we input a vector of complex numbers $\underline{x} = (x_0, \ldots, x_{N-1})$ into the discrete Fourier transform, then the output is a vector of complex numbers $\underline{y} = (y_0, \ldots, y_{N-1})$ defined by

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{i2\pi jk/N}. \tag{1}$$

Note that here and in the next couple of lectures I'll use i to denote $\sqrt{-1}$ and not an index like j or k. There will be lots of indicies coming up soon!

1.2 Quantum Fourier transform (quantum)

If we input a quantum state $|x\rangle = \sum_{j=0}^{N-1} x_j |j\rangle$ into the quantum Fourier transform, then the output is a quantum state $|y\rangle = \sum_{j=0}^{N-1} y_k |k\rangle$ whose coefficients y_k are the discrete Fourier transform of the amplitudes x_j. In other words, the computational basis states transform as

$$|j\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{i2\pi jk/N} |k\rangle . \tag{2}$$

To implement the quantum Fourier transform (QFT) we take $N = 2^n$ and use the basis states $\{|0\rangle, \ldots, |2^n - 1\rangle\}$ from the basis states of an n-qubit quantum computer.

How do we build a quantum circuit to implement the QFT? For the simple case of a qubit the QFT is just the Hadamard operation:

$$H: \quad |0\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{3}$$

$$|1\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{4}$$

For multiqubit systems representing $N = 2^n$ basis states (or levels) it becomes more involved. To see this we need to consider a different representation of the QFT compared to that given in Eq. (2)

1.3 Binary representations

The following binary representations will be useful for what follows:

- $j = j_1 j_2 j_3 \dots j_n$ is the binary representation of integer j, where $j = j_1 2^{n-1} + j_2 2^{n-2} + \dots j_n 2^0$.
- $0.j_\ell j_{\ell+1} \dots j_m$ is the binary fraction representation of the fraction j, written in $m - \ell$ bits, where $j = j_\ell/2 + j_{\ell+1}/4 + \dots j_m/2^{m-\ell+1}$, with ℓ and m being any integers we choose ($\ell \leq m$).

With these we can show that (see Appendix)

$$|j\rangle = |j_1 j_2 \dots j_n\rangle \overset{QFT}{\rightarrow} \frac{(|0\rangle + e^{i2\pi 0.j_n}|1\rangle)(|0\rangle + e^{i2\pi 0.j_{n-1}j_n}|1\rangle)\dots(|0\rangle + e^{i2\pi 0.j_1 j_2 \dots j_n}|1\rangle)}{2^{n/2}}. \tag{5}$$

Below is a quantum circuit that can achieve this transformation:

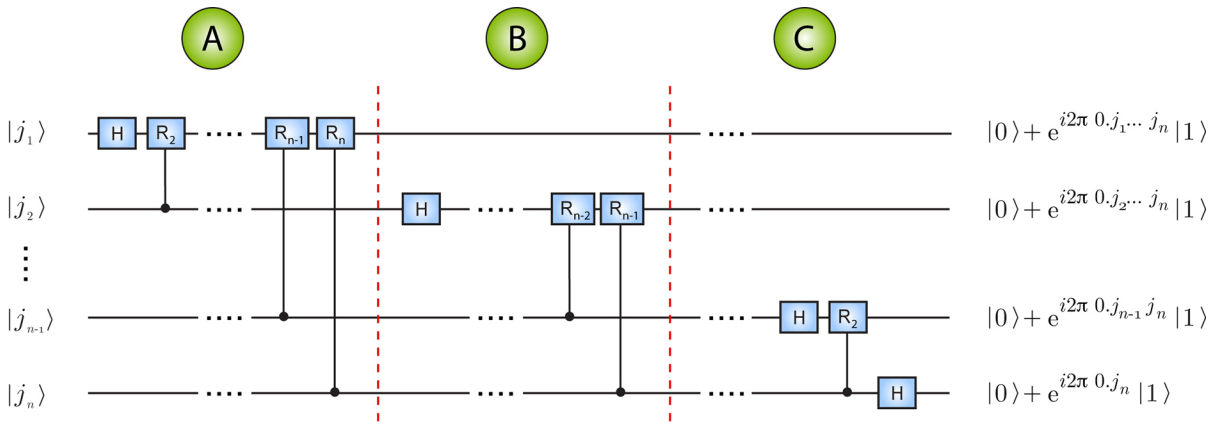

Figure 1: Quantum circuit representing the quantum Fourier transform. The output states are left unnormalised for brevity.

Here the rotation $R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{i2\pi/2^k} \end{pmatrix}$.

