Sampling Theory 101

Oliver Kreylos

kreylos@cs.ucdavis.edu

Center for Image Processing and Integrated Computing (CIPIC)

Department of Computer Science

University of California, Davis

Outline

- Things that typically go wrong
- Mathematical model of sampling
- How to do things better

Things That Typically Go Wrong

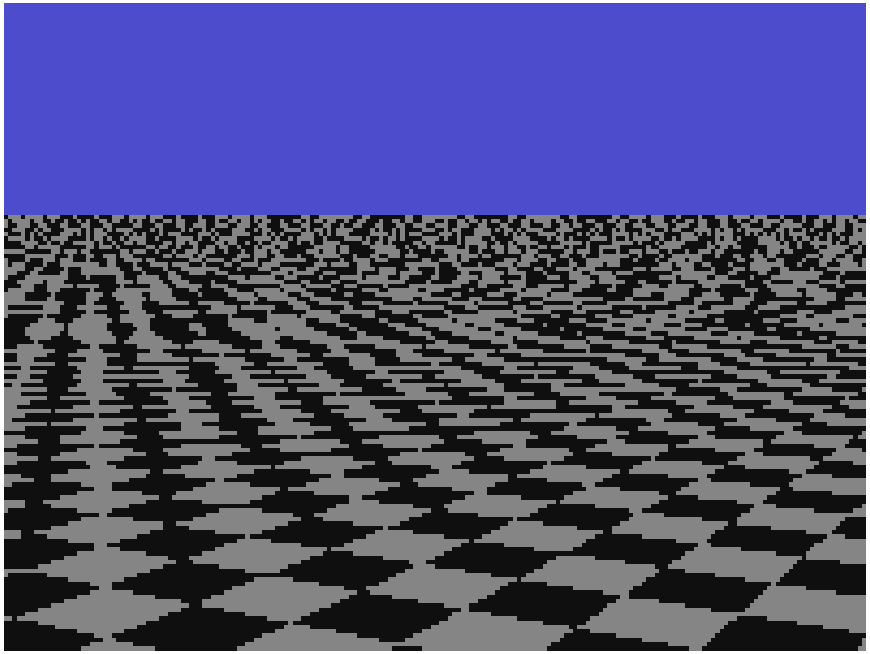
Where Does Sampling Occur?

Almost all data we are dealing with is discrete

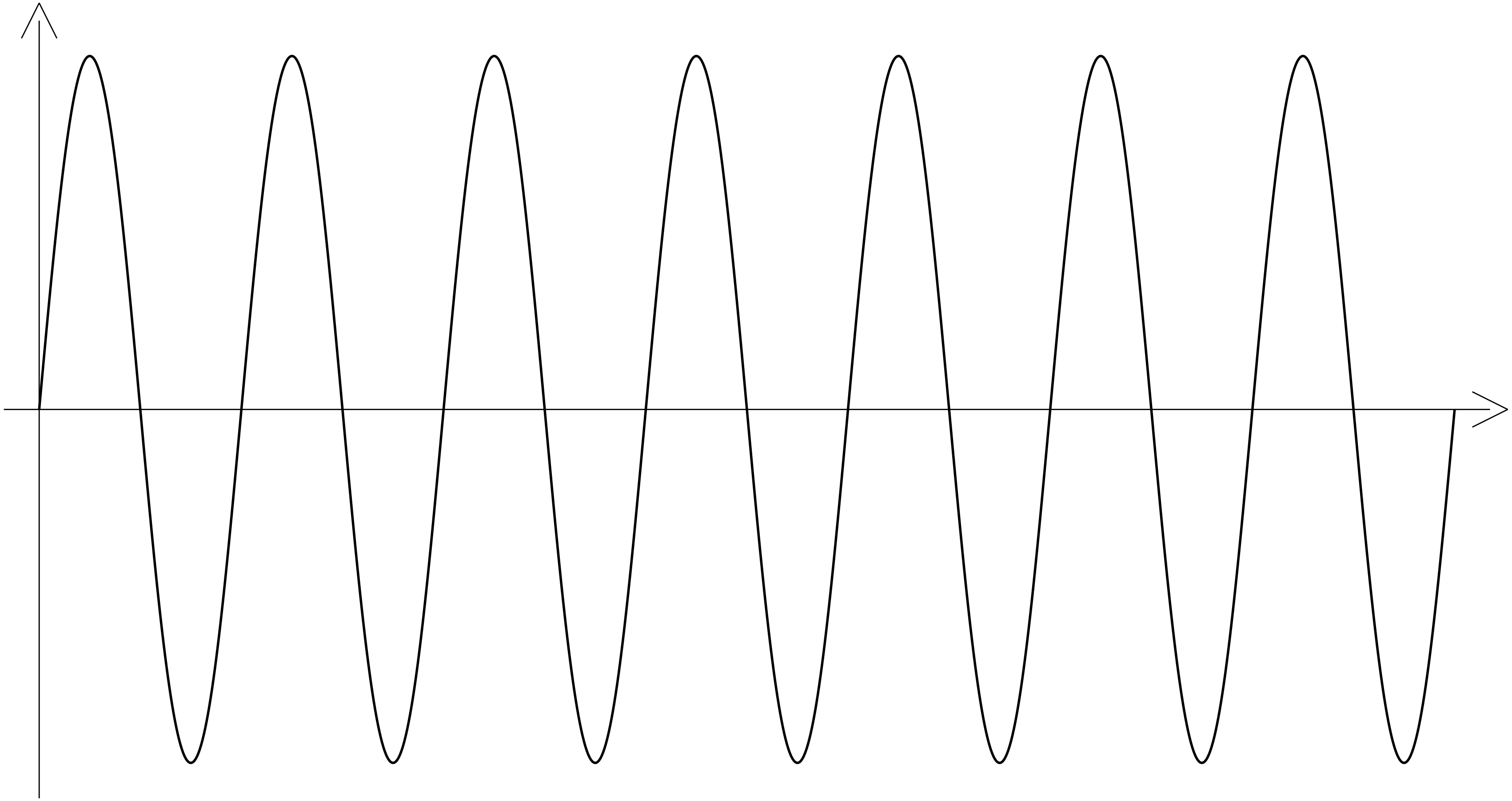
- Evaluation of sampled functions at arbitrary sites
- Volume rendering
- Isosurface extraction
- Ray tracing
- . . .

What Went Wrong Here?

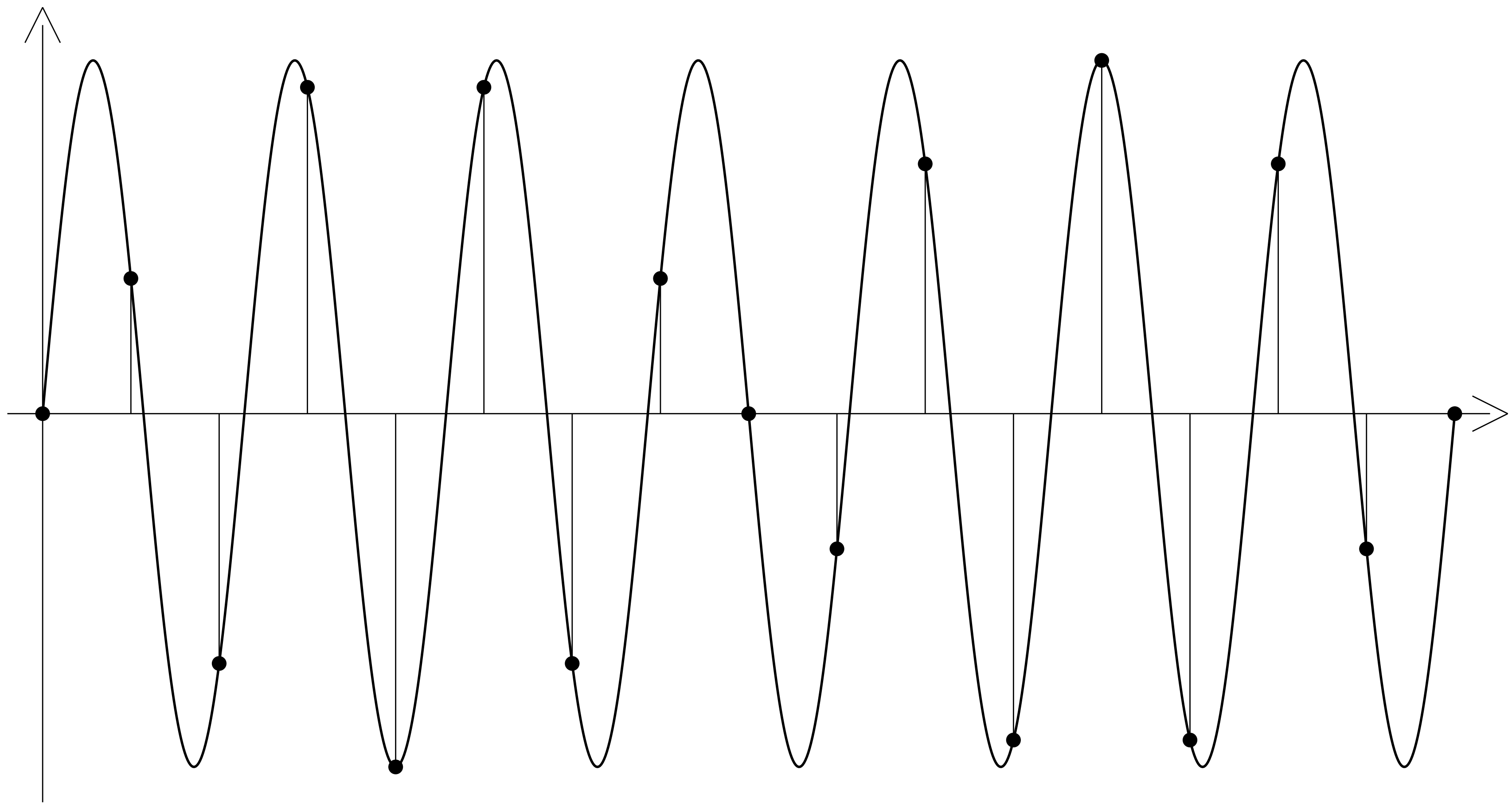
Typical ray tracing example:



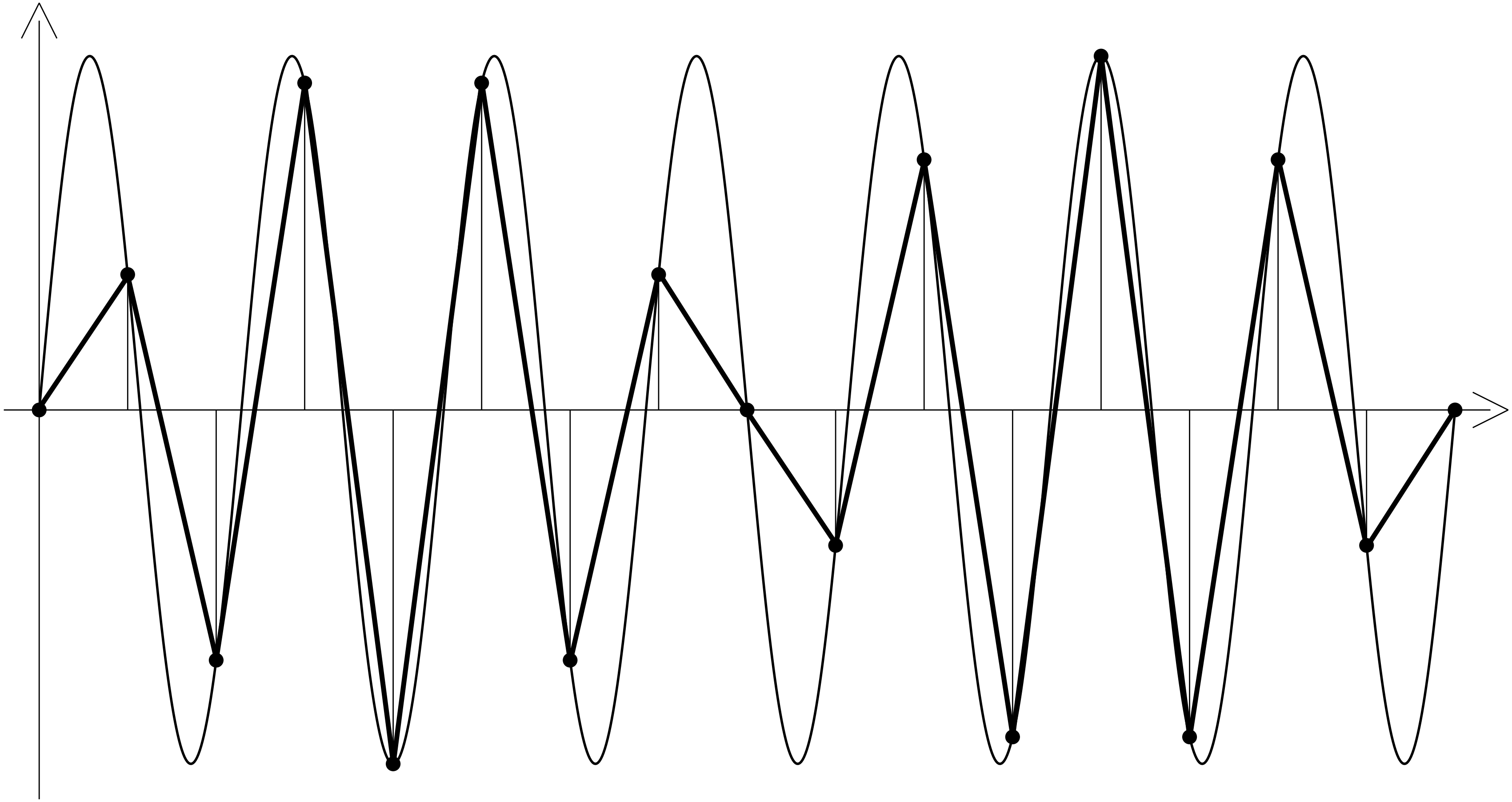
Sampling and Reconstruction



Sampling and Reconstruction



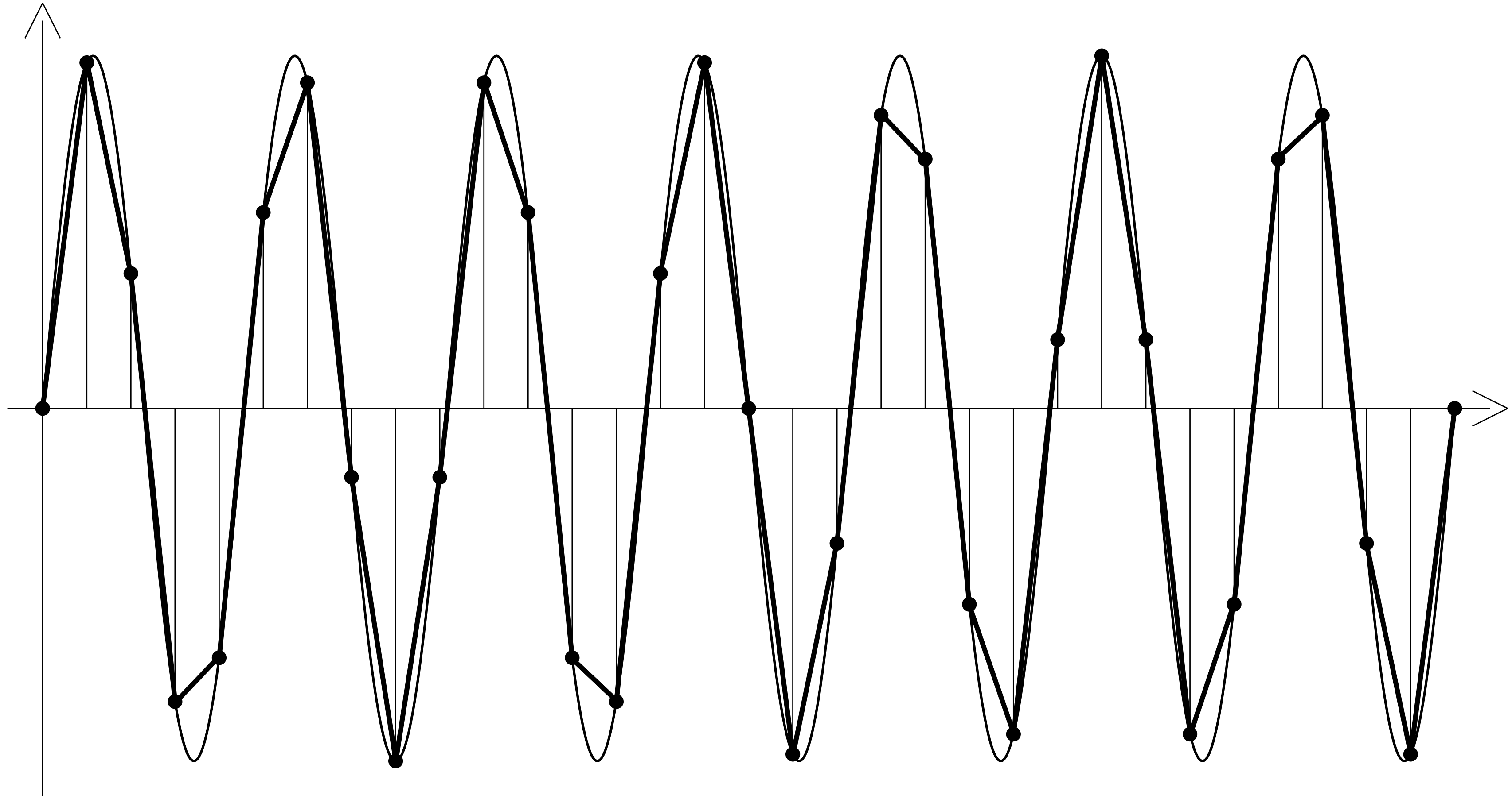
Sampling and Reconstruction



Sampling and Reconstruction

- Undersampling?
- Sampling “below Nyquist rate”?
- Quick solution: Double sampling rate

Sampling and Reconstruction



Sampling and Reconstruction

- Things get better, but are still bad
- Isn't sampling above the Nyquist rate supposed to solve all problems?

Sampling Theory

To understand what went wrong on the last slide, we need a mathematical model of sampling.

Mathematical Model of Sampling

Choice of Domain

- Process of sampling and reconstruction is best understood in frequency domain
- Use *Fourier transform* to switch between time and frequency domains
- Function in time domain: *signal*
- Function in frequency domain: *spectrum*

Fourier Transform I

- Many functions $f\colon \mathbf{R} \to \mathbf{R}$ can be written as sums of sine waves

$$f(x) = \sum_{\omega} a_{\omega} \sin(\omega x + \theta_{\omega})$$

- $\omega = 2\pi \cdot \mathit{frequency}$ is angular velocity,
- a_{ω} is amplitude, and
- θ_{ω} is phase shift

Fourier Transform II

- Moving to complex numbers simplifies notation:

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

  (*Euler identity*)

- One complex coefficient $c_\omega e^{i\omega x}$ encodes both amplitude and phase shift
- Moving to integral enlarges class of representable functions

Fourier Transform III

- Now, for almost all f:

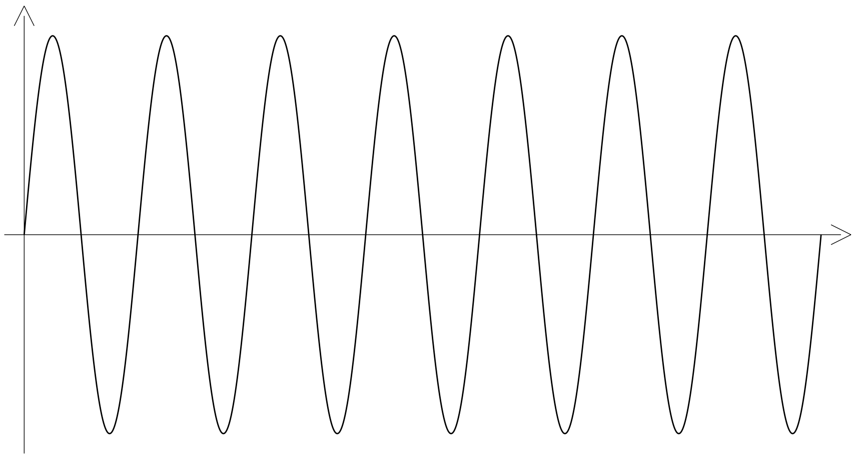
$$f(x) = \int F(\omega)e^{i\omega x}\,\mathrm{d}\omega$$

- $F(\omega)$ is the spectrum of $f(x)$, and the above operator is the *inverse Fourier transform*
- Its inverse, the *Fourier transform*, is

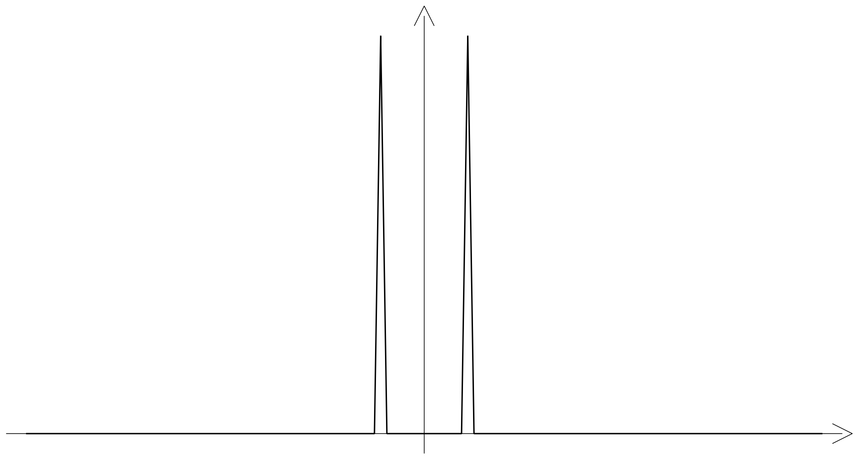
$$F(\omega) = \int f(x)e^{-i\omega x}\,\mathrm{d}x$$

Some Signals and Their Spectra

Single sine wave $f(x) = \sin(\omega x)$:

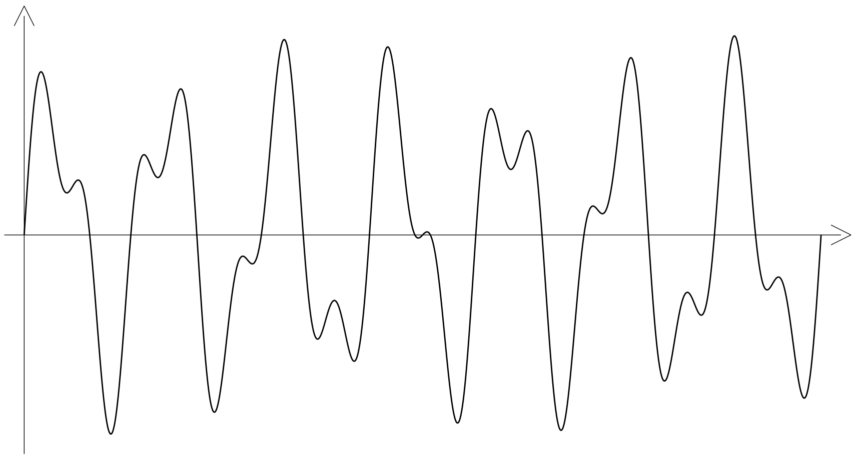


$\longleftrightarrow$

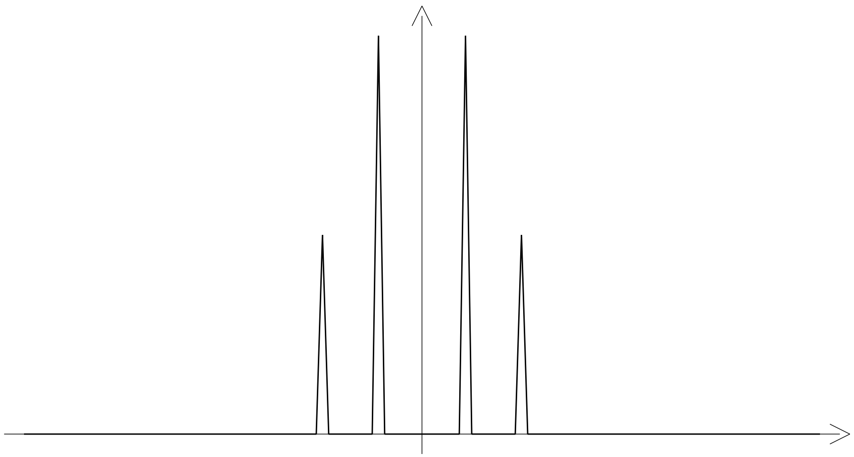


Some Signals and Their Spectra

Sum of two sine waves
$f(x) = \sin(\omega x) + 0.5\sin(2\omega x)$:

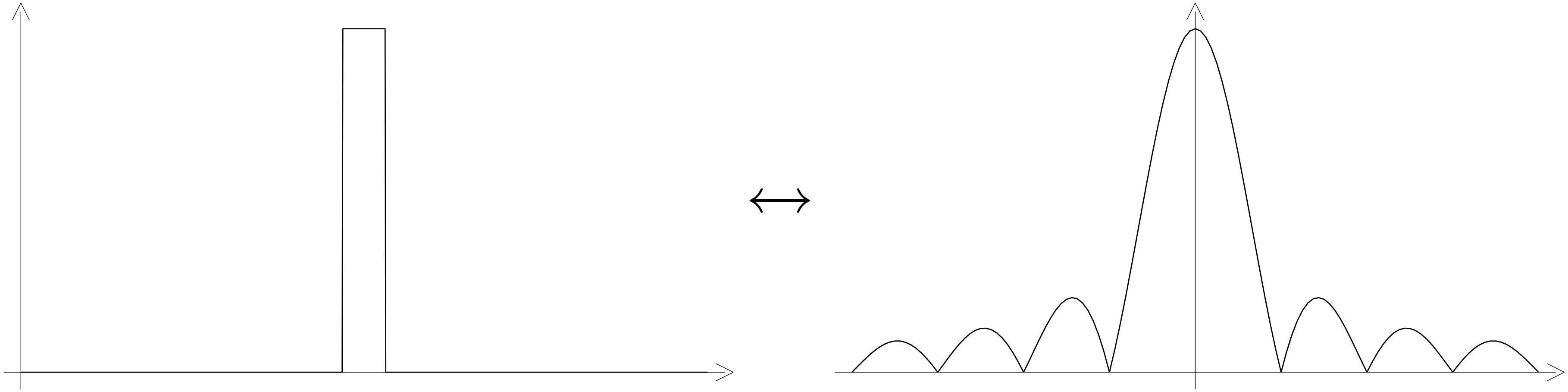


$\longleftrightarrow$



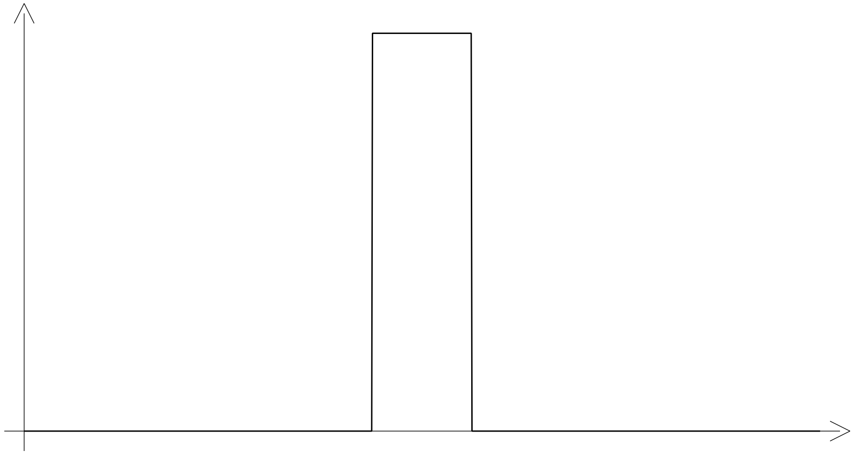
Some Signals and Their Spectra

Box function:

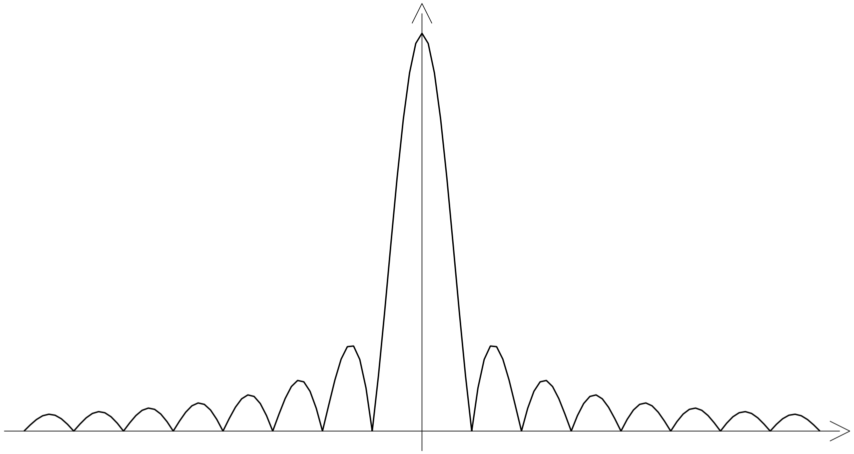


Some Signals and Their Spectra

Wider box function:

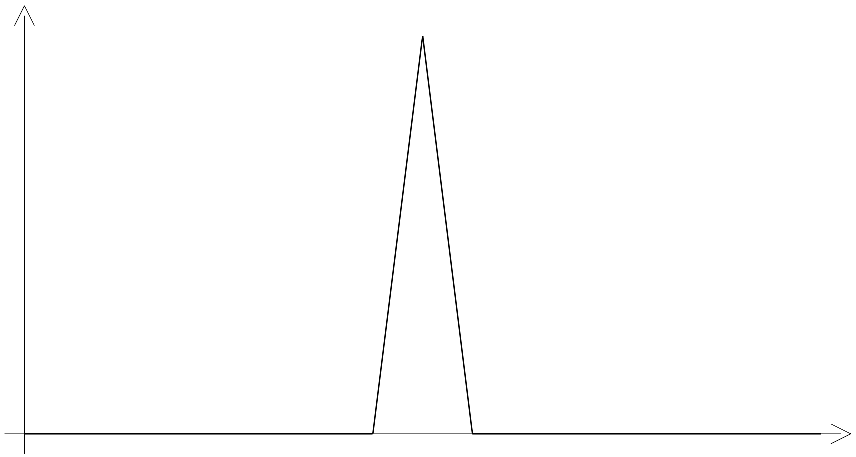


$\longleftrightarrow$

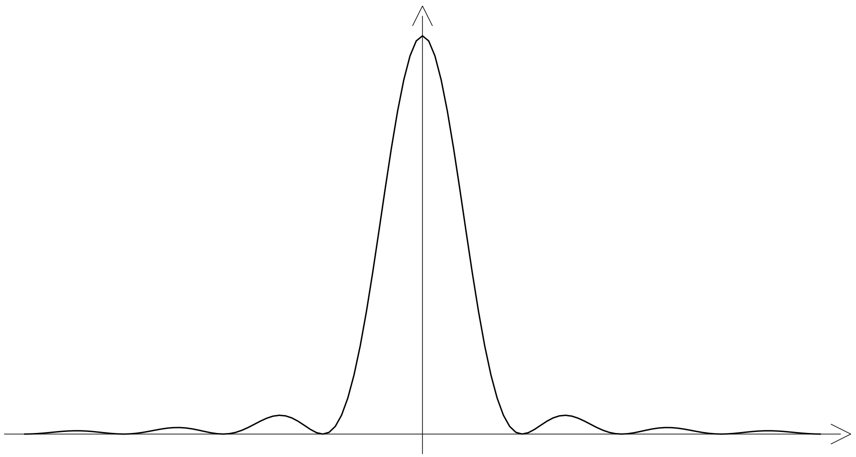


Some Signals and Their Spectra

Triangle function:

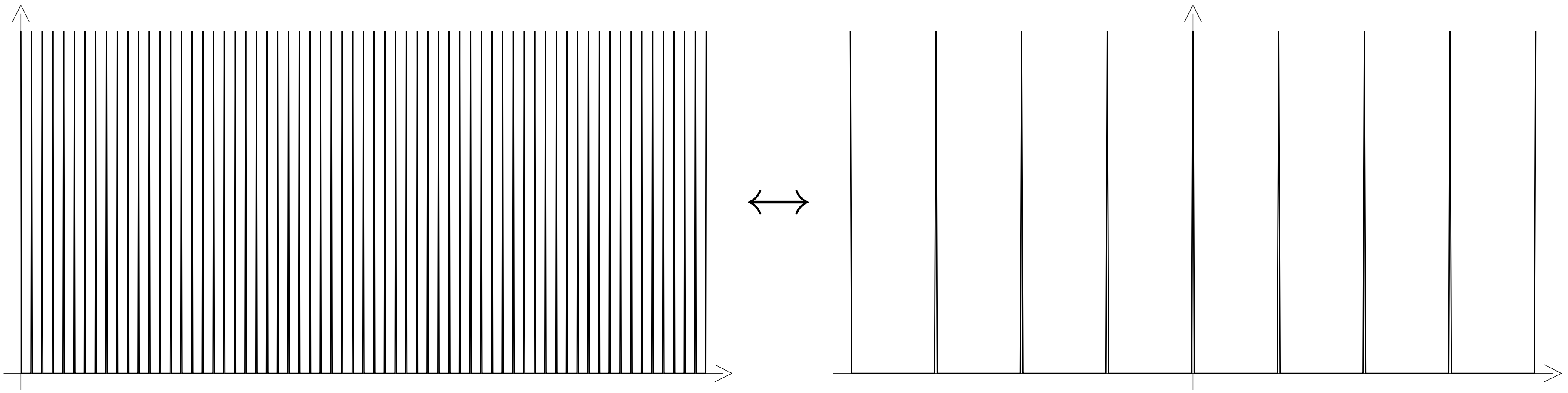


$\longleftrightarrow$



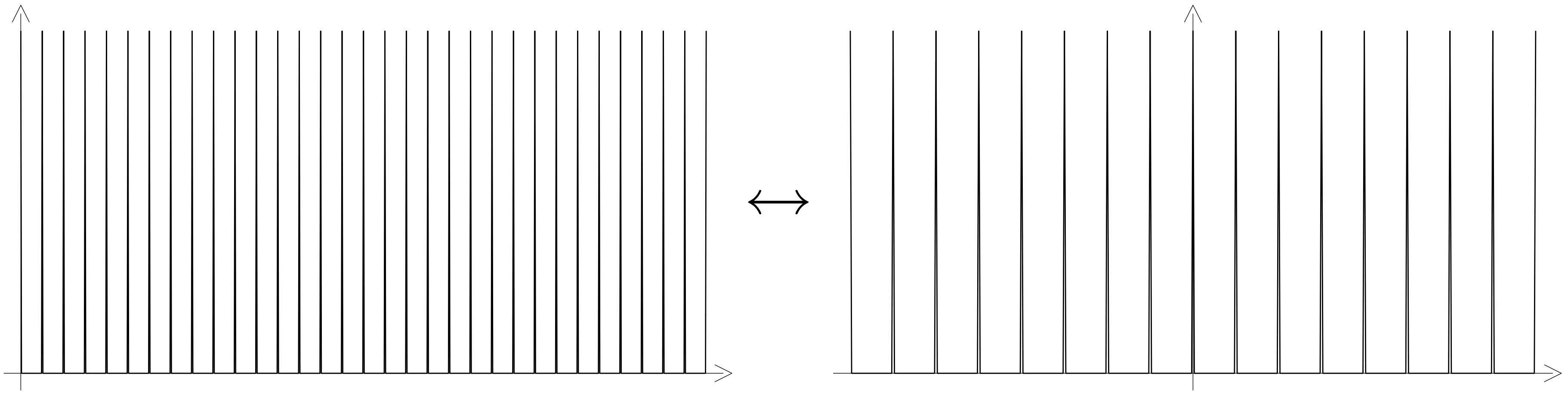
Some Signals and Their Spectra

Comb function:



Some Signals and Their Spectra

Wider comb function:



Convolution I

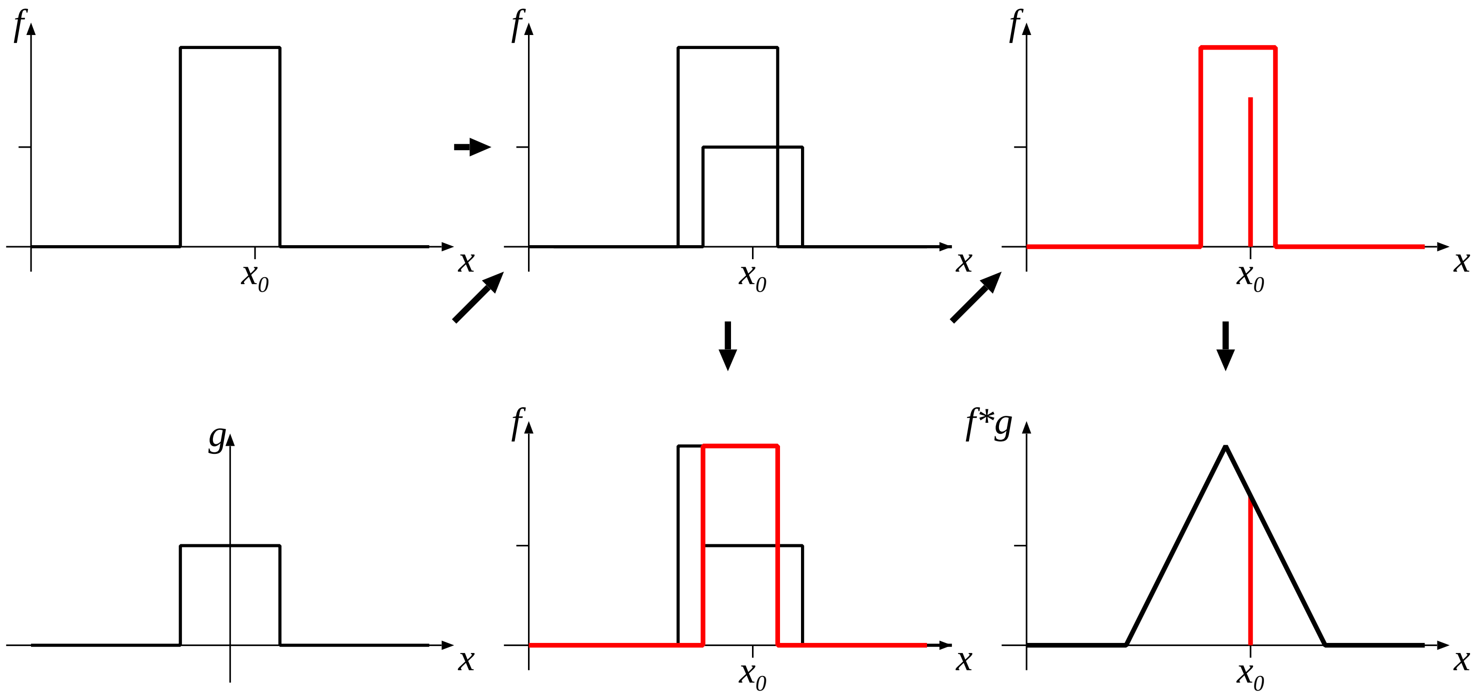
- The *convolution operator* is a generalized formula to express weighted averaging of an input signal f and a *weight function* or *filter kernel* g:

$$(f * g)(x) = \int f(t) \cdot g(x - t)\, \mathrm{d}t$$

- One important application of convolution is reconstructing sampled signals

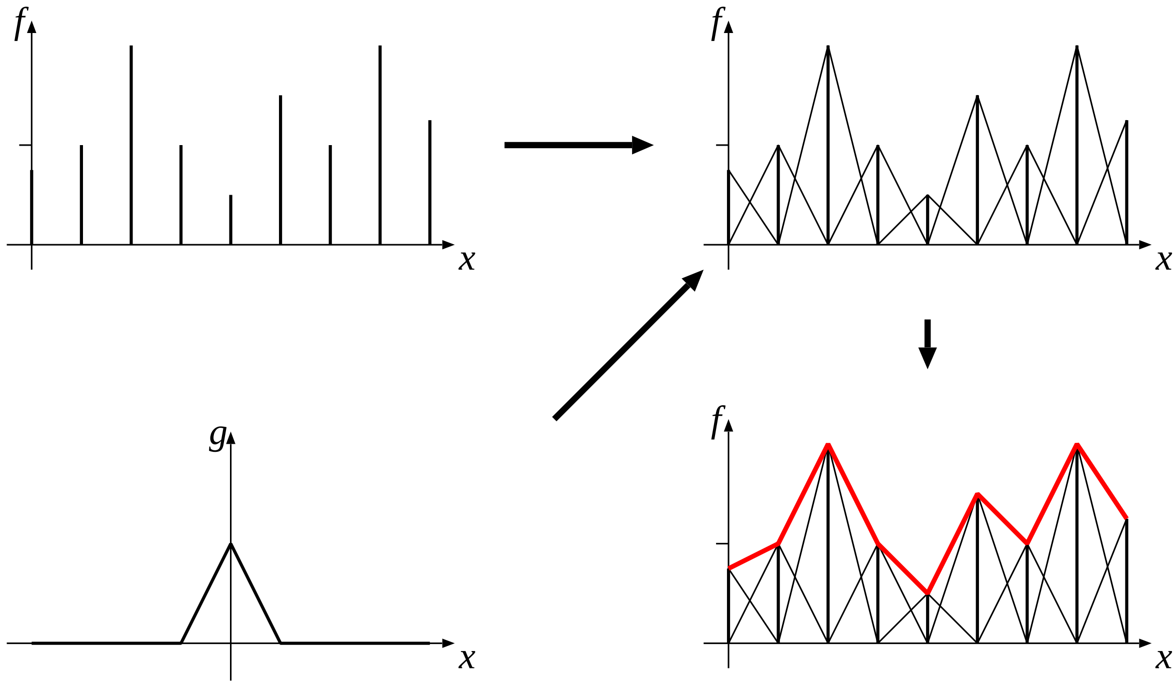
Convolution II

$$(f * g)(x_0) = \int f(t) \cdot g(x_0 - t)\,\mathrm{d}t$$



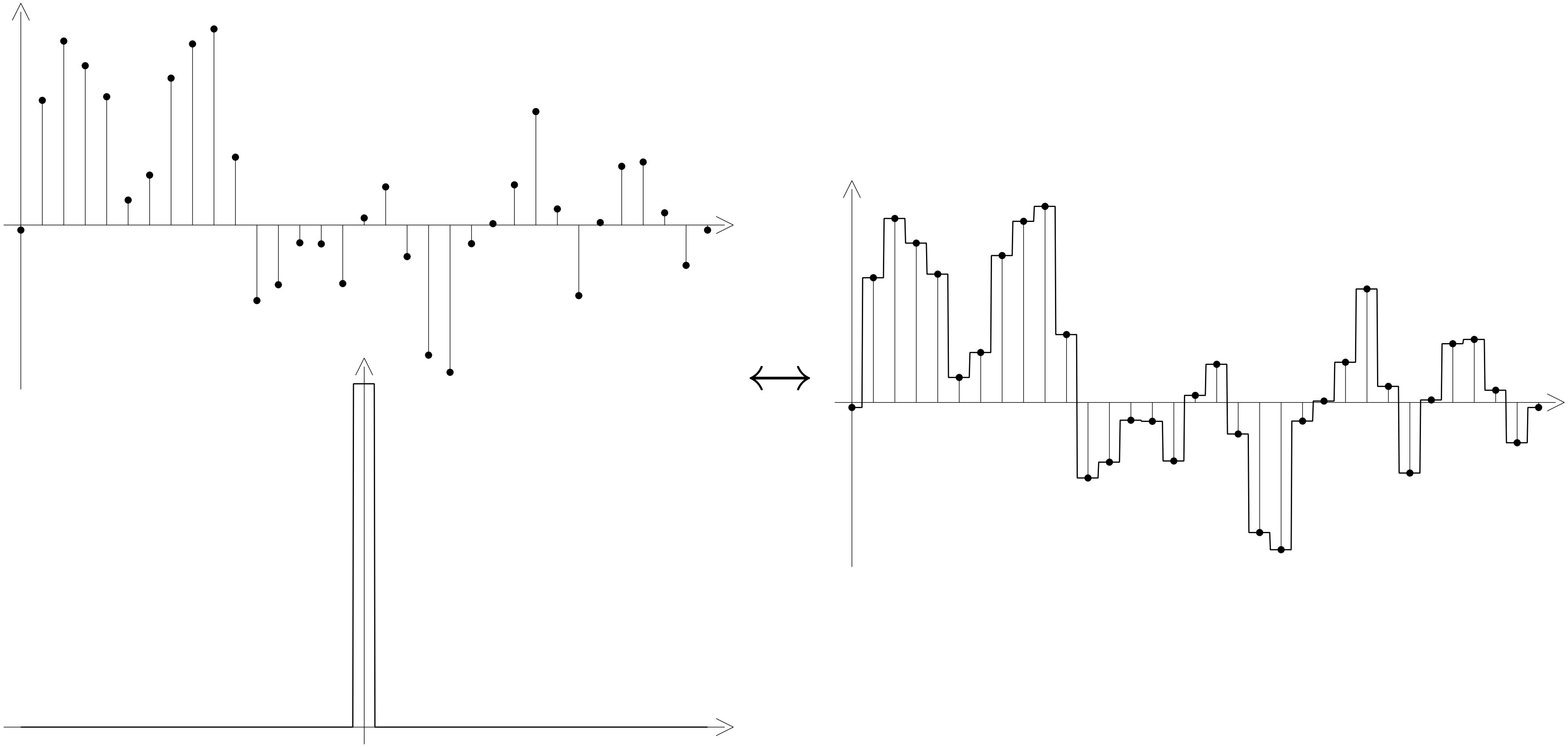
Convolution III

Linear interpolation can be interpreted as convolution:



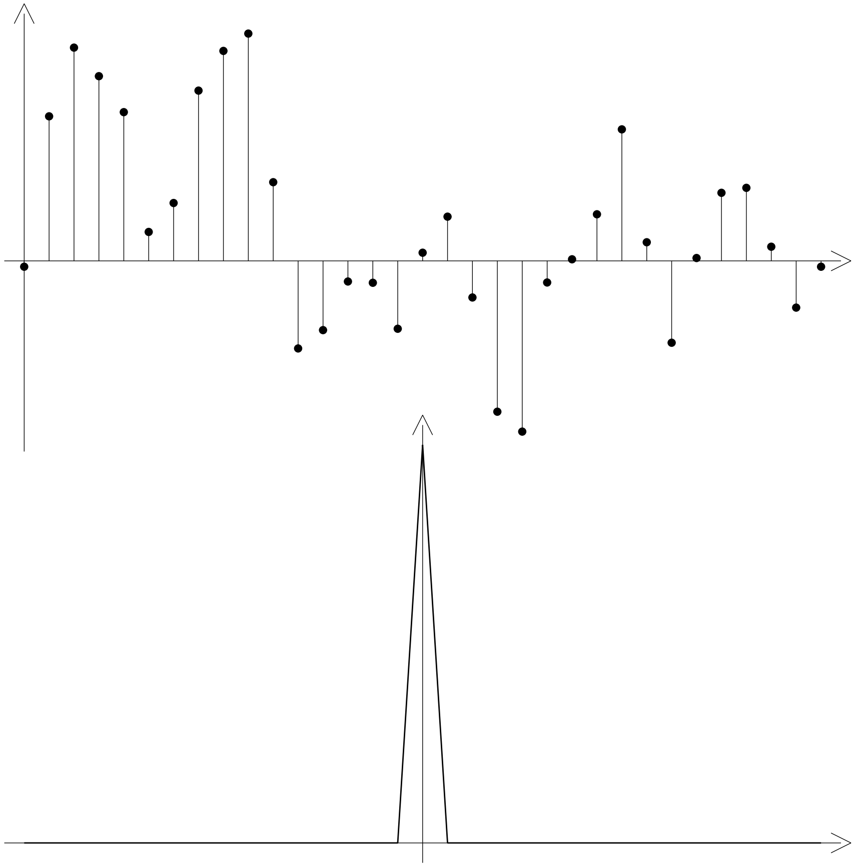
Function Reconstruction

Constant interpolation:

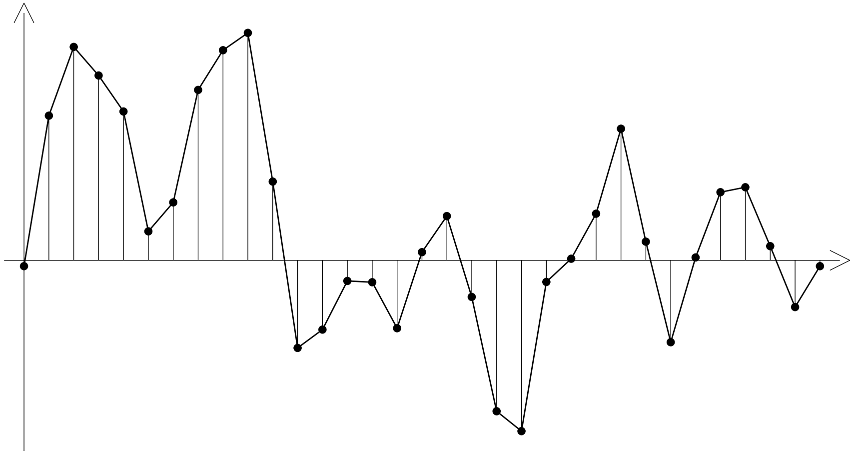


Function Reconstruction

Linear interpolation:

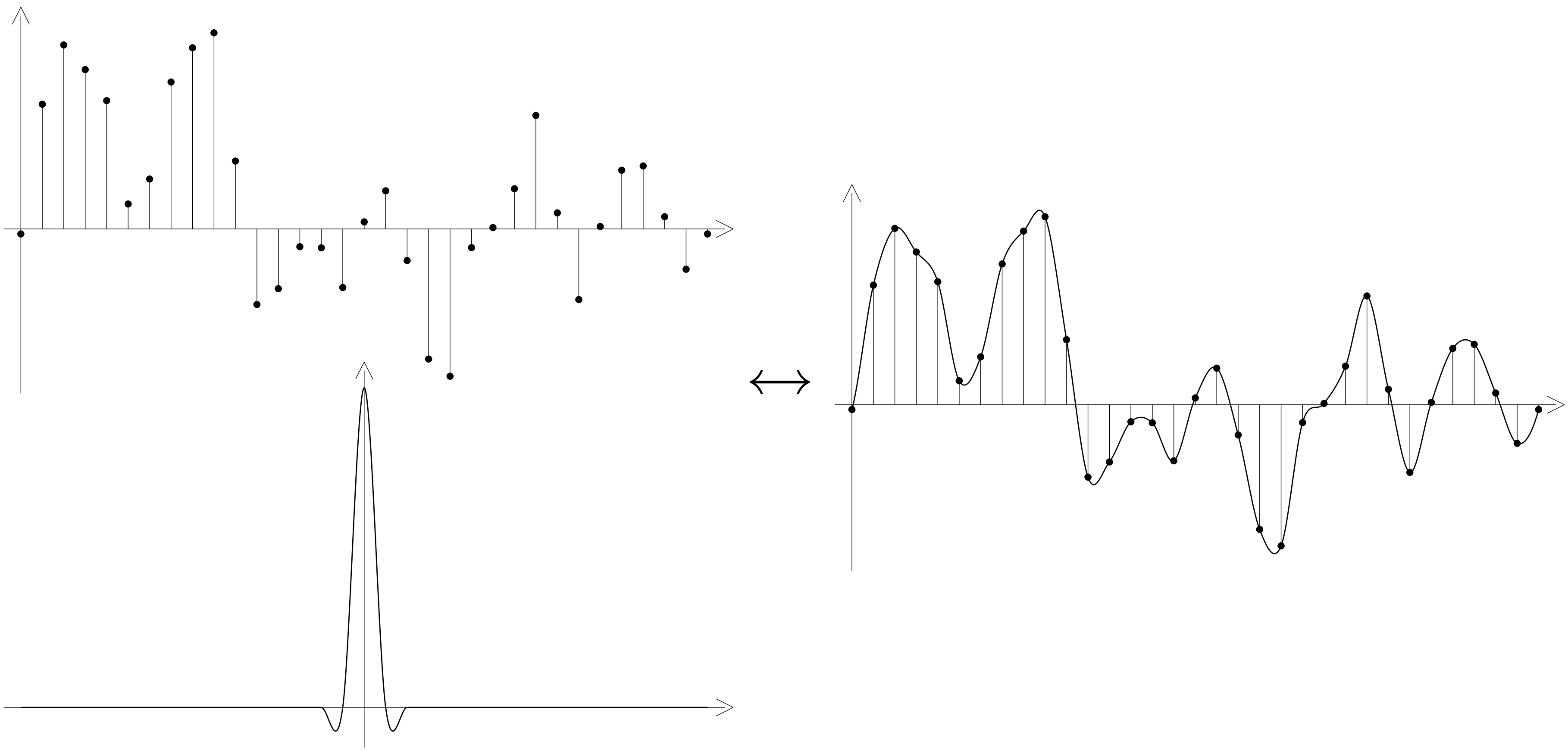


$\longleftrightarrow$



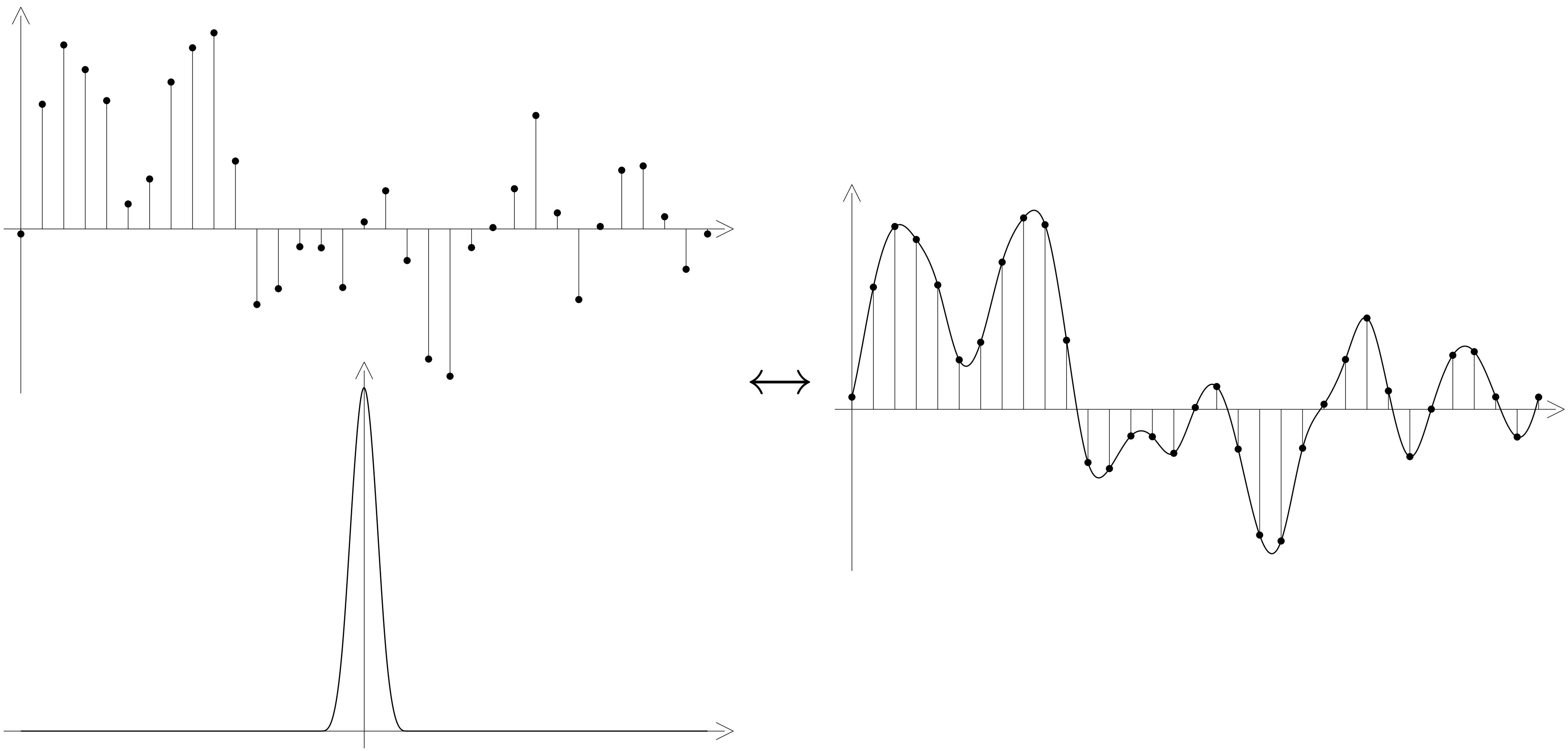
Function Reconstruction

Catmull-Rom interpolation:



Function Reconstruction

Cubic B-spline approximation:



The Convolution Theorem

- The *Convolution Theorem* relates convolution and Fourier transform:

$$(f * g) \;\leftrightarrow\; F \cdot G$$

- Convolutions can be computed by going through the frequency domain:

$$(f * g) = \mathrm{IFT}(\mathrm{FT}(f) \cdot \mathrm{FT}(g))$$

Sampling in Time Domain

- Sampling a function f means multiplying it with a comb function c with *tap distance* d (or sample frequency $\omega = 2\pi/d$):

$$s = f \cdot c$$

- Reconstructing a sampled function means convolving it with a suitable filter kernel

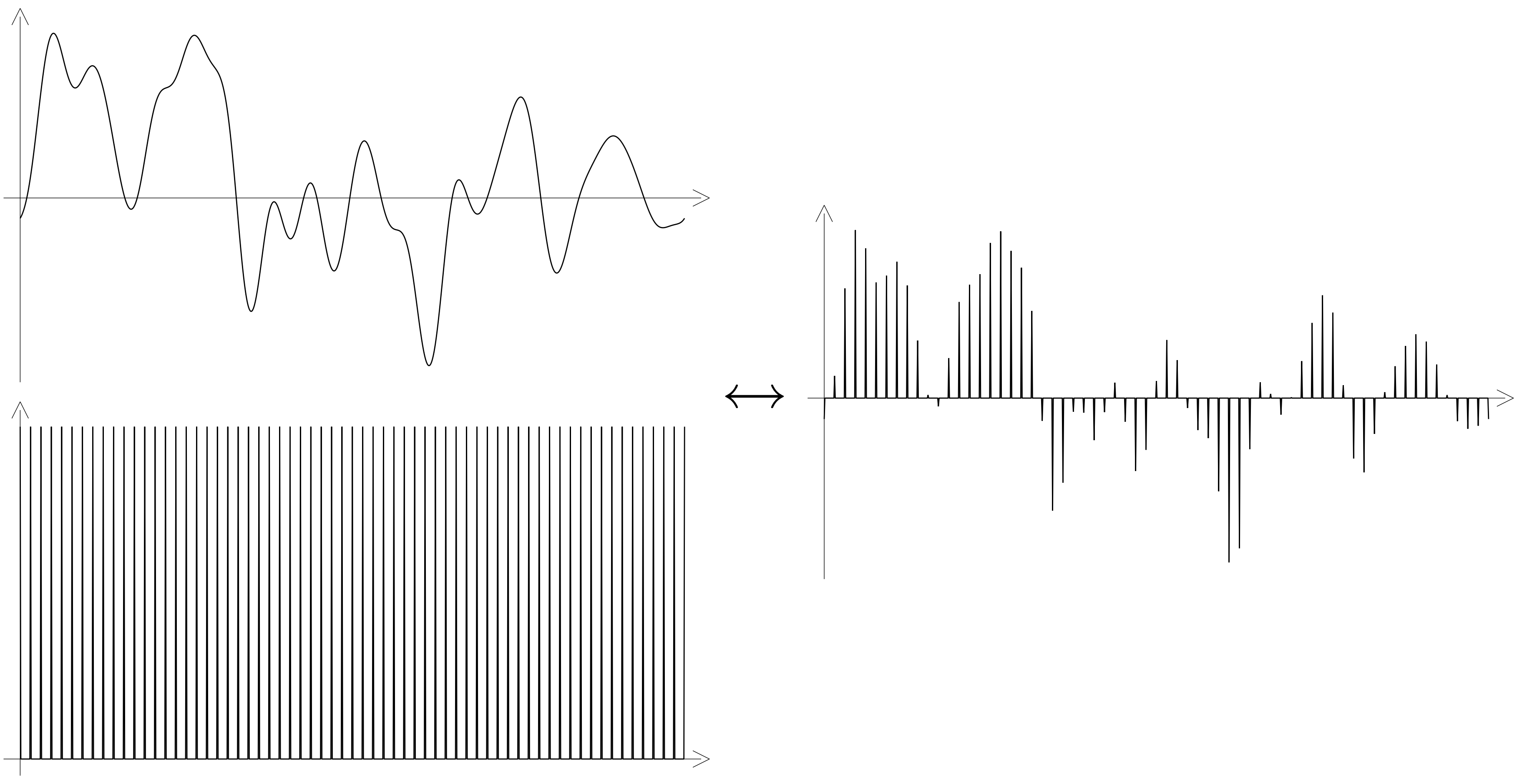
$$f' = s * k$$

- What happens to the function's spectrum in the process?

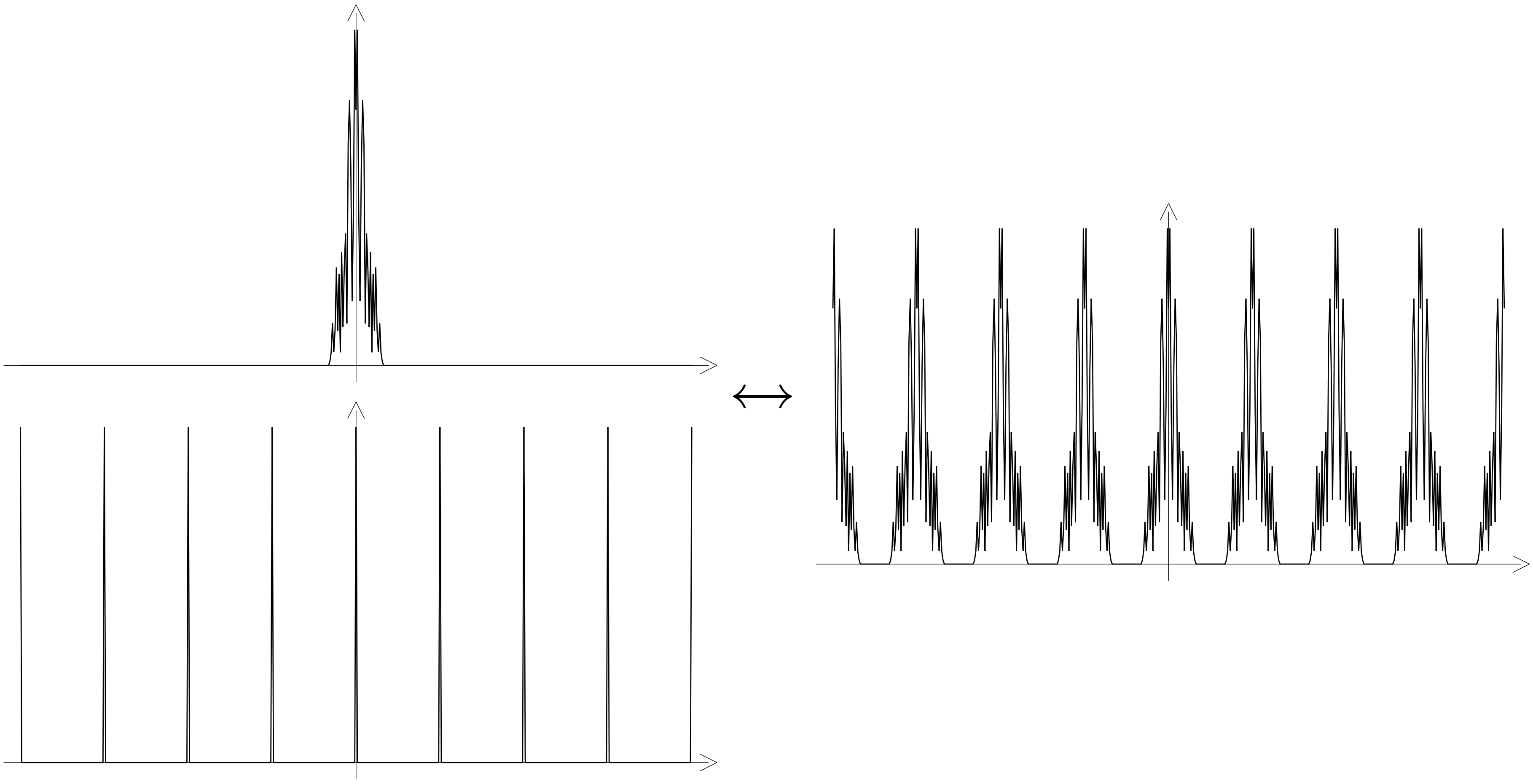
Sampling in Frequency Domain

- The spectrum of s is the convolution of F and C, the spectra of f and c
- C is a comb function with tap distance $2\pi/d$
- S consists of shifted copies of F, each $2\pi/d$ apart

Sampling Process - Time Domain



Sampling Process - Frequency Domain



Sampling in Frequency Domain

- If the sampling frequency is ω, the replicated copies of f's spectrum are ω apart
- If the highest-frequency component in f is ω_f, f's spectrum covers the interval $[-\omega_f, \omega_f]$
- If $\omega_f \geq \omega/2$, then the replicated spectra overlap!
- This means, information is lost

