Quantum information and computing

Lecture 7: Quantum Fourier transform

Jani-Petri Martikainen
Jani-Petri.Martikainen@helsinki.fi
http://www.helsinki.fi/~jamartik

Department of Physical Sciences
University of Helsinki
Demotivation

- Despite my promise, I don’t think we will be able to proceed to the factoring algorithm quite yet.
- Let us instead focus on the quantum Fourier transform...
Quantum algorithms and their relations

Quantum search

Fourier transform

Hidden subgroup problem

Quantum counting

Discrete log

Order–finding

Factoring

Search for crypto keys

Break cryptosystems (RSA)

Speedup for some NP problems

Statistics: mean, median, min
QFT and its applications

- Prime factorization of an $n$-bit integer takes $\exp(\mathcal{O}(n^{1/3} \log^{2/3} n))$ operations using the so-called number field sieve.

- This is exponential in $n$ so factorization is considered intractable problem for a classical computer.

- Quantum algorithm factorizes using $\mathcal{O}(n^2 \log n \log \log n)$ operations...this is exponentially faster!

- We now discuss the quantum Fourier transform which a key ingredient for factoring and many other interesting quantum algorithms.

- QFT does not, however, speed up the classical task of computing Fourier transforms.
Quantum Fourier transform

Discrete Fourier transform takes the input vector of complex numbers $x_0 \ldots x_{N-1}$ into a transformed vector

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi i j k / N}$$  \hspace{1cm} (1)

Quantum Fourier transform is the same transformation with somewhat different notation. QFT transform is a linear operator and acts on the orthonormal basis states $|0\rangle, \ldots, |N-1\rangle$ according to

$$|j\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k / N} |k\rangle$$  \hspace{1cm} (2)
Quantum Fourier transform

- Equivalently, the action on an arbitrary state is

\[
\sum_{j=0}^{N-1} x_j |j\rangle \rightarrow \sum_{k=0}^{N-1} x_k |k\rangle
\]  

(3)

where amplitudes \( y_k \) are the Fourier transform of amplitudes \( x_j \).

- Transformation is also unitary (show this) so it can be implemented as the dynamics for a quantum computer.

- Take \( N = 2^n \) and the computational basis \( |0\rangle \ldots |2^n - 1\rangle \) for an \( n \)-qubit quantum computer.

- Write the state \( |j\rangle \) in binary representation \( j = j_1j_2\ldots j_n \) so that \( j = j_12^{n-1} + j_22^{n-2} + \cdots + j_n2^0 \)
Quantum Fourier transform

- Also, we use the notation \( 0.j_l j_{l+1} \ldots j_m \) for the binary fraction \( j_l/2 + j_{l+1}/4 \ldots + j_m/2^{m-l+1} \)

- Fourier transform in the product representation

\[
|j_1, \ldots j_n\rangle \rightarrow \frac{(|0\rangle + e^{2\pi i 0.j_n} |1\rangle) (|0\rangle + e^{2\pi i 0.j_{n-1}j_n} |1\rangle) \ldots (|0\rangle + e^{2\pi i 0.j_1j_2\ldots j_n} |1\rangle)}{2^n/2}
\]

- This is very useful and you can even take this as a definition of the quantum Fourier transform

- Equivalence of Eqs. (2) and (4) follows from elementary algebra
Quantum Fourier transform

\[ \ket{j} \rightarrow \frac{1}{2^{n/2}} \sum_{k=0}^{2^{n-1}} e^{2\pi i j k / 2^n} \ket{k} = \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} e^{2\pi i j (\sum_{l=1}^{n} k_l 2^{-l})} \ket{k_1 \cdots k_n} \]

\[ = \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \bigotimes_{l=1}^{n} e^{2\pi i j k_l 2^{-l}} \ket{k_l} = \frac{1}{2^{n/2}} \bigotimes_{l=1}^{n} \left[ \sum_{k_l=0}^{1} e^{2\pi i j k_l 2^{-l}} \ket{k_l} \right] \]

\[ = \frac{1}{2^{n/2}} \bigotimes_{l=1}^{n} \left[ \ket{0} + e^{2\pi i j 2^{-l}} \ket{1} \right] \]

\[ = \frac{\left( \ket{0} + e^{2\pi i 0 \cdot j_n} \ket{1} \right) \left( \ket{0} + e^{2\pi i 0 \cdot j_{n-1} j_n} \ket{1} \right) \cdots \left( \ket{0} + e^{2\pi i 0 \cdot j_1 j_2 \cdots j_n} \ket{1} \right)}{2^{n/2}} \] (4)

Here we used on the first line that each \( k_i = \{0, 1\} \) in binary representation.

