Chapter 2: Fourier Analysis and Signal Processing

Prof. Dr.-Ing. Thomas Schultz

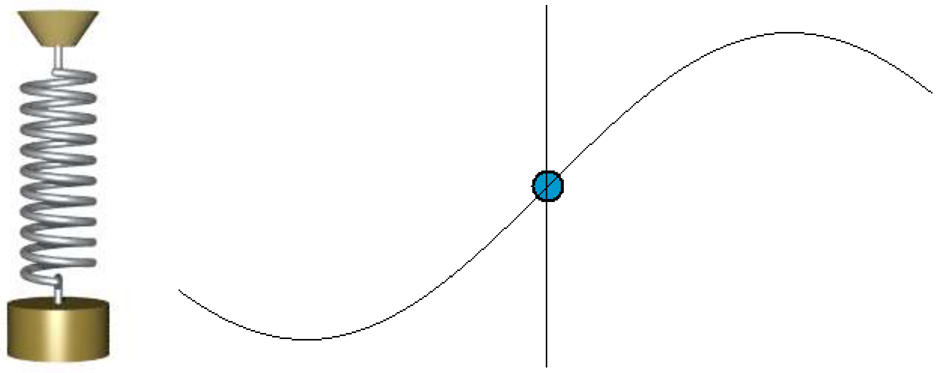
URL: http://cg.cs.uni-bonn.de/schultz/
E-Mail: schultz@cs.uni-bonn.de
Office: Friedrich-Hirzebruch-Allee 6, Room 2.117

October 16/20, 2025

2.1 Basics of Fourier Analysis

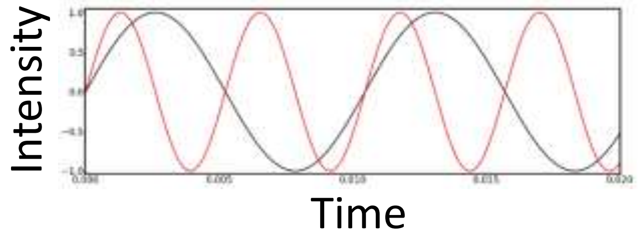
Periodic Functions, Idea of Time vs Frequency Domain

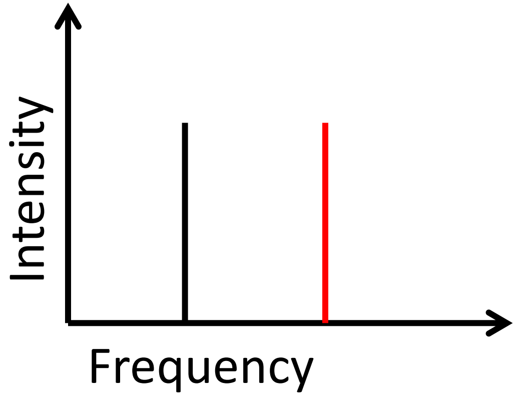
- Consider **periodic functions** $f(x + p) = f(x)$ with period p
  - **Fundamental period:** Smallest positive p
- Some of the most basic **periodic systems** (e.g., harmonic oscillator) are described by (co)sines



