A Fast-Intuition Guide to Fourier Analysis: Continuous, Discrete, and the FFT

Z.P.

December 11, 2025

Contents

1 Philosophy and Big Picture

1.1 Why Fourier Analysis Exists

Central Philosophy

Any sufficiently nice function — a signal, a physical field, a vibration, a sound waveform, or even financial time series — can be re-expressed as a superposition of oscillations (in a linear system, *superposition* means that the total output is the sum of the outputs produced by each component separately). Each oscillation has a frequency, an amplitude, and a phase. This representation is powerful because moving between the "time domain" (how the function behaves moment-by-moment) and the "frequency domain" (how much of each oscillatory mode the function contains) reveals structure and meaning that is otherwise hidden.

Fourier analysis was historically motivated by questions about heat conduction (Fourier, 1822), but has since become a unifying language of mathematics, physics, and engineering. The central conceptual leap is that oscillations are not merely special signals; they form a *basis* in which *arbitrary* signals can be expanded.

The word "basis" should immediately evoke the geometry of vector spaces: once a basis is known, any element of the space can be expressed uniquely as a linear combination of the basis vectors. Fourier analysis extends this idea to functions. Instead of Euclidean vectors and basis vectors like $(1, 0)$ and $(0, 1)$, we have infinite-dimensional function spaces and elementary oscillatory modes such as $e^{i\omega t}$.

The frequency-domain description of a signal is often more interpretable than the original time-domain description. Slowly varying trends correspond to low-frequency content, rapid fluctuations correspond to high-frequency content, and periodicities show up as sharp spectral peaks. Thus, Fourier analysis does not merely compute coefficients; it *reveals structure*.

From the point of view of computation, this representation allows efficient manipulation of convolution, filtering, smoothing, and system identification. Many operations that are costly in the time domain become cheap in the frequency domain because convolution turns into multiplication:

$$\mathcal{F}(f * g) = \mathcal{F}(f) \cdot \mathcal{F}(g).$$

This duality is central to modern digital signal processing, image processing, numerical PDEs, and communication systems.

Overall, Fourier analysis exists because it:

- provides a universal coordinate system for functions;
- reveals periodicity, structure, and complexity by decomposing signals into constituent oscillatory modes;

- dramatically simplifies operations such as convolution, filtering, and differential equations when viewed in the frequency domain;
- offers both theoretical clarity and computational efficiency (especially via the FFT).

1.2 Signals, Information, and Decomposition

When we say a signal "contains information," we implicitly mean that some aspect of the signal is relevant while other aspects represent noise, redundancy, or high-frequency detail. Fourier decomposition separates a function into components that naturally isolate different scales of variation.

Two important conceptual statements:

Key Intuition

Decomposition into oscillatory modes is not arbitrary. The set of exponentials $\{e^{i2\pi kt/T}\}$ is orthogonal under a natural inner product. Orthogonality guarantees that the coefficients we compute are the *right ones* — the projection of the signal onto each oscillatory mode. It also guarantees that reconstruction is exact.

Why is orthogonality so essential? In finite-dimensional vector spaces, if we decompose a vector along non-orthogonal directions, computing the coefficients becomes unstable and reconstruction is harder. Orthogonality makes projection stable, unique, and conceptually clear.

Similarly, in function spaces, oscillatory modes $e^{i\omega t}$ are mutually orthogonal over an appropriate interval. Orthogonality ensures that the Fourier coefficients

$$A(\omega) = \langle f, e^{i\omega t} \rangle$$

truly measure *how much* of frequency ω is present in the signal. Because the inner product isolates exactly one oscillatory component at a time, reconstruction by summing all components is lossless.

Thus, Fourier analysis is not merely a computational trick: it is the canonical orthogonal decomposition of a surprisingly wide class of functions. The fact that exponentials are eigenfunctions of many linear time-invariant systems only strengthens the interpretation: physical systems "naturally speak Fourier."

1.3 Continuous vs Discrete Views

Broadly speaking, Fourier analysis comes in two flavors:

1. The *continuous Fourier transform (CFT)*, which describes non-sampled, real-time or spatially continuous signals.
2. The *discrete Fourier transform (DFT)*, which describes sampled, finite-length sequences, and is the theoretical foundation of digital signal processing.

At first glance, these two settings may look unrelated. One involves integrals, the other involves finite sums over N samples. However, they are deeply connected. The DFT is not a mere approximation of the continuous transform — it is the exact orthogonal decomposition of a vector in $\mathbb{C}^N$ using a finite set of harmonic exponentials:

$$X_k = \sum_{n=0}^{N-1} x_n \, e^{-2\pi ikn/N}.$$

Sampling theory provides the precise bridge between the continuous and discrete worlds. Under proper bandlimiting assumptions (Nyquist), a continuous-time signal can be perfectly captured by its samples, and the DFT provides frequency-domain access to the same fundamental information contained in the CFT. Without proper sampling, aliasing folds high-frequency content onto lower frequencies and distorts interpretation.

This dual perspective becomes especially powerful when paired with the Fast Fourier Transform (FFT), which makes the DFT computationally practical at scale.

1.4 The Fundamental Idea

The philosophical core of Fourier theory is the following statement:

$$f(t) = \sum_k A_k \, e^{i2\pi kt/T}.$$

The expression should be read as follows:

- Each term $e^{i2\pi kt/T}$ is a pure oscillation at frequency k/T; - The coefficient A_k tells us how much of this oscillation is present; - Adding all of them reproduces the original function $f(t)$ exactly (under suitable assumptions about smoothness or integrability).

Why these exponentials? Because the complex exponential is a natural oscillatory building block:

$$e^{i\omega t} = \cos(\omega t) + i \sin(\omega t),$$

encoding both sine and cosine in a single analytic object. Most importantly, exponentials are *orthogonal* over an interval, and they are eigenfunctions of the differential operator:

$$\frac{d}{dt} e^{i\omega t} = i\omega e^{i\omega t}.$$

Thus, if a physical system is governed by linear differential equations with constant coefficients, the solutions propagate oscillations independently. Each frequency evolves on its own — a profound structural simplification.

Oscillatory Building Blocks

Each $e^{i\omega t}$ (or $\cos(\omega t)$) is a pure oscillation at frequency ω. Every real-world signal can be decomposed into a linear combination of such building blocks. Higher frequencies oscillate faster; lower frequencies oscillate more slowly.

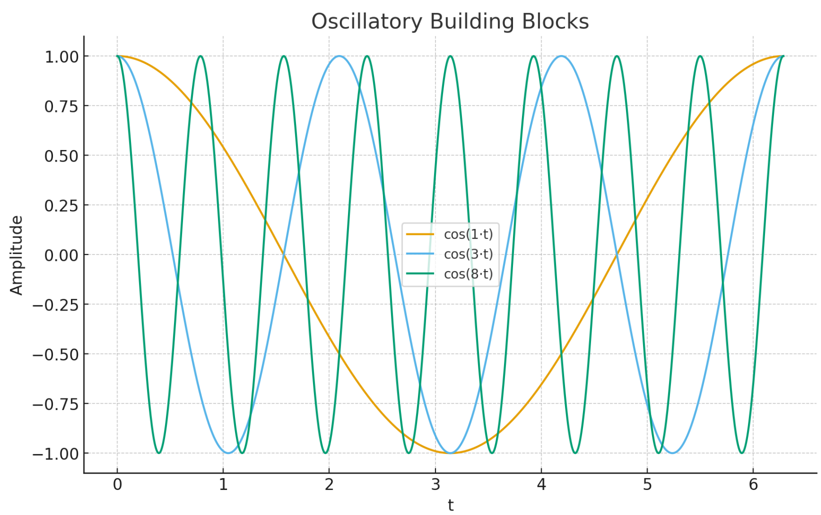

Figure 1: Individual oscillatory building blocks: three pure tones $\cos(\omega t)$ at different frequencies.

Superposition

Adding oscillatory building blocks is like adding musical notes: the total waveform is just a linear combination of different frequencies. The coefficients in a Fourier representation control the amplitude and phase of each note. The entire complexity of $f(t)$ is produced by superposition.

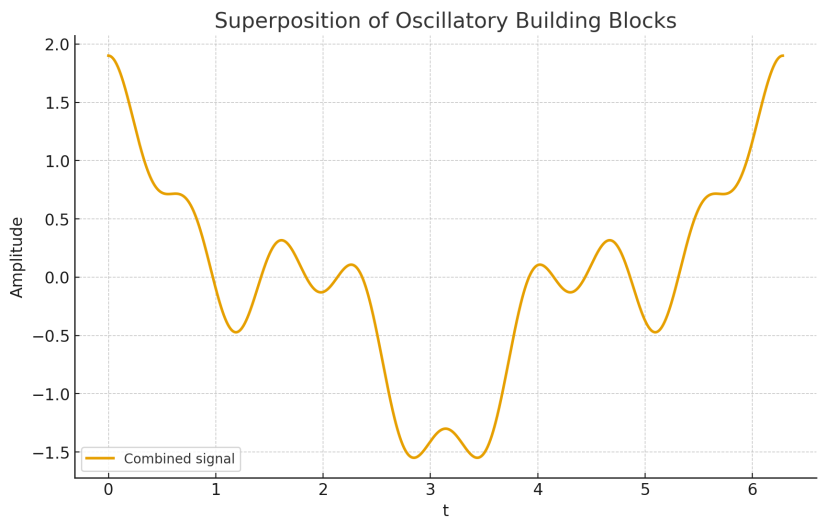

Figure 2: Superposition of oscillatory building blocks: combining different frequencies yields a complex waveform.

Oscillation as Rotation

A complex exponential $e^{i\omega t}$ is most naturally seen as a phasor: a constant-length vector rotating at angular speed ω in the complex plane. If we stack this rotation over time, we obtain a helix wrapping around a cylinder. The real part is $\cos(\omega t)$, the imaginary part is $\sin(\omega t)$. Thus, a sinusoid is the *shadow* of a rotating phasor. Frequency is spin rate, amplitude is radius, and phase is angular offset.