There are 3 main sections:

Section A

- Apply a Hadamard operation to the first qubit. This gives the state $\frac{1}{2^{1/2}}(|0\rangle + e^{i2\pi 0.j_1}|1\rangle)|j_2 \dots j_n\rangle$, because when $j_1 = 0$ we have $0.j_1 = 0 \to |+\rangle$ and when $j_1 = 1$ we have $0.j_1 = 1/2 \to |-\rangle$.
- Apply a controlled R_2 gate. This gives the state $\frac{1}{2^{1/2}}(|0\rangle + e^{i2\pi 0.j_1 j_2}|1\rangle)|j_2 \dots j_n\rangle$, because $0.j_1 j_2 = j_1/2 + j_2/4$ and if $j_2 = 0$ there is no controlled rotation $j_2/4 = 0$, but if $j_2 = 1$ a controlled rotation $R_2 = \begin{pmatrix} 1 & 0 \\ 0 & e^{i2\pi/4} \end{pmatrix}$ is applied, adding a phase $2\pi j_2/4$ to the phase of qubit 1.
- Applying $R_3 \dots R_n$ gates gives the state $\frac{1}{2^{1/2}}(|0\rangle + e^{i2\pi 0.j_1 \dots j_n}|1\rangle)|j_2 \dots j_n\rangle$.

Section B

Here we do a similar procedure for qubit 2. This gives

$$\frac{1}{2^{2/2}}(|0\rangle + e^{i2\pi 0.j_1 \dots j_n}|1\rangle)(|0\rangle + e^{i2\pi 0.j_2 \dots j_n}|1\rangle)|j_3 \dots j_n\rangle\,. \tag{6}$$

Section C

This is similar to Sections A and B all the way up until the last qubit which gives

$$\frac{1}{2^{n/2}}(|0\rangle + e^{i2\pi 0.j_1 \dots j_n}|1\rangle)(|0\rangle + e^{i2\pi 0.j_2 \dots j_n}|1\rangle) \dots (|0\rangle + e^{i2\pi 0.j_n}|1\rangle). \tag{7}$$

Here we've used $\frac{n(n+1)}{2}$ gates. Now we just need to swap the order of the qubits (reverse them all), which requires $n/2$ swap gates. The above circuit provides an asymptotic 'tight bound' $\Theta(n^2)$ algorithm to complete the quantum Fourier transform (in terms of the runtime). We'll look into runtimes in more detail in the third lecture. Note that while the best classical algorithms for computing the discrete Fourier transform on 2^n elements require $\Theta(n2^n)$ gates, *i.e.* exponentially more operations, in the QFT case we don't ever have access to the amplitudes until a measurement is made. Therefore we need a repeated number of runs (plus measurements) to find all the amplitudes, which destroys any speedup we gained. Thus the quantum Fourier transform by itself cannot provide any advantage over classical methods.

2 Quantum phase estimation algorithm

Suppose that a unitary operation U has an eigenvector $|u\rangle$ with eigenvalue $e^{i2\pi\varphi}$, where φ is unknown. Here $\varphi \in [0, 1] \to 2\pi\varphi \in [0, 2\pi]$. For example, the unitary operation

$$U = R_z^\phi = \begin{pmatrix} e^{-i\phi} & 0 \\ 0 & e^{i\phi} \end{pmatrix} \tag{8}$$

has eigenvectors $|u\rangle = |0\rangle$ or $|1\rangle$, and one finds that $U|0\rangle = e^{-i\phi}|0\rangle$ or $U|1\rangle = e^{i\phi}|1\rangle$, with $\phi = 2\pi\varphi$.

The goal of the phase estimation algorithm is to estimate φ when we don't know U (or $|u\rangle$) but have available black boxes capable of preparing $|u\rangle$ and applying controlled U operations. The use of black boxes

here means that the phase estimation algorithm should be thought of as a subroutine or 'module' which when combined with other modules makes up a complete algorithm.

Two registers are required for the algorithm:

1. The first register contains t-qubits to read out the value of φ (estimate).
2. The second register begins in the state $|u\rangle$.

