

# Spectral Analysis

---

## Fourier transform and the frequency domain

**Lorenz Schwarz**

Karlsruhe University of Arts and Design (HfG)

Winter Semester 2024/25

[Course info](#)

[← Chapters](#) · [Download PDF ↓](#)

*For iPhone/iPad users, the PDF download is recommended.*

# SPECTRAL ANALYSIS

---

Fourier transform and frequency domain

Lorenz Schwahn · HfG Karlsruhe · CC BY 4.0

# Sound in different domains

---

An audio signal can be described from different perspectives, depending on which aspect of sound is being analyzed.

- Time-based descriptions reveal changes over time
- Frequency-based descriptions reveal spectral content

→ *Both views describe the same signal, but reveal different information.*

Lorenz Schwarz · H20 Karlsruhe · CC BY 4.0

# Time domain vs. frequency domain

---

Audio signals are commonly represented in two domains:

- Time domain:
  - Amplitude as a function of time
- Frequency domain:
  - Distribution of energy across frequencies (magnitude and phase)

(Analog signals are continuous; digital signals are discrete.)

# Time domain

---

The time domain shows how a signal's amplitude changes over time.

- Analog signals: continuous in time and amplitude
- Digital signals: discrete samples in time and amplitude

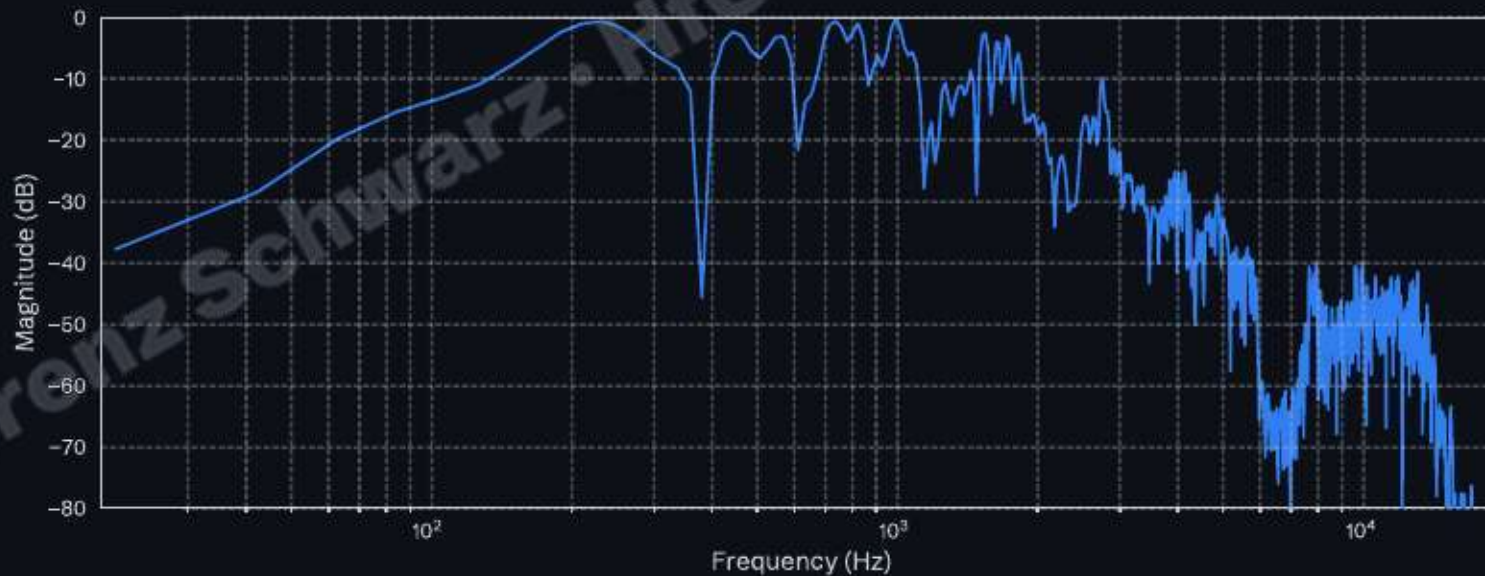
→ *The time domain reveals transients, timing, and amplitude changes.*



# Frequency domain

The frequency domain describes a signal in terms of its frequency components rather than time.

