CS 599 P1: Introduction to Quantum Computation — Boston University, Fall 2025
Instructor: Alexander Poremba — **Scribe:** Ethan Cappelleri

LECTURE #10: QUANTUM FOURIER TRANSFORM

In this lecture, we introduce the *Quantum Fourier Transform* (QFT). The QFT is a generalization of the Hadamard transform and plays a central role in several quantum algorithms, including Shor's factoring algorithm. This lecture will provide us with the necessary toolkit to move toward algorithms that provide genuine computational speedups without relying on oracle models.

1 From the Hadamard Transform to the Fourier Transform

The Hadamard gate is one of the simplest yet most important quantum gates: for $x \in \{0,1\}$, we have

$$H|x\rangle = \frac{1}{\sqrt{2}} \sum_{y\in\{0,1\}} (-1)^{x\cdot y}|y\rangle.$$

Intuitively, H maps the computational basis states $|0\rangle$ and $|1\rangle$ to equal superpositions:

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = |+\rangle, \qquad H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |-\rangle.$$

As we saw in previous lectures, the relative *phase* encodes information about the original state.

Mathematically, the Hadamard gate can be interpreted as the *Fourier transform over the group* $\mathbb{Z}_2$; this is the set $\{0,1\}$ which is equipped with a group operation corresponding to addition modulo 2. Note $\mathbb{Z}_2$ forms an abelian group, and H acts as its *character* transform, mapping group elements $x \in \mathbb{Z}_2$ to linear combinations of characters $\chi_y(x) = (-1)^{x\cdot y}$. As we will see, this perspective becomes especially useful when generalizing the notion of a Hadamard transform to multiple qubits or larger groups.

Multi-qubit Hadamard. For n qubits, the Hadamard gate extends naturally as the tensor product $H^{\otimes n}$, acting independently on each of the qubits:

$$H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} (-1)^{\langle x,y\rangle}|y\rangle,$$

where $\langle x,y\rangle = x_1y_1 \oplus x_2y_2 \oplus \cdots \oplus x_ny_n$ is the inner product modulo 2. This transformation turns computational basis states into equal superpositions, with relative phases determined by the bitwise inner product. In other words, $H^{\otimes n}$ performs a discrete Fourier transform over the group $(\mathbb{Z}_2)^n$.

This property is extremely useful in quantum algorithms: it allows one to *access information stored in the phases* of a quantum state, rather than just its amplitudes. In many algorithms, including Simon's and Shor's, this phase information is the key to extracting hidden structure efficiently. The *Quantum Fourier Transform* will provide us with the right tool to access phase information in higher-dimensional systems.

Example: Quantum information in the phase

Consider the single-qubit state

$$|\psi(\theta)\rangle = \frac{1}{\sqrt{2}}\big(\,|0\rangle + e^{i\theta}\,|1\rangle\,\big), \qquad \theta \in [0, 2\pi).$$

Measuring in the computational basis yields $|0\rangle$ or $|1\rangle$ with equal probability, giving no information about θ. However, applying a Hadamard before measurement transforms the state into

$$H\,|\psi(\theta)\rangle = \frac{1}{2}\Big((1 + e^{i\theta})\,|0\rangle + (1 - e^{i\theta})\,|1\rangle\,\Big),$$

so the measurement probabilities now depend explicitly on θ. This illustrates how we can use the Hadamard transformation to access *important phase information* hidden in the amplitudes.

Beyond Qubits: Toward Higher Dimensions. While the Hadamard works for qubits with computational basis $\{|0\rangle, |1\rangle\}$, we often encounter higher-dimensional systems (or "qudits") for $N \geq 2$ with basis states

$$\{|0\rangle, |1\rangle, \ldots, |N-1\rangle\}.$$

How can we generalize the Hadamard transform to higher-dimensional Hilbert spaces? The answer is the *Quantum Fourier Transform* (QFT), which generalizes the Hadamard from $N = 2$ to any dimension $N \geq 2$, operating over the group $\mathbb{Z}_N = \{0, 1, \ldots, N-1\}$ with addition modulo N.