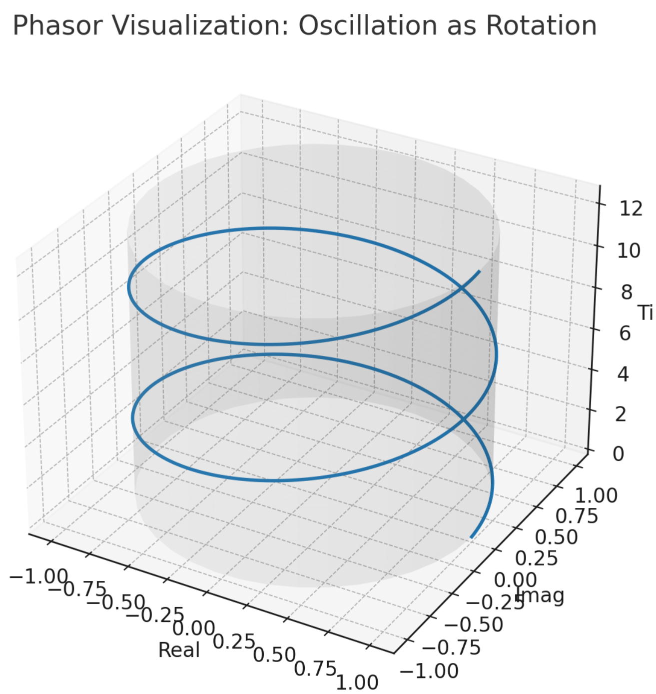

Figure 3: Phasor visualization: a pure oscillation $e^{i\omega t}$ as a rotating vector in the complex plane. As time advances, the trajectory forms a helix around a cylinder. The waveform observed in time is the projection of this rotation onto the real axis.

Superposition in the Complex Plane

Each pure oscillation $e^{i\omega t}$ is a rotating phasor of constant magnitude. When two phasors with close frequencies are added, the resultant vector does not rotate uniformly — its length slowly grows and shrinks. This slow radial modulation is the beat frequency. The real-time waveform is simply the projection of this evolving phasor onto the real axis.

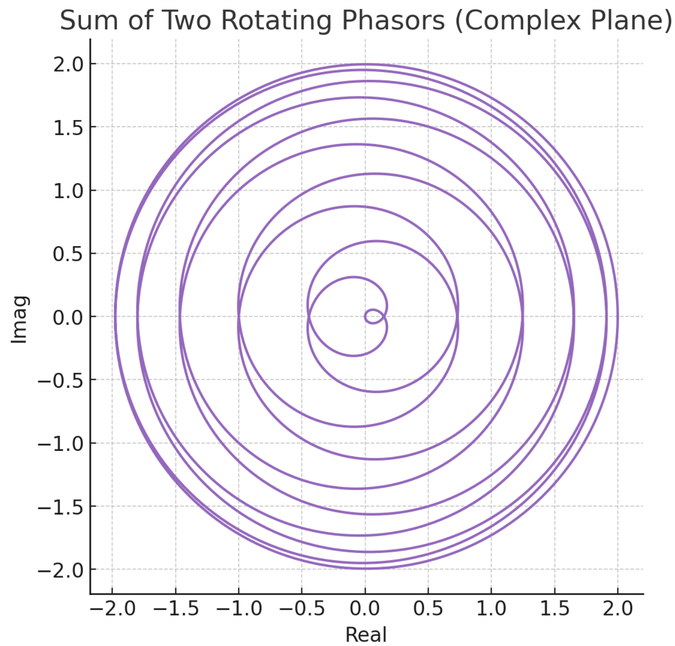

Figure 4: Sum of two rotating phasors in the complex plane. Close frequencies yield a slowly varying resultant amplitude, creating a spiral-like pattern.

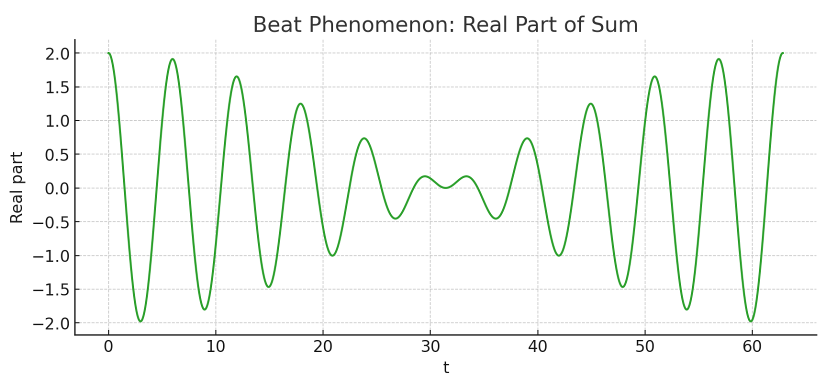

Figure 5: Beat phenomenon: the real part of the sum shows slow modulation of amplitude. The beat frequency is $|\omega_1 - \omega_2|$.

Why Exponentials Form a Basis

If you perturb a physical system, its long-term response often decomposes into damped or pure oscillatory modes. Each mode evolves independently because the exponential is an eigenfunction of the underlying operator. That is why Fourier modes enter everywhere: waves, heat propagation, acoustics, quantum mechanics, electrical circuits, time series analysis, and more.

1.5 Geometry of Oscillations

A pure tone $e^{i\omega t}$ can be viewed as uniform circular motion in the complex plane. At every time t, the value of $e^{i\omega t}$ is a point on the unit circle making angular speed ω:

$$e^{i\omega t} = \cos(\omega t) + i\sin(\omega t).$$

Geometrically:

- The real part is horizontal projection; - The imaginary part is vertical projection; - Magnitude is constant (always 1); - Phase grows linearly with t.

Thus, adding two oscillations is simply adding two rotating vectors. A complicated signal corresponds to the superposition of many rotating vectors with different angular speeds and amplitudes. The Fourier transform is the process of identifying these speeds and amplitudes.

From this perspective, frequency-domain coefficients describe how strongly each individual rotating vector contributes to the global motion. Low-frequency vectors rotate slowly; high-frequency vectors rotate rapidly. If a function varies slowly in time, then most of its energy is associated with slowly rotating vectors (low-frequency content). Conversely, sharp edges or rapid fluctuations correspond to significant high-frequency content.

This geometric viewpoint is extremely useful when developing intuition for:

- Why cosine waves can reconstruct arbitrary signals,
- Why convolution becomes multiplication,
- Why phase matters (alignment of rotating vectors),
- Why spectral leakage occurs when signals are not perfectly periodic.

2 Mathematical Preliminaries

2.1 Complex Numbers and Polar Form

Complex numbers play a foundational role in Fourier analysis because oscillations are expressed naturally using exponentials of the form $e^{i\omega t}$. Any complex number $z = x + iy$ can be represented in two ways:

$$z = x + iy \quad \text{(Cartesian form)}, \qquad z = r\,e^{i\theta} \quad \text{(polar form)}.$$

Here, $r = |z| = \sqrt{x^2 + y^2}$ is the *magnitude* and $\theta = \arg(z)$ is the *phase angle*. Polar form gives a geometric interpretation: r tells you how long the complex number is, and θ tells you its direction in the complex plane.

Geometric Interpretation of Complex Numbers

A complex number is not just a pair (x, y) or a symbol $x + iy$ — it is a point (or arrow) in the complex plane. The polar form $re^{i\theta}$ expresses that point in terms of its length r and direction θ. Multiplying complex numbers adds angles and scales lengths, making exponential notation the natural language for oscillations and rotations.

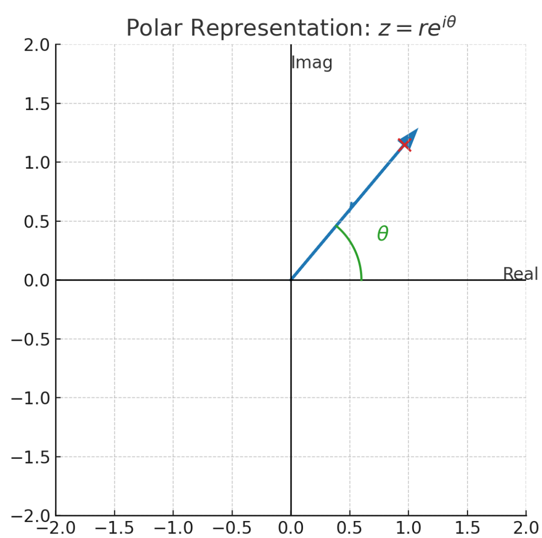

Figure 6: Polar representation of a complex number $z = re^{i\theta}$. The magnitude r is the distance from the origin, and θ is the angle with respect to the real axis.

The exponential representation has a remarkable identity known as Euler's formula:

$$e^{i\theta} = \cos\theta + i\sin\theta.$$

Thus, a rotating phasor with angular speed ω at time t is

$$e^{i\omega t} = \cos(\omega t) + i\sin(\omega t).$$

The three quantities that appear throughout Fourier analysis are:

- **Frequency:** ω or $2\pi f$ controls how rapidly the oscillation rotates around the unit circle.
- **Amplitude:** $|A|$ determines how large its contribution is when combined with other oscillations.
- **Phase:** ϕ determines the orientation (or time shift) of the oscillation.

Amplitude, Frequency, Phase

A single oscillatory mode looks like

$$Ae^{i(\omega t+\phi)}.$$

The complex number A encodes the amplitude $|A|$ and initial phase ϕ, while ω controls the speed of rotation. A real sinusoid is just the real part of this rotation. Thus, every oscillatory component has three interpretable parameters: how big it is (amplitude), how fast it oscillates (frequency), and how shifted it is in time (phase).

Polar representation is especially valuable when multiplying complex exponentials, since magnitudes multiply and phases add:

$$(re^{i\theta})(se^{i\phi}) = (rs)e^{i(\theta+\phi)}.$$

This algebraic simplicity underpins how convolution, filtering, and linear systems behave in the frequency domain.

Another object that will recur is the set of Nth roots of unity:

$$\omega_N^k = e^{2\pi ik/N}, \qquad k = 0, \ldots, N-1.$$

These are equally spaced points on the unit circle, forming a regular N-gon in the complex plane. Each root represents a complex exponential with frequency k/N, sampled at N equally spaced time instants over one period.