Step 1

Below is a figure showing a circuit diagram of the algorithm:

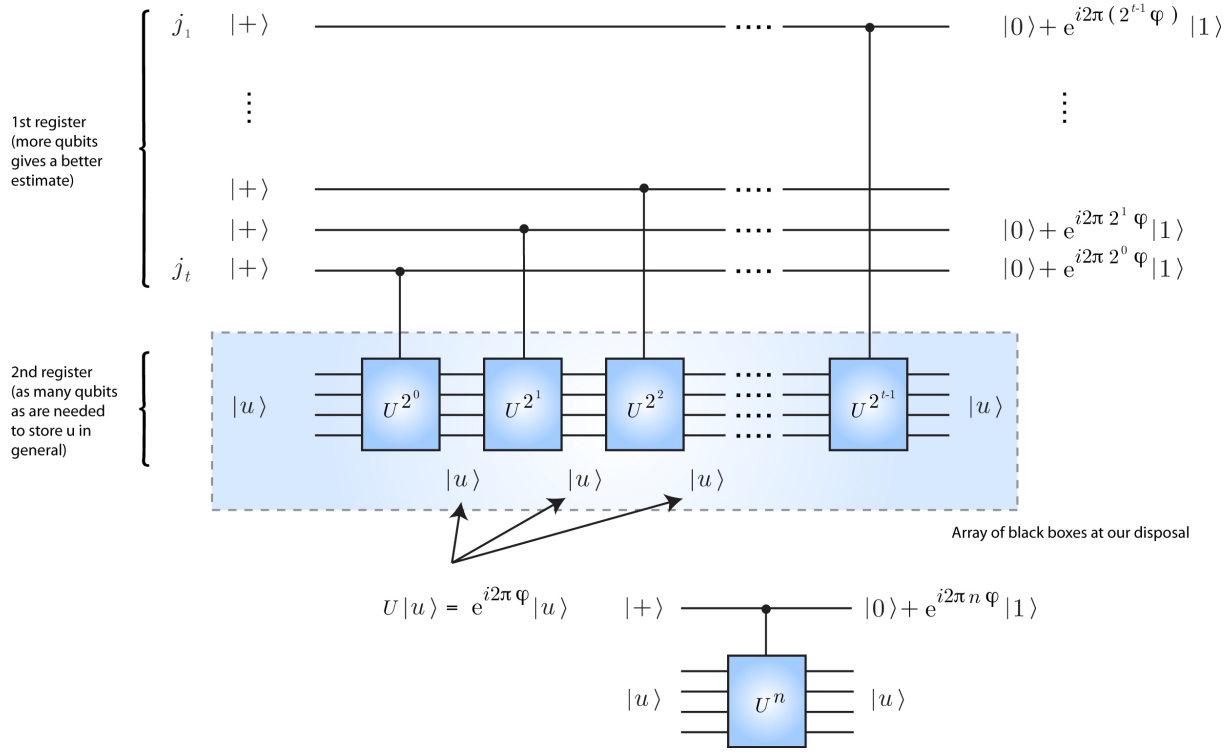

Figure 2: Quantum circuit representing the quantum phase estimation algorithm (QPE). The inset shows what happens in general to states input to one of the black boxes.

Note that $|j\rangle\,|u\rangle \overset{QPE}{\rightarrow} |j\rangle\,U^j\,|u\rangle$, *i.e.* if we input the arbitrary state $|j\rangle$ instead of $|+\rangle^{\otimes t}$ into the QPE circuit. Here, $|j\rangle = |j_1 j_2 j \ldots j_t\rangle$ and if $j_\ell = 1$ then $U^{2^{(t-\ell)}}$ is applied and therefore $U^{\sum_{\ell=1}^{t} j_\ell 2^{(t-\ell)}} = U^j$, by definition of the binary representation. Note that this is regardless of $|u\rangle$ as the operation depends on j only. We'll need these insights for later on.

Step 2

Let's assume that φ can be expressed in terms of exactly t bits and take the phases from the output states of the QPE circuit, inspecting them a bit closer

$$\begin{aligned} e^{i2\pi 2^0\varphi} &= e^{i2\pi\varphi} = e^{i2\pi 0.\varphi_1\varphi_2\ldots\varphi_t} \\ e^{i2\pi 2^1\varphi} &= e^{i2\pi 2^1(\varphi_1/2+\varphi_2/4+\cdots+\varphi_t/2^t)} \\ &= e^{i2\pi\varphi_1}e^{i2\pi(\varphi_2/2+\cdots+\varphi_t/2^{t-1})} = e^{i2\pi 0.\varphi_2\ldots\varphi_t} \\ &\vdots \\ e^{i2\pi 2^{t-1}\varphi} &= e^{i2\pi 2^{t-1}(\varphi_1/2+\varphi_2/4+\cdots+\varphi_t/2^t)} \\ &= e^{i2\pi\varphi_t/2} = e^{i2\pi 0.\varphi_t}. \end{aligned} \tag{9}$$