- In acoustic waves, (co)sines correspond to pure tones, frequencies to pitch

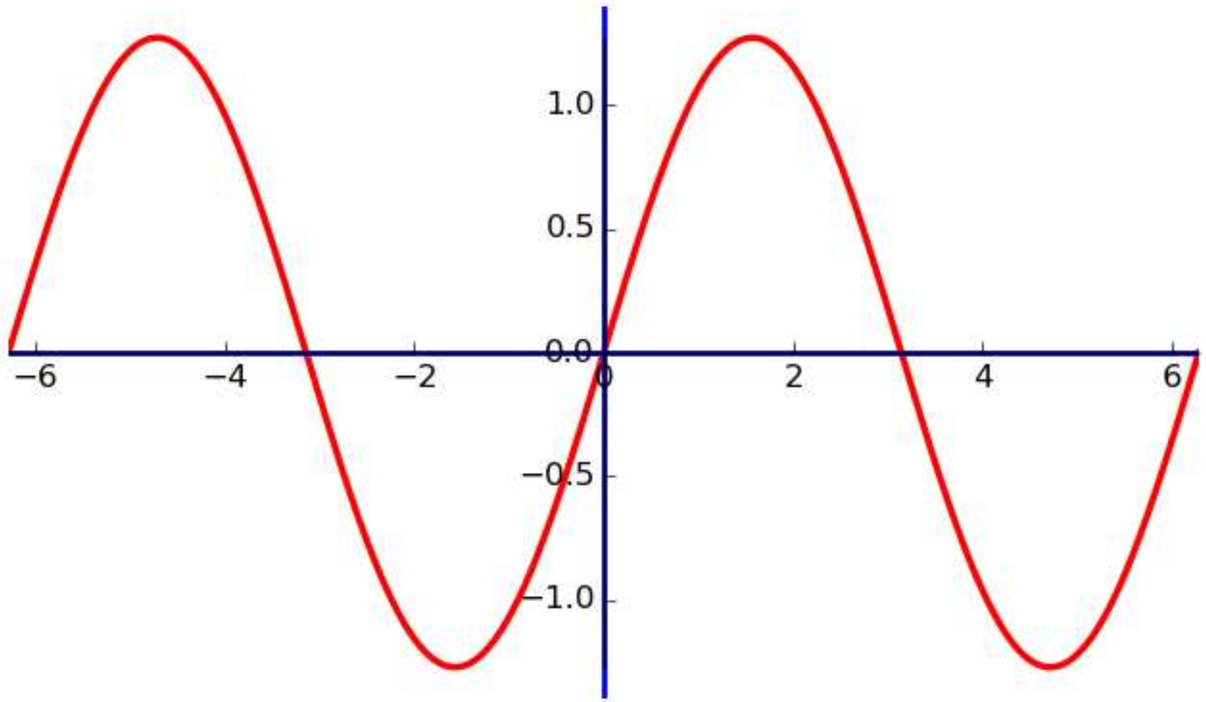
600 Hz 1200 Hz





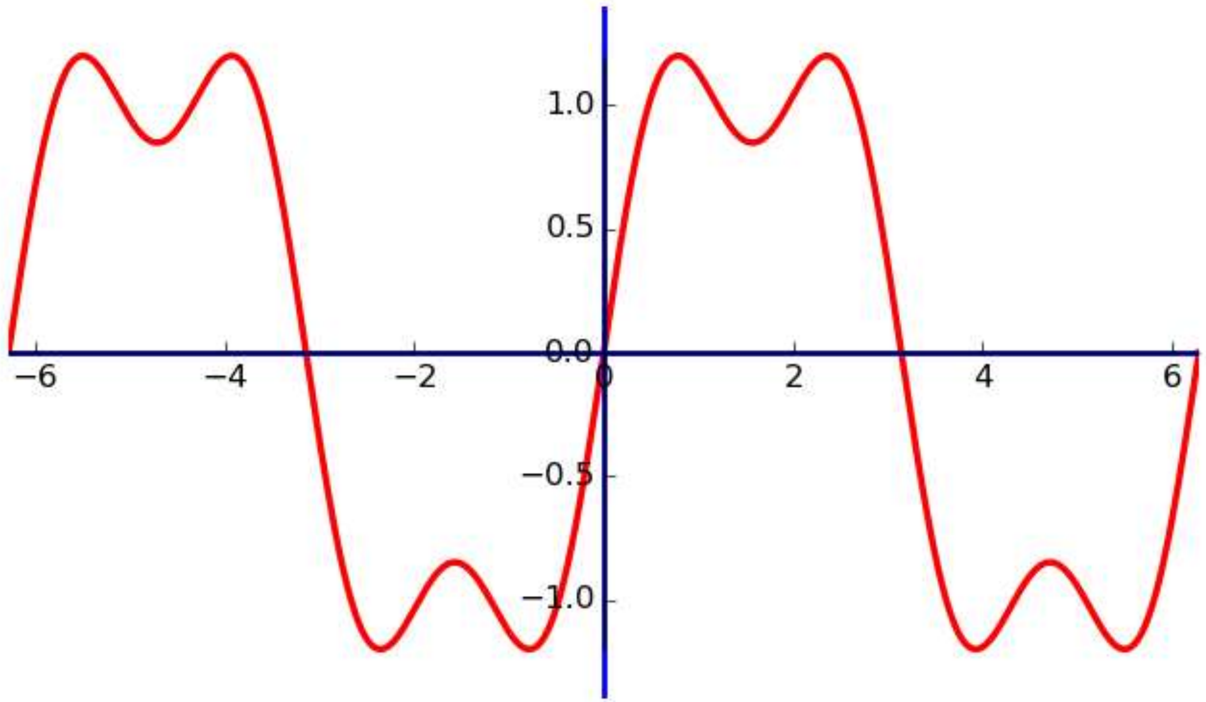
Building a Square Wave from Sines

$$f(x) = \frac{4}{\pi}\sin x$$



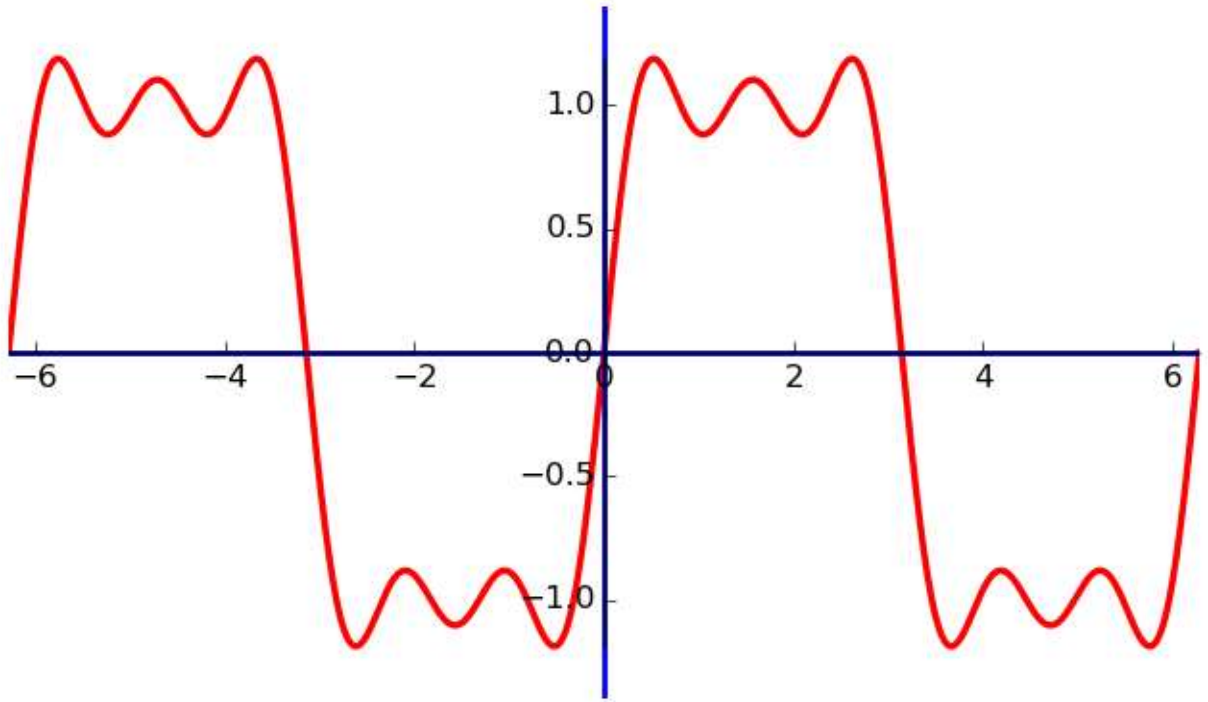
Building a Square Wave from Sines

$$f(x) = \frac{4}{\pi}\left(\sin x + \frac{1}{3}\sin 3x\right)$$



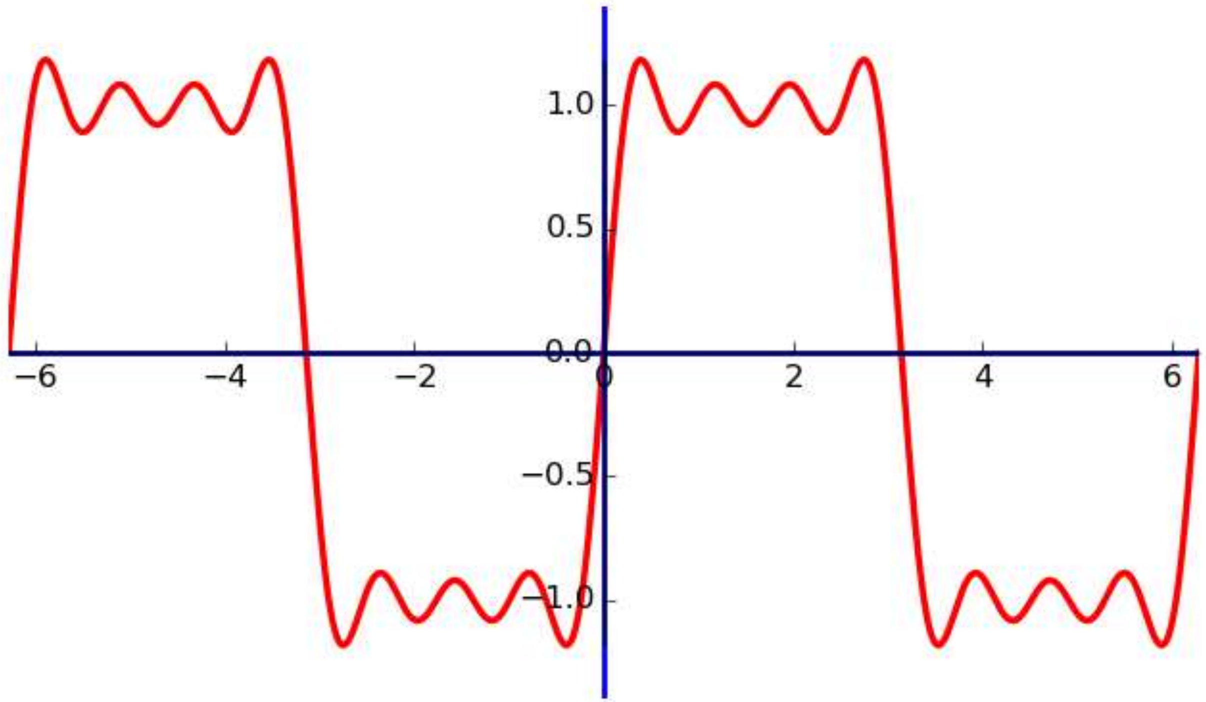
Building a Square Wave from Sines

$$f(x) = \frac{4}{\pi}\left(\sin x + \frac{1}{3}\sin 3x + \frac{1}{5}\sin 5x\right)$$



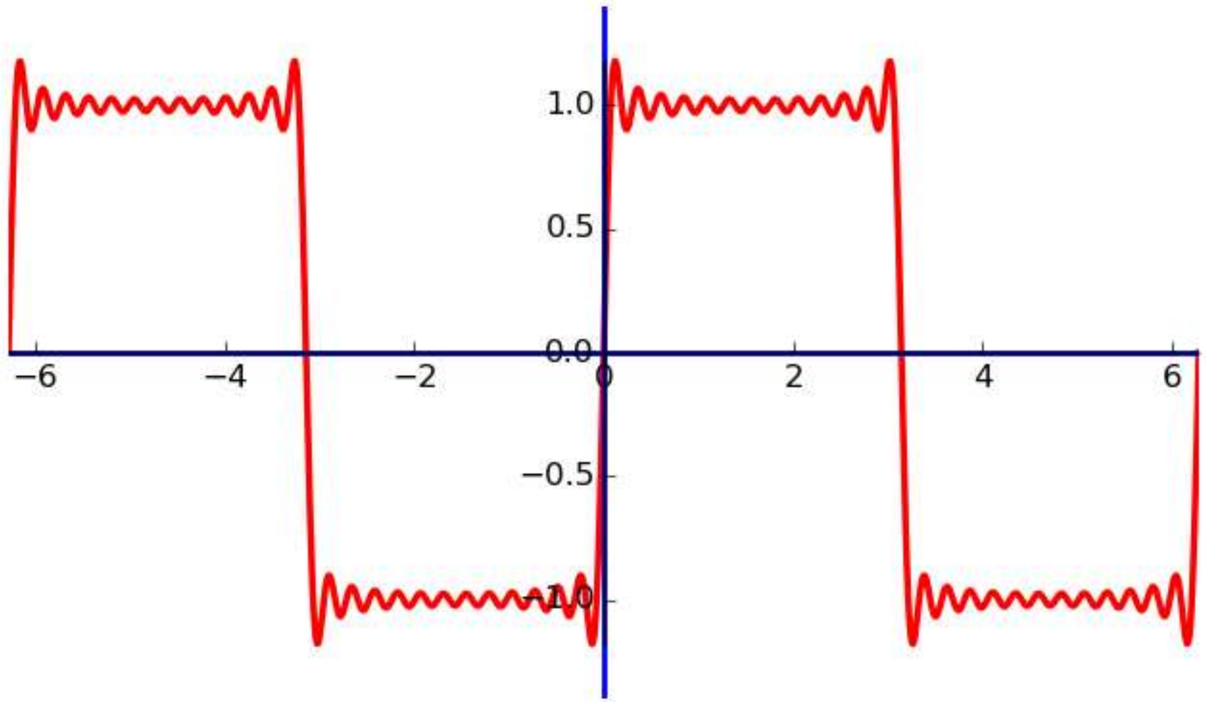
Building a Square Wave from Sines

$$f(x) = \frac{4}{\pi}\left(\sin x + \frac{1}{3}\sin 3x + \frac{1}{5}\sin 5x + \frac{1}{7}\sin 7x\right)$$



Building a Square Wave from Sines

$$f(x) = \frac{4}{\pi}\left(\sin(x) + \frac{1}{3}\sin(3x) + \ \cdots + \frac{1}{25}\sin(25x)\right)$$

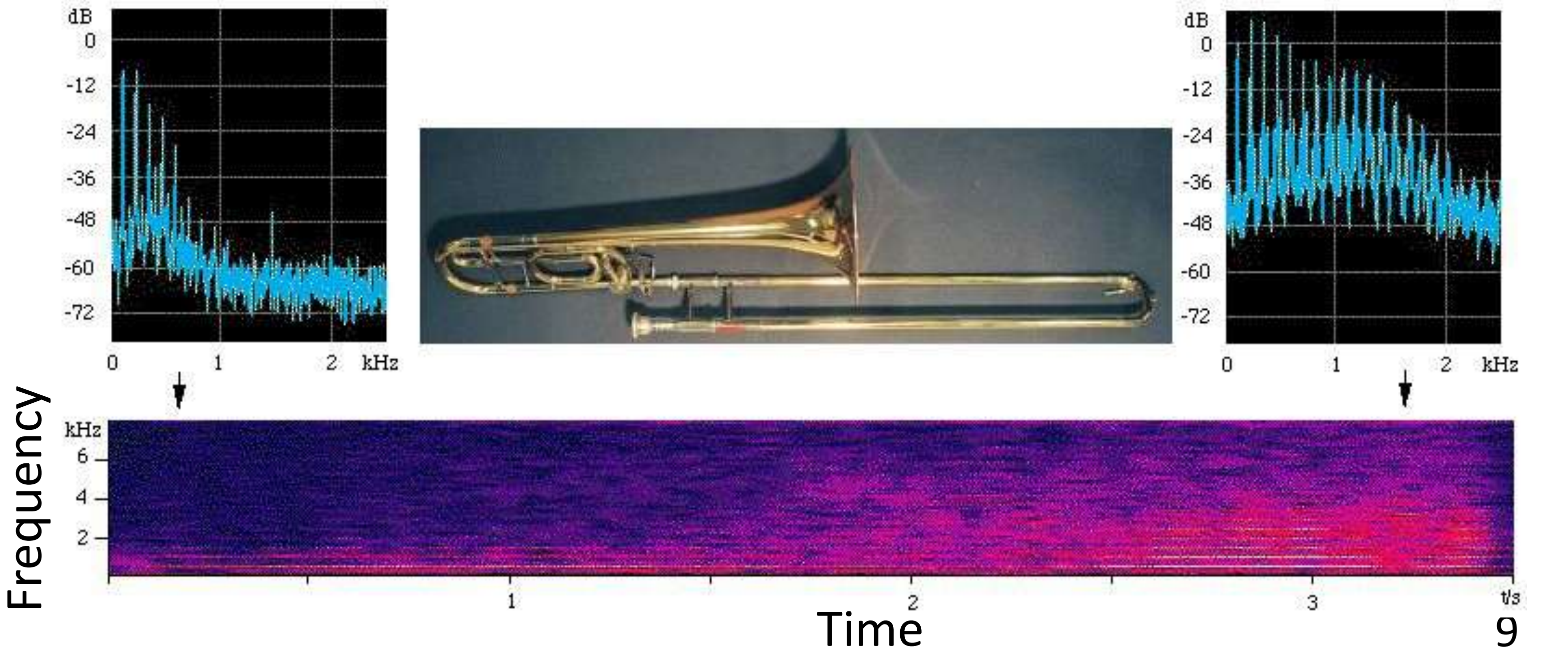


Intuition: Additive Synthesis and Fourier Analysis

Hammond Organ is based on **additive synthesis**

- (Almost) any periodic function can be approximated by adding sines and cosines of different frequencies
- Fourier Series, inverse Fourier Transform