Geometric Meaning of Roots of Unity

The Nth roots of unity are not abstract algebraic items — they are *rotations on the complex circle*. Starting from $1 = e^{2\pi i\cdot 0/N}$, each successive power multiplies by $e^{2\pi i/N}$, rotating counterclockwise by exactly one Nth of a turn. After N multiplications, the phasor has returned to 1, completing a full cycle. These N phasors form a perfectly symmetric, equally spaced set around the circle: a regular polygon.

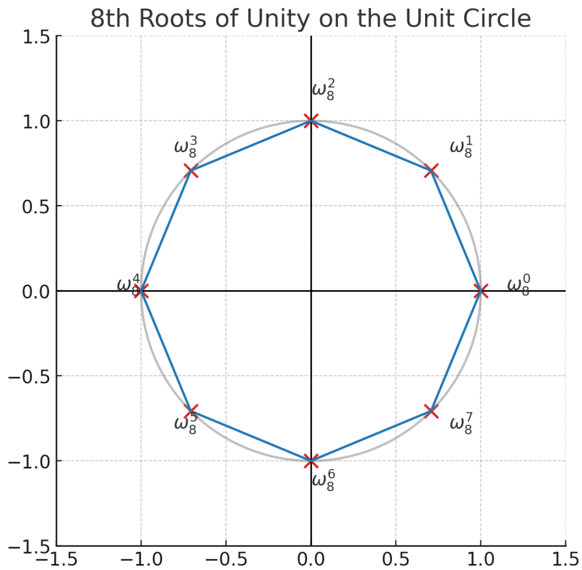

Figure 7: 8th roots of unity on the complex unit circle. Each point is $e^{2\pi ik/8}$, equally spaced at 45° increments. Connecting them forms a regular octagon.

Why Roots of Unity Matter for the DFT

The discrete Fourier transform decomposes a length-N signal into N frequency components. The kth component uses the complex exponential

$$e^{2\pi ikn/N} = \omega_N^{kn},$$

which is simply the kth root of unity raised to the nth time sample. Thus: - **The DFT matrix is built entirely from powers of the Nth root of unity**. - Each column of the DFT corresponds to a pure rotating phasor sampled at N time instants. - Orthogonality of these sampled exponentials is the geometric reason the DFT coefficients do not interfere with one another. The more symmetric these roots are around the unit circle, the cleaner the frequency decomposition becomes. Fourier analysis on finite data is, at its core, geometry on the unit circle.

Discrete Phasor Sampling

A DFT basis vector at frequency index k is obtained by sampling the pure oscillation $e^{2\pi ikt/N}$ at integer time steps $t = 0, 1, \ldots, N-1$. Each sample is a point on the unit circle, forming a rotating phasor pattern. This pattern is exactly the kth column of the DFT matrix. Higher k means the phasor spins faster around the circle, and the samples are spaced more densely in angle.

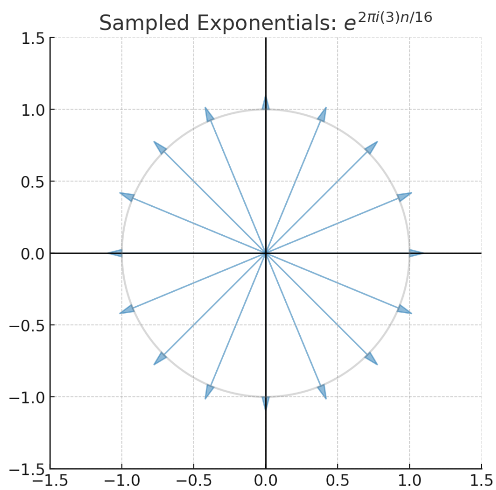

Figure 8: Sampled complex exponential $e^{2\pi i(k)n/N}$ for $k = 3$, $N = 16$. Each arrow represents a discrete phasor sample on the unit circle. These N samples form the geometric core of a DFT column.

Orthogonality as Phase-Walk Cancellation

When we multiply one sampled phasor by the conjugate of another, the result is a slow phasor that rotates with the phase difference $(k_1 - k_2)$. Summing these vectors traces a polygon in the complex plane. If the two frequencies differ, the polygon wraps evenly, causing the total to nearly cancel. This geometric cancellation is exactly the statement that distinct DFT columns are orthogonal under the discrete inner product.

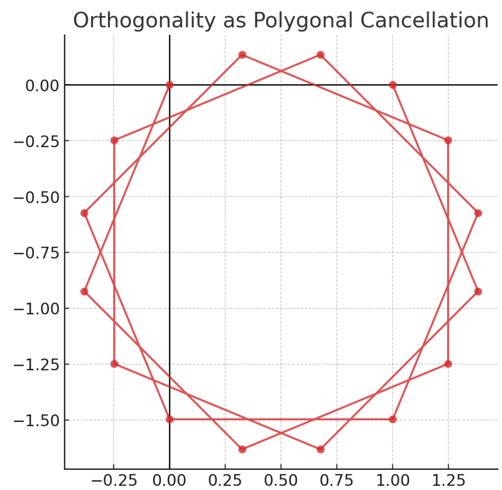

Figure 9: Orthogonality of sampled exponentials visualized as polygonal cancellation. The cumulative sum of $e^{2\pi i k_1 n/N}\,\overline{e^{2\pi i k_2 n/N}}$ forms a closed polygon when $k_1 \neq k_2$, showing that their inner product cancels.

DFT Columns are Sampled Rotating Modes

The kth DFT basis vector is obtained by evaluating $e^{2\pi i k n/N}$ at $n = 0, \ldots, N-1$. The real part produces a cosine-like shape with k oscillations across N samples. The imaginary part produces a sine-like shape. Every discrete time signal is expressed as a linear combination of these sampled oscillations.

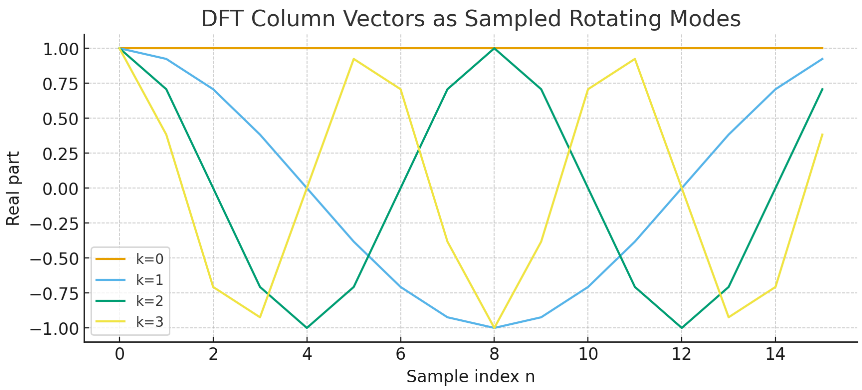

Figure 10: DFT column vectors interpreted as sampled rotating modes. Higher frequency index k corresponds to a faster oscillation in the time samples. The kth DFT column is the discretized phasor $e^{2\pi ikn/N}$.

The DFT is not mysterious: each column is a rotating phasor sampled at N successive time instants. Orthogonality arises because different rotation rates correspond to different "phase walks" around the unit circle. When summed, their directions distribute evenly, cancelling one another. The Fourier coefficients measure how strongly the input signal aligns with each rotating mode, exactly as inner products measure projection in Euclidean geometry.

2.2 Harmonic Exponentials

A **harmonic exponential** is a complex sinusoid of the form

$$e^{i\omega t} = \cos(\omega t) + i\sin(\omega t),$$

where ω is a real angular frequency. These functions are called "harmonic" because they oscillate smoothly and periodically, and "exponential" because their algebraic behavior is governed by exponential laws.

Why Harmonic Exponentials Are Fundamental

A harmonic exponential behaves like a rotating vector in the complex plane. Its magnitude is constant, and its phase increases linearly in time. Because the exponential form converts oscillation into simple multiplication,

$$e^{i\omega_1 t} \cdot e^{i\omega_2 t} = e^{i(\omega_1+\omega_2)t},$$

harmonic exponentials form a frequency algebra that is dramatically simpler than working with sines and cosines directly.

Harmonic exponentials serve as **eigenfunctions of linear time-invariant (LTI) systems**: if $\mathcal{L}$ is a linear differential operator with constant coefficients, then

$$\mathcal{L}\{e^{i\omega t}\} = \lambda(\omega)e^{i\omega t},$$

where $\lambda(\omega)$ is a scalar multiplier (the frequency response). In other words, harmonic exponentials pass through an LTI system unchanged in shape—only their amplitude and phase are modified. This makes them the perfect basis functions for analysis, filtering, control, and modulation.

title

- Harmonic exponentials provide a **polar representation** of oscillation, - They convert differentiation into multiplication, - They diagonalize convolution and LTI operators, - Their orthogonality guarantees unique spectral decomposition, - Their constant magnitude and linear phase evolution make them the canonical oscillatory basis for Fourier analysis.

Every Fourier series and Fourier transform represents a signal as a superposition of harmonic exponentials:

$$f(t) = \sum_k A_k e^{i\omega_k t} \quad \text{or} \quad f(t) = \int_{-\infty}^{\infty} F(\omega)e^{i\omega t}\, d\omega,$$

meaning that any sufficiently regular function can be expressed as a weighted combination of harmonic building blocks. The coefficients A_k (or $F(\omega)$) measure how strongly the signal aligns with each frequency mode, providing a complete, orthogonal, and geometrically interpretable representation of structure across scales.

2.3 Orthogonality and Inner Products

The language of Fourier analysis is most transparent when described using the inner product. In finite dimensions, the inner product of vectors $u, v \in \mathbb{C}^N$ is

$$\langle u, v\rangle = \sum_{n=0}^{N-1} u_n\, \overline{v_n}.$$

Two vectors are *orthogonal* if $\langle u, v\rangle = 0$. Orthogonality guarantees independent directions: projecting onto one direction does not interfere with projection onto another.