- shows how much energy each frequency contributes



# Spectrum

---

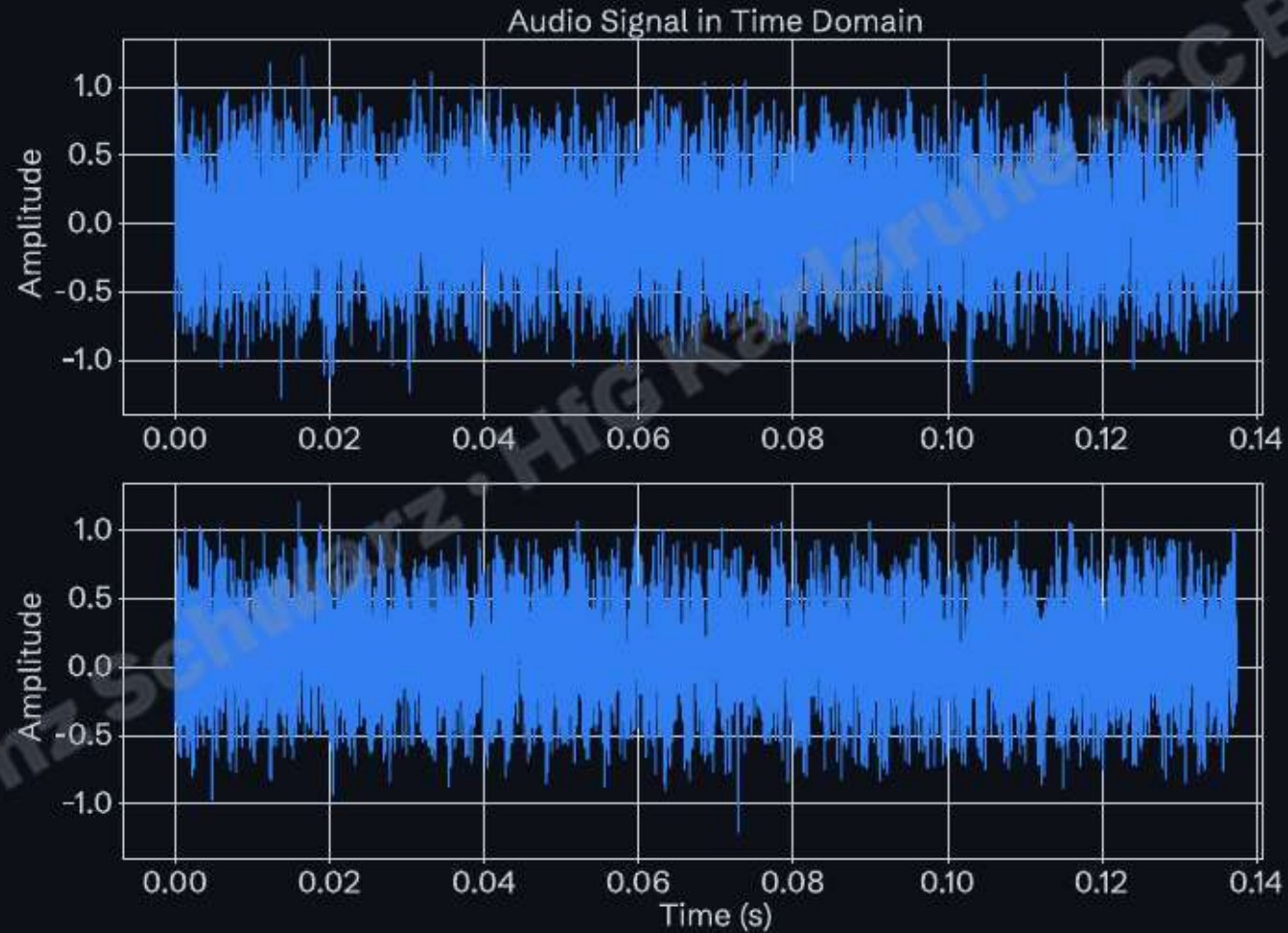
The spectrum shows a moment of a signal's frequency content.

- Amplitude (or magnitude) as a function of frequency
- Optionally includes phase information

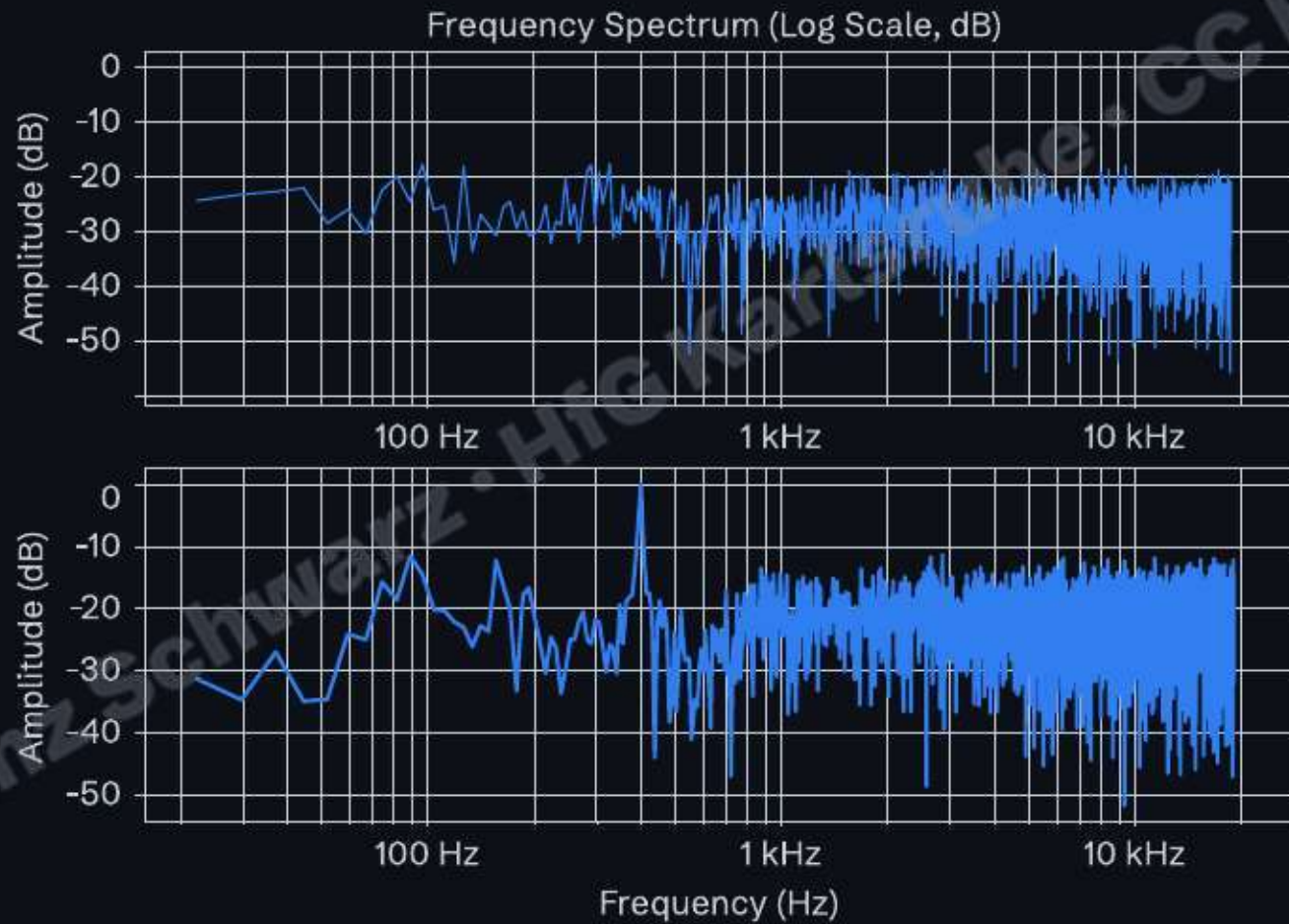
→ *The spectrum is the primary tool for analyzing timbre and harmonic structure.*