And on the last line that \( 2\pi i j 2^{-l} = 2\pi i \sum_{k=1}^{n} j_k 2^{n-k-l} \) from which we see that we have terms \( e^{2\pi i \times \text{integer}} = 1 \) if \( k \leq n - l \) and only \( k > n - l \) contribute non-trivially and give rise to the binary fractions of the last line.
Quantum Fourier transform

- The product representation of the Fourier transform makes it easy to derive an efficient circuit for the quantum Fourier transform

- We define

\[ R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix} \] (5)

(Note the special cases \( R_2 = S \) and \( R_3 = T \) which give us the phase and \( \pi/8 \) gates)

- SEE THE DIAGRAM!
Quantum Fourier transform

- Why it works? Take the $|j_1 \ldots j_n\rangle$ as an input. The first Hadamard gate produces a state

$$\frac{1}{2^{1/2}} \left( |0\rangle + e^{2\pi i 0 \cdot j_1} |1\rangle \right) |j_2 \ldots j_n\rangle$$

(6)

since $e^{2\pi i 0 \cdot j_1} = -1$ when $j_1 = 1$ and $+1$ otherwise.

- Then apply the controlled-$R_2$ gate on the first qubit and get

$$\frac{1}{2^{1/2}} \left( |0\rangle + e^{2\pi i 0 \cdot j_1 j_2} |1\rangle \right) |j_2 \ldots j_n\rangle$$

(7)

since $0 \cdot j_1 j_2 = j_1/2 + j_2/4$. The last term gives rise to the term that we need in $R_2$ operation and it gives something only if the 2nd qubit is set.
Quantum Fourier transform

We continue by applying the controlled-$R_3$ to controlled-$R_n$ gates to the first qubit and in the end we have a state

$$\frac{1}{2^{1/2}} (|0\rangle + e^{2\pi i 0.j_1j_2\ldots j_n}|1\rangle) |j_2 \ldots j_n\rangle$$

(8)

This is the last term in the product representation of the Fourier transformation.

Next we move to the 2nd qubit and apply the Hadamard gate to it...

$$\frac{1}{2^{2/2}} (|0\rangle + e^{2\pi i 0.j_1j_2\ldots j_n}|1\rangle) (|0\rangle + e^{2\pi i 0.j_2}|1\rangle) |j_3 \ldots j_n\rangle$$

(9)
Quantum Fourier transform

- Then the sequence of controlled-$\text{R}_2$ to $\text{R}_{n-1}$ gates produces the state

$$\frac{1}{2^{2/2}} \left( |0\rangle + e^{2\pi i 0.j_1j_2\ldots j_n} |1\rangle \right) \left( |0\rangle + e^{2\pi i 0.j_2\ldots j_n} |1\rangle \right) |j_3 \ldots j_n\rangle$$

- We continue in this way for each qubit and get

$$\frac{1}{2^{n/2}} \left( |0\rangle + e^{2\pi i 0.j_1j_2\ldots j_n} |1\rangle \right) \left( |0\rangle + e^{2\pi i 0.j_2\ldots j_n} |1\rangle \right) \ldots \left( |0\rangle + e^{2\pi i 0.j_n} |1\rangle \right)$$

- Using swap operations in the end we can reorder the qubits so that we have the desired output

$$\frac{1}{2^{n/2}} \left( |0\rangle + e^{2\pi i 0.j_n} |1\rangle \right) \left( |0\rangle + e^{2\pi i 0.j_{n-1}j_n} |1\rangle \right) \ldots \left( |0\rangle + e^{2\pi i 0.j_1j_2\ldots j_n} |1\rangle \right)$$

of the quantum Fourier transform
Quantum Fourier transform

- Incidentally this construction also proves that the quantum Fourier transform is unitary, since each gate in the circuit is unitary.

- Number of gates?: Hadamard + controlled rotations of the first qubit gives \( n \) gates, then there are \( n - 1 \) gates for the second qubit.

\[ n + (n - 1) + (n - 2) \cdots + 1 = \frac{n(n + 1)}{2} \] gates in total.

At most \( \frac{n}{2} \) swaps are required... so overall we find that this circuit provides an algorithm with \( \mathcal{O}(n^2) \) performance.

- In contrast, the best classical algorithm (fast Fourier transform) uses \( \mathcal{O}(n2^n) \) gates and therefore requires exponentially more operations to perform.

- Sounds terrific?
Quantum Fourier transform

Unfortunately, there is no known way to speed up the computation of Fourier transforms of classical data.

