# Fourier Transform
# Lecture 2

Yonghao Lee

November 9, 2025

# Contents

# 1 The Core Idea: Signals and Bases

The main idea of this lecture is that any signal (like an image or a sound file) can be represented in multiple ways, called "bases." A **transform** is simply a change from one basis to another.

## 1.1 Standard Basis (Local)

This is the normal, pixel-by-pixel or sample-by-sample representation. Each data point describes a single, **local** point in space or time.

## 1.2 Fourier Basis (Global)

This represents the *exact same signal* as a **weighted sum of sine and cosine waves**. Each wave in this "recipe" is **global**, meaning it stretches across the entire signal.

# 2 The Fourier Transform

**Key Concept 2.1** (Fourier's Idea)**:** *The* ***Fourier Transform*** *is the mathematical tool (the "recipe book") that tells you exactly which frequencies, amplitudes, and phases are needed to build any given signal from a sum of simple waves.*

*It was based on Joseph Fourier's 1807 idea that* ***any periodic function*** *can be perfectly represented as a sum of sines and cosines.*

# 3 The Math Toolkit: Waves & Complex Numbers

To work with waves efficiently, we use two key tools:

## 3.1 Properties of a Wave

- **Amplitude ($A$):** The height or "strength" of the wave.
- **Frequency ($\omega$):** The number of cycles per second (pitch).
- **Phase ($\varphi$):** The starting offset or "shift" of the wave.

## 3.2 Euler's Formula

Working with sines and cosines separately is clumsy. Euler's formula provides a compact way to store both the amplitude and phase of a wave in a single **complex number**.

$$e^{i\alpha} = \cos(\alpha) + i\sin(\alpha) \tag{1}$$

A complex number $z = R \cdot e^{i\alpha}$ can represent a wave where:

- $R$ **(Magnitude):** Represents the wave's **Amplitude** ($R = \sqrt{a^2 + b^2}$).

- $\alpha$ **(Phase):** Represents the wave's **Phase** ($\alpha = \tan^{-1}(b/a)$).

# 4 The 1D Discrete Fourier Transform (DFT)

For digital signals (a discrete list of $N$ samples), we use the DFT.

## 4.1 The Formulas

The DFT is a pair of equations that let you move between the "time domain" and the "frequency domain."

- **Forward DFT (Analysis):** $f(x) \to F(u)$
  This formula **analyzes** a signal $f(x)$ (time domain) and finds its frequency "recipe" $F(u)$ (frequency domain).
  $$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{\frac{-2\pi i u x}{N}} \tag{2}$$
- **Inverse DFT (Synthesis):** $F(u) \to f(x)$
  This formula **reconstructs** the original signal $f(x)$ by adding up all the sine/cosine waves from its recipe $F(u)$.
  $$f(x) = \sum_{u=0}^{N-1} F(u) e^{\frac{2\pi i u x}{N}} \tag{3}$$

## 4.2 Key Properties of the DFT

- $F(0)$**:** The first coefficient ($u = 0$) is the **average value** of the entire signal.
- **Linearity:** The transform of a sum of signals is just the sum of their individual transforms. This is why a complex signal with three frequencies produces a spectrum with three distinct peaks.
- **Symmetry (for Real Signals):** When the input signal is "real" (has no imaginary parts, like all audio), the resulting frequency spectrum is symmetric. The magnitude is a mirror image: $|F(u)| = |F(N-u)|$.
- **Redundancy:** Because of this symmetry, all the information is contained in just the first half of the spectrum (from 0 to $N/2$). The second half is a redundant mirror image.

# 5 The Problem: Stationary vs. Non-Stationary Signals

The standard DFT has one critical, show-stopping flaw.

- **Stationary Signal:** A signal whose frequency content is constant over time (e.g., three notes played as a chord). The DFT works perfectly here.

- **Non-Stationary Signal:** A signal whose frequency *changes* over time (e.g., three notes played as a melody, one after another).

***Note:*** *The standard DFT* ***cannot tell the difference*** *between the chord and the melody. It provides* ***frequency localization*** *(it tells you what frequencies were present) but* ***NO time localization*** *(it cannot tell you when they occurred).*

# 6 The Solution: The Short-Time Fourier Transform (STFT)

To analyze real-world, non-stationary signals like speech and music, we need to know *when* the frequencies change. The **STFT** solves this.

## 6.1 The STFT Process

The idea is to slice the signal into small, overlapping windows and analyze each window individually, assuming the signal is "stationary" *inside* that short window.

1. **Window:** Choose a small time window (e.g., 25 milliseconds).
2. **FFT:** Apply the Fourier Transform (using the fast **FFT** algorithm) to the signal *only inside* that window.
3. **Slide:** Move the window a small amount to the right (a "hop," e.g., 10ms).
4. **Repeat:** Go back to step 2 and repeat this process for the entire length of the signal.

# 7 The Spectrogram: Visualizing the STFT

**Definition 7.1** (Spectrogram)**:** *The result of the STFT is a 2D map called a* ***Spectrogram****. It is a heatmap with:*

- ***X-axis:*** *Time*
- ***Y-axis:*** *Frequency*
- ***Color:*** *Amplitude/Energy (e.g., blue is "cold" = 0 energy, red/yellow is "hot" = high energy)*

## 7.1 Spectrogram in Practice

- **Log Scale:** Human hearing is logarithmic. A raw, linear-scale spectrogram hides most of the details. We almost always plot the **log of the magnitude** ($\log(|F(u)| + 1)$) to "boost" the quiet-but-important details and make them visible.
- **The Time-Frequency Tradeoff:** You must choose a window size, which involves a critical compromise:

  - **Long Window (Narrowband):** Gives you excellent **frequency resolution** (you can see exact musical pitches as clear horizontal lines) but poor **time resolution** (you don't know exactly when a sound event happened).
  - **Short Window (Wideband):** Gives you excellent **time resolution** (you can see percussive "clicks" as clear vertical lines) but poor **frequency resolution** (the frequencies get smeared into blurry horizontal bands).

# 8 Applications

Once you have the spectrogram, you can manipulate it and then reconstruct the audio.

## 8.1 Audio Equalizer

An equalizer is a direct spectrogram manipulator. The sliders for different frequency bands (e.g., 64 Hz, 1K Hz) are just a set of multipliers that are applied to the corresponding **rows (frequencies)** of the spectrogram.

## 8.2 Speech Fast-Forward (Time-Scale Modification)

- **Naïve Method:** Just playing audio samples faster (or dropping them) also raises the frequency, creating the "chipmunk effect."
- **STFT Method:**
  1. Calculate the spectrogram.
  2. "Sample" the spectrogram in the time dimension by **dropping columns** (e.g., throw away every other time-slice). This makes the spectrogram shorter (faster) in time.
  3. Reconstruct the audio from this new, shorter spectrogram.
- **Result:** Because the frequency information (the Y-axis) was never changed, the **pitch remains the same**.

  ***Note:*** *This reconstruction requires a complex step called "phase correction" to make sure the new, squeezed-together windows add up properly without canceling each other out.*