In function spaces, we extend the same idea. If functions f and g live on an interval $[0, T]$, a natural inner product is

$$\langle f, g\rangle = \int_0^T f(t)\,\overline{g(t)}\, dt.$$

The complex exponential family

$$\{e^{i2\pi kt/T}\}_{k\in\mathbb{Z}}$$

is orthogonal with respect to this inner product:

$$\langle e^{i2\pi kt/T}, e^{i2\pi \ell t/T}\rangle = 0 \quad \text{if } k \neq \ell.$$

Why Orthogonality Matters

Orthogonality ensures that each oscillatory mode isolates its own frequency content. When computing the Fourier coefficient $A_k = \langle f, e^{i2\pi kt/T}\rangle$, all other modes vanish automatically. This makes projection stable, exact, and conceptually clean. Reconstruction is a sum of mutually independent building blocks.

The complex exponential basis is therefore the canonical orthogonal basis for periodic functions. The independence of oscillatory directions is the essence of "superposition" in signal space.

2.4 Vector Spaces of Functions

The framework above can be understood by analogy with finite-dimensional vector spaces. Consider $\mathbb{R}^3$: any vector v can be decomposed uniquely into a linear combination of orthonormal basis vectors e_1, e_2, e_3:

$$v = (v\cdot e_1)e_1 + (v\cdot e_2)e_2 + (v\cdot e_3)e_3.$$

Because the basis is orthonormal, the coefficients are just inner products.

Similarly, a function $f(t)$ can be decomposed into orthogonal exponential basis functions:

$$f(t) = \sum_k A_k e^{i2\pi kt/T},$$

where $A_k = \langle f, e^{i2\pi kt/T}\rangle$ is the projection coefficient.

Infinite-Dimensional Analogy

Think of a function as a point in an infinite-dimensional space. The set $\{e^{i2\pi kt/T}\}$ plays the role of orthonormal axes. Each axis corresponds to a frequency. The Fourier transform gives coordinates of f along each axis. The sum over all axes reconstructs the point exactly.

A consequence of orthonormality is a form of energy preservation known as Parseval's identity:

$$\int_0^T |f(t)|^2 dt = \sum_k |A_k|^2.$$

Energy in time equals energy in frequency. This identity clarifies why amplitude is so meaningful: $|A_k|^2$ literally measures the contribution of frequency k to the total signal energy.

2.5 Convolution: Intuition First

Suppose we have two signals, f and g. Their convolution is

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)\, g(t - \tau)\, d\tau.$$

Conceptually, convolution measures "sliding overlap." As g slides across f, the integral measures how well the two functions line up.

Sliding Overlap

Imagine fixing f and sliding a time-reversed version of g across it. At each position, convolution multiplies values where they overlap and integrates the result. Where features align, convolution is large; where they mismatch, it is small.

Why is convolution central to signals and systems? Because any linear time-invariant (LTI) system can be described using convolution with its impulse response. Input f produces output $f * g$. The convolution operator summarizes how the system responds to all possible inputs.

The entire power of Fourier analysis emerges from the following identity:

$$\mathcal{F}(f * g) = \mathcal{F}(f) \cdot \mathcal{F}(g).$$

In other words, convolution in time corresponds to multiplication in frequency. This is the reason filtering, smoothing, and linear system analysis become vastly simpler in the frequency domain.

The connection between convolution and oscillations comes from eigenfunctions: $e^{i\omega t}$ is an eigenfunction of every LTI system. When you feed an LTI system a pure sinusoid, you get back a sinusoid at the same frequency — only its amplitude/phase change. Thus, oscillations are the natural eigenmodes of linear filtering.

2.6 Sampling Theory (Preview)

In practice, most data we encounter is discrete: audio samples, pixel intensities, weekly financial prices, sensor readings, and so on. The discrete Fourier transform is not a mere numerical approximation to the continuous transform — it is the exact Fourier analysis of a finite vector in $\mathbb{C}^N$.

Sampling is a map from a continuous signal $f(t)$ to a sequence $\{f(nT_s)\}$ where T_s is the sampling interval. If the signal is bandlimited to less than half the sampling frequency (Nyquist rate), then no information is lost: the continuous signal can be reconstructed by ideal low-pass filtering.

Nyquist and Aliasing

Sampling a continuous signal without sufficient sampling rate causes *aliasing*: high-frequency content folds onto lower frequencies, creating ambiguity. Proper sampling preserves frequency content exactly, and the DFT gives access to that information through a finite set of orthogonal complex exponentials.

A key insight is that the DFT

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi ikn/N}$$

is a perfectly valid linear operator on $\mathbb{C}^N$, independent of any limiting argument. Given a finite vector, this decomposition is exact, and its frequency-domain coefficients capture amplitude, frequency, and phase for each discrete Fourier mode:

$$X_k = |X_k| e^{i\phi_k}.$$

The magnitude $|X_k|$ measures how strongly the kth discrete frequency contributes to the signal. The phase ϕ_k encodes alignment (or time shift). The index k corresponds to a frequency of k/N cycles per sample interval.

Later, we will see how these ideas unify continuous and discrete Fourier theory, and how efficient algorithms like the FFT make such decompositions fast enough to be used everywhere in modern computing.

3 The Continuous Fourier Transform (CFT)

3.1 Fourier Series: Periodic Case

We begin with the case of a function f that is periodic with period $T > 0$:

$$f(t+T) = f(t) \quad \text{for all } t.$$

The central statement of Fourier series is that, under suitable regularity conditions, f can be written as

$$f(t) = \sum_{k=-\infty}^{\infty} c_k \, e^{i2\pi kt/T},$$

where the coefficients c_k tell us how much of each harmonic (frequency k/T) is present in f.

Orthogonality of Complex Exponentials

Consider the space of square-integrable T-periodic functions on $[0, T]$ with inner product

$$\langle f, g \rangle = \frac{1}{T} \int_0^T f(t) \, \overline{g(t)} \, dt.$$

The functions

$$\phi_k(t) = e^{i2\pi kt/T}, \quad k \in \mathbb{Z},$$

are orthogonal with respect to this inner product. Indeed, for integers k, ℓ:

$$\langle \phi_k, \phi_\ell \rangle = \frac{1}{T} \int_0^T e^{i2\pi kt/T} \, \overline{e^{i2\pi \ell t/T}} \, dt = \frac{1}{T} \int_0^T e^{i2\pi(k-\ell)t/T} \, dt.$$

If $k = \ell$, the integrand is 1, so

$$\langle \phi_k, \phi_k \rangle = \frac{1}{T} \int_0^T 1 \, dt = 1.$$

If $k \neq \ell$, we have

$$\frac{1}{T} \int_0^T e^{i2\pi(k-\ell)t/T} \, dt = \frac{1}{T} \left[\frac{T}{i2\pi(k-\ell)} e^{i2\pi(k-\ell)t/T} \right]_0^T = \frac{1}{i2\pi(k-\ell)} \left(e^{i2\pi(k-\ell)} - 1 \right) = 0,$$

since $e^{i2\pi(k-\ell)} = 1$.

Thus,

$$\langle \phi_k, \phi_\ell \rangle = \begin{cases} 1, & k = \ell, \\ 0, & k \neq \ell. \end{cases}$$

Fourier Coefficients as Projections

Because $\{\phi_k\}_{k\in\mathbb{Z}}$ is an orthonormal family, the coefficient c_k in $f(t) = \sum_k c_k\phi_k(t)$ is just the inner product

$$c_k = \langle f, \phi_k\rangle = \frac{1}{T}\int_0^T f(t)\, e^{-i2\pi kt/T}\, dt.$$

Each c_k isolates the contribution of exactly one frequency k/T to the overall function.

Geometric Picture

Each mode $e^{i2\pi kt/T}$ is a pure oscillation at frequency k/T:

$$e^{i2\pi kt/T} = \cos\left(\frac{2\pi k}{T}t\right) + i\sin\left(\frac{2\pi k}{T}t\right).$$

The coefficient c_k is a complex number:

$$c_k = |c_k|e^{i\phi_k},$$

where:

- $|c_k|$ is the *amplitude* of the kth frequency component,
- ϕ_k is its *phase* (how shifted it is in time).

Fourier Series as a Musical Chord

Think of $f(t)$ as a complex sound that repeats every T seconds. The harmonics $e^{i2\pi kt/T}$ are pure notes at frequencies k/T. The Fourier coefficients c_k tell you how loud (amplitude) and how shifted (phase) each note is. The full sound is the chord formed by all these notes played together.

The Fourier series representation is thus: - A coordinate system indexed by integer frequencies k/T, - With coordinates c_k encoding amplitude and phase.

3.2 Continuous Fourier Transform

Fourier series handle periodic functions on a finite interval. Many signals in practice, however, are not periodic, or are better thought of as defined on the whole real line. The *continuous Fourier transform* extends the Fourier idea to non-periodic functions.

For a sufficiently nice function $f : \mathbb{R} \to \mathbb{C}$, the Fourier transform is defined by

$$F(\omega) = \mathcal{F}\{f\}(\omega) = \int_{-\infty}^{\infty} f(t)\, e^{-i\omega t}\, dt.$$

Intuition from Fourier Series on Expanding Intervals

Imagine first restricting f to a finite interval $[-L, L]$ and artificially declaring it to be periodic with period $2L$. On $[-L, L]$, you can expand this periodic version in a Fourier series:

$$f_L(t) \approx \sum_{k=-\infty}^{\infty} c_k^{(L)} e^{i\pi k t/L}.$$

The spacing between adjacent frequencies is $\Delta\omega = \pi/L$. As $L \to \infty$, the discrete frequency grid $\omega_k = \pi k/L$ becomes denser and denser, essentially giving a continuum of frequencies. The Fourier coefficients $c_k^{(L)}$ then behave like samples of a continuous function in frequency, scaled by $\Delta\omega$:

$$c_k^{(L)} \approx \frac{\Delta\omega}{2\pi} F(\omega_k).$$

In the limit $L \to \infty$, the sum

$$\sum_k c_k^{(L)} e^{i\omega_k t}$$

turns into an integral over all real ω. This heuristic leads exactly to the Fourier transform and its inverse.

Frequency as a Continuous Variable

In Fourier series, frequency is discrete: only integer multiples of a base frequency appear. In the Fourier transform, we allow signals that are not periodic, so there is no fundamental base period. As a result, *every* real frequency ω is allowed, and the spectrum becomes a continuous function $F(\omega)$.