Lorenz Schwarz · HfG Karlsruhe · CC BY 4.0

Which signal contains a 400 Hz sine?



The second spectrum shows a spike at 400Hz



# Timbre and spectrum

---

## Timbre (perceptual):

The sonic quality that distinguishes instruments playing the same pitch

## Spectrum (technical):

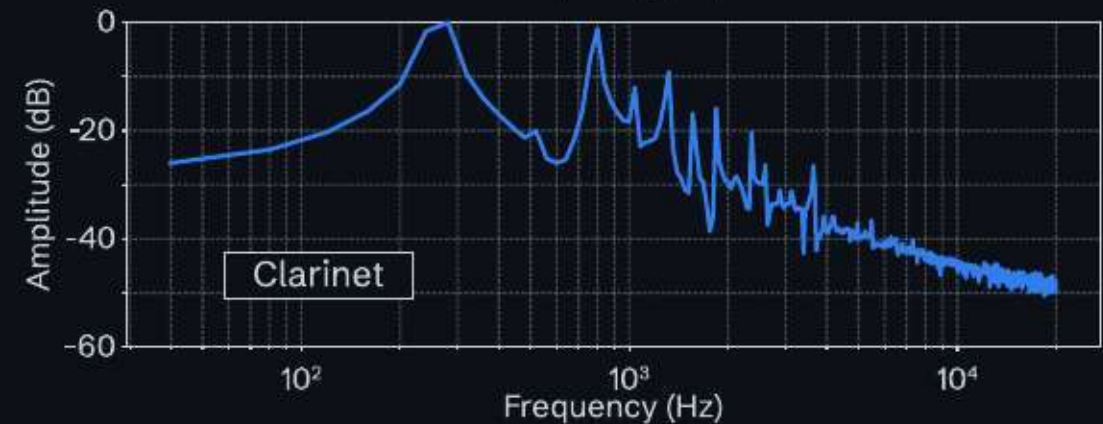
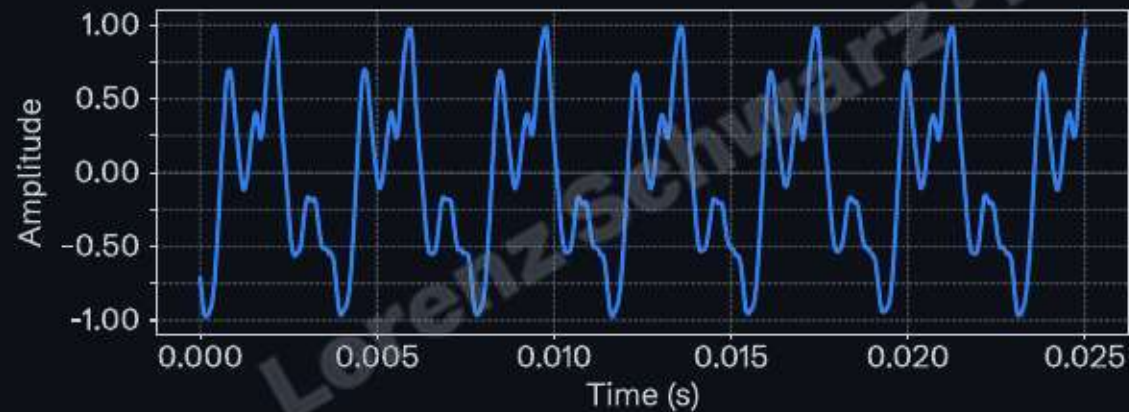
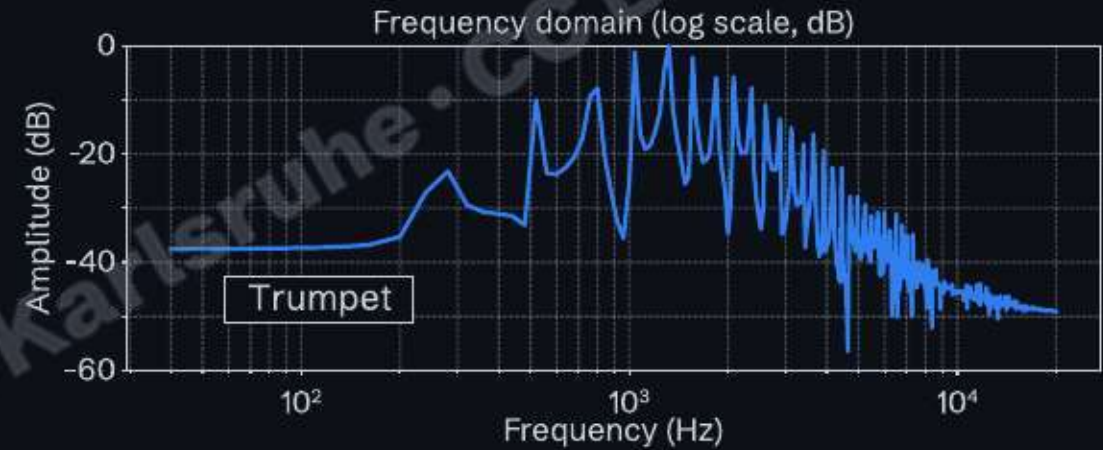
The distribution of frequency components and their amplitudes

