# Week 8: The Quantum Fourier Transform

COMS 4281 (Fall 2025)

# Admin

1. No worksheet/quiz this week.

# Last time

- Introduction to Quantum Fourier Transform
- Midterm

# Last time

- Consider the Hilbert space $\mathbb{C}^N$ with standard basis $\{|0\rangle, \ldots, |N-1\rangle\}$.
- Discrete Fourier Transform $F_N$ is a **unitary matrix** mapping standard basis $\{|0\rangle, \ldots, |N-1\rangle\}$ to Fourier basis $\{|f_0\rangle, |f_1\rangle, \ldots, |f_{N-1}\rangle\}$ where

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

where $\omega_N = \exp(2\pi i/N)$ is the ***N*'th root of unity**.

- Let $|\psi\rangle = \sum_{j=0}^{N-1} \psi_j |j\rangle$ be a signal represented in the **standard basis**.
- In the Fourier basis, $|\psi\rangle$ can be written as $|\psi\rangle = \sum_{j=0}^{N-1} \hat{\psi}_j |f_j\rangle$ for some **Fourier coefficients** $\hat{\psi}_j$.

- Let $|\psi\rangle = \sum_{j=0}^{N-1} \psi_j |j\rangle$ be a signal represented in the **standard basis**.
- In the Fourier basis, $|\psi\rangle$ can be written as $|\psi\rangle = \sum_{j=0}^{N-1} \hat{\psi}_j \, |f_j\rangle$ for some **Fourier coefficients** $\hat{\psi}_j$.
- Given the amplitudes $\{\psi_0, \psi_1, \ldots, \psi_{N-1}\}$, how to compute Fourier coefficients $\{\hat{\psi}_0, \hat{\psi}_1, \ldots, \hat{\psi}_{N-1}\}$?

They are related by a *unitary* transformation $F_N$ (which is the DFT). It maps $|j\rangle \mapsto |f_j\rangle$. The *inverse DFT* is $F_N^\dagger$, which maps $|f_j\rangle \rightarrow |j\rangle$.

Applying $F_N^\dagger$ to $|\psi\rangle = \sum_{j=0}^{N-1} \psi_j |j\rangle$ yields the vector

$$F_N^\dagger |\psi\rangle = \sum_{j=0}^{N-1} \hat{\psi}_j F_N^\dagger |f_j\rangle = \sum_{j=0}^{N-1} \hat{\psi}_j |j\rangle .$$

The **Fourier coefficients** of $|\psi\rangle$ have been turned into *amplitudes* in the standard basis of $|\hat{\psi}\rangle$.

# Quantum Fourier Transform

# Quantum Fourier Transform

Let $N = 2^n$. The **Quantum Fourier Transform (QFT)** is a quantum algorithm for implementing the $n$-qubit unitary $F_N$ (technically, its inverse $F_N^\dagger$) while taking only $\mathrm{poly}(n)$ time.

QFT maps quantum states $|\psi\rangle$ to their Fourier transform state $|\hat{\psi}\rangle = \sum_j \hat{\psi}_j |j\rangle$.

# Quantum Fourier Transform

Let $N = 2^n$. The **Quantum Fourier Transform (QFT)** is a quantum algorithm for implementing the $n$-qubit unitary $F_N$ (technically, its inverse $F_N^\dagger$) while taking only $\mathrm{poly}(n)$ time.

QFT maps quantum states $|\psi\rangle$ to their Fourier transform state $|\hat{\psi}\rangle = \sum_j \hat{\psi}_j |j\rangle$.

Here, we associate $|j\rangle$ with the $n$-qubit state $|j_1 j_2 \cdots j_n\rangle$, the binary representation of $j$.

# Quantum Fourier Transform

The best classical algorithm for computing DFT of a $N$-dimensional vector is called the **Fast Fourier Transform (FFT)**, which takes $O(N \log N)$ time.

Does this constitute an exponential speedup?

# Quantum Fourier Transform

Not exactly. The Fast Fourier Transform gets an input that is a **classical** $N$-dimensional vector $|\psi\rangle$, and outputs another $N$-dimensional vector $|\hat{\psi}\rangle$.

On the other hand, QFT gets an input vector in **quantum form**, and produces an output vector in quantum form.

Not exactly. The Fast Fourier Transform gets an input that is a **classical** $N$-dimensional vector $|\psi\rangle$, and outputs another $N$-dimensional vector $|\hat{\psi}\rangle$.

On the other hand, QFT gets an input vector in **quantum form**, and produces an output vector in quantum form.

The Fourier coefficients $\{\hat{\psi}_j\}_j$ are not readily accessible: measuring $|\hat{\psi}\rangle$ produces basis vector $|j\rangle$ with probability $|\hat{\psi}_j|^2$, but afterwards all other information about the Fourier coefficients is lost.

# The Fourier Transform unitary $F_2$

Example: $N = 2$ (single-qubit unitary)

$$F_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} .$$

We've seen this before...

# The Fourier Transform unitary $F_4$

Example: $N = 4$ (two qubits). Reordering the rows/columns according to $|00\rangle$, $|10\rangle$ , $|01\rangle$, $|11\rangle$.