Amplitudes of the quantum computer cannot be directly accessed by measurement! So we cannot determine the Fourier amplitudes of the original data.

Also, there is in general no way to prepare the initial state efficiently.

Finding uses for the QFT is therefore more subtle.
Phase estimation

- The QFT is the key to a general procedure known as phase estimation.
- This in turn is the key for many quantum algorithms.
- Suppose that the unitary operator $U$ has an eigenvector $|u\rangle$ with an eigenvalue $e^{2\pi i \phi}$, where $\phi$ is unknown. In phase estimations we want to estimate $\phi$.
- Assume, for the moment, that we have black boxes (oracles) capable of preparing $|u\rangle$ and performing controlled-$U^{2^j}$ operations for non-negative integer $j$'s.
Phase estimation

- The use of black boxes indicates that phase estimation is not a complete quantum algorithm. Think of it as a module or a subroutine which can be used to perform interesting computational tasks.

- We use two registers. The first register contains $t$ qubits initially in the state $|0\rangle$.

- How we choose $t$ depends on the accuracy we require and on the probability of success we want...we will see how these things work out soon.

- The second register begins in the state $|u\rangle$ and contains as many bits as necessary to store $|u\rangle$. 

Department of Physical Sciences, University of Helsinki
http://theory.physics.helsinki.fi/~quantumgas/ - p. 16/25
Phase estimation

- Apply the circuit...SEE DIAGRAM

- First apply a Hadamard transformation on the first register, followed by the controlled-$U$ operation on the second register, with $U$ raised to successive powers of two.

- The first register ends up in the state

$$\frac{1}{2^{t/2}} \left( |0\rangle + e^{2\pi i 2^{t-1} \phi} |1\rangle \right) \left( |0\rangle + e^{2\pi i 2^{t-2} \phi} |1\rangle \right) \cdots \left( |0\rangle + e^{2\pi i 2^0 \phi} |1\rangle \right)$$

$$= \frac{1}{2^{t/2}} \sum_{k=0}^{2^t-1} e^{2\pi i \phi k} |k\rangle$$  \hspace{1cm} (10)

- The second register stays in the state $|u\rangle$ through out, since it was an eigenstate of $U$. 

Department of Physical Sciences, University of Helsinki

Phase estimation

The second stage of the phase estimation is to apply the inverse Fourier transform on the first register.

This is done by reversing the previous circuit that computed the Fourier transform. ($O(t^2)$ steps)

At third stage read out the state of the first register by doing measurement in the computational basis. This provides a pretty good estimate on $\phi$.

Why is this?

Suppose $\phi$ can be expressed exactly in $t$ bits as

$$\phi = 0.\phi_1\phi_2 \cdots \phi_t$$
Phase estimation

Then state of the first register prior to inverse FT is

\[ \frac{1}{2^{t/2}} \left( |0\rangle + e^{2\pi i 0 \cdot \phi_t} |1\rangle \right) \left( |0\rangle + e^{2\pi i 0 \cdot \phi_{t-1}} |1\rangle \right) \cdots \left( |0\rangle + e^{2\pi i 0 \cdot \phi_1 \phi_2 \cdots \phi_t} |1\rangle \right) \]

The second stage was the inverse FT, but comparing the above state with the definition of the Fourier transform (product form) shows that the output of the inverse FT is nothing but the state \( |\phi_1 \phi_2 \cdots \phi_t\rangle \) i.e. measurement in the computational basis gives \( \phi \) exactly!

In phase estimation we can estimate the phase \( \phi \) of the eigenvalue of \( U \) corresponding to the eigenstate \( |u\rangle \). At the heart of this we have the inverse FT performing

\[ \frac{1}{2^{t/2}} \sum_{j=0}^{2^t-1} e^{2\pi i \phi j} |j\rangle |u\rangle \to |\tilde{\phi}\rangle |u\rangle \quad (11) \]
Phase estimation: performance

What happens if $\phi$ cannot be expressed exactly in $t$ bits?

It turns out that then the above procedure gives a pretty good approximation to $\phi$ with high probability.

To see this, let $b$ be an integer in the range from 0 to $2^t - 1$ such that $b/2^t = 0.b_1\ldots b_t$ is the best $t$ bit approximation to $\phi$ which is less than $\phi$.

Therefore, difference $\delta = \phi - b/2^t$ satisfies $0 \leq \delta \leq 2^{-t}$.