The (forward) **Fourier Transform** allows us to analyze a periodic function in terms of its frequency contents



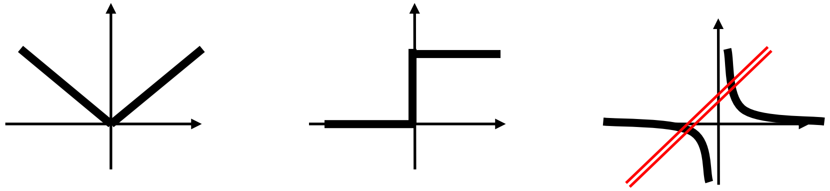
Examples from Nathaniel C. Wilcox (via Wikipedia, CC) and Joe Wolfe

Fourier Theorem for Periodic Functions

- **Theorem:** For a 2π periodic, piecewise monotone, bounded real function $f(x)$, values at points of continuity x equal those of its Fourier Series

$$f(x) = a_0 + \sum_{n=1}^{\infty} (a_n \cos(nx) + b_n \sin(nx))$$

Examples:



Joseph Fourier

- Alternative notation, considering that cos(0)=1 and sin(0)=0:

$$f(x) = \sum_{n=0}^{\infty} (a_n \cos(nx) + b_n \sin(nx))$$

Scalar Product on 2π-periodic Functions

- **Definition:** We define a scalar product on the space H of all 2π periodic real functions that fulfill the conditions above as

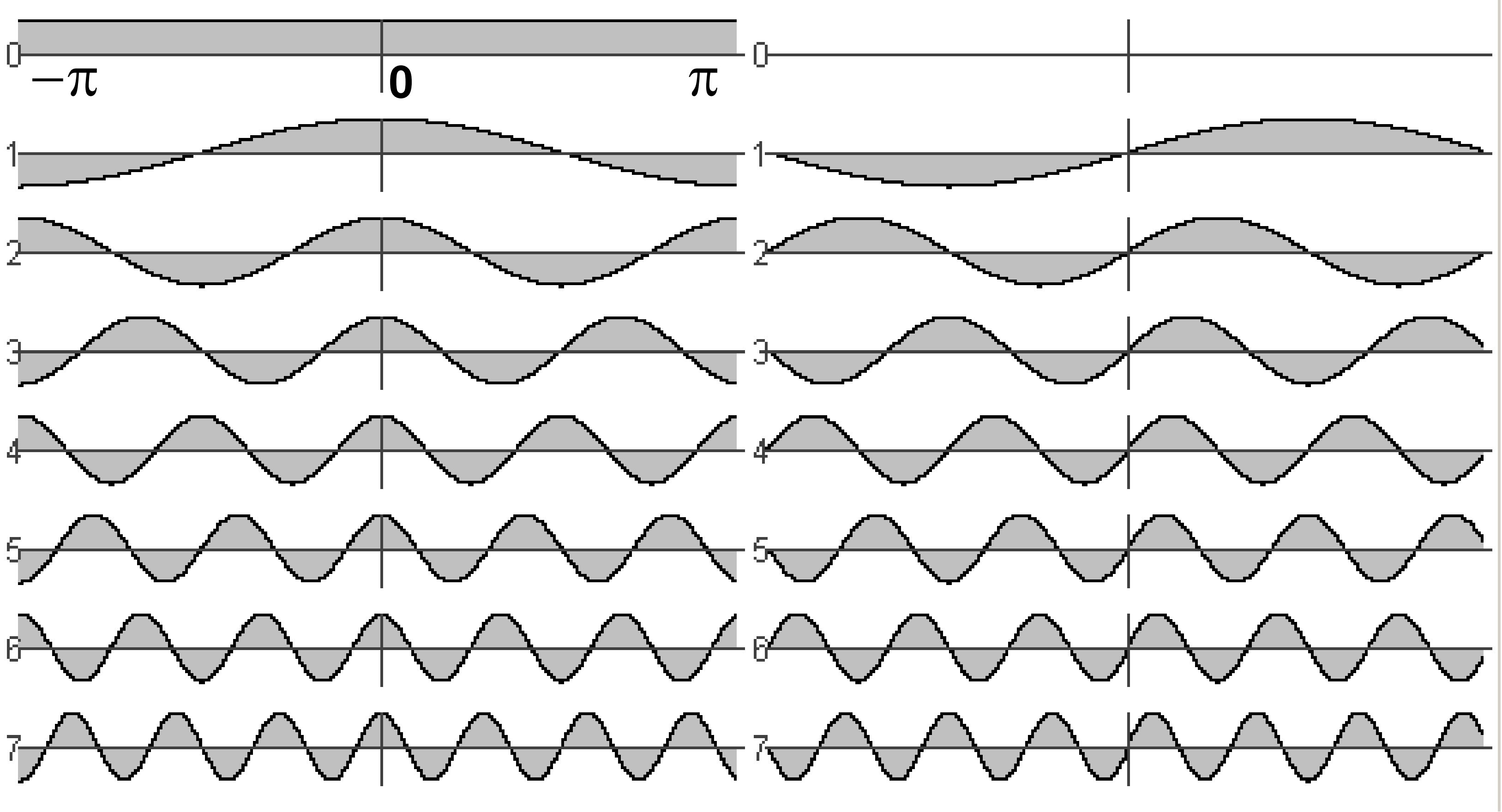
$$\langle f, g \rangle := \int_{-\pi}^{\pi} f(x)g(x)\,dx$$

- **Observation:** The functions

$$u_n(t) := \cos(nt)\,, n = 0,1,2,\ldots$$
$$v_n(t) := \sin(nt)\,, n = 0,1,2,\ldots$$

define orthogonal function sequences in *H.*

Sine und Cosine Functions

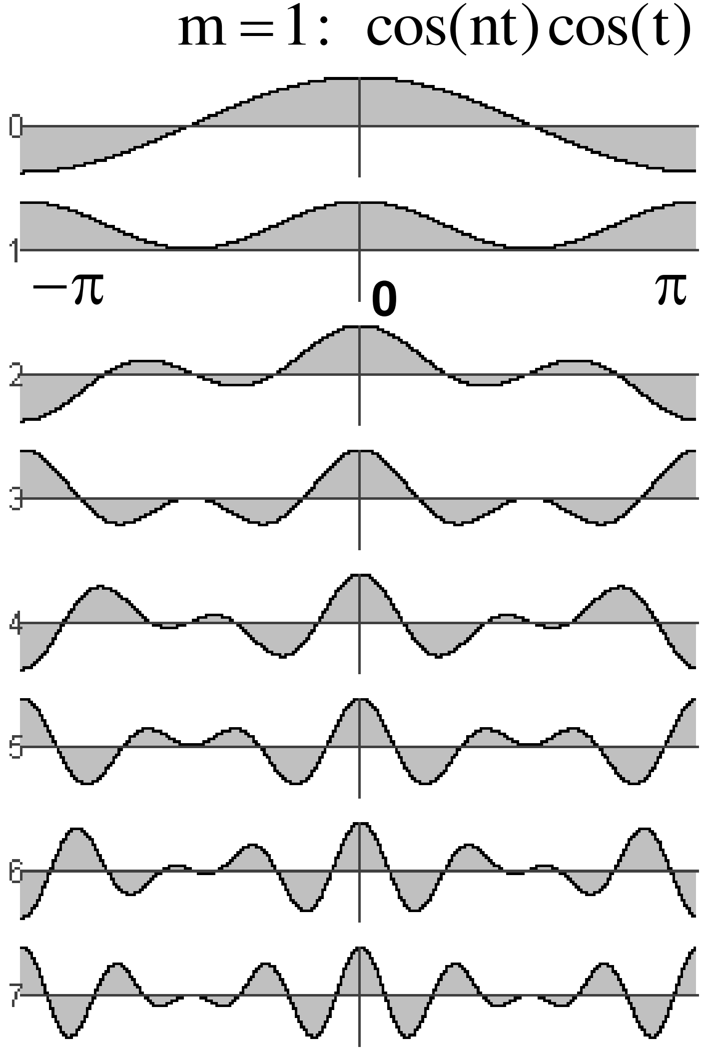


$u_n(t) := \cos(nt)\,, n = 0,1,2,\ldots$ $v_n(t) := \sin(nt)\,, n = 0,1,2,\ldots$

Illustration: Orthogonality of Cosine Functions

$$\langle u_n, u_m \rangle = \int_{-\pi}^{\pi} \cos(nt)\cos(mt)\,dt$$

$$= \begin{cases} 0 & \text{if } m \neq n \\ \pi & \text{if } m = n = 1,2,\ldots \\ 2\pi & \text{if } m = n = 0 \end{cases}$$