## Listen to the same pitch (C4 $\approx$ 261.6 Hz):

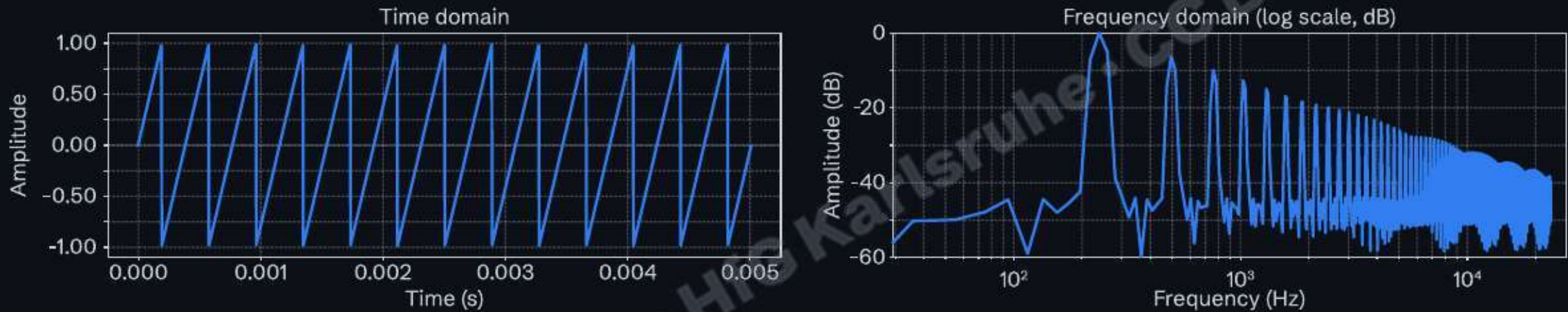
Trumpet:  Play    Clarinet:  Play

→ *Same pitch, different timbre*

# Comparing clarinet and trumpet at 260 Hz



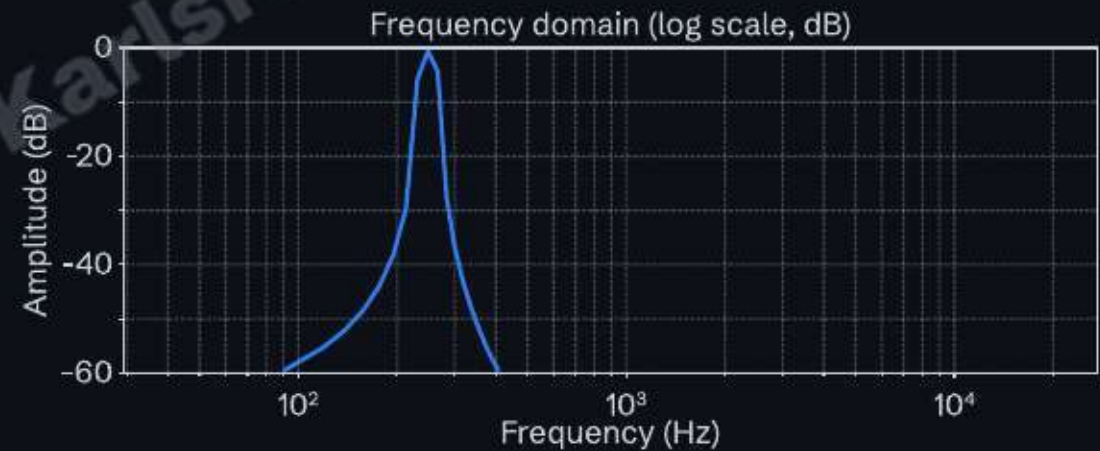
# Understanding spectra with a sawtooth wave



- Each peak in the spectrum represents one sine wave (partial or harmonics)
  - Harmonic series 260, 520, 780, 1040... Hz
- ▶ Play sawtooth wave C4  $\approx$  260 Hz

# Pure tone (sine wave)

A [sine wave](#) is a single frequency component, the fundamental building block of all sounds

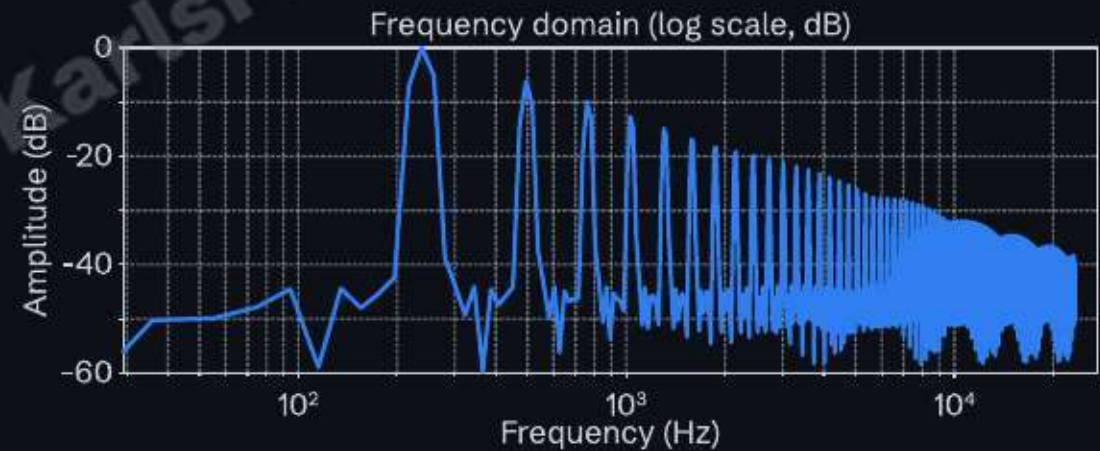


$$x(t) = A \sin(2\pi f_0 t + \varphi)$$

[View sine wave on Desmos](#)