2 The Discrete Fourier Transform

Before introducing the *Quantum Fourier Transform* (QFT), it is useful to recall its classical counterpart, the *Discrete Fourier Transform* (DFT). The DFT is a fundamental tool in signal processing, allowing us to decompose a discrete signal into its constituent frequency components. Intuitively, any discrete signal in time can be viewed as a sum of periodic oscillations of different frequencies, and the DFT provides a systematic method to extract these frequencies.

Formally, let $f : \{0, 1, \ldots, N-1\} \to \mathbb{C}$ be a discrete signal of length N (say, over N distinct *time* steps). The DFT produces a new sequence $\widehat{f} : \{0, 1, \ldots, N-1\} \to \mathbb{C}$ defined by

$$\widehat{f}(y) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} f(x)\, e^{2\pi i x y/N}, \qquad y = 0, 1, \ldots, N-1.$$

Here, $e^{2\pi i x y/N}$ are the *complex exponentials* that serve as the basis functions of the transform.

From a linear algebra perspective, the DFT can be interpreted as a change of basis in the vector space $\mathbb{C}^N$. The standard basis consists of vectors $\{\mathbf{e}_0, \mathbf{e}_1, \ldots, \mathbf{e}_{N-1}\}$, where $\mathbf{e}_x$ has a 1 in position x and zeros elsewhere. The DFT changes to the *Fourier basis* (or "frequency domain") consisting of vectors

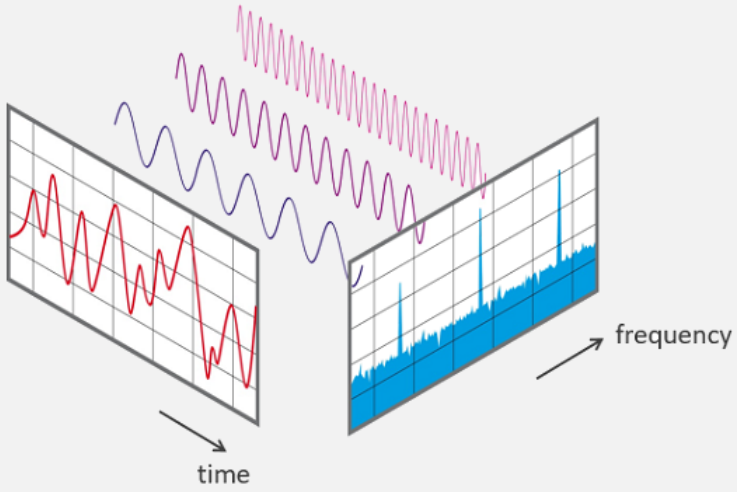
$$\mathbf{f}_y = \frac{1}{\sqrt{N}}\big(1, e^{2\pi i y/N}, e^{2\pi i 2y/N}, \ldots, e^{2\pi i (N-1) y/N}\big)^\top,$$

for $y = 0, 1, \ldots, N-1$. In this basis, the components $\widehat{f}(y)$ measure how strongly the corresponding frequency contributes to the original signal.

Thus, the DFT provides a compact and elegant way to encode frequency information, a perspective that generalizes directly to quantum states in the form of the QFT.

The Discrete Fourier Transform

The discrete Fourier transform[a] allows us to decompose a discrete signal (say, discrete time steps) as a sum of periodic oscillations across different frequencies which make up the original signal. This can be viewed as a *change of basis* from the *time domain* to the *frequence domain*, as in the figure below:[b]



[a] For some intuition, see this 3Blue1Brown video for how to visualize the Fourier transform:
https://www.youtube.com/watch?v=spUNpyF58BY

[b] Image credit: This figure was taken from the following source:
https://www.nti-audio.com/en/support/know-how/fast-fourier-transform-fft