# The Fourier Transform unitary $F_4$

Example: $N = 4$ (two qubits). Reordering the rows/columns according to $|00\rangle$, $|10\rangle, |01\rangle$, $|11\rangle$.

$$F_4 = \frac{1}{\sqrt{4}} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & i & -i \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -i & i \end{pmatrix} .$$

Can you spot recursive structure?

# The Fourier Transform unitary $F_4$

Example: $N = 4$ (two qubits). Reordering the rows/columns according to $|00\rangle$, $|10\rangle$, $|01\rangle$, $|11\rangle$.

$$F_4 = \frac{1}{\sqrt{4}} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & i & -i \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -i & i \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} F_2 & A_2 F_2 \\ F_2 & -A_2 F_2 \end{pmatrix}$$

where

$$A_2 = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

# The Fourier Transform unitary $F_N$

For general $N = 2^n$, if we order the columns where all the "even" columns are on the left, and all the "odd" columns are on the right, then

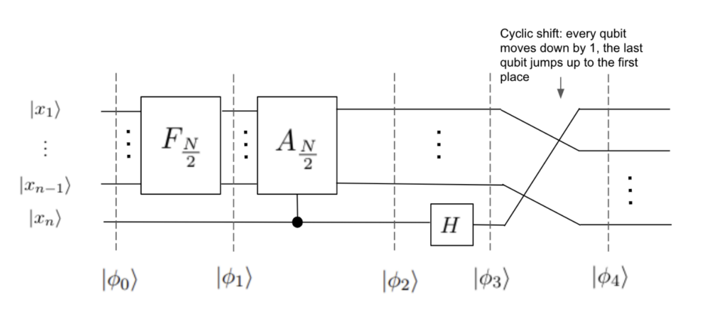
$$F_N = \frac{1}{\sqrt{2}} \begin{pmatrix} F_{\frac{N}{2}} & A_{\frac{N}{2}} F_{\frac{N}{2}} \\ F_{\frac{N}{2}} & -A_{\frac{N}{2}} F_{\frac{N}{2}} \end{pmatrix}$$

where

$$A_{N/2} = \begin{pmatrix} 1 & & & & \\ & \omega_N & & & \\ & & \omega_N^2 & & \\ & & & \ddots & \\ & & & & \omega_N^{N/2-1} \end{pmatrix}$$

# The Fourier Transform circuit

The recursive formula for $F_N$ inspires a recursive construction for the QFT circuit. Assume we already have circuits for the unitaries $F_{N/2}$ and $A_{N/2}$. Then the the circuit for $F_N$ looks like



# The Phase Circuit $A_{N/2}$

How is the circuit for $A_{N/2}$ implemented?

Note that the unitary acts on $n-1$ qubits, and for all $y \in \{0,1\}^{n-1}$ the unitary maps

$$|y_1, \ldots, y_{n-1}\rangle \mapsto \omega_N^{\mathrm{toint}(y)} |y_1, \ldots, y_{n-1}\rangle$$

where

$$\mathrm{toint}(y) = y_1 2^{n-2} + y_2 2^{n-3} + \cdots + y_{n-1}$$

# The Phase Circuit $A_{N/2}$

Expanding and regrouping we get

$$\begin{aligned}|y_1, \ldots, y_{n-1}\rangle &\mapsto \omega_N^{2^{n-2} \cdot y_1} \omega_N^{2^{n-3} \cdot y_2} \cdots \omega_N^{y_{n-1}} |y_1, \ldots, y_{n-1}\rangle \\ &= \left(\omega_N^{2^{n-2} \cdot y_1} |y_1\rangle\right)\left(\omega_N^{2^{n-3} \cdot y_2} |y_2\rangle\right) \cdots \left(\omega_N^{y_{n-1}} |y_{n-1}\rangle\right)\end{aligned}$$

# The Phase Circuit $A_{N/2}$

It is just a tensor product unitary!

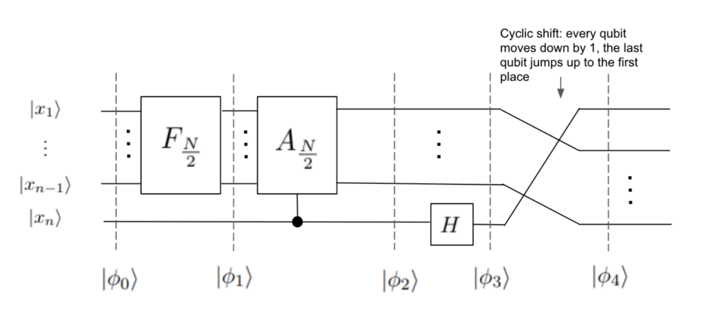
$$A_{N/2} = P(1/4) \otimes P(1/8) \otimes \cdots \otimes P(1/2^n)$$

where

$$P(\varphi) = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i \varphi} \end{pmatrix}$$

# The Fourier Transform circuit

**Complexity analysis**: Let $T(n)$ denote the number of gates used to construct $F_N$. It satisfies $T(n) = T(n-1) + O(n)$. Unrolling the recursion, we get $T(n) = O(n^2)$.



# Analysis of the Quantum Fourier Transform circuit

Why does it work?

It's a couple pages of algebra... take a look at the scribe notes, or in Nielsen/Chuang Chapter 5 if you're interested!

# Next time

Phase Estimation, the RSA Cryptosystem, and Shor's Algorithm.