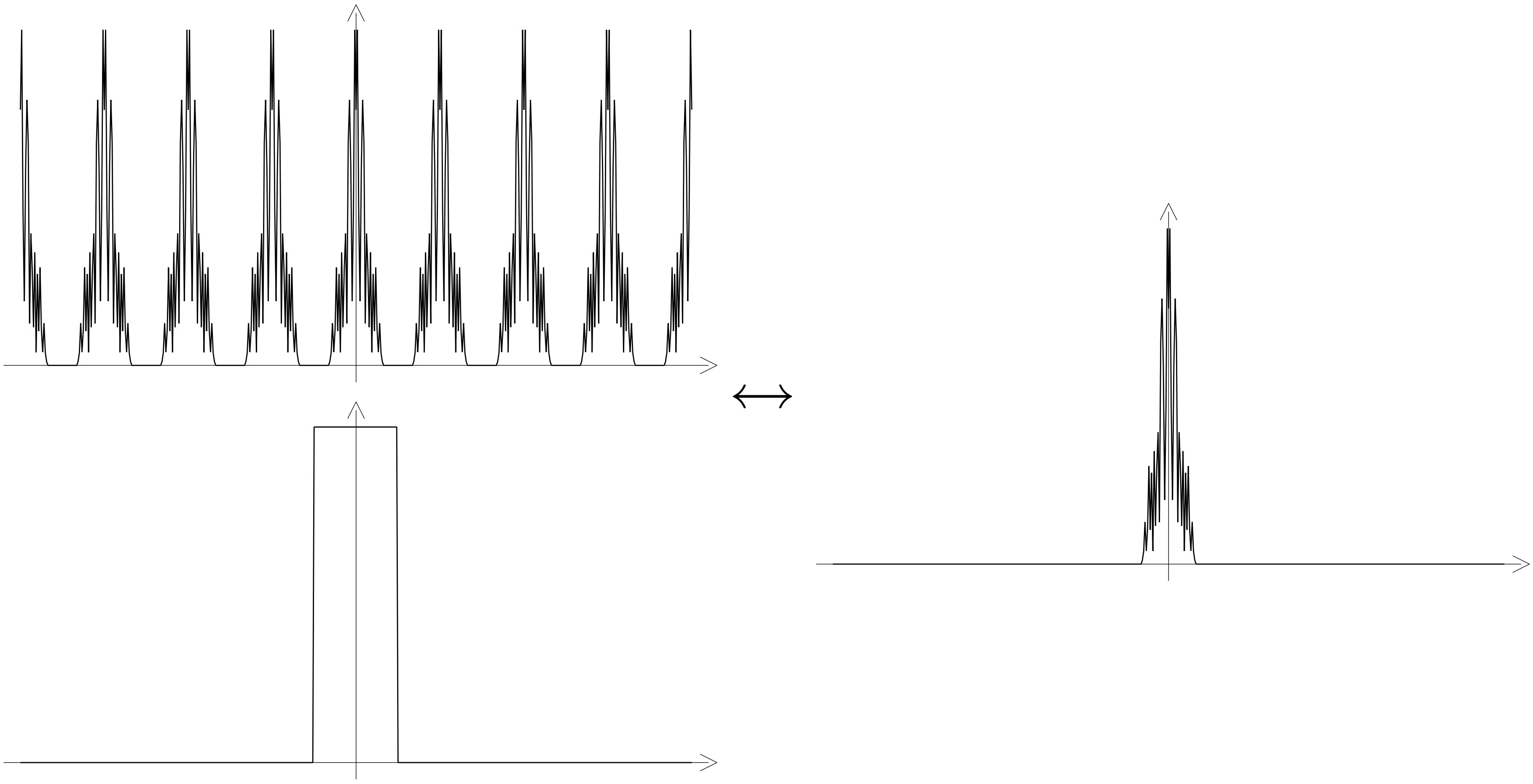
The Sampling Theorem

- "If f is a frequency-limited function with maximum frequency ω_f, then f must be sampled with a sampling frequency larger than $2\omega_f$ in order to be able to exactly reconstruct f from its samples."
- This theorem is sometimes called *Shannon's Theorem*
- $2\omega_f$ is sometimes called *Nyquist rate*

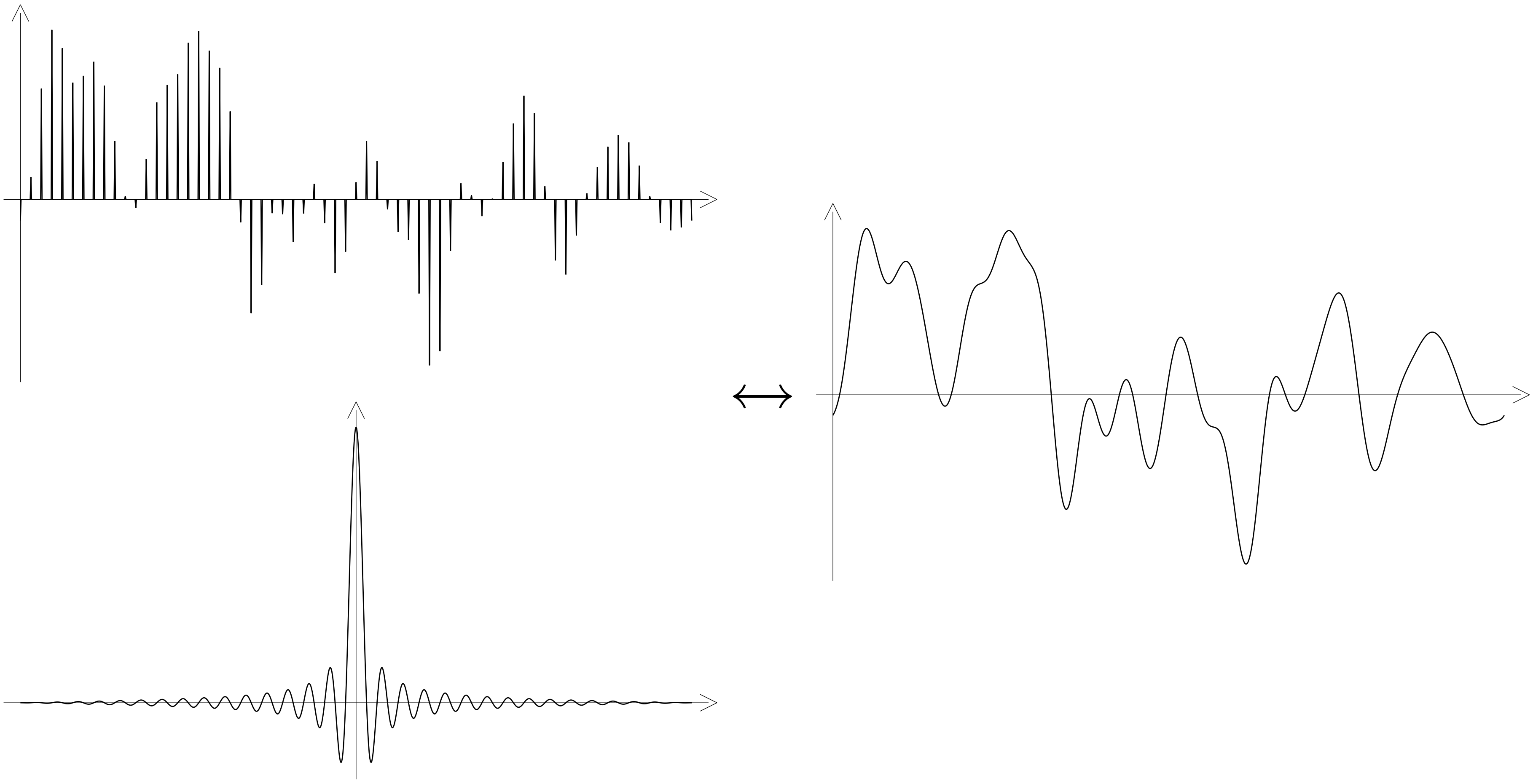
Reconstruction

- The only difference between F and S is that S is made of shifted copies of F
- If those extra, higher-frequency, spectra could be removed from S, f would “magically” reappear
- Reconstructing f means low-pass-filtering s!

Reconstruction Process - Frequency Do



Reconstruction Process - Time Domain



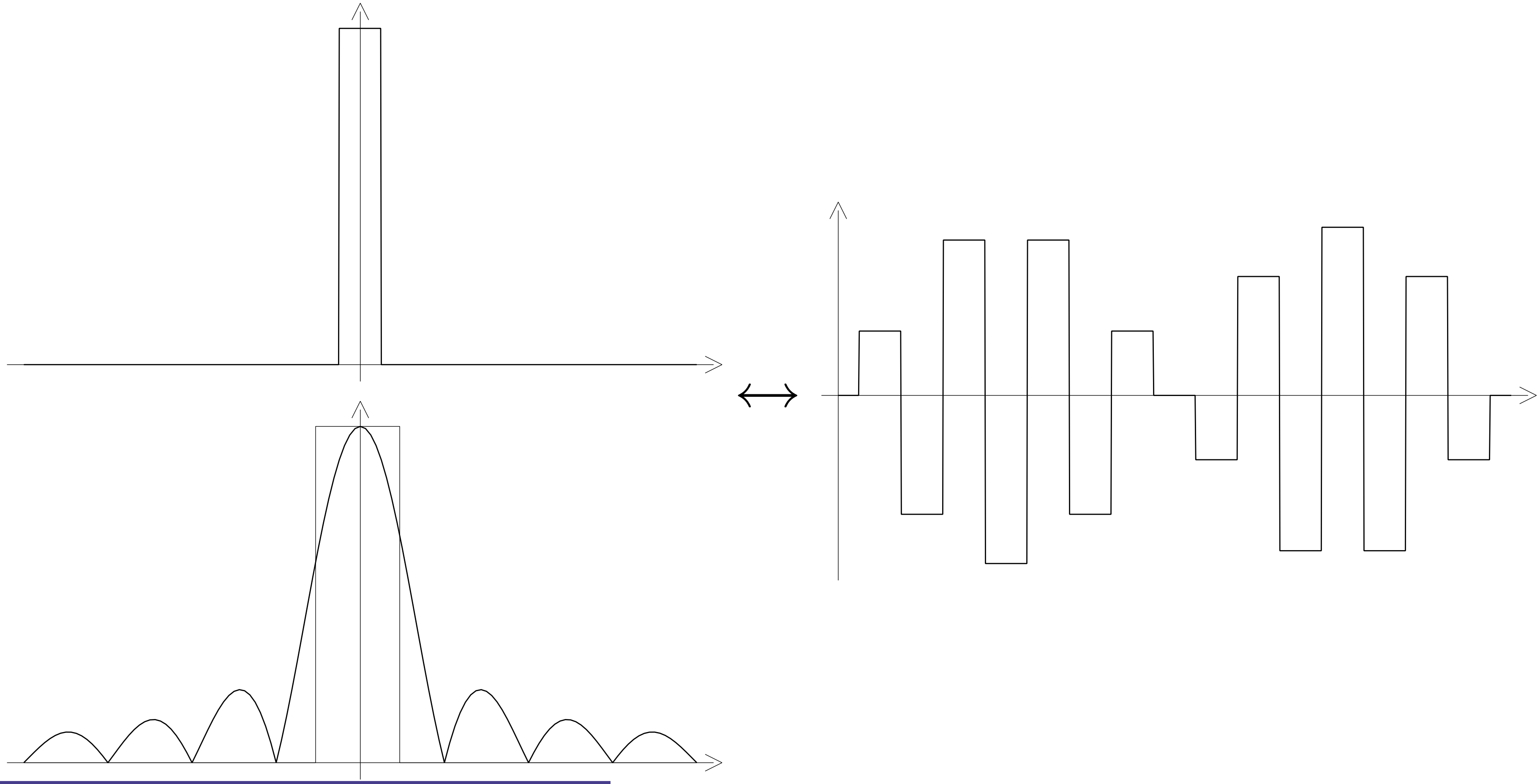
Reconstruction Problem

- Practical problem: The optimal reconstruction filter, the sinc function, has infinite support
- Evaluating it in the time domain involves computing a weighted average of infinitely many samples
- Practical reconstruction filters must have finite support
- But: A perfect low-pass filter with finite support does not exist!

The Quest For Better Filters

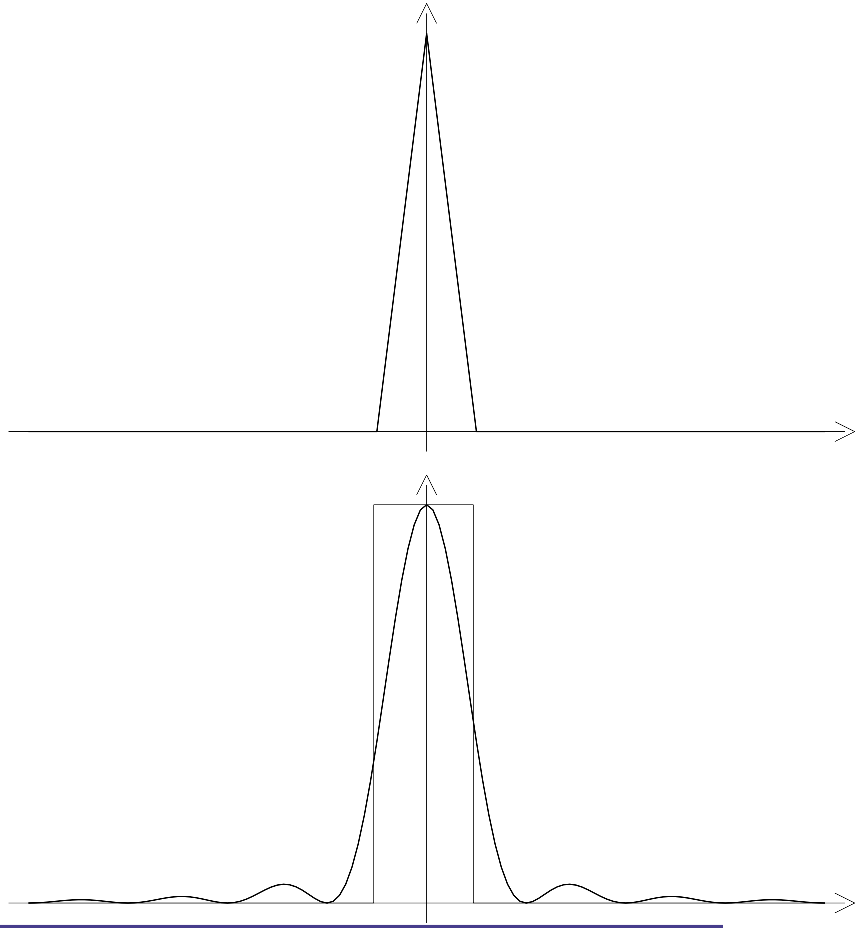
Commonly Used Filters

Constant filter:

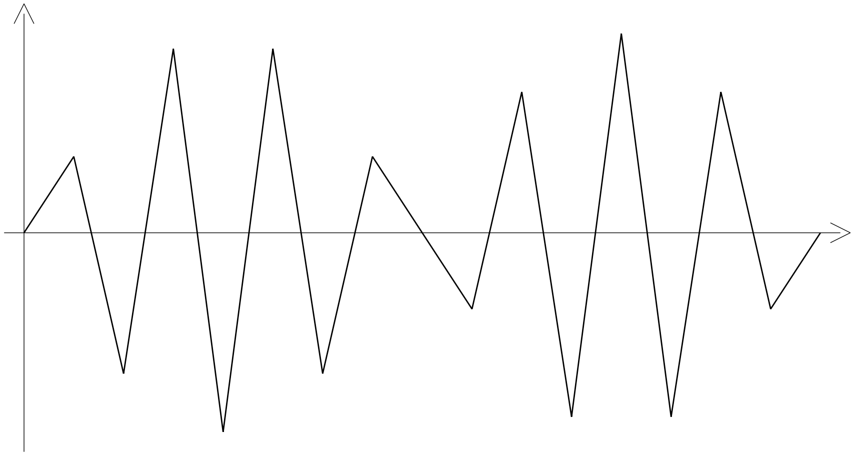


Commonly Used Filters

Linear filter:

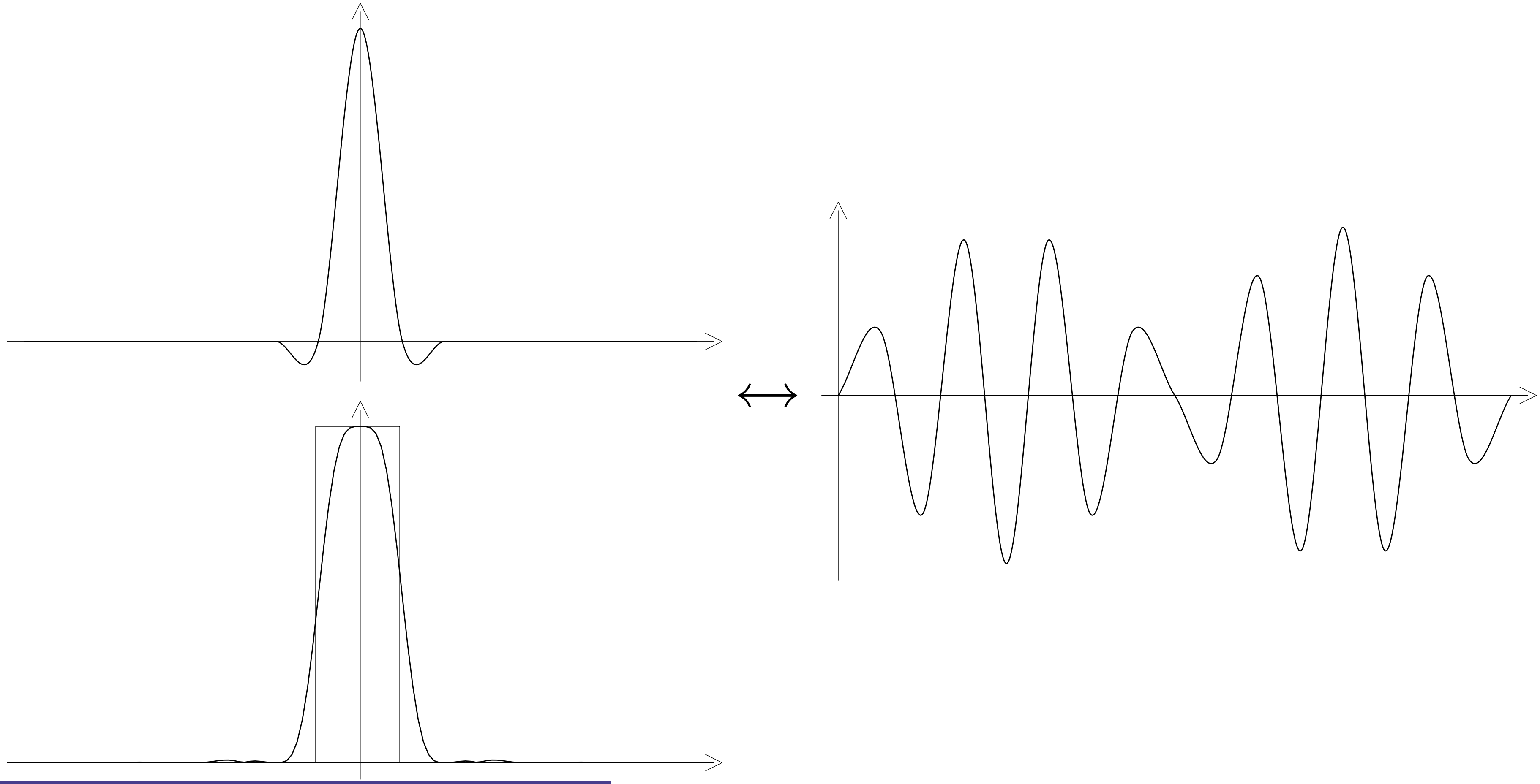


$\leftrightarrow$



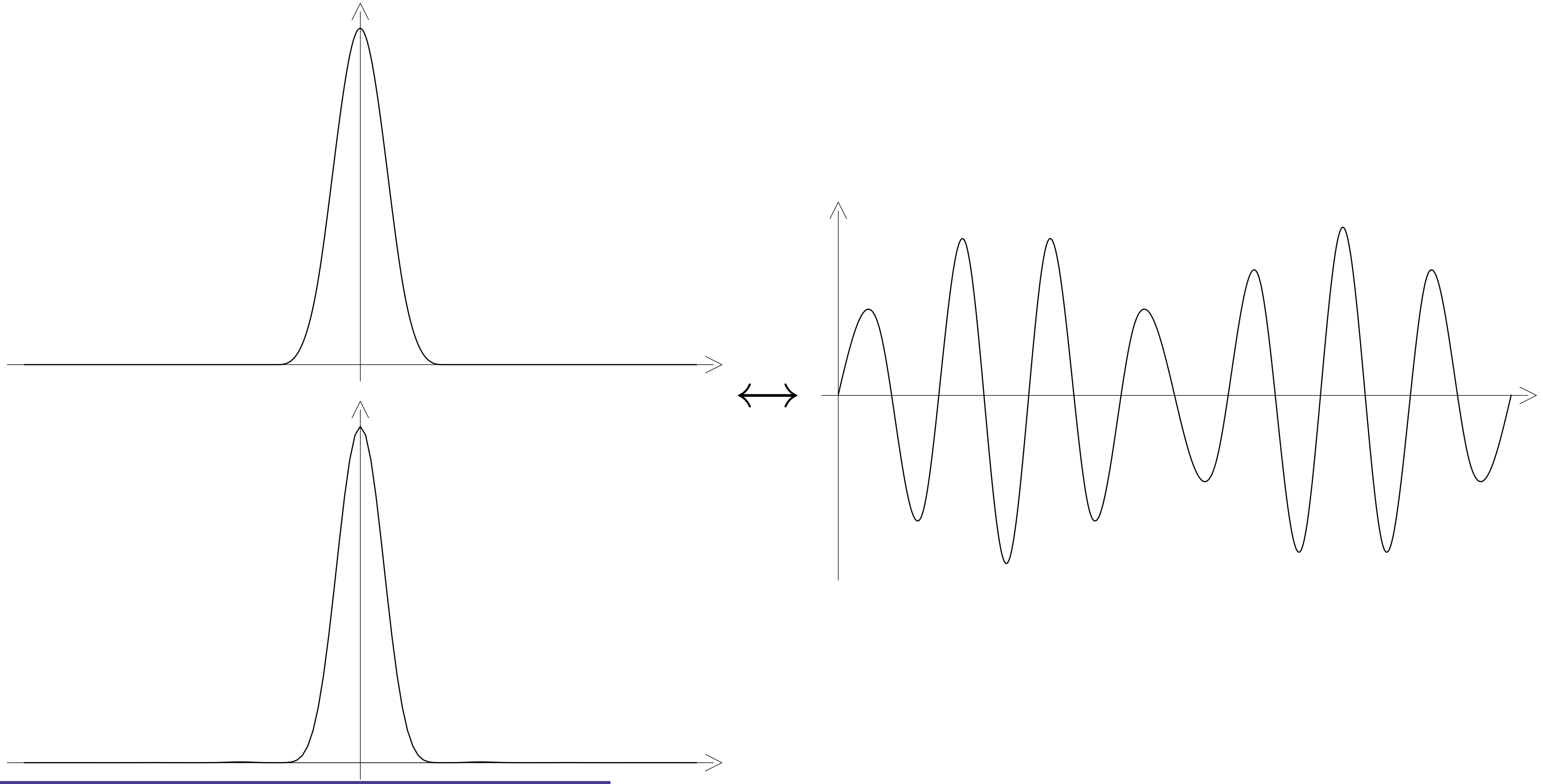
Commonly Used Filters

Catmull-Rom filter:



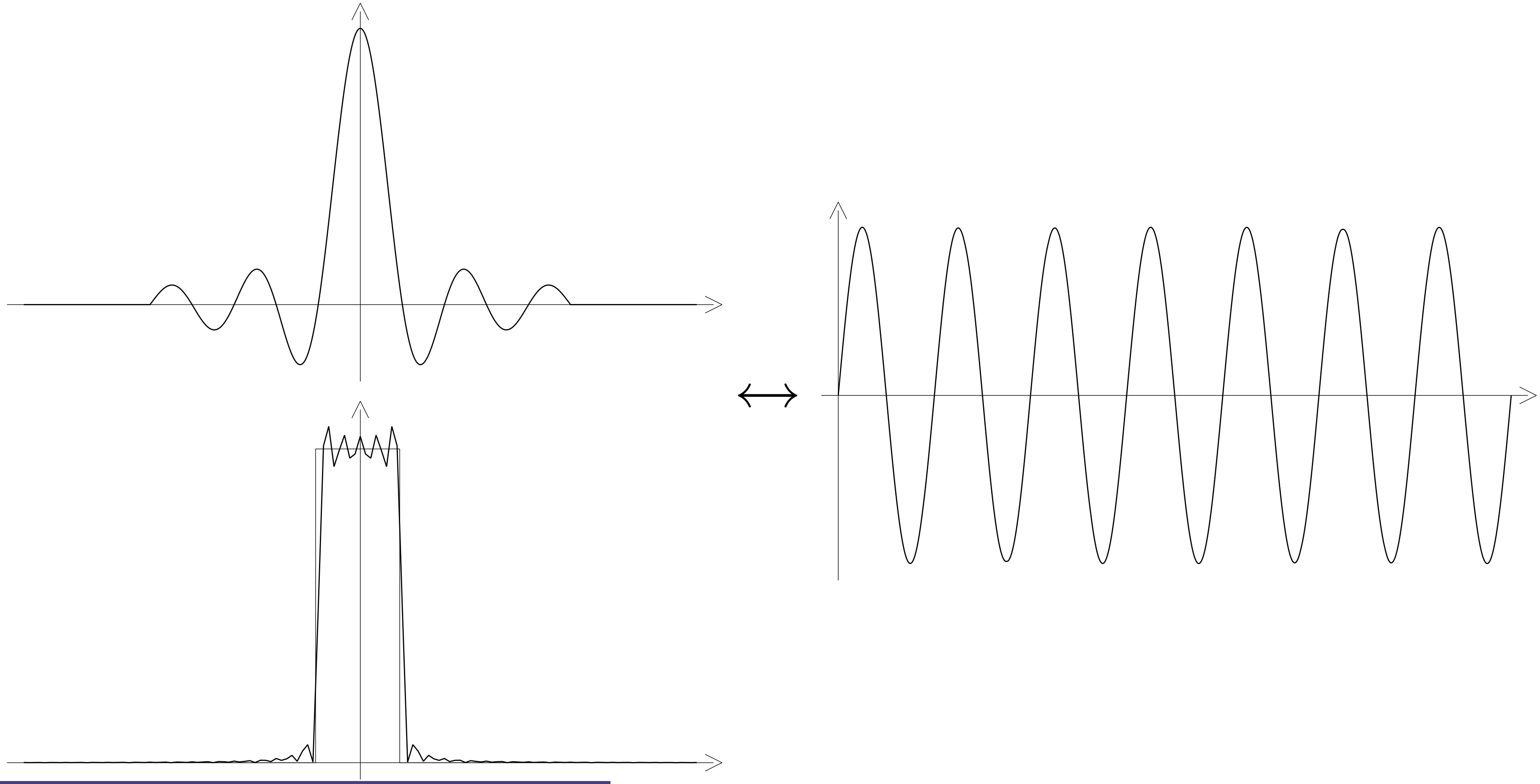
Commonly Used Filters

B-Spline filter:



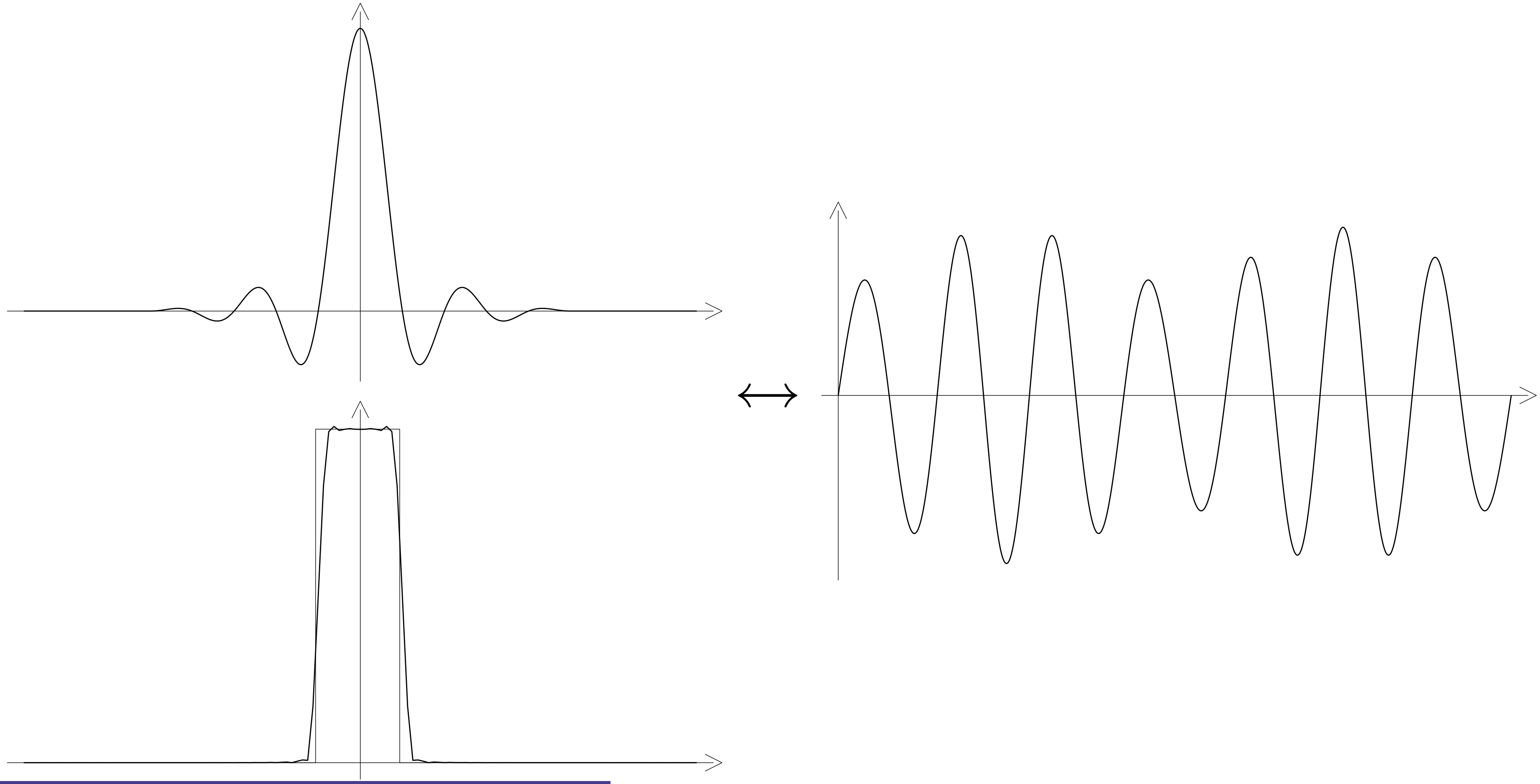
Sampling Theory Filters

Truncated sinc filter:



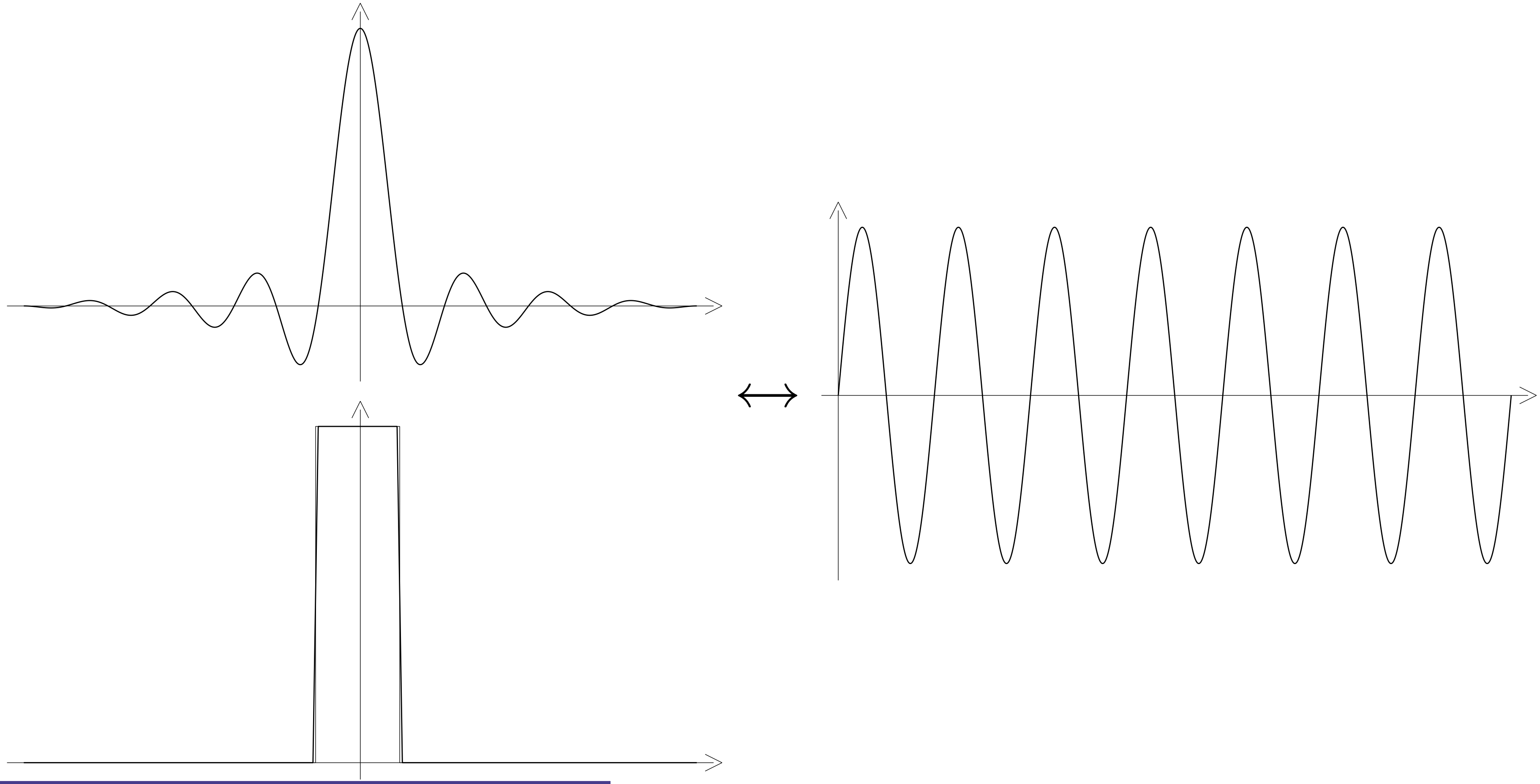
Sampling Theory Filters

Lanczos filter:



Sampling Theory Filters

Full sinc filter:



Sampling Theory Applications

Function Reconstruction

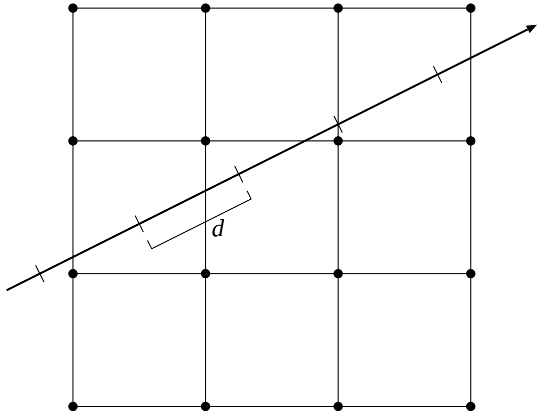
- Sampling theory filters (Lanczos etc.) can be used to reconstruct n-dimensional functions sampled on regular grids
- Oversampling improves reconstruction quality with “bad” filters, but at a high cost

Function Resampling

- When resampling a sampled function to a different grid (shifted and/or scaled), it is important to use a good reconstruction filter
- When downsampling a function, it is important to use a good low-pass filter to cut off frequencies above half the new sampling frequency

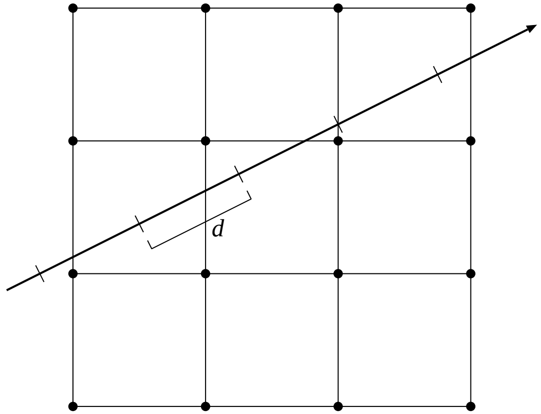
Volume Rendering

- Basic question: How many samples to take per cell?



Volume Rendering

- Basic question: How many samples to take per cell?

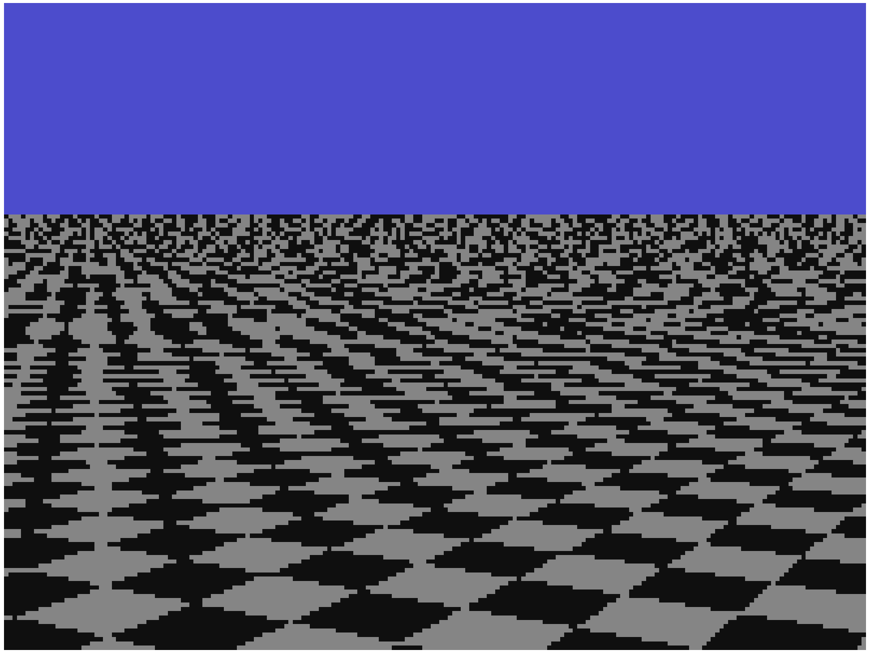


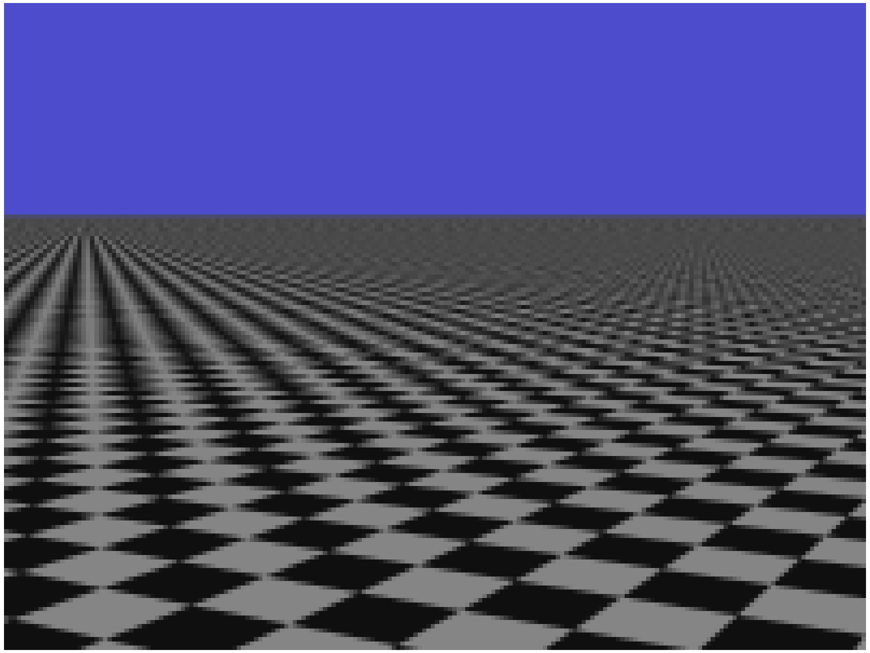
- Sampling theory says "1," but integration along ray might demand more

Ray Tracing

- In ray tracing, the function to be sampled is typically not frequency-limited
- Practical solution: Shoot multiple rays per pixel (oversampling) and sample down to image resolution
- Using a good low-pass filter is important

Ray Tracing





Conclusions

Conclusions

- Knowing the mathematical background of sampling and reconstruction helps to understand common problems and devise solutions
- Ad-hoc solutions usually do not work well
- **This stuff should be taught in class!**

The End