Roots of Unity. A *primitive Nth root of unity* is defined as

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

This complex number satisfies $\omega_N^N = 1$ and generates all other Nth roots of unity through its powers:

$$1,\ \omega_N,\ \omega_N^2,\ \ldots,\ \omega_N^{N-1}.$$

These roots are the solutions to the polynomial equation $z^N = 1$ in the complex plane. They lie *equally spaced* around the unit circle, forming the vertices of a regular N-gon. Under multiplication, the set of roots

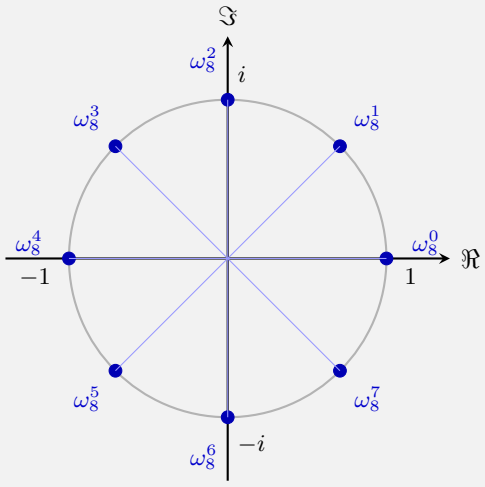
$$\{1, \omega_N, \omega_N^2, \ldots, \omega_N^{N-1}\}$$

forms a *cyclic group* of order N, with ω_N as a generator.

Roots of unity play a central role in Fourier analysis because the complex exponentials $e^{2\pi ixy/N} = \omega_N^{xy}$ appear in both the classical discrete Fourier transform and the quantum Fourier transform. They serve as the "basis functions" in which signals or quantum states are decomposed, allowing frequency components or phase information to be extracted systematically.

Example: 8th roots of unity

Let $N = 8$. Then, the 8th roots of unity $\omega_8^k = e^{2\pi i \frac{k}{8}}$ for $k = 0, 1, \ldots, 7$ partition the unit sphere on the complex plan into 8 equally spaced rotations in the complex plane.



Matrix Form of the DFT. Recall that a discrete signal $f : \{0, 1, \ldots, N-1\} \to \mathbb{C}$ can be thought of as a complex-valued vector in $\mathbb{C}^N$ with

$$f = \begin{pmatrix} f(0) \\ f(1) \\ \vdots \\ f(N-1) \end{pmatrix}.$$

Therefore, the DFT can be represented as a unitary change of basis $\mathsf{FT}_N : \mathbb{C}^N \to \mathbb{C}^N$ acting on vectors which over basis vectors labeled by $\mathbb{Z}_N$. Its entries are

$$(\mathsf{FT}_N)_{y,x} = \frac{1}{\sqrt{N}} \omega_N^{xy} = \frac{1}{\sqrt{N}} e^{2\pi i xy/N}, \qquad x, y = 0, 1, \ldots, N-1,$$

where $\omega_N = e^{2\pi i/N}$ is a primitive Nth root of unity. In matrix form, this gives

$$\mathsf{FT}_N = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & \omega & \omega^2 & \ldots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \ldots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \ldots & \omega^{(N-1)(N-1)} \end{pmatrix}.$$

Applying FT_N to a signal f explicitly expresses the DFT as a change of basis:

$$\widehat{f} = \mathsf{FT}_N f.$$

where the Fourier transform of f is once again a complex vector $\widehat{f} \in \mathbb{C}^N$ with

$$\widehat{f} = \begin{pmatrix} \widehat{f}(0) \\ \widehat{f}(1) \\ \vdots \\ \widehat{f}(N-1) \end{pmatrix}.$$

For $N = 2$, this reduces to the familiar single-qubit Hadamard gate:

$$\mathsf{FT}_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & \omega_2 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

where $\omega_2 = -1$. This highlights the connection between the DFT and the Hadamard transform: both are unitary changes of basis, with F_2 performing a Fourier transform over $\mathbb{Z}_2$.