3.3 Inverse Transform

The Fourier transform is not just a one-way compression into the frequency domain. Under suitable conditions on f, we can reconstruct f from F via the inverse transform:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)\, e^{i\omega t}\, d\omega.$$

This pair of formulas,

$$F(\omega) = \int_{-\infty}^{\infty} f(t)\, e^{-i\omega t}\, dt, \quad f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)\, e^{i\omega t}\, d\omega,$$

captures the idea that $\{e^{i\omega t}\}_{\omega\in\mathbb{R}}$ form a kind of *continuous basis* for functions. The transform $F(\omega)$ gives the "coordinate density" of f with respect to this basis.

Completeness of the Exponential Basis

The inverse transform expresses f as a continuous superposition of complex exponentials:

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)\, e^{i\omega t}\, d\omega.$$

Every frequency ω contributes a rotating phasor $e^{i\omega t}$, scaled by amplitude $|F(\omega)|$ and phase $\arg F(\omega)$. The collection of all such contributions reconstructs the original signal exactly.

Formally, questions of existence and uniqueness are handled in the context of $L^2(\mathbb{R})$ (square-integrable functions) and distributions, but for many engineering and applied mathematics purposes, it suffices to remember: if f is reasonably well-behaved (integrable, no pathological growth), then the Fourier transform pair above holds in a pointwise or L^2 sense.

3.4 Properties

The Fourier transform enjoys a collection of structural properties, many of which mirror familiar operations on functions.

Let $\mathcal{F}\{f\}(\omega) = F(\omega)$.

Linearity

For scalars a, b and functions f, g:

$$\mathcal{F}\{af + bg\}(\omega) = aF(\omega) + bG(\omega).$$

Time Shifting

If $f(t - t_0)$ is f shifted in time by t_0, then

$$\mathcal{F}\{f(t - t_0)\}(\omega) = e^{-i\omega t_0} F(\omega).$$

A shift in time multiplies the spectrum by a pure phase factor.

Frequency Shifting (Modulation)

If we modulate f by $e^{i\omega_0 t}$:

$$\mathcal{F}\{e^{i\omega_0 t} f(t)\}(\omega) = F(\omega - \omega_0).$$

Multiplying by a complex exponential shifts the spectrum in frequency.

Differentiation in Time

Differentiating f corresponds to multiplying by $i\omega$ in frequency:

$$\mathcal{F}\{f'(t)\}(\omega) = i\omega F(\omega).$$

Repeated differentiation gives higher powers of $i\omega$. This is the reason differential equations become algebraic in the Fourier domain.

Convolution $\leftrightarrow$ Multiplication

If $(f * g)(t) = \int f(\tau)g(t-\tau)\,d\tau$, then

$$\mathcal{F}\{f * g\}(\omega) = F(\omega)G(\omega).$$

Conversely,

$$\mathcal{F}\{f \cdot g\}(\omega) = \frac{1}{2\pi}(F * G)(\omega).$$

We will revisit this in more detail shortly.

Energy Conservation (Parseval/Plancherel)

For suitable f,

$$\int_{-\infty}^{\infty} |f(t)|^2\,dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} |F(\omega)|^2\,d\omega.$$

Total energy in time equals total energy in frequency. The magnitude squared $|F(\omega)|^2$ thus has the interpretation of an energy density across frequencies.

Amplitude and Phase in the Continuous Spectrum

At each frequency ω, the value

$$F(\omega) = |F(\omega)|e^{i\phi(\omega)}$$

tells you:

- how strong the oscillation at frequency ω is (*amplitude* $|F(\omega)|$),
- how shifted or aligned it is relative to a reference (*phase* $\phi(\omega)$).

The total signal is a continuum of such rotating phasors, each with its own amplitude and phase, summing up to $f(t)$.

3.5 Examples

We now compute a few canonical Fourier transforms. These will later become reference pairs.

Square Pulse (Rectangular Function)

Let

$$f(t) = \begin{cases} 1, & |t| \le \frac{T}{2}, \\ 0, & \text{otherwise}. \end{cases}$$

Then

$$F(\omega) = \int_{-T/2}^{T/2} e^{-i\omega t}\, dt = \left[\frac{e^{-i\omega t}}{-i\omega} \right]_{-T/2}^{T/2} = \frac{2\sin(\omega T/2)}{\omega}.$$

Often this is written using the normalized sinc function:

$$F(\omega) = T\,\mathrm{sinc}\left(\frac{\omega T}{2\pi}\right), \quad \mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}.$$

Sharp Edges $\Rightarrow$ High Frequencies

A square pulse has abrupt jumps (discontinuities). These sharp edges require a wide range of high-frequency components to approximate. The sinc spectrum decays like $1/\omega$, significantly slower than, say, a Gaussian.

Gaussian

Let

$$f(t) = e^{-t^2/(2\sigma^2)}.$$

Then a beautiful fact is that its Fourier transform is also a Gaussian:

$$F(\omega) = \sigma\sqrt{2\pi}\, e^{-\sigma^2\omega^2/2}.$$

The Gaussian is "its own Fourier transform" up to scaling, making it an ideal window in time-frequency analysis.

Triangular Function

The convolution of two square pulses is a triangle. Because convolution in time corresponds to multiplication in frequency, the Fourier transform of a triangular pulse is the square of a sinc function. Triangular functions therefore have spectra that decay faster (like $1/\omega^2$).

Dirac Delta

The Dirac delta $\delta(t)$ is not a classical function but a distribution. It is defined by the property

$$\int_{-\infty}^{\infty} \delta(t)\,\varphi(t)\,dt = \varphi(0)$$

for all test functions φ. Its Fourier transform is

$$\mathcal{F}\{\delta\}(\omega) = \int_{-\infty}^{\infty} \delta(t)\, e^{-i\omega t}\, dt = 1.$$

Conversely, the Fourier transform of 1 is (up to constants) a delta at the origin:

$$\mathcal{F}\{1\}(\omega) = 2\pi\delta(\omega).$$

These dualities are the backbone of many transform identities.

3.6 Convolution Theorem

The convolution theorem is one of the most important results in Fourier analysis. It explains why it is often much easier to work in the frequency domain when dealing with linear systems and filters.

Statement

Let $h = f * g$ be the convolution of f and g:

$$h(t) = (f * g)(t) = \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau.$$

Then the Fourier transform of h is

$$\mathcal{F}\{f * g\}(\omega) = F(\omega)\, G(\omega),$$

where $F = \mathcal{F}\{f\}$ and $G = \mathcal{F}\{g\}$.

Proof Sketch

Starting from the definition,

$$\begin{aligned} \mathcal{F}\{f * g\}(\omega) &= \int_{-\infty}^{\infty} (f * g)(t)\, e^{-i\omega t}\, dt \\ &= \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau \right] e^{-i\omega t}\, dt. \end{aligned}$$

Assuming we can interchange the order of integration (Fubini's theorem),

$$\mathcal{F}\{f * g\}(\omega) = \int_{-\infty}^{\infty} f(\tau) \left[\int_{-\infty}^{\infty} g(t-\tau)\, e^{-i\omega t}\, dt \right] d\tau.$$

Make the change of variables $u = t - \tau$ (so $t = u + \tau$):

$$\begin{aligned} \int_{-\infty}^{\infty} g(t-\tau)\, e^{-i\omega t}\, dt &= \int_{-\infty}^{\infty} g(u)\, e^{-i\omega(u+\tau)}\, du \\ &= e^{-i\omega\tau} \int_{-\infty}^{\infty} g(u)\, e^{-i\omega u}\, du \\ &= e^{-i\omega\tau} G(\omega). \end{aligned}$$

Substituting back,

$$\begin{aligned}\mathcal{F}\{f * g\}(\omega) &= \int_{-\infty}^{\infty} f(\tau)\, e^{-i\omega\tau}\, G(\omega)\, d\tau \\ &= G(\omega) \int_{-\infty}^{\infty} f(\tau)\, e^{-i\omega\tau}\, d\tau \\ &= F(\omega)\, G(\omega).\end{aligned}$$

Filtering in the Frequency Domain

If g is the impulse response of a linear time-invariant system, then $f * g$ is the system output when f is the input. The convolution theorem says:

$$\mathcal{F}\{f * g\}(\omega) = F(\omega)\, G(\omega).$$

So to compute the output spectrum, you do not need to convolve in time; you simply multiply spectra frequency by frequency. This is the core reason FFT-based filtering is so powerful: long convolutions in time become simple, pointwise multiplications in frequency.

In later sections, we will see how the discrete analog of this theorem, combined with the FFT, leads to extremely efficient algorithms for convolution, correlation, and filtering in digital signal processing.

4 Sampling, Aliasing, and Bandlimiting

4.1 The Sampling Operator

Suppose a continuous signal $f(t)$ is observed only at discrete time instants $t = nT_s$, where T_s is the sampling interval and $n \in \mathbb{Z}$. Ideal sampling can be expressed using an impulse train:

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s),$$

so that the sampled signal is

$$f_s(t) = f(t)\,s(t) = \sum_{n=-\infty}^{\infty} f(nT_s)\,\delta(t - nT_s).$$

Sampling as Time-Gated Observation

Sampling is conceptually simple: we only observe the value of $f(t)$ at integer multiples of T_s, and all intermediate information is discarded. The Dirac impulses serve as exact "memory markers" of those values. Everything we later do with the sampled signal depends entirely on the sequence $\{f(nT_s)\}$.

A central principle is that sampling in time corresponds to *periodization in frequency*. Let $F(\omega)$ be the continuous Fourier transform of $f(t)$. The Fourier transform of the impulse train is another impulse train:

$$\mathcal{F}\{s(t)\}(\omega) = \frac{2\pi}{T_s} \sum_{m=-\infty}^{\infty} \delta\left(\omega - \frac{2\pi m}{T_s}\right).$$

Thus the spectrum of the sampled signal is

$$F_s(\omega) = \mathcal{F}\{f\,s\}(\omega) = \frac{1}{2\pi}(F * \mathcal{F}\{s\})(\omega) = \frac{1}{T_s} \sum_{m=-\infty}^{\infty} F\left(\omega - \frac{2\pi m}{T_s}\right),$$

a **sum of shifted copies** of the original spectrum.