- Proof is based on product-to-sum identities

$$\cos\theta\cos\phi = \frac{\cos(\theta-\phi)+\cos(\theta+\phi)}{2}$$

Orthogonality of Sine and Cosine Functions

Similarly,

$$\langle v_n, v_m \rangle = \int_{-\pi}^{\pi} \sin(nt)\sin(mt)\,dt = \begin{cases} 0 & \text{if } m \neq n \\ \pi & \text{if } m = n = 1,2,\ldots \\ 0 & \text{if } m = n = 0 \end{cases}$$

$$\langle u_n, v_m \rangle = \int_{-\pi}^{\pi} \cos(nt)\sin(mt)\,dt = 0 \quad \forall n, m = 0,1,2,\ldots$$

Computing the coefficient *a₀*

- The coefficient a_0 can be computed by considering

$$\begin{aligned}\langle f, u_0\rangle &= \left\langle \sum_{n=0}^{\infty} a_n u_n + b_n v_n\,,\ u_0 \right\rangle \\ &= \langle a_0 u_0, u_0\rangle \\ &= a_0 2\pi\end{aligned}$$

- Solving for a_0 yields

$$a_0 = \frac{1}{2\pi}\langle f, u_0\rangle = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)\cos(0x)\,dx = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)dx$$

Computing the coefficients a_m *(m>0)*

- Similarly, for $m = 1,2, \ldots$

$$\begin{aligned}\langle f, u_m \rangle &= \left\langle \sum_{n=0}^{\infty} a_n u_n + b_n v_n \, , \, u_m \right\rangle \\ &= \langle a_m u_m, u_m \rangle \\ &= a_m \pi\end{aligned}$$

- From this:

$$a_m = \frac{1}{\pi} \langle f, u_m \rangle = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(mx) \, dx$$

Computing the coefficients b_m

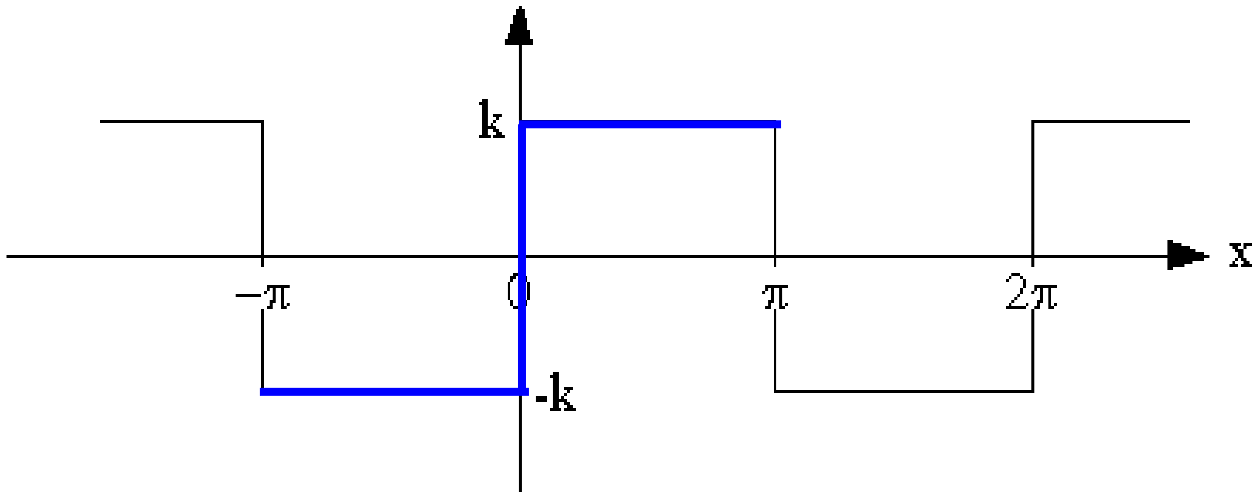
- In complete analogy, for $m = 1,2, \ldots$

$$
\begin{aligned}
\langle f, v_m \rangle &= \left\langle \sum_{n=0}^{\infty} a_n u_n + b_n v_n \, , v_m \right\rangle \\
&= \langle b_m v_m, v_m \rangle \\
&= b_m \pi
\end{aligned}
$$

- From this:

$$b_m = \frac{1}{\pi} \langle f, v_m \rangle = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin(mx) \, dx$$

Example: Periodic Rectangle Function



$$f(x) = \begin{cases} -k & \text{if} \quad -\pi < x < 0 \\ k & \text{if} \quad 0 < x < \pi \end{cases} \quad \text{and} \quad f(x) = f(x + 2\pi)$$

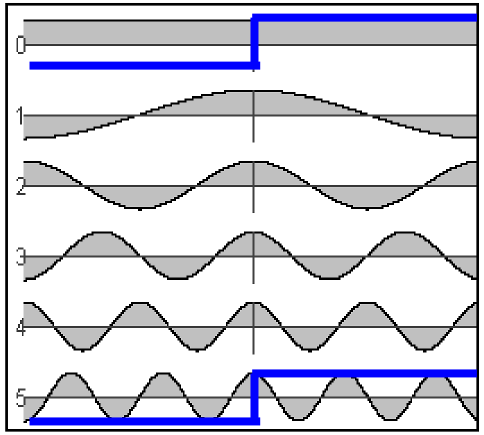
Example: Periodic Rectangle Function

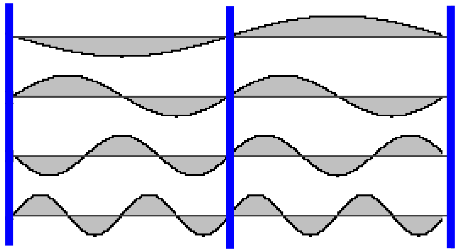
$a_0 = 0$ trivial

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(nx)\, dx$$

$$= \frac{1}{\pi} \left[\int_{-\pi}^{0} (-k) \cos(nx)\, dx + \int_{0}^{\pi} k \cos(nx)\, dx \right]$$

$$= \frac{1}{\pi} \left[-k \frac{\sin(nx)}{n} \bigg|_{-\pi}^{0} + k \frac{\sin(nx)}{n} \bigg|_{0}^{\pi} \right] = 0$$





since
$\sin(nx) = 0$ for all
$x = \ldots, -\pi, 0, \pi, \ldots$

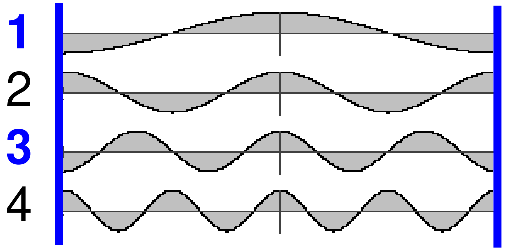
Example: Periodic Rectangle Function

$$b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin(nx)\,dx = \frac{1}{\pi}\left[\int_{-\pi}^{0}(-k)\sin(nx)\,dx + \int_{0}^{\pi} k\sin(nx)\,dx\right]$$

$$= \frac{1}{\pi}\left[k\frac{\cos(nx)}{n}\bigg|_{-\pi}^{0} - k\frac{\cos(nx)}{n}\bigg|_{0}^{\pi}\right]$$

$$= \frac{k}{n\pi}\left(\cos 0 - \cos(-n\pi) - \cos(n\pi) + \cos 0\right)$$

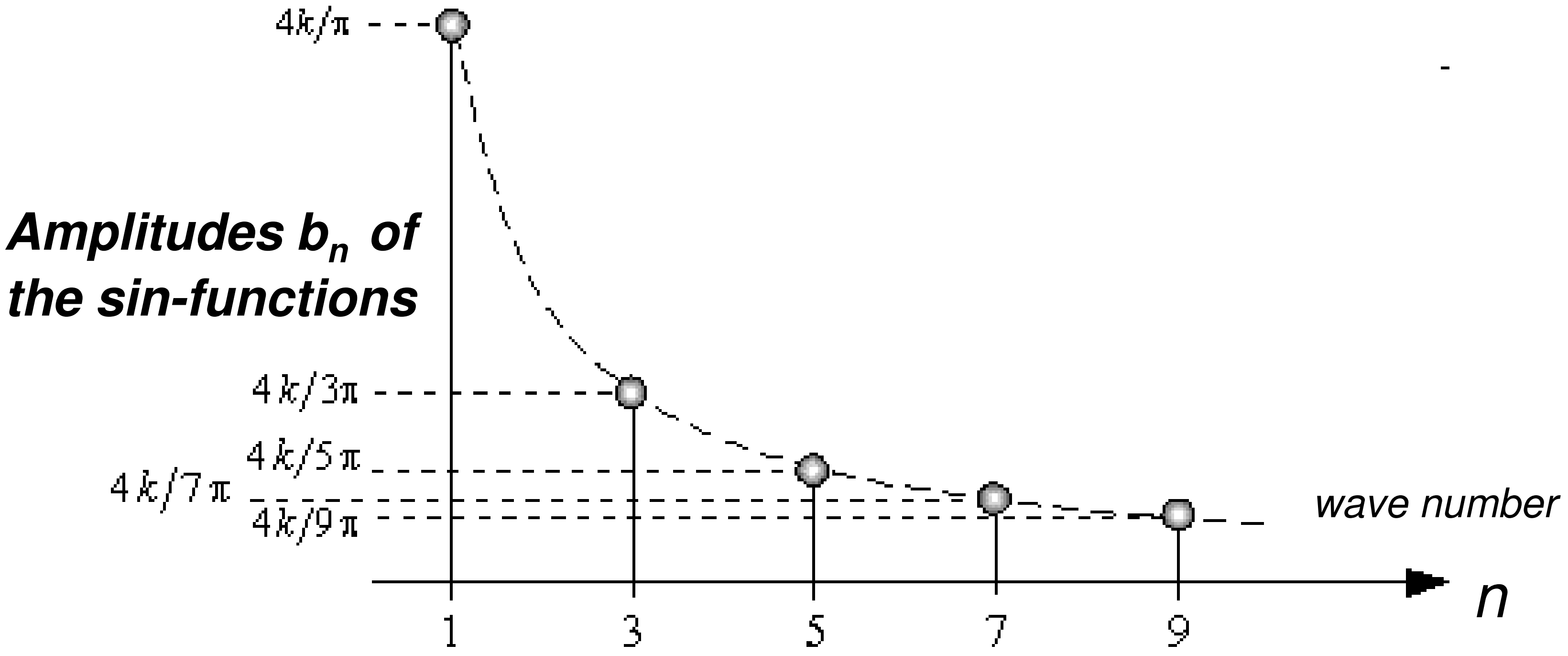
$$= \frac{2k}{n\pi}\left(1 - \cos(n\pi)\right) = \begin{cases} \dfrac{4k}{n\pi} & \text{if } n \text{ odd} \\ 0 & \text{if } n \text{ even} \end{cases}$$