3 Quantum Fourier Transform

The *Quantum Fourier Transform* (QFT) is the quantum analogue of the classical discrete Fourier transform. The QFT is a unitary operation acting on computational basis states $|x\rangle$ as

$$\mathsf{QFT}_N|x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i x y/N} |y\rangle, \qquad x = 0, 1, \ldots, N-1.$$

By linearity, the QFT extends naturally to any quantum state $|\psi\rangle \in \mathbb{C}^N$ as follows:

$$|\psi\rangle = \sum_{x=0}^{N-1} \alpha_x |x\rangle \quad \mapsto \quad \mathsf{QFT}_N|\psi\rangle = \sum_{y=0}^{N-1} \widehat{\alpha}_y |y\rangle, \quad \text{where} \quad \widehat{\alpha}_y = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} \alpha_x \, e^{2\pi i x y/N}.$$

Note that we can compactly represent the roots of unity as $\omega_N = e^{2\pi i/N}$ so that for $x, y \in \mathbb{Z}_N$:

$$\omega_N^{xy} = e^{2\pi i x y/N}.$$

Intuitively, the QFT maps the amplitudes α_x in the computational basis to new amplitudes $\widehat{\alpha}_y$ in the *Fourier basis*. This change of basis is what allows quantum algorithms to extract periodicity and phase information efficiently. It is straightforward to verify that QFT_N is *unitary* operation; this requires the following orthogonality property of the Fourier characters over $\mathbb{Z}_N$:

Fact: Orthogonality of Fourier characters

For any integer $N \geq 2$, the roots of unity $\omega_N = e^{2\pi i/N}$ give rise to so-called *Fourier characters* $\chi_y(x) = \omega_N^{x \cdot y} \in \mathbb{C}$ which satisfy the following orthogonality property:

$$\sum_{y \in \mathbb{Z}_N} \omega_N^{x \cdot y} \cdot \omega_N^{-x' \cdot y} = N \cdot \delta_{x,x'}, \qquad \text{for all } x, x' \in \mathbb{Z}_N.$$

Here, $\delta_{x,x'}$ is the Kronecker delta which is equal to 1, if $x = x'$, and equal to 0, if $x \neq x'$.

The unitarity of the Fourier transform ensures that inner products between vectors are preserved, which makes it a valid *change of basis*. Moreover, unitarity also ensures that the transformation is reversible, a key requirement for quantum computation.

4 Quantum Circuit Construction

In practice, we often restrict to $N = 2^n$ for the quantum Fourier transform. This simplifies both the analysis and the implementation. In this case, the computational basis states are labeled as

$$|0\rangle, |1\rangle, \ldots, |2^n - 1\rangle.$$

Each integer $y \in \{0, 1, \ldots, 2^n - 1\}$ can be expressed as an n-bit binary string

$$y = y_{n-1}y_{n-2}\ldots y_0, \quad \text{with } y = \sum_{k=0}^{n-1} y_k 2^k, \quad y_k \in \{0, 1\}.$$

4.1 Action of the QFT on a basis state

Applying the QFT to a basis state $|x\rangle$, with $x \in \mathbb{Z}_{2^n}$, gives

$$|x\rangle \quad \mapsto \quad \mathsf{QFT}_N |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{Z}_{2^n}} \omega_{2^n}^{x \cdot y} |y\rangle, \qquad \omega_{2^n} = e^{2\pi i/2^n}.$$

Using the binary decomposition of y, we can rewrite the exponent as

$$x \cdot y = x \cdot \sum_{k=0}^{n-1} y_k 2^k,$$

so that the QFT maps a logical basis state $|x\rangle$ to its Fourier transform via

$$|x\rangle \mapsto \mathsf{QFT}_N |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{Z}_{2^n}} \prod_{k=0}^{n-1} \omega_{2^n}^{x \cdot y_k 2^k} |y_{n-1}, \ldots, y_0\rangle.$$

4.2 Factorization into qubit states

Since each $y_k \in \{0, 1\}$, the sum over y factorizes into independent sums over each qubit:

$$|x\rangle \longmapsto \frac{1}{\sqrt{2^n}} \bigotimes_{k=0}^{n-1} \left(\sum_{y_k=0}^{1} \omega_{2^n}^{x \cdot y_k 2^k} |y_k\rangle \right).$$