Time Sampling $\Rightarrow$ Frequency Replication

Sampling a signal multiplies it by an impulse train in time. In frequency, this corresponds to convolving the original spectrum with another impulse train, creating *periodic replicas* of $F(\omega)$ spaced by $\frac{2\pi}{T_s}$. All of aliasing, Nyquist, and reconstruction phenomena stem from this spectral replication.

The preceding identity is a concrete manifestation of the Poisson summation formula — sampling collapses an infinite function into a discrete sequence, and in the frequency domain that collapse becomes exact periodic structure.

4.2 Nyquist–Shannon Sampling Theorem

The Nyquist–Shannon theorem formalizes when sampling preserves all information.

Let $f(t)$ be **bandlimited**: its spectrum $F(\omega)$ is zero outside the interval

$$|\omega| < \omega_{\max}.$$

If we choose a sampling period T_s such that

$$\frac{2\pi}{T_s} > 2\omega_{\max},$$

or equivalently

$$f_s = \frac{1}{T_s} > \frac{2\omega_{\max}}{2\pi},$$

then the replicas of $F(\omega)$ produced by sampling do **not overlap**. In that case, all spectral energy remains separated, and $f(t)$ can be perfectly recovered from its discrete samples.

Nyquist Criterion

The sampling frequency must exceed twice the highest frequency present in $f(t)$. This ensures that the replicated spectra produced by sampling do not overlap. No overlap means no ambiguity; all sampled information remains uniquely decodable.

Conversely, if the sampling rate is too low, the periodic copies of $F(\omega)$ overlap. This causes **frequency wrap-around**: high-frequency components appear in the wrong region of the sampled spectrum. This phenomenon is known as aliasing and is fundamental to digital signal processing.

4.3 Aliasing

Aliasing occurs when different continuous-time frequencies become indistinguishable after sampling. In the frequency domain, this happens because overlapping replicas add coherently:

$$F_s(\omega) = \frac{1}{T_s} \sum_m F\left(\omega - \frac{2\pi m}{T_s}\right).$$

If two components ω_1 and ω_2 satisfy

$$\omega_2 = \omega_1 + \frac{2\pi m}{T_s},$$

then their sampled spectra land in the same frequency band and **cannot be separated**.

Geometric Interpretation of Aliasing

Sampling compresses the infinite frequency axis into a repeating band of width $\frac{2\pi}{T_s}$. Imagine taking the real line and folding it modulo this band width. Frequencies that differ by integer multiples of $\frac{2\pi}{T_s}$ now coincide. That folding is aliasing.

Example: Sampling a High-Frequency Sine

Let $f(t) = \sin(\omega t)$ and sample at rate $f_s = \frac{1}{T_s}$. If ω exceeds half the sampling rate, then after sampling it becomes indistinguishable from a lower frequency:

$$\sin(\omega t) \longleftrightarrow \sin((\omega - 2\pi k/T_s)t)\,.$$

Even though the original sinusoid oscillates faster, the sampled sequence appears as if it oscillated slowly. Digital sensors, audio equipment, and oscilloscopes all encounter this phenomenon.

Aliasing is **not noise** — it is the deterministic result of undersampling.

4.4 Reconstruction

Perfect reconstruction is possible whenever the sampling rate satisfies Nyquist's condition and the original signal is bandlimited. The idea is to isolate the central copy of $F_s(\omega)$ using an ideal low-pass filter:

$$F(\omega) = \begin{cases} F_s(\omega), & |\omega| < \omega_{\max}, \\ 0, & \text{otherwise.} \end{cases}$$

Transforming back to time gives

$$f(t) = \sum_{n=-\infty}^{\infty} f(nT_s)\,\mathrm{sinc}\left(\frac{t - nT_s}{T_s}\right),$$

where

$$\mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}.$$

Sinc Interpolation

Each sample $f(nT_s)$ contributes a shifted sinc function. The continuous-time signal is reconstructed as a weighted sum of those sinc waveforms. Provided the signal was bandlimited and sampled above Nyquist, this reconstruction is mathematically exact.

The interpolation kernel $\mathrm{sinc}\left(\frac{t-nT_s}{T_s}\right)$ follows directly from the inverse Fourier transform of an ideal rectangular low-pass filter. Its infinite support reflects the fact that perfect bandlimiting in frequency has infinite support in time — a manifestation of the uncertainty principle.

5 The Discrete Fourier Transform (DFT)

5.1 Definition

Let $x = (x_0, \ldots, x_{N-1}) \in \mathbb{C}^N$. The discrete Fourier transform (DFT) produces another vector $X = (X_0, \ldots, X_{N-1})$ defined by

$$X_k = \sum_{n=0}^{N-1} x_n \, e^{-2\pi i k n/N}, \qquad k = 0, \ldots, N-1.$$

The DFT as a Change of Basis

The DFT expresses the vector x in terms of oscillatory basis vectors

$$\phi_k(n) = e^{-2\pi i k n/N}.$$

Each ϕ_k is a sampled complex exponential with discrete frequency index k. The coefficient X_k is the inner product

$$X_k = \langle x, \phi_k \rangle,$$

measuring how strongly x aligns with the kth oscillation. The DFT is therefore a **change of coordinates from the standard basis in $\mathbb{C}^N$ to the frequency basis**.

Because the samples are periodic with period N in the exponent, these basis vectors are **Nth roots of unity traced through time**:

$$\phi_k(n) = \omega_N^{kn}, \qquad \omega_N = e^{-2\pi i/N}.$$

5.2 Roots of Unity

Recall the Nth roots of unity:

$$\omega_N^k = e^{2\pi i k/N}, \qquad k = 0, \ldots, N-1.$$

Geometrically, these are equally spaced points on the unit circle forming a regular N-gon. The discrete exponential

$$\phi_k(n) = \omega_N^{kn}$$

evaluates the kth phasor at the nth sample.

Orthogonality via Discrete Rotation

Distinct frequency indices $k \neq \ell$ produce sampled phasors whose phase walks differ. When we compute their inner product,

$$\sum_{n=0}^{N-1} \omega_N^{(k-\ell)n},$$

the rotation is symmetric around the circle, causing cancellation. The sum collapses to 0. This is the discrete analog of orthogonality for Fourier series, and is what makes frequency decomposition exact.

Orthogonality can be written compactly:

$$\sum_{n=0}^{N-1} \omega_N^{(k-\ell)n} = \begin{cases} N, & k = \ell, \\ 0, & k \neq \ell. \end{cases}$$

Thus the frequency basis vectors are orthogonal up to a factor of N.

5.3 Inverse DFT

Since the frequency vectors $\phi_0, \ldots, \phi_{N-1}$ form a complete orthogonal basis, we can reconstruct x from its coefficients:

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \, e^{2\pi i k n/N}.$$

The factor $\frac{1}{N}$ comes from normalization of discrete orthogonality:

$$\frac{1}{N} \sum_{k=0}^{N-1} e^{2\pi i (k-\ell)n/N} = \delta_{n\ell}.$$

The Inverse DFT

The forward DFT is

$$X_k = \sum_{n=0}^{N-1} x_n \, e^{-2\pi i k n/N},$$

and the inverse DFT is

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \, e^{2\pi i k n/N}.$$

Together they guarantee perfect reconstruction. The forward transform measures projection onto each sampled oscillation; the inverse transform recombines them to reassemble the original vector.

5.4 Matrix Form

Let $\mathbf{F}_N$ be the $N \times N$ matrix with entries

$$(\mathbf{F}_N)_{k,n} = e^{-2\pi ikn/N}.$$

Then the DFT is simply

$$\mathbf{X} = \mathbf{F}_N\,\mathbf{x}.$$

The inverse transform is

$$\mathbf{x} = \frac{1}{N}\,\mathbf{F}_N^*\,\mathbf{X},$$

where $\mathbf{F}_N^*$ is the conjugate transpose.

Unitary Scaling

If we define

$$\mathbf{U}_N = \frac{1}{\sqrt{N}}\mathbf{F}_N,$$

then $\mathbf{U}_N$ is unitary:

$$\mathbf{U}_N^*\mathbf{U}_N = \mathbf{I}.$$

Thus the DFT becomes a unitary change of basis in $\mathbb{C}^N$, completely preserving length and orthogonality. Inner products, norms, and energy are identical in both bases.

The DFT matrix is a Vandermonde matrix generated by powers of the root ω_N. Its eigenvectors and eigenvalues have deep number-theoretic structure, but for signal processing purposes, its primary meaning is that it is **the matrix of all sampled rotating phasors**.

5.5 Parseval in Finite Dimension

Since $\mathbf{U}_N$ is unitary, it preserves Euclidean energy:

$$\|x\|_2^2 = \frac{1}{N}\sum_{k=0}^{N-1} |X_k|^2.$$

That is,

$$\sum_{n=0}^{N-1} |x_n|^2 = \frac{1}{N}\sum_{k=0}^{N-1} |X_k|^2.$$

Discrete Parseval Identity

Energy in time equals energy in frequency. The magnitude $|X_k|^2$ represents how much of the signal's energy resides in the kth discrete mode. This is the finite-dimensional counterpart of Parseval for continuous Fourier transforms.

5.6 Computational Complexity

A naive computation of the DFT requires N inner products, each of which is a sum of N complex multiplications:

$$\text{Cost} = O(N^2).$$

When N is large (audio, video, radar, wireless, large FFT engines), this quadratic complexity is prohibitive. For $N = 10^6$, a naive DFT requires 10^{12} operations — far beyond real-time computation for typical systems.

Why $O(N^2)$ Matters

If N is the number of samples in a digital signal (e.g. $N = 44{,}100$ for one second of audio), computing its DFT naively would require on the order of two billion operations per second. This is unacceptable for real-time processing.

The Fast Fourier Transform (FFT) exploits **symmetry and reuse** in the roots of unity to reduce the cost to $O(N \log N)$ — an exponential improvement. We will study its derivation shortly.