Example: Periodic Rectangle Function

Final result in frequency space:

$$f(x) = \frac{4k}{\pi} \sum_{m \geq 0} \frac{\sin(\,(2m+1)\,x)}{2m+1}$$

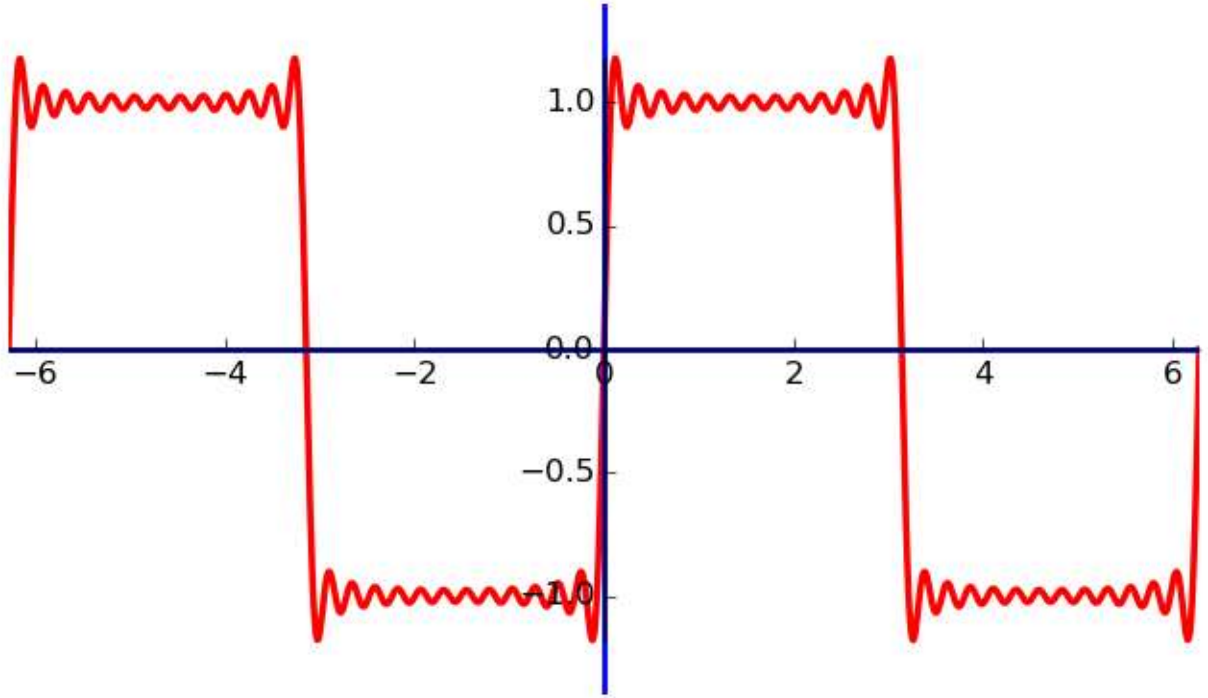


Example: Periodic Rectangle Function

- **Reminder**: Plot of partial sum

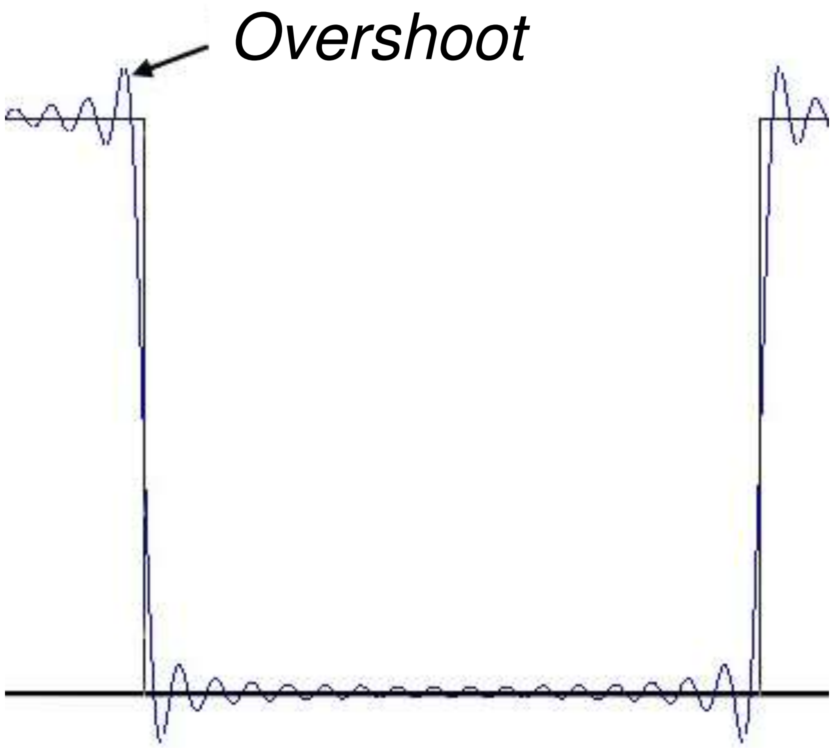
$$f(x) = \frac{4}{\pi}\left(\sin x + \frac{1}{3}\sin 3x + \frac{1}{5}\sin 5x + \cdots \right)$$

up to $n = 25$:



Convergence at Finite Discontinuities

- We allowed for a finite number of finite discontinuities
- At the discontinuity, the Fourier series behaves like a sinc-function sinc(x)=sin(x)/x.
- **Gibbs Phenomenon:**
  The first overshoot and undershoot combined are about 18% of the magnitude of the discontinuity
- **In other words:** Better approximation decreases the width, but not the height.



Example: Das Gibbs Phenomenon

- In Magnetic Resonance Imaging, Gibbs Phenomenon can lead to visible oscillations around sharp edges
  - We will understand why in Chapter 3

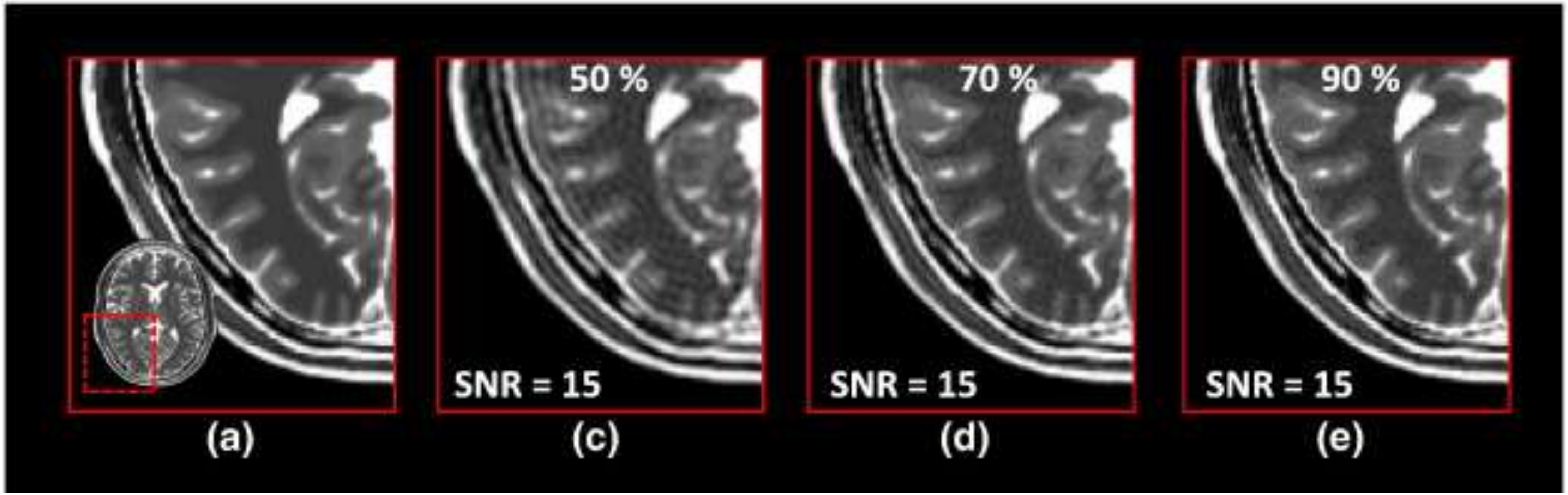
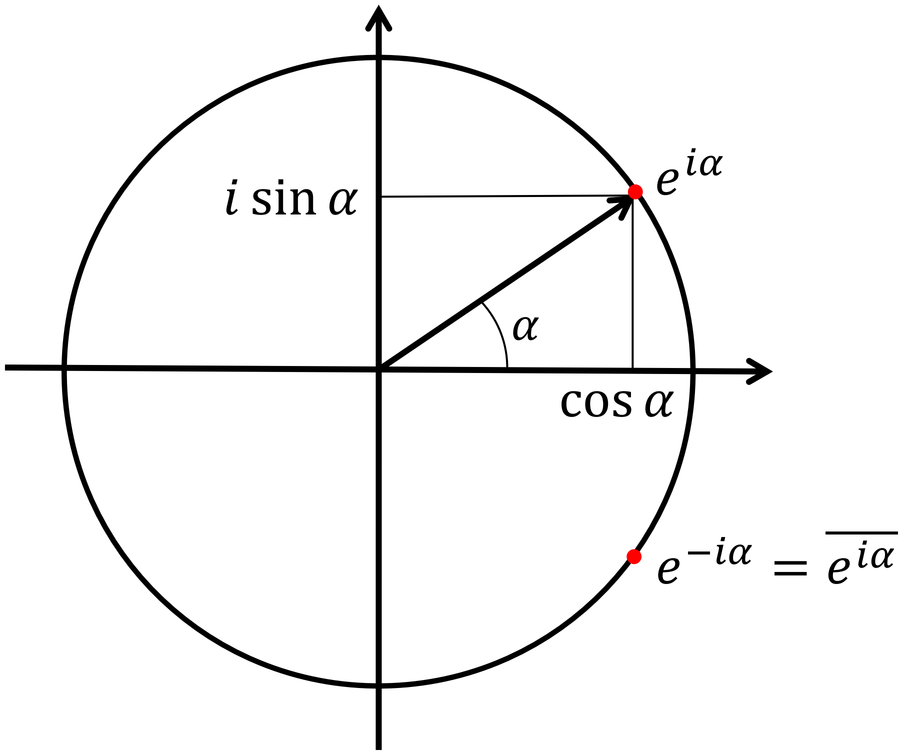