Applying the inverse FT to the state in Eq.(10) gives

$$\frac{1}{2^t} \sum_{k,l=0}^{2^t-1} e^{-\frac{2\pi i kl}{2^t}} e^{2\pi i \phi k} |l\rangle$$

(12)
Phase estimation: performance

Let $\alpha_l$ be the amplitude of $|(b + l)\text{mod}2^t\rangle$ i.e.

$$\alpha_l = \frac{1}{2^t} \sum_{k=0}^{2^t-1} \left( e^{2\pi i (\phi-(b+l)/2^t)} \right)^k$$  \hspace{1cm} (13)

This is a geometric series so

$$\alpha_l = \frac{1}{2^t} \left( \frac{1 - e^{2\pi i (2^t\phi-(b+l))}}{1 - e^{2\pi i (\phi-(b+l)/2^t)}} \right) = \frac{1}{2^t} \left( \frac{1 - e^{2\pi i (2^t\delta-l)}}{1 - e^{2\pi i (\delta-l/2^t)}} \right)$$  \hspace{1cm} (14)

Suppose we measure $m$ in the end. We wish to bound the probability of obtaining a value $m$ such that $|m - b| > e$ where $e$ is the positive integer characterizing our tolerance for errors.
Phase estimation: performance

- The probability for such unfortunate measurement is

\[ p(|m-b| > e) = \sum_{-2^{t-1} \leq l \leq -(e+1)} |\alpha_l|^2 + \sum_{(e+1) \leq l \leq 2^{t-1}} |\alpha_l|^2 \]  \hspace{1cm} (15)

- For any real \( \theta \), \( |1 - \exp(i\theta)| \leq 2 \) so

\[ |\alpha_l| \leq 2/(2^t|1 - e^{2\pi i(\delta - l/2^t)}|) \]

- Also \( |1 - \exp(i\theta)| \geq 2|\theta|/\pi \) when \(-\pi \leq \theta \leq \pi\). But when \(-2^{t-1} \leq l \leq 2^{t-1}\) we have \(-\pi \leq 2\pi(\delta - l/2^t) \leq \pi\) and we find a bound

\[ |\alpha_l| \leq \frac{1}{2^{t+1}(\delta - l/2^t)} \]  \hspace{1cm} (16)
Phase estimation: performance

- Combining we find

\[ p(|m - b| > e) \leq \frac{1}{4} \left[ \sum_{l=-2^{t-1}+1}^{-(e+1)} \frac{1}{(l - 2^t \delta)^2} + \sum_{l=e+1}^{2^t-1} \frac{1}{(l - 2^t \delta)^2} \right] \]

(17)

- And then recalling that \(0 \leq 2^t \delta \leq 1\) we obtain

\[ p(|m - b| > e) \leq \frac{1}{4} \left[ \sum_{l=-2^{t-1}+1}^{-(e+1)} \frac{1}{l^2} + \sum_{l=e+1}^{2^t-1} \frac{1}{(l - 1)^2} \right] \]

\[ \leq \frac{1}{2} \sum_{l=e}^{2^t-1-1} \frac{1}{l^2} \leq \frac{1}{2} \int_{e-1}^{2^t-1-1} \frac{1}{l^2} \mathrm{d}l \]

\[ = \frac{1}{2} \int_{e-1}^{2^t-1-1} \frac{1}{l^2} \mathrm{d}l \]

\[ = 1/[2(e - 1)] \]

(18)
Phase estimation: performance

- Suppose we wish to approximate $\phi$ to an accuracy $2^{-n}$ so we choose $e = 2^{t-n} - 1$.

- By using $t = n + p$ qubits in the phase estimation algorithm we see that the probability of obtaining an approximation correct to this accuracy is at least $1 - 1/2(2^p - 2)$.

- Therefore, to obtain a good approximation to $\phi$ with probability of success at least $1 - \epsilon$ we choose

$$t = n + \left\lceil \log(2 + \frac{1}{2\epsilon}) \right\rceil$$  \hspace{1cm} (19)

- In order to use phase estimation algorithm we must be able to prepare the eigenstate $|u\rangle$. Suppose we don’t know how.
Phase estimation: performance

- Rather than preparing \( |u\rangle \) we prepare \( |\psi\rangle = \sum_u c_u |u\rangle \).
- Suppose that eigenstate \( |u\rangle \) has the eigenvalue \( e^{2\pi i \phi_u} \).
- Running the phase estimation algorithm will give a state \( \sum_u c_u |\tilde{\phi}_u\rangle |u\rangle \), where \( \tilde{\phi}_u \) approximates \( \phi_u \).
- Reading out the first register will give us a good approximation to \( \phi_u \), where \( u \) is chosen randomly with probability \( |c_u|^2 \).
- So we don’t have to be able to prepare the eigenstate perfectly, if we accept some additional randomness into the algorithm.