# Complex tones (example: sawtooth)

Musical instrument sounds and basic waveforms (except sine) contain many sine waves ([click for graphing calculator](#)).



$$x(t) = \frac{2A}{\pi} \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n} \sin(2\pi n f_0 t)$$

# Fourier transform

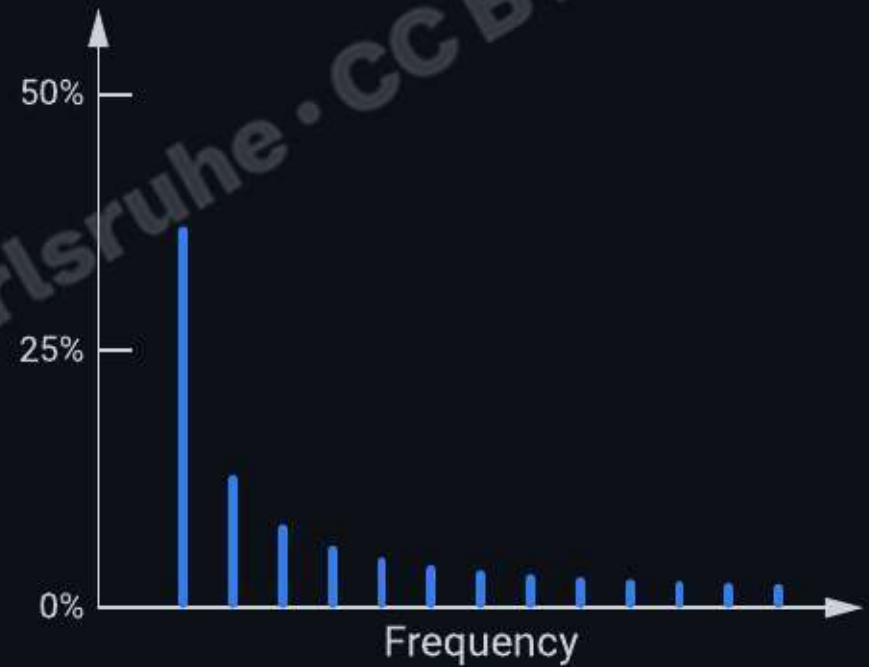
---

Decomposes a signal from the time domain (waveform) into the frequency domain (spectrum):

- Reveals the individual sine wave components and their amplitudes (and phases)
- Shows which frequencies are present and how strong they are

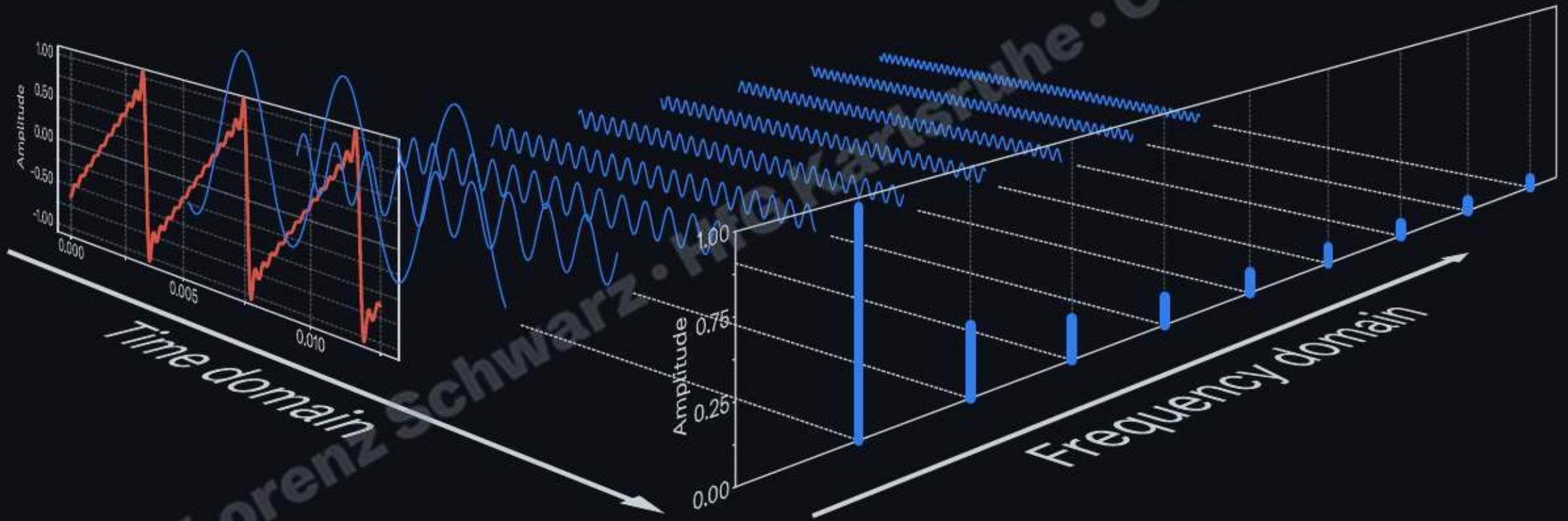
→ *Any complex sound can be represented as a sum of sine waves.*

# Fourier series of a saw tooth wave (approximation)



Lorenz Schwarz · HfG Karlsruhe · CC BY 4.0

# Fourier transform of a sawtooth wave



# Time-domain signals and spectral analysis

---

A time-domain signal  $x(t)$  represents amplitude values over time, either continuous or discrete  $x(n)$ . The spectrum  $X(\omega)$  is a weighting function that describes how harmonic components are combined to reconstruct the time-domain signal as a sum.