Image Source: [Perrone et al., NeuroImage 2015]

Euler's Formula

Link between complex exponential and trigonometric functions:



$$\boxed{e^{i\alpha} = \cos\alpha + i\sin\alpha} \qquad i = \sqrt{-1}$$

Polar form with magnitude r and angle ϕ:

$$z = re^{i\phi} = r(\cos\phi + i\sin\phi)$$

$$\cos\alpha = \frac{e^{i\alpha} + e^{-i\alpha}}{2}$$

$$\sin\alpha = -i\frac{e^{i\alpha} - e^{-i\alpha}}{2}$$

Fourier Series in Complex Notation

$$f(x) = \sum_{n=0}^{\infty} (a_n \cos(nx) + b_n \sin(nx))$$

$$= \sum_{n=0}^{\infty} \left(a_n \frac{e^{inx} + e^{-inx}}{2} - ib_n \frac{e^{inx} - e^{-inx}}{2} \right)$$

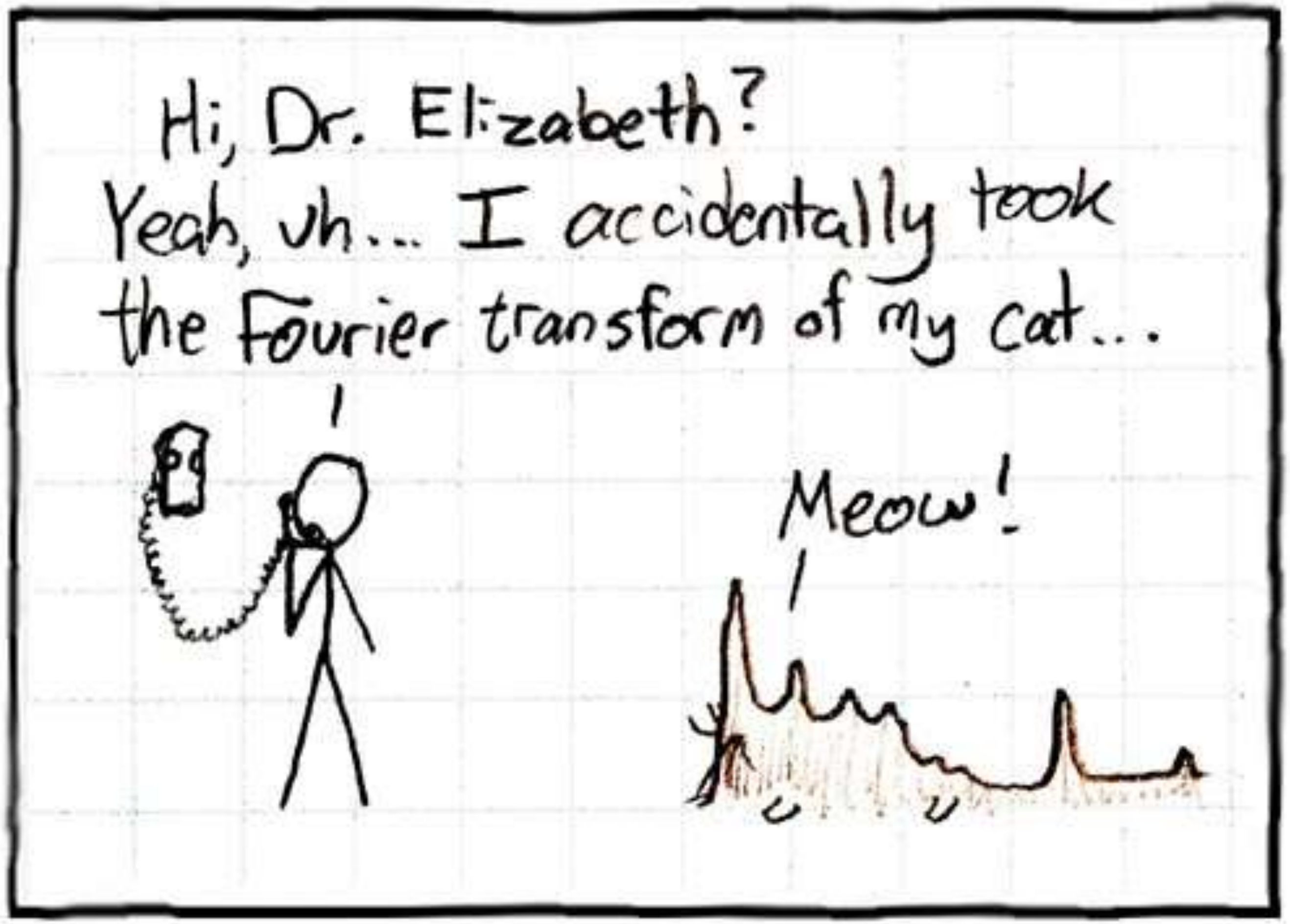
$$= \sum_{n=0}^{\infty} \left(\frac{a_n - ib_n}{2} e^{inx} + \frac{a_n + ib_n}{2} e^{-inx} \right) = \sum_{n=-\infty}^{\infty} c_n e^{inx}$$

A Fourier Series is real if and only if $c_n = \overline{c_{-n}}$

Initial Summary: Fourier Series

- (Almost) any periodic function can be written as an infinite sum of **Fourier basis functions**
  – Intuition: Decomposition into constituent frequencies
  – Real functions: sine and cosine
  – Complex functions: complex exponentials
    - Interpretation: magnitude and phase

Nerd Joke Alert



Source: https://xkcd.com/26/

The Fourier Transformation

- Generalizing Fourier Series to non-periodic functions *f(x)* yields the **Fourier Transform:**

inverse transform

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{inx} \qquad f(x) = \int_{-\infty}^{\infty} F(u)\, e^{2\pi iux}\, du$$

$$c_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-inx} dx \qquad F(u) = \int_{-\infty}^{\infty} f(x)\, e^{-2\pi iux}\, dx$$

forward transform

Fourier Duality

- Time/space and frequency domains are duals
- **Theorem:** If *F(u)* is the Fourier transform of *f(x)*, then *f(-x)* is the Fourier transform of *F(u)*.
- **Proof:**

Given:

$$F(u) = \int_{-\infty}^{\infty} f(x)\, e^{-2\pi iux}\, dx$$

forward transform of *f(x)*

$$\Leftrightarrow f(x) = \int_{-\infty}^{\infty} F(u)\, e^{2\pi iux}\, du$$

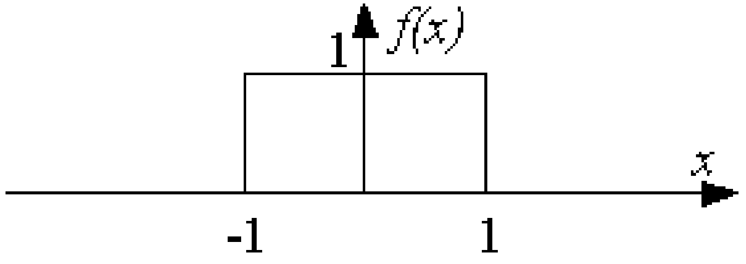
inverse transform of *F(u)*

Thus:

$$f(-x) = \int_{-\infty}^{\infty} F(u)\, e^{-2\pi iux}\, du$$

= forward transform of *F(u)*

Example: Fourier Transform of Box Function



$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i u x} dx = \int_{-1}^{1} e^{-2\pi i u x} dx = -\frac{1}{2\pi i u}\left[e^{-2\pi i u x}\right]_{-1}^{1}$$

$$= \frac{i}{2\pi u}\left(e^{-2\pi i u} - e^{2\pi i u}\right) = \frac{1}{\pi u}\sin(2\pi u) = 2\frac{\sin(2\pi u)}{2\pi u}$$

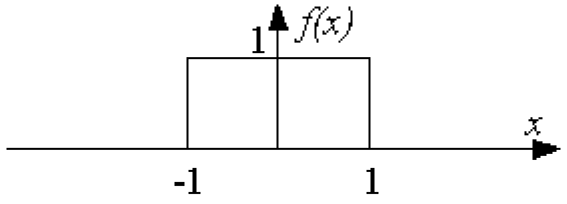
Reminder: $\sin\alpha = -i\frac{e^{i\alpha} - e^{-i\alpha}}{2}$