Tidying this up a bit, the output of the circuit for the first register is:

$$\frac{(|0\rangle + e^{i2\pi 0.\varphi_t}|1\rangle)(|0\rangle + e^{i2\pi 0.\varphi_{t-1}\varphi_t}|1\rangle)\ldots(|0\rangle + e^{i2\pi 0.\varphi_1\varphi_2\ldots\varphi_t}|1\rangle)}{2^{t/2}}. \tag{10}$$

If we apply the inverse quantum Fourier transform, QFT†, we get the state at the output registers as $|\varphi_1\ldots\varphi_t\rangle_1|u\rangle_2$. Thus, we can read off the value of the phase φ from a computational basis measurement of the first register! In general, if φ cannot be expressed in t-bits we have

$$\frac{1}{2^{t/2}}\sum_{j=0}^{2^t-1}|j\rangle|u\rangle \overset{QPE}{\rightarrow} \frac{1}{2^{t/2}}\sum_{j=0}^{2^t-1}e^{i2\pi\varphi j}|j\rangle|u\rangle \tag{11}$$

as $|j\rangle|u\rangle \overset{QPE}{\rightarrow} |j\rangle U^j|u\rangle$. After applying QFT† this gives the output state $|\tilde{\varphi}\rangle|u\rangle$, where $\tilde{\varphi}$ is a close estimate to φ in the computational basis. It turns out that in order to successfully obtain φ accurate to n bits with probability of success at least $1-\epsilon$ we must choose

$$t = n + \lceil \log_2(2 + \frac{1}{2\epsilon})\rceil, \tag{12}$$

where $\lceil x\rceil$ is the ceiling function, *i.e.* the smallest integer $\geq x$.

The quantum Fourier transform and quantum phase estimation algorithm on their own aren't really that useful. But, we'll see in the next lecture how they can be used as part of Shor's algorithm to break RSA and other types of public-key encryption that are widely used on the internet and in the military, industry and commerce.

References

- N. D. Mermin, *Quantum Computer Science: An introduction*, Cambridge University Press, Cambridge (2007).
- M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge (2000).

- J. Preskill, *Quantum Information lecture notes*, http://www.theory.caltech.edu/people/preskill/ph229/#lecture (2004).
- R. B. Griffiths and C.-S. Niu, *Semiclassical Fourier Transform for Quantum Computation*, Phys. Rev. Lett. **76**, 3228 (1996).
- M. Dobscek, G. Johansson, V. Shumeiko and G. Wendin, *Arbitrary accuracy iterative quantum phase estimation algorithm using a single ancillary qubit: A two-qubit benchmark*, Phys. Rev. A (Rapid) **76**, 030306 (2007).

Appendix

Proof of Eq. (5)

$$|j\rangle \overset{QFT}{\rightarrow} \frac{1}{2^{n/2}} \sum_{k=0}^{2^{n-1}} e^{i2\pi jk/2^n} |k\rangle = \frac{1}{2^{n/2}} \underbrace{\sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1}}_{\sum_{k=0}^{2^n-1}} e^{i2\pi j(\sum_\ell^n k_\ell 2^{-\ell})} |k_1 \ldots k_n\rangle \,, \tag{13}$$

where we've used the binary representation of k and $e^{i2\pi j 2^n} = 1, \forall j$. The right hand side of Eq. (13) can be written as

$$\begin{aligned} &= \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \otimes_{\ell=1}^{n} e^{i2\pi jk_\ell 2^{-\ell}} |k_\ell\rangle \\ &= \frac{1}{2^{n/2}} \otimes_{\ell=1}^{n} \left[\sum_{k_\ell=0}^{1} e^{i2\pi jk_\ell 2^{-\ell}} |k_\ell\rangle \right] \\ &= \frac{1}{2^{n/2}} \otimes_{\ell=1}^{n} \left(|0\rangle + e^{i2\pi j 2^{-\ell}} |1\rangle \right). \end{aligned} \tag{14}$$

Now we have that $j2^{-\ell} = \frac{1}{2^\ell}(j_1 2^{n-1} + j_2 2^{n-2} + \ldots)$, which for $\ell = 1$ gives $j_1 2^{n-2} + j_2 2^{n-3} + \ldots j_n 2^{-1}$. As $n \geq 2$, only the last term contributes to the phase factor in Eq. (14) and by definition $0.j_n = j_n/2$ from the binary fraction representation. Thus, the phase factor for $\ell = 1$ is $e^{i2\pi 0.j_n}$. For $\ell = 2$ we have $j_{n-1}2^{-1} + j_n 2^{-2} \equiv 0.j_{n-1}j_n$. Following through for all ℓ we recover the right hand side of Eq. (5) $\square$