QUANTUM COMPUTATION AND QUANTUM INFORMATION: THE QUANTUM FOURIER TRANSFORM

1

We consider the linear map in $\mathbb{C}^N$ which acts on the computational basis as

$$|j\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2i\pi jk}{N}} |k\rangle$$

Let A be the matrix of the transformation in the computational basis.

$$\forall (k,l) \in [\![0, N-1]\!]^2, \quad a_{kl} = \frac{1}{\sqrt{N}} e^{\frac{2i\pi kl}{N}}$$

The adjoint matrix $A^\dagger$ is then

$$\begin{aligned}\forall (k,l) \in [\![0, N-1]\!]^2, \quad b_{kl} &= a_{lk}^* \\ &= \frac{1}{\sqrt{N}} e^{-\frac{2i\pi kl}{N}}\end{aligned}$$

We compute the coefficient k, l of the product $AA^\dagger$:

$$\begin{aligned}\forall (k,l) \in [\![0, N-1]\!]^2, \quad c_{kl} &= \sum_{j=0}^{N-1} a_{kj} b_{jl} \\ &= \frac{1}{N} \sum_{j=0}^{N-1} e^{\frac{2i\pi j}{N}(k-l)} \\ &= \frac{1}{N} \sum_{j=0}^{N-1} (e^{\frac{2i\pi}{N}(k-l)})^j \\ &= \begin{cases} \dfrac{1}{N} \dfrac{1-(e^{\frac{2i\pi}{N}(k-l)})^N}{1-e^{\frac{2i\pi}{N}(k-l)}} = 0 & \text{if } e^{\frac{2i\pi}{N}(k-l)} \neq 1, \\ 1 & \text{if } e^{\frac{2i\pi}{N}(k-l)} = 1. \end{cases} \\ &= \begin{cases} 0 & \text{if } k \neq l, \\ 1 & \text{if } k = l. \end{cases} \\ &= \delta_{kl}\end{aligned}$$

which shows that $AA^\dagger = A^\dagger A = I$ i.e. A is unitary.

2

Here the dimension of the state space is $N = 2^n$. The Fourier transform of the n qubit state $|00\dots0\rangle$ is

$$A|0\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle$$

we can write k in binary $k_{n-1} \dots k_1 k_0$

$$A|0\rangle = \frac{1}{2^{n/2}} \sum_{k_0, k1, \dots, k_{n-1}=0}^{1} |k_{n-1} \dots k_1 k_0\rangle$$

or in product representation,

$$= \frac{1}{2^{n/2}} \underbrace{(|0\rangle + |1\rangle)(|0\rangle + |1\rangle)\dots(|0\rangle + |1\rangle)}_{n \text{ qubits}}$$

3

Let $N = 2^n$ and $Y = (y_k)_{k\in[\![0,N-1]\!]}$ be the classical fourier transform of $X = (x_k)_{k\in[\![0,N-1]\!]}$.

$$\forall k \in [\![0, N-1]\!], \quad y_k = \sum_{j=0}^{N-1} e^{\frac{2i\pi kj}{2^n}} x_j$$

The factor $\frac{1}{\sqrt{N}}$ is omitted for clarity. We can write j in binary $j_{n-1}\dots j_1 j_0$

$$\begin{aligned}
y_k &= \sum_{j_0,j_1,\dots,j_{n-1}=0}^{1} e^{\frac{2i\pi k(2^{n-1}j_{n-1}+\dots+2j_1+j_0)}{2^n}} x_j \\
&= \sum_{j_1,\dots,j_{n-1}=0}^{1} e^{\frac{2i\pi k(2^{n-1}j_{n-1}+\dots+2j_1)}{2^n}} x_{j_{n-1}\dots j_1 0} + \sum_{j_1,\dots,j_{n-1}=0}^{1} e^{\frac{2i\pi k(2^{n-1}j_{n-1}+\dots+2j_1+1)}{2^n}} x_{j_{n-1}\dots j_1 1} \\
&= \sum_{j_1,\dots,j_{n-1}=0}^{1} e^{\frac{2i\pi k(2^{n-1}j_{n-1}+\dots+2j_1)}{2^n}} x_{j_{n-1}\dots j_1 0} + e^{\frac{2i\pi k}{2^n}} \sum_{j_1,\dots,j_{n-1}=0}^{1} e^{\frac{2i\pi k(2^{n-1}j_{n-1}+\dots+2j_1)}{2^n}} x_{j_{n-1}\dots j_1 1} \\
&= \sum_{j_1,\dots,j_{n-1}=0}^{1} e^{\frac{2i\pi k(2^{n-2}j_{n-1}+\dots+j_1)}{2^{n-1}}} x_{j_{n-1}\dots j_1 0} + e^{\frac{2i\pi k}{2^n}} \sum_{j_1,\dots,j_{n-1}=0}^{1} e^{\frac{2i\pi k(2^{n-2}j_{n-1}+\dots+j_1)}{2^{n-1}}} x_{j_{n-1}\dots j_1 1}
\end{aligned}$$

We see the first sum is the k^{th} coefficient of the FT of the sequence $(x_{2k})_{k\in[\![0,N/2-1]\!]}$ and the second is the k^{th} coefficient of the FT of $(x_{2k+1})_{k\in[\![0,N/2-1]\!]}$. This shows that to compute FT of sequence of length N, we have to compute 2 FT of sequence of length $\frac{N}{2}$ and do $2N$ complex additions/multiplications. The complexity of the operation $T(N)$ follows the recurrence:

$$T(N) = 2T(\frac{N}{2}) + 2N$$

We can use the Master theorem [1]:

Theorem. Let $a \geqslant 1$ and $b > 1$ be constants, let $f(n)$ be a function, and let $T(n)$ be defined on the non negative integers by the recurrence

$$T(n) = aT(\frac{n}{b}) + f(n)$$

where we interpret $\frac{n}{b}$ to mean either $\lfloor \frac{n}{b} \rfloor$ or $\lceil \frac{n}{b} \rceil$. Then $T(n)$ has the following asymptotic bounds:

(1) If $f(n) = O(n^{\log_b a-\epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.
(2) If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a}\log n)$.
(3) If $f(n) = \Omega(n^{\log_b a+\epsilon})$ for some constant $\epsilon > 0$, and if $af(\frac{n}{b}) \leqslant cf(n)$ for some constant $c < 1$ and n sufficiently large, then $T(n) = \Theta(f(n))$.

Here we are in the second case of the theorem, so $T(N) = \Theta(N\log(N)) = \Theta(n2^n)$.

[1] Thomas H. Cormen and Charles E. Leiserson : *Introduction to algorithms*, MIT Press (2009)