$$\boxed{\operatorname{sinc}(x) = \frac{\sin(x)}{x}}$$

Box Function in Time and Frequency Domain

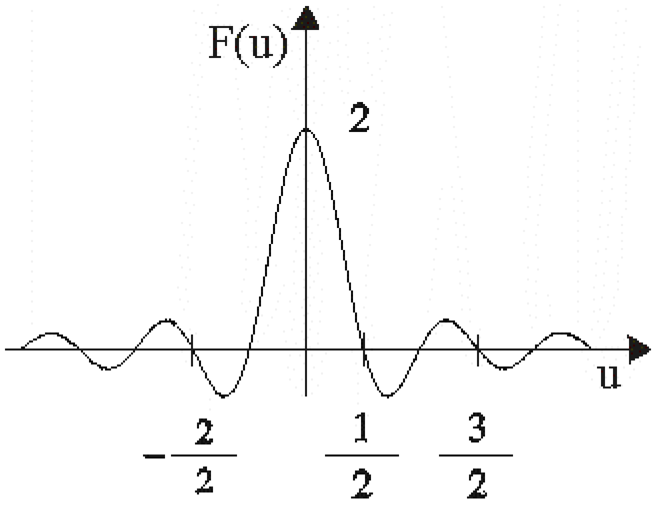
- Time (space) domain

$$f(x) = \begin{cases} 1, x \in [-1,1] \\ 0, \text{ otherwise} \end{cases}$$



- Frequency domain

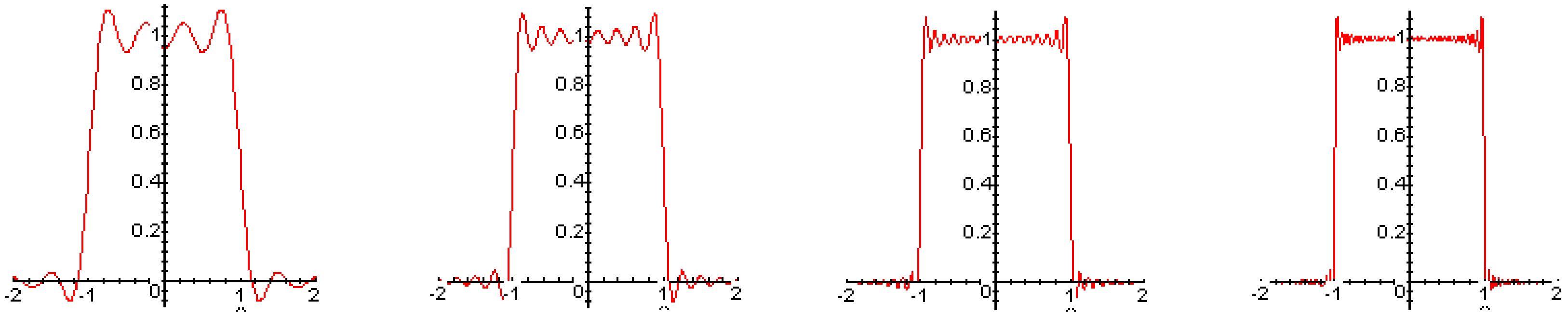
$$F(u) = 2\frac{\sin(2\pi u)}{2\pi u}$$



Reconstructing the Box Function

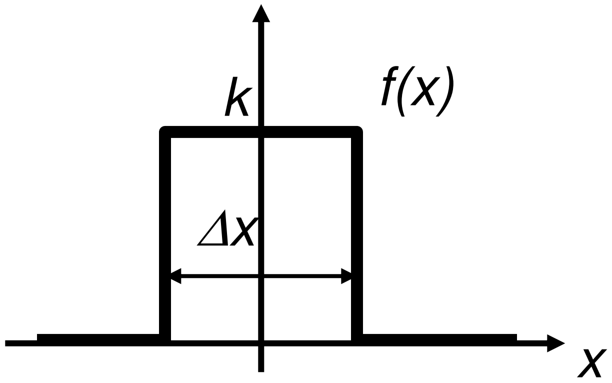
- Inverse Fourier Transform:

$$f(x) = \int_{-\infty}^{\infty} 2\frac{\sin(2\pi u)}{2\pi u} e^{2\pi i u x} du$$



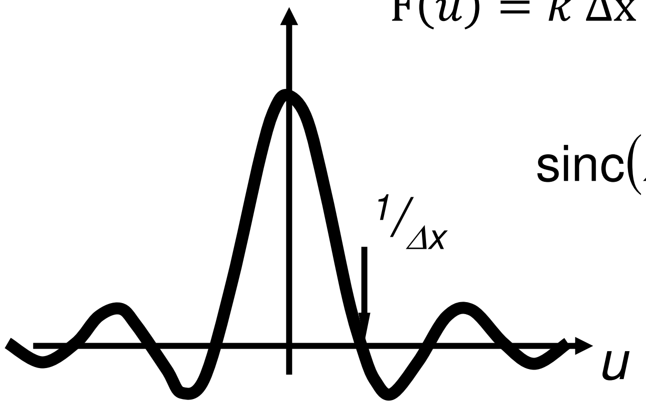
$\int_{-1}^{1} \ldots du$ $\int_{-2}^{2} \ldots du$ $\int_{-4}^{4} \ldots du$ $\int_{-8}^{8} \ldots du$

Fourier Pair: Box and Sinc



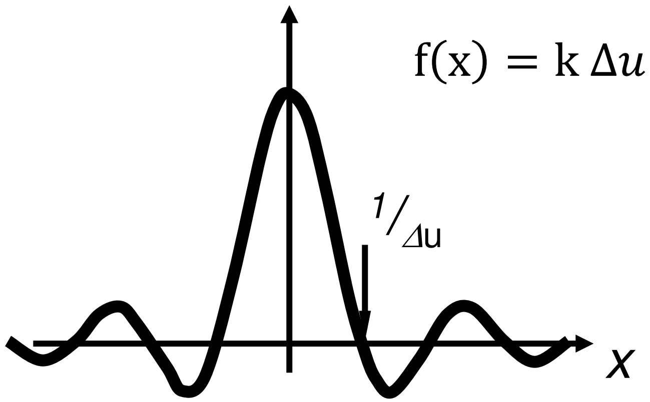
$\Leftrightarrow$

$$\mathrm{F}(u) = k\ \Delta\mathrm{x}\ \mathrm{sinc}(\pi u \Delta\mathrm{x})$$

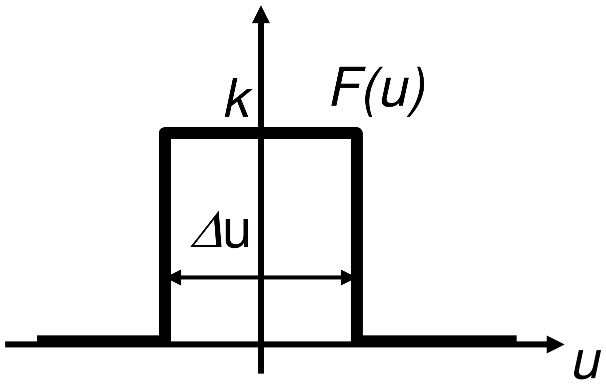


$$\mathrm{sinc}(x) = \frac{\sin(x)}{x}$$

$$\mathrm{f}(\mathrm{x}) = \mathrm{k}\ \Delta u\ \mathrm{sinc}(\pi \mathrm{x} \Delta u)$$



$\Leftrightarrow$



Note: For even functions *f(-x)=f(x)*, we can neglect the sign flip that otherwise occurs in the Fourier duality.

Dirac Delta

- The Dirac delta $\delta(x)$ has the property

$$\int_{-\infty}^{\infty} \delta(x)dx = 1$$

- Multiplying a function $f(x)$ with $\delta(x-a)$ and integrating over x yields the function value at a:

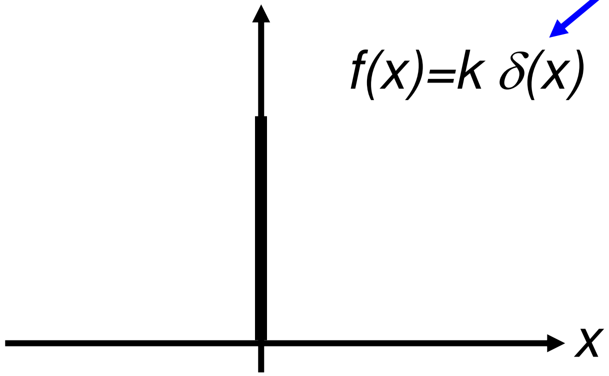
$$\int_{-\infty}^{\infty} f(x)\delta(x-a)dx = f(a)$$

- Informally, $\delta(x)$ is often thought of as a function that is $\delta(x) = 0$ for all x except $\delta(0) = \infty$. Strictly speaking, it is not a function.
- Graphical depiction of $\delta(x-a)$: Vertical bar at $x = a$

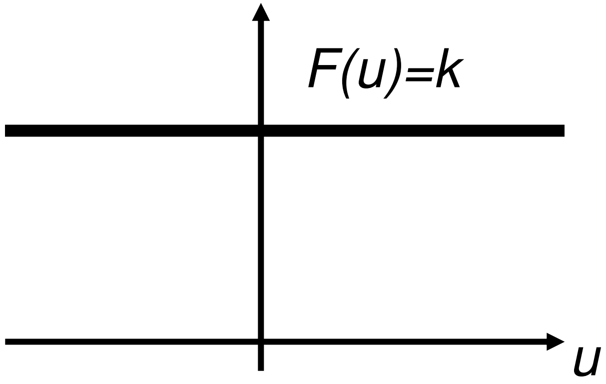
Dual Pair: Constant and Dirac Delta

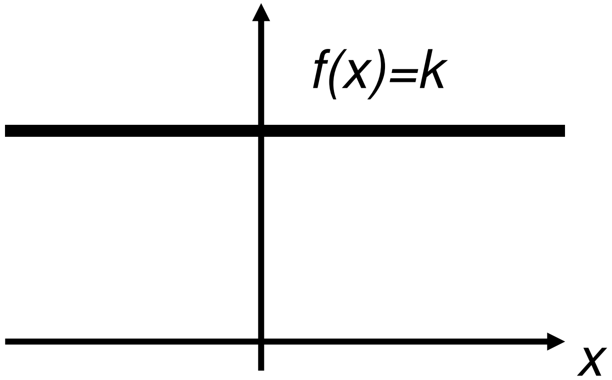
$$F(u) = \int_{-\infty}^{\infty} f(x)\, e^{-2\pi i u x}\, dx$$