- Input: time-domain signal  $x(t)$
- Output: frequency-domain spectrum  $X(\omega)$

→ *The spectrum represents the amplitude and phase of each frequency component.*

# Fourier Transform

---

Fourier transform (analysis formula):

- Break the signal into its frequency components

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt, \quad \omega \in (-\infty, \infty)$$

Inverse transform (synthesis formula):

- Rebuild the signal from its frequency components.

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega)e^{j\omega t} d\omega$$

# Euler's formula and complex numbers

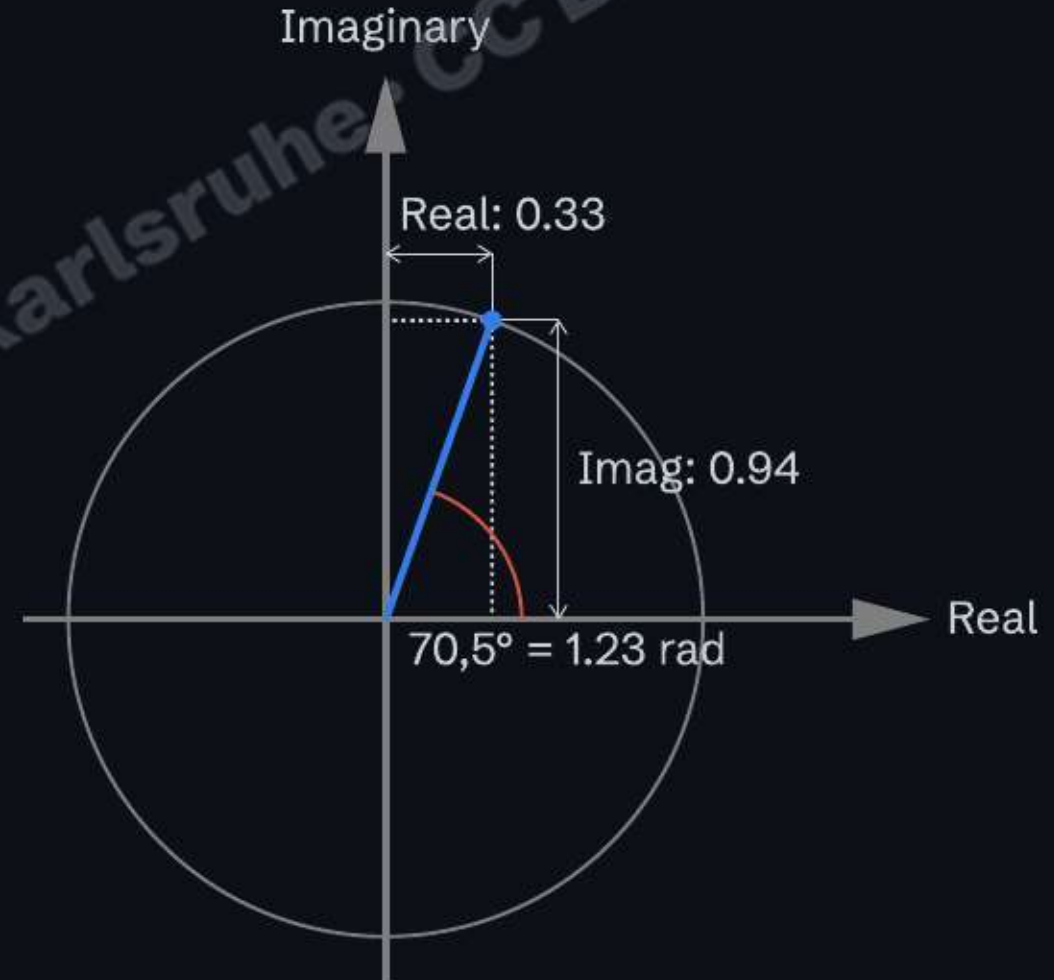
A complex exponential combines cosine and sine into a single expression representing sinusoidal components:

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

where  $j = \sqrt{-1}$  is the imaginary unit.

- **Real part:**  $\cos(\omega t)$  — cosine component
- **Imaginary part:**  $j \sin(\omega t)$  — sine component

→ *This representation is fundamental to the Fourier transform, allowing efficient encoding of both amplitude and phase.*



# Discrete Fourier transform (DFT)

---

For digital audio, the DFT is used, which operates on discrete, finite-duration signals by replacing the continuous integral with a finite sum. The Fast Fourier Transform (FFT) is an efficient algorithm for computing the DFT.

$$X(\omega_k) = \sum_{n=0}^{N-1} x(t_n) e^{-j\omega_k t_n}, \quad k = 0, 1, 2, \dots, N-1$$

inverse DFT:

$$x(t_n) = \frac{1}{N} \sum_{k=0}^{N-1} X(\omega_k) e^{j\omega_k t_n}, \quad n = 0, 1, 2, \dots, N-1$$

# Discrete Fourier transform (DFT)

---

- No calculus needed — uses finite sums, avoids infinities.
- Assumes finite, sampled signals: Digital processing uses sampled signals.

→ *DFT is simpler and more computationally relevant than FT.*

Lorenz Schwarz · HfG Karlsruhe · CC BY 4.0

# Quantities of the DFT formula:

---

$$\sum_{n=0}^{N-1} = f(0) + f(1) + \dots + f(N - 1)$$

$x(t_n)$  = input signal amplitude at time  $t_n$  (sec)  
 $t_n$  =  $nT = nth$  sampling instant (sec),  $n$  an integer  $\geq 0$   
 $T$  = sampling interval (sec)  
 $X(\omega_k)$  = spectrum of  $x$  at frequency  $\omega_k$   
 $\omega_k$  =  $k\Omega = kth$  frequency sample (rad/s)  
 $\Omega$  =  $\frac{2\pi}{NT}$  = radian-frequency sampling interval (rad/s)  
 $f_s$  =  $1/T$  = sampling rate (samples/second, or Hertz (Hz))  
 $N$  = number of time samples = number of frequency samples (integer)

# Fast Fourier Transform (FFT)

---

The FFT is an efficient algorithm for computing the DFT, reducing computational complexity from  $N^2$  to  $N \log N$  operations.