5.7 Worked Numerical Example ($N = 8$)

Let $N = 8$ and

$$x = (1, 0, -1, 0, 1, 0, -1, 0).$$

We expect strong contributions at the frequencies corresponding to $k = 1$ and $k = 3$, based on symmetry and periodic structure.

$$X_k = \sum_{n=0}^{7} x_n \, e^{-2\pi i k n/8}.$$

For clarity, let $\omega = e^{-2\pi i/8} = e^{-i\pi/4}$ so

$$e^{-2\pi i k n/8} = \omega^{kn}.$$

Compute a few terms:

$$\begin{aligned}
X_0 &= x_0 + x_1 + \cdots + x_7 = (1 + 0 - 1 + 0 + 1 + 0 - 1 + 0) = 0,\\
X_1 &= x_0\omega^0 + x_1\omega^1 + x_2\omega^2 + \cdots + x_7\omega^7 = 1 - \omega^2 + \omega^4 - \omega^6,\\
X_2 &= 1 + \omega^4 + 1 + \omega^4 = 2(1 + \omega^4),\\
&\vdots
\end{aligned}$$

Using $\omega^4 = e^{-i\pi} = -1$, we get

$$X_2 = 2(1 - 1) = 0,$$

and similarly

$$X_6 = 0.$$

One finds

$$X_1 = 4, \qquad X_3 = 0, \qquad X_5 = 4, \qquad X_7 = 0,$$

and all others vanish.

Geometric Meaning

The pattern $(1, -1, 1, -1, \dots)$ has periodicity 2. Its strongest alignment is with the rotating phasor ω^n (frequency index $k = 1$) and its conjugate ω^7 (frequency index $k = 7$). Orthogonality makes projection exact and isolates frequency energy perfectly.

Interpretation

The DFT has decomposed the finite sequence into two pure frequency components at indices $k = 1$ and $k = 7$, fully capturing the alternating sign structure. The zero magnitudes at other frequencies indicate no alignment with those oscillatory modes.

6 The Fast Fourier Transform (FFT)

6.1 Why Divide and Conquer Works

Recall the length–N DFT:

$$X_k = \sum_{n=0}^{N-1} x_n \, \omega_N^{kn}, \qquad \omega_N = e^{-2\pi i/N}.$$

Assume N is even. Separate indices into evens and odds:

$$X_k = \sum_{m=0}^{N/2-1} x_{2m} \, \omega_N^{k(2m)} + \sum_{m=0}^{N/2-1} x_{2m+1} \, \omega_N^{k(2m+1)}.$$

Use the identity

$$\omega_N^{2mk} = \omega_{N/2}^{mk},$$

so

$$X_k = E_k + \omega_N^k \, O_k,$$

where

$$E_k = \sum_{m=0}^{N/2-1} x_{2m} \, \omega_{N/2}^{mk}, \qquad O_k = \sum_{m=0}^{N/2-1} x_{2m+1} \, \omega_{N/2}^{mk}.$$

Recursive Decomposition

The N–point DFT is reduced to two $(N/2)$–point DFTs: one built from the even samples, the other from the odd samples. Since these smaller transforms are reused for all output indices, the total cost collapses dramatically.

Additionally,

$$X_{k+N/2} = E_k - \omega_N^k O_k.$$

Thus each pair (E_k, O_k) generates **two outputs**. This two–output structure is called a *butterfly*.

The asymptotic cost satisfies:

$$T(N) = 2\,T(N/2) + O(N), \qquad T(N) = O(N \log N).$$

6.2 Derivation of Cooley–Tukey

The Cooley–Tukey FFT evaluates the DFT by recursively halving N, computing smaller transforms, and combining results using the twiddle factors ω_N^k.

A butterfly operation is given by:

$$X_k = E_k + \omega_N^k O_k, \qquad X_{k+N/2} = E_k - \omega_N^k O_k.$$

Butterfly Geometry

A butterfly takes two intermediate values (even and odd), rotates the odd one by a unit phasor, and forms two outputs as their sum and difference. The symmetry enables reuse: one pair of intermediate results produces two DFT outputs.

Each stage of the FFT consists of many such butterflies arranged in parallel.

6.3 Recursive Algorithm

The FFT recursion reduces a size–N transform to two transforms of size $N/2$ plus $O(N)$ combination work. Repeating this halving until N reaches 1 yields $O(\log N)$ recursive levels, each performing $O(N)$ work.

From $O(N^2)$ to $O(N \log N)$

The FFT replaces quadratic work with recursive reuse. The transformation is exact and exploits symmetry in the roots of unity. This is one of the greatest algorithmic accelerations in applied mathematics.

6.4 Iterative Form and Bit-Reversal

The recursive FFT depends on rearranging data so that each stage's butterflies operate on contiguous blocks. When written iteratively, this requires **bit-reversal reordering**.

If $N = 2^M$, each index n has an M–bit representation. Bit reversal maps

$$(n_{M-1} \dots n_1 n_0)_2 \quad \mapsto \quad (n_0 n_1 \dots n_{M-1})_2.$$

Bit Reversal for Data Layout

The iterative FFT performs butterfly operations in fixed stages. To allow efficient in-place computation, samples must be rearranged so that the subproblems are adjacent in memory. Bit reversal accomplishes this reordering once, enabling all subsequent stages to operate locally and efficiently.

This structure dramatically improves cache locality and is central to high-performance FFT implementations.

6.5 Numerical Stability Considerations

Although FFTs are composed of unitary operations, floating–point arithmetic introduces: - rounding error in twiddle factors, - finite precision accumulation through $\log N$ stages, - scaling requirements in fixed-point implementations, - and potential dynamic range issues for very large N.

Practical Numerical Stability

FFT is stable in double precision: total error tends to grow like $O(\log N)$, not exponentially. This property explains why FFTs are robust tools in scientific computing and high–accuracy digital signal processing.

6.6 Geometry of the Inverse FFT

The inverse DFT is

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \, e^{2\pi i k n/N}.$$

Since

$$e^{2\pi i k n/N} = \overline{e^{-2\pi i k n/N}},$$

the inverse transform can be interpreted geometrically as:

1. Reverse rotation direction (complex conjugation), 2. Reuse forward FFT machinery, 3. Reverse rotation again (conjugation), 4. Apply scaling by $1/N$.

Inverse as Conjugated Forward Transform

Forward and inverse FFT differ only by rotation direction and normalization. Complex conjugation flips rotation direction in the complex plane, and $1/N$ accounts for orthogonal scaling. No new transform is required.

6.7 FFT as Linear Operator

Define the normalized Fourier matrix

$$\mathbf{U}_N = \frac{1}{\sqrt{N}} \mathbf{F}_N,$$

where $\mathbf{F}_N$ is the unnormalized DFT matrix. Then

$$\mathbf{U}_N^* \mathbf{U}_N = \mathbf{I},$$

which shows that the FFT is a **unitary operator** when normalized.

Structured Orthonormal Factorization

The FFT factorizes a unitary matrix into $O(N \log N)$ sparse block operations (butterflies). Each butterfly is a two–dimensional orthonormal rotation and reflection. Efficiency arises because symmetry in roots of unity permits massive reuse and factorization.

7 Filtering in Frequency Domain

7.1 Low-pass, High-pass, Band-pass

A discrete-time signal $x = (x_0, \dots, x_{N-1})$ has DFT coefficients X_k. Each X_k encodes amplitude and phase at frequency index k. Filtering in the frequency domain means altering these coefficients before applying the inverse DFT.

A **low-pass filter** retains only the lowest frequency coefficients (slow variations):

$$X'_k = X_k \quad \text{for } |k| \le K,$$

and zeroes out the rest. A **high-pass filter** does the opposite: it removes slow variations and retains rapid oscillations. A **band-pass filter** keeps coefficients within a selected frequency band.

Magnitude vs Phase

The magnitude $|X_k|$ determines how strong a frequency component is, while the phase $\arg(X_k)$ controls its alignment in time. Filtering decisions are usually made in terms of magnitude, but phase must be preserved when reconstructing real-world signals to avoid distortion.

Filtering in the frequency domain is simply coefficient-level editing. Reconstruction then happens via inverse FFT:

$$x'_n = \frac{1}{N} \sum_{k=0}^{N-1} X'_k \, e^{2\pi i k n / N}.$$

7.2 Frequency-domain Denoising

Random noise typically exhibits rapid fluctuations in time, producing large energy in **high-frequency** modes. Signals of scientific or practical interest (temperature curves, heart rates, financial trajectories, speech formants, sensor drift, etc.) usually vary more slowly, dominating the **low-frequency** region.

DFT as a Microscope

The DFT separates a signal into resolvable oscillatory modes. Low-frequency modes capture long-term structure (smooth changes), while high-frequency modes isolate abrupt oscillations (noise, jitter, quantization artifacts, measurement uncertainty). This spectral separation enables denoising by simple truncation.

Thus frequency-domain denoising consists of: 1. computing X_k, 2. removing or attenuating high-k coefficients, 3. reconstructing x'_n.

This is conceptually simpler and often more stable than time-domain smoothing.

7.3 Practical Low-Frequency Reconstruction

One effective truncation scheme for real-valued data uses Hermitian symmetry. Suppose x is real, so

$$X_{N-k} = \overline{X_k}.$$

To retain only K lowest frequencies while maintaining symmetry, define

$$X'_k = \begin{cases} X_k, & k \le K, \\ 0, & K < k < N-K, \\ X_{N-k}, & k \ge N-K, \end{cases}$$

and reconstruct

$$x'_n = \frac{1}{N} \sum_{k=0}^{N-1} X'_k \, e^{2\pi i k n/N}.$$

Long-term Trend vs High-frequency Residual

Low-frequency coefficients capture aggregate structure, drift, seasonal behavior, or slowly varying modulation. High-frequency coefficients capture abrupt transitions, noise bursts, and fine texture. By truncating to $K \ll N$, reconstruction x' yields a smooth trend, and $x - x'$ isolates fine-scale residuals.

This technique is widely used in: - denoising, - trend extraction, - feature separation, - and exploratory signal analysis.

7.4 Windowing and Leakage

If a signal x is not exactly periodic within the observation window, then truncating it with a rectangular window (i.e., finite-length sampling) introduces sharp edges. Sharp edges act like discontinuities and distribute energy across many frequency bins, creating *spectral leakage*.

A **window function** is a smooth taper applied before computing the DFT:

$$x_n^{(w)} = w_n x_n,$$

where w_n is Hann, Hamming, Blackman, Kaiser, etc.