Next, by introducing the single-qubit rotations as

$$|z_k\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle + \omega_{2^n}^{x \cdot 2^k} |1\rangle \right), \quad \text{for } k = 0, 1, \ldots, n-1.$$

we can write the output as

$$|x\rangle \longmapsto |z_{n-1}\rangle \otimes |z_{n-2}\rangle \otimes \cdots \otimes |z_0\rangle.$$

Notice that for $y_k = 0$ or 1, the amplitude is either 1 or $\omega_{2^n}^{x \cdot 2^k}$, so each factor is essentially a Hadamard gate followed by controlled phase rotations, encoding the contribution of higher-order bits of x into the phase.

A more explicit expression using the binary expansion of x as $x = x_{n-1} \ldots x_0$ is

$$|z_k\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i \sum_{j=0}^{n-1} x_j 2^{k-j-n}} |1\rangle \right),$$

which clearly shows how the phase for qubit k depends on all higher-order bits x_j with $j \leq k$.

4.3 Circuit implementation

This factorization into single-qubit rotation states $\{|z_k\rangle\}$ leads to an efficient QFT circuit which is made up of only single-qubit Hadamard gates and controlled phase rotations:

- Apply a Hadamard gate to the most significant qubit. In quantum circuit notation, this is the gate

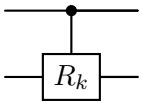
- For each qubit j, apply controlled rotations with respect to $R_k = \mathrm{diag}(1, e^{2\pi i/2^k})$ with

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix} = |0\rangle\langle 0| + e^{2\pi i/2^k}|1\rangle\langle 1|$$

  from all qubits with higher significance (i.e., $k < j$), where the controlled-R_k operation is given by

$$\text{controlled-}R_k = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes R_k.$$

  In quantum circuit notation, this is the gate represented by

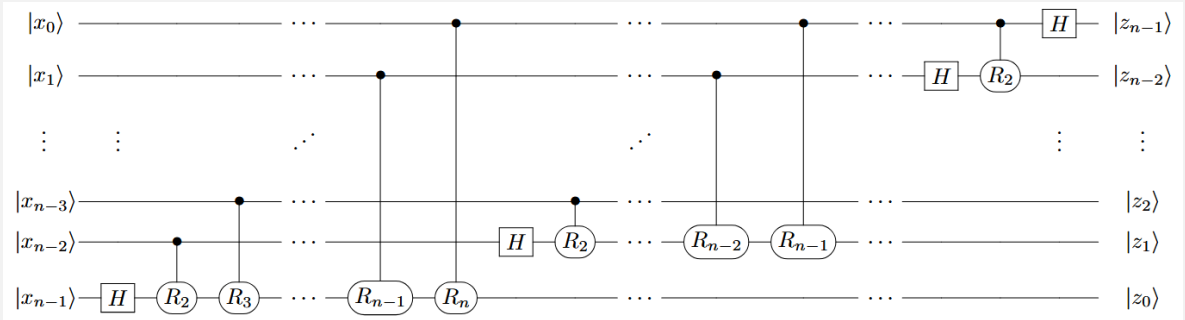


- Repeat this for all qubits in order of decreasing significance.
- (Optional:) Apply a *bit-reversal* at the end to reorder the output qubits into standard binary order.

Quantum circuit construction. Below, is the full quantum circuit for QFT_{2^n}.

Quantum circuit: Quantum Fourier Transform over $\mathbb{Z}_{2^n}$.

The following quantum circuit on n qubits is an exact and efficient implementation of the quantum Fourier transform QFT_{2^n}. The circuit relies on the previous factorization into single-qubit states and relies only on single-qubit Hadamard gates and controlled phase rotations:



Circuit complexity. This construction shows that the QFT can be implemented efficiently using $O(n^2)$ gates, and with approximate small-angle rotations, the complexity can be reduced further to $O(n \log n)$. This efficient implementation is central to the success of algorithms like Shor's factoring algorithm.