This allows efficient computation for:

- Real-time spectrum analysis
- Frequency-domain filtering and equalization
- Convolution-based processing (reverberation, time-stretching)

→ *The FFT is an algorithmic optimization of the DFT computation.*

# Frequency bins

---

The FFT produces discrete frequency values called bins, each representing a specific frequency component. Each bin contains amplitude and phase information for its frequency component.

**Frequency of bin  $k$ :**

$$f_k = \frac{k}{N} f_s$$

where  $k = 0, 1, 2, \dots, N - 1$  is the bin index,  $f_s$  is the sampling rate, and  $N$  is the FFT size.

For real signals: Number of bins =  $\frac{N}{2} + 1$  (due to symmetry)

# Frequency resolution

---

Frequency resolution (bin spacing) determines how finely the spectrum is divided:

$$\Delta f = \frac{f_s}{N}$$

where  $f_s$  is the sampling rate and  $N$  is the FFT size.

- Larger FFT size  $\rightarrow$  smaller  $\Delta f \rightarrow$  better frequency resolution
- Frequencies separated by less than  $\Delta f$  cannot be distinguished

## Example

---

Sampling rate: 44.1 kHz

FFT 1024-sample

$$\Delta f = \frac{44100}{1024} \approx 43 \text{ Hz per bin}$$

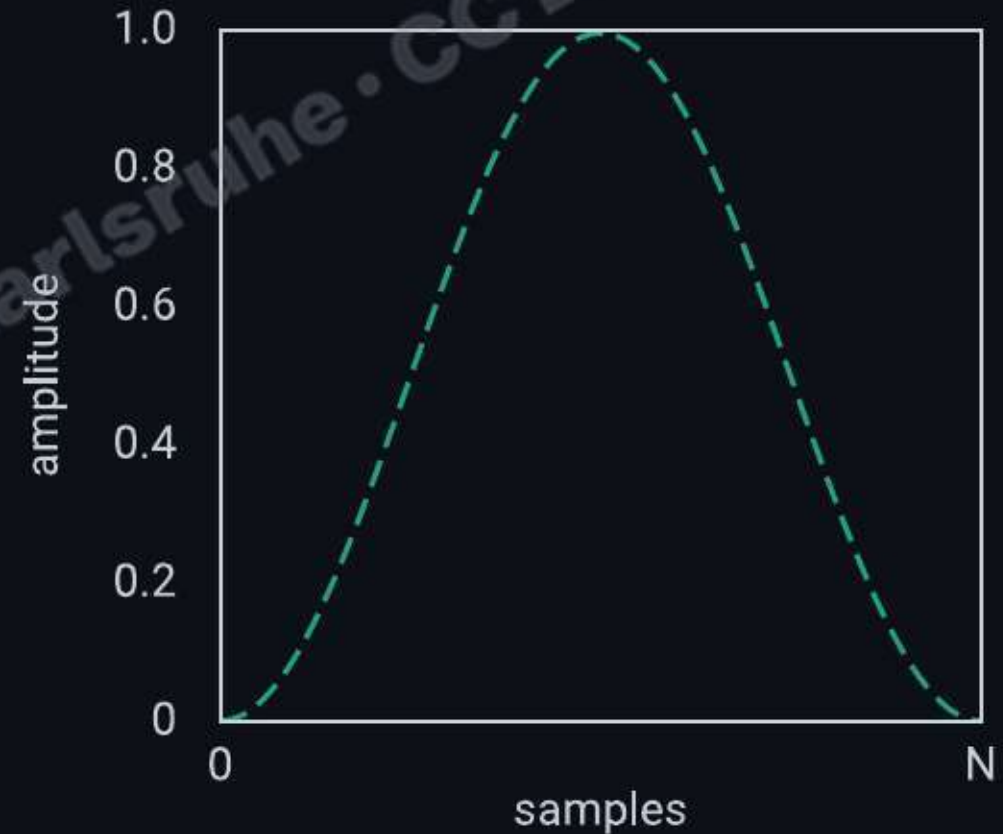
This means frequencies within a frequency band 43 Hz fall in the same bin and cannot be distinguished.

# Window function

Tapering function that smoothly reduces signal amplitude to zero at analysis window boundaries, minimizing discontinuities and spectral leakage.

- Applied when signals contain non-integer periods within the FFT window
- Typically symmetrical, bell-shaped functions
- Common types: Hann, Hamming, Blackman-Harris

→ *Trade-off: Reduced leakage vs. reduced frequency resolution.*



Hann window

# Spectral leakage and windowing

---

Spectral leakage occurs when the analysis window doesn't contain an exact integer number of wave cycles.

- The signal appears discontinuous at window edges
- This discontinuity creates artificial frequency components
- Energy 'leaks' from the true frequency into neighboring bins

→ *A window function tapers the signal smoothly to zero at the edges.*

# Window characteristics in frequency domain

Each window function has a characteristic frequency response with a main lobe and side lobes:

- **Main Lobe:**
  - Central peak determining frequency resolution. Width measured between first zeros (null points).
- **Side Lobes:**
  - Secondary peaks flanking the main lobe. Height (in dB) indicates leakage suppression quality.
- **Trade-off:**
  - Lower side lobes require wider main lobes, reducing frequency resolution.