Leakage Intuition

A rectangular window has an abrupt jump at its boundaries, equivalent to multiplying by a step function. In frequency, multiplication by a step corresponds to convolution with a broad sinc, spreading spectral energy far beyond the main lobe. Smooth tapers reduce the side-lobes at the expense of main-lobe widening.

Tradeoff: - smoother windows (Blackman, Blackman–Harris, Kaiser) have excellent leakage suppression but worse frequency resolution, - less smooth windows (rectangular, Hamming) have sharper resolution but stronger sidelobes.

Choosing a window involves a balance between frequency resolution and leakage suppression.

7.5 Spectral Estimation

Finite-length data produce noisy estimates of the true spectrum. The periodogram

$$\widehat{S}_k = \frac{1}{N}|X_k|^2$$

is a raw estimate, but has high variance.

Variance reduction is achieved by **Welch averaging**: - split the signal into overlapping segments, - window each segment, - compute DFT of each, - average the magnitudes squared.

Welch Method

Welch estimation reduces variance by averaging spectra of multiple windowed segments. Although resolution decreases slightly, spectral peaks become clearer, and noise-induced fluctuations become less pronounced.

8 Advanced FFT Topics

8.1 Mixed-Radix FFT

The standard radix-2 FFT assumes that the length N of the data is a power of two. When N is not a power of two, one may still accelerate the DFT by decomposing N into its prime factors:

$$N = r_1 r_2 \cdots r_m.$$

Each stage of the FFT exploits recursive factorization of the DFT matrix into smaller blocks; these blocks correspond to radix-r_j butterflies.

A **radix-2 FFT** uses $r_j = 2$ repeatedly, ideal for $N = 2^M$. A **radix-3 FFT** uses $r_j = 3$ repeatedly. A **mixed-radix FFT** allows arbitrary combinations (e.g., $N = 2 \cdot 3 \cdot 5 \cdot 2$), building butterfly structures of mixed sizes.

Why Mixed Radix Helps

The same divide-and-conquer principle applies for any factorization of N. If $N = ab$, then a length-N DFT can be decomposed into a subtransforms of length b and b subtransforms of length a, with sparse twiddle-factor coupling. Mixed-radix FFTs generalize the logic of radix-2 FFTs but with more flexible building blocks.

Mixed-radix FFTs are used when: - signals have arbitrary lengths, - N has large prime factors (sometimes padded), - or performance tuning benefits from specific butterfly sizes.

Modern FFT libraries automatically choose optimal radix strategies to minimize arithmetic and maximize memory locality.

8.2 Real-FFT Optimization

When x_n is real-valued, its DFT satisfies Hermitian symmetry:

$$X_{N-k} = \overline{X_k}.$$

Therefore, only half of the frequency coefficients are independent. A **real FFT** computes all relevant data using roughly half the arithmetic of a complex FFT, with no loss of information.

Symmetry as Free Compression

Real-valued data store no unique information at negative frequency indices, because complex conjugation enforces redundancy: low-frequency phases mirror their high-frequency counterparts. FFT algorithms exploit this structure internally to halve the computational load.

In time-domain applications: - forward real FFTs compress frequency data, - inverse real FFTs expand through symmetric reconstruction.

8.3 Multidimensional FFT

Higher-dimensional data (images, volumetric scans, PDE fields, radar arrays, tomography) require multidimensional transforms.

A **2D FFT** for an $M \times N$ grid applies a 1D FFT along each row, then applies a 1D FFT along each column. This factorization preserves the same $O(N \log N)$ scaling in each dimension:

$$T(M, N) = O(MN(\log M + \log N)).$$

A **3D FFT** applies the same logic in three orthogonal directions. Multidimensional FFTs are widely used in: - image processing, - deconvolution, - holography, - 3D fluid simulation, - MRI reconstruction, - and spectral PDE solvers.

FFT for Higher-Dimensional Convolution

The DFT diagonalizes convolution in any dimension. Filtering an image with a large 2D kernel is dramatically cheaper via FFT than via direct convolution, especially for large spatial supports. The same benefit extends to 3D data.

8.4 FFT for Convolution

Let f and g be discrete signals. Their convolution is

$$(f * g)_n = \sum_m f_m \, g_{n-m}.$$

Direct convolution is $O(N^2)$ when f and g have length N.

The convolution theorem states:

$$\mathcal{F}\{f * g\} = \mathcal{F}f \cdot \mathcal{F}g, \qquad \mathcal{F}^{-1}\{\mathcal{F}f \cdot \mathcal{F}g\} = f * g.$$

Why FFT Makes Convolution Fast

Convolution in the time domain corresponds to pointwise multiplication in the frequency domain. Instead of N^2 pairwise multiplications, we: 1. compute $\mathcal{F}f$ and $\mathcal{F}g$ using $O(N \log N)$ each, 2. multiply them pointwise in $O(N)$, 3. reconstruct via inverse FFT in $O(N \log N)$. Total cost is $O(N \log N)$, a tremendous improvement for large N.

This is foundational in: - digital filtering, - speech/audio processing, - seismic analysis, - large-kernel image filtering, - fast polynomial multiplication, - and spectral numerical PDE methods.

FFT for Large Convolutions

FFT-based filtering replaces expensive sliding correlation with sparse spectral multipliers. Long convolutions become tractable, stable, and frequently faster than any time-domain alternative.

9 Applications

9.1 Signal Processing

Fourier analysis is foundational in digital signal processing (DSP). Time-domain data often exhibit a mixture of long-term structure and rapid oscillations. FFT-based frequency-domain techniques allow one to:

- remove high-frequency noise (spectral truncation),
- isolate oscillatory features,
- perform compression by retaining only the most energetic Fourier modes,
- analyze smoothness, periodicity, or modulation patterns,
- perform rapid convolution-based filtering,
- extract harmonics for feature identification.

Spectral Separation

Low-frequency coefficients capture bulk structure, slow modulation, and trend-like dynamics. High-frequency coefficients capture fine-scale detail, jitter, or stochastic fluctuations. Filtering in the Fourier domain converts denoising into selective frequency editing.

Spectral compression is widely used in audio, image transforms, seismic profiling, radar, and biomedical signals.

9.2 Communication Systems

Modern communication systems encode information by modulating a carrier sinusoid. Fourier analysis on finite windows provides:

- instantaneous spectral energy distribution,
- phase-sensitive demodulation,
- channel equalization in frequency domain,
- multi-carrier waveform construction (e.g., OFDM).

Frequency-Domain Modulation

Amplitude, phase, or frequency modulation of a carrier correspond to controlled manipulations of Fourier coefficients. A band-pass filter isolates a modulated channel by suppressing unrelated frequencies.

Orthogonal Frequency-Division Multiplexing (OFDM) uses FFT and inverse FFT for efficient multi-carrier signaling: each subcarrier corresponds to a distinct orthogonal Fourier mode, enabling simultaneous transmission with minimal mutual interference.

FFT also powers channel estimation, pilot synchronization, adaptive equalization, and spectral shaping.

9.3 Numerical PDEs

Spatially periodic PDEs can be solved spectrally:

$$u_t = \mathcal{L}u + \mathcal{N}(u),$$

where $\mathcal{L}$ is linear and diagonalizable in Fourier space. For instance, the Poisson equation

$$-\Delta u = f,$$

on a periodic domain becomes

$$|k|^2 \widehat{u}_k = \widehat{f}_k,$$

so that solving for u is trivial in Fourier space:

$$\widehat{u}_k = \frac{\widehat{f}_k}{|k|^2}.$$

Diagonalization for PDE Solves

The Fourier transform diagonalizes the Laplacian under periodic boundary conditions. What is a dense differential operator in physical space becomes a diagonal multiplier in frequency space. FFT-based solvers provide enormous acceleration relative to naive finite-difference inversion.

Spectral methods are used in:

- fluid dynamics (Navier–Stokes),
- semi-implicit evolution equations,
- spectral Poisson solvers,
- turbulence modeling,
- pseudo-spectral simulation of nonlinear waves.

Periodic geometry allows FFT-based inversion in $O(N \log N)$, a dramatic improvement over $O(N^3)$ direct linear solves in multidimensional grids.

9.4 Econometric / Financial Time Series

Economic and financial data exhibit multi-scale temporal behavior: long-run drift, seasonal patterns, cyclical macro structure, and high-frequency volatility. Fourier analysis decomposes these effects linearly and orthogonally.

- trend extraction via low-frequency truncation,
- seasonality identification,
- volatility isolation through high-pass residuals,
- cross-spectral analysis between commodities or exchange rates,
- frequency-domain PCA for factor discovery,
- structural decomposition under stationarity.

Fourier as Time-Series Microscope

Financial time series contain components operating at separate frequencies: macroeconomic cycles (very low frequency), weekly or seasonal effects (intermediate), and market microstructure noise (high-frequency). Fourier separation makes each component visible and interpretable.

9.5 Crude Oil Time-Series Example: FFT Decomposition and Reconstruction

Let x_n denote daily crude oil prices on a fixed observation window. Compute the DFT:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i k n/N}.$$

Low-frequency truncation for trend extraction:

$$X'_k = \begin{cases} X_k, & k \le K, \\ 0, & K < k < N-K, \\ X_{N-k}, & k \ge N-K. \end{cases}$$

Reconstruction by inverse FFT:

$$x'_n = \frac{1}{N} \sum_{k=0}^{N-1} X'_k e^{2\pi i k n/N}.$$

Here: - x'_n is the **smooth trend component**, - the residual

$$r_n = x_n - x'_n$$

captures **high-frequency price movement**, daily jitter, and shorter-term structural shifts.

Spectral Trend Extraction

Long-run crude oil price trajectories are contained in the lowest Fourier modes. Shorter-term dynamics reside in mid/high frequencies. Spectral filtering separates structural macro trends from micro fluctuations without ad hoc smoothing assumptions.

This approach supports:

- volatility decomposition,
- seasonal cycle identification,
- regression on isolated frequency bands,
- risk modeling via spectral features,
- fractal or multi-scale analysis.

Geometric Interpretation of Financial Fluctuations

Low-frequency Fourier coefficients act like a macroeconomic baseline. High-frequency coefficients encode intra-market microstructure. Their orthogonality gives rigorous decomposition with clean attribution of variance to temporal scales.