Dirac delta

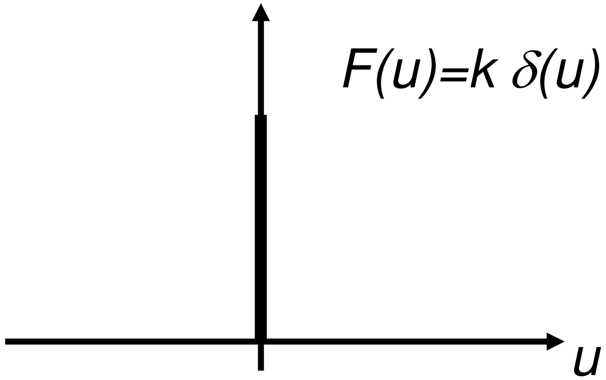


$\Leftrightarrow$





$\Leftrightarrow$

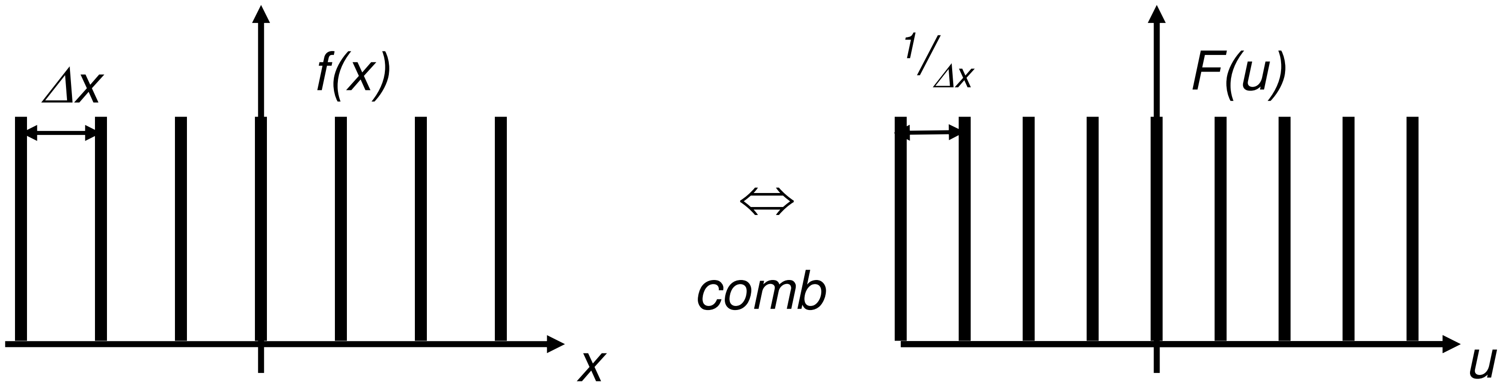


Fourier Transform of the Comb

- A **comb** $Ш_{\Delta x}$ denotes an infinite series of Δx spaced Dirac δ:

$$Ш_{\Delta x}(x) = \sum_{k=-\infty}^{\infty} \delta(x - k\Delta x)$$

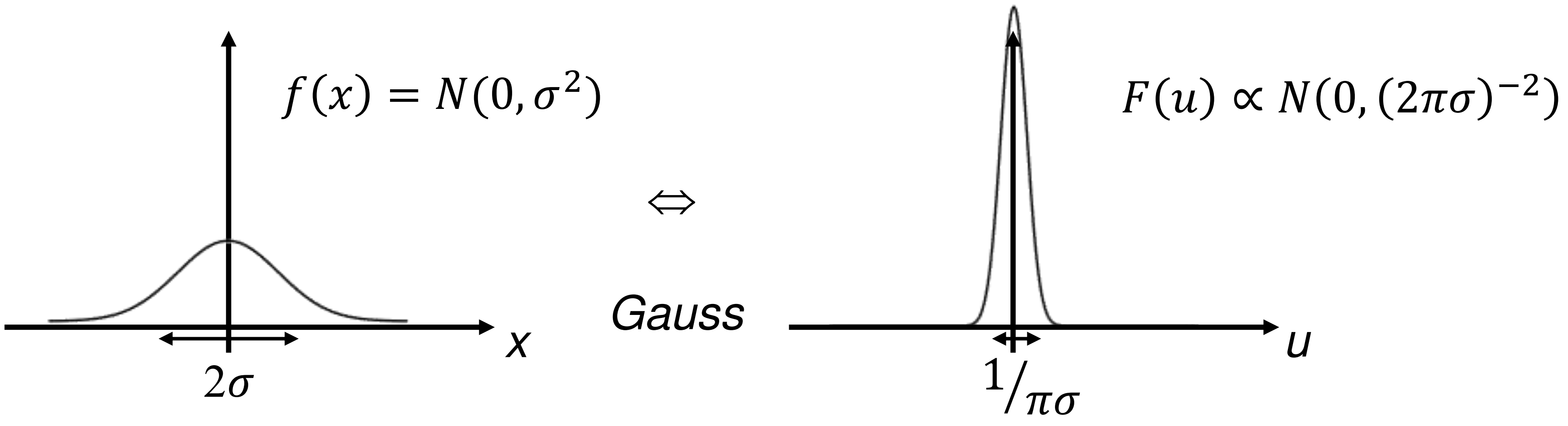
- The Fourier transform of $Ш_{\Delta x}$ is $\mathcal{F}\{Ш_{\Delta x}\} = \frac{1}{\Delta x} Ш_{\frac{1}{\Delta x}}$



Fourier Transform of the Gaussian

The Fourier Transform of a Gaussian $\mathcal{N}(0, \sigma^2)$ is

$$\mathcal{F}\{\mathcal{N}(0, \sigma^2)\} = e^{-\frac{1}{2}\frac{u^2}{\sigma_u^2}} \text{ with } \sigma_u = \frac{1}{2\pi\sigma}$$

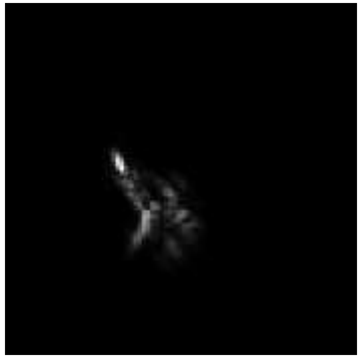


Properties of the Fourier Transform

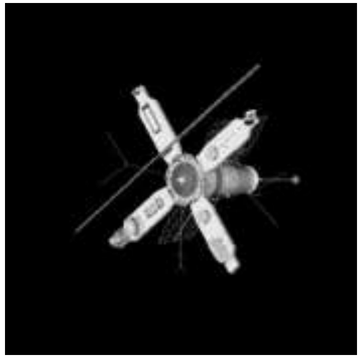
Property	Time / Space	Frequency Domain
Linearity	$c_1 \cdot f(x) + c_2 \cdot g(x)$	$c_1 \cdot F(u) + c_2 \cdot G(u)$
Time scaling	$f(c \cdot x)$	$\frac{1}{\lvert c \rvert} F\left(\frac{u}{c}\right)$
Space/Time shift	$f(x-x_0)$	$F(u)e^{-i2\pi u x_0}$
Time derivative	$df(x)/dx$	$i2\pi u \cdot F(u)$
Complex conjugation	$\overline{f(x)}$	$\overline{F(-u)}$
Even signals	$f(-x)=f(x)$	$F(-u)=F(u)$
Odd signals	$f(-x)=-f(x)$	$F(-u)=-F(u)$
Convolution	$f(x)*g(x)$	$F(u) \cdot G(u)$
Modulation	$f(x) \cdot g(x)$	$F(u)*G(u)$

Convolution: Intuition

- **Convolution** is an operation on two given functions h, g that produces another function f
  - *Intuition*: f obtained as weighted sum of shifted versions of g. The weights of different shifts are given by h
    - *Example*: Photograph with motion blur
  - In practice, "kernel" h often has much smaller support than g
    - Note that h and g are interchangeable, since convolution is *commutative*



$*$



$=$

h g f

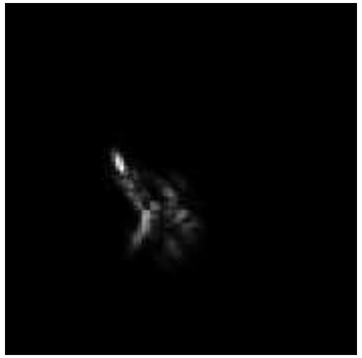
Image Source: PhD Thesis of Michael Hirsch

Continuous Convolution

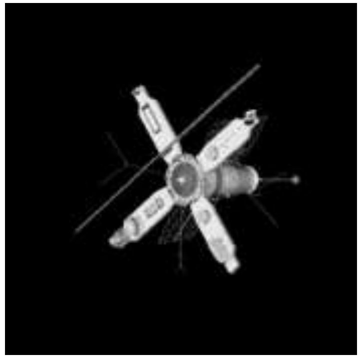
- **Definition:** Convolution of continuous functions g and h:

$$f(x) = (h * g)(x) = \int_{-\infty}^{\infty} h(\xi) \cdot g(x - \xi) d\xi$$

  – Intuition: Negative ξ should shift *g* to the left
  – Convolution is associative and commutative

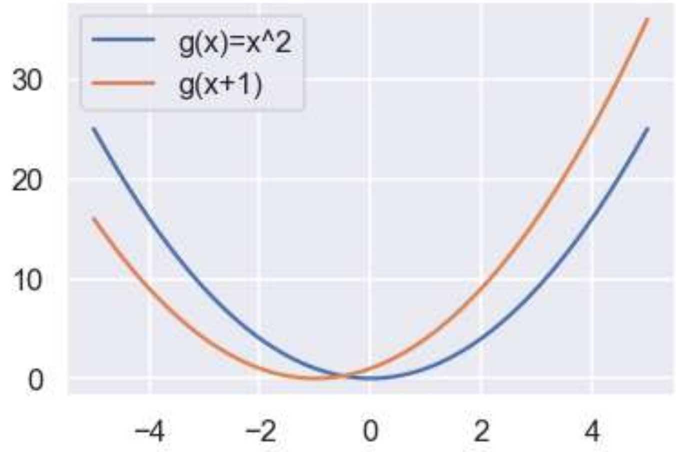


*



h

g



Convolution Theorem

- **Theorem:** $f(x) = g*h$ corresponds to $F(u) = G \cdot H$
- **Proof**:

$$F(u) = \int_{x=-\infty}^{\infty} \int_{\xi=-\infty}^{\infty} g(\xi) h(x-\xi) d\xi \, e^{-2\pi i u x} \, dx$$
$$= \int_{\xi=-\infty}^{\infty} g(\xi) \int_{x=-\infty}^{\infty} h(x-\xi) \, e^{-2\pi i u x} \, dx \, d\