Rectangular (-13 dB side lobes) vs. Hann (-31 dB side lobes)

# Selecting a window function

| Window                     | Main lobe width       | Side lobe level     | Use case  |
|----------------------------|-----------------------|---------------------|---|
| Rectangular<br>(no window) | Narrowest<br>(2 bins) | Highest<br>(-13 dB) | Maximum frequency resolution,<br>integer number of periods    |
| Hann                       | Medium<br>(4 bins)    | -31 dB              | General purpose,<br>good balance of resolution and leakage    |
| Hamming                    | Medium<br>(4 bins)    | -42 dB              | Better side lobe suppression,<br>8-bit systems, telephony     |
| Blackman-Harris            | Widest<br>(6 bins)    | -92 dB<br>(4-term)  | High dynamic range,<br>very low leakage critical applications |

Trade-off: Better side lobe suppression = wider main lobe = reduced frequency resolution.

# FFT size (window size)

---

Number of samples per FFT computation. Determines the time-frequency resolution trade-off:

- Larger size: Better frequency resolution, worse time resolution
- Smaller size: Better time resolution, worse frequency resolution

Common sizes: 256, 512, 1024, 2048, 4096 (powers of 2)

| FFT Size     | Frequency Resolution | Time Resolution      |
|--------------|----------------------|----------------------|
| Small (256)  | Poor (coarse bins)   | Good (fast response) |
| Large (4096) | Good (fine bins)     | Poor (slow response) |

# Applications of the Fourier transform

---

Theoretical approaches:

- Organs: Additive synthesis for sound creation.
- Tone Wheels: Used in the Telharmonium by Thaddeus Cahill (1898).

Spectral audio signal processing:

- Additive synthesis
- Digital filter design
- Vocoder: Manipulation of speech and audio signals.

# Applications of spectral analysis

---

- Analysis: spectrum analyzers in DAWs
- Processing: convolution reverb, spectral effects, noise reduction, time stretching
- Synthesis: additive synthesis, spectral resynthesis

Lorenz Schwarz · HfG Karlsruhe · CC BY 4.0

# Spectrogram, sonogram

---

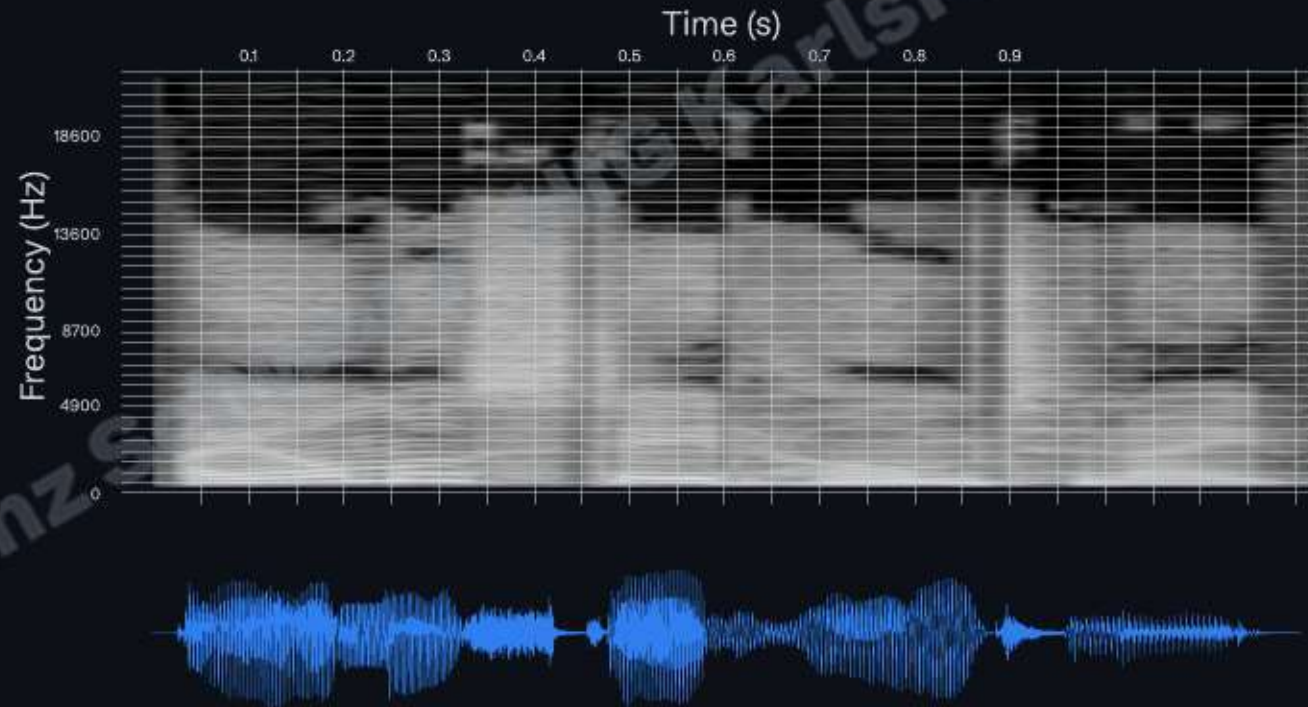
A time-varying visual representation of a signal's frequency content:

- **X-axis:** Time
- **Y-axis:** Frequency
- **Color/brightness:** Amplitude

Lorenz Schwarz · HfG Karlsruhe · CC BY 4.0

# Spectrogram

Spectrograms reveal temporal evolution of spectral content. Helpful for analyzing speech, music, and environmental sounds.



# Copyright and Licensing

---

Original content: © 2025 Lorenz Schwarz

Licensed under [CC BY 4.0](#). Attribution required for all reuse.

Includes: text, diagrams, illustrations, photos, videos, and audio.

**Third-party materials:** Copyright respective owners, educational use.

**Contact:** [lschwarz@hfg-karlsruhe.de](mailto:lschwarz@hfg-karlsruhe.de)

[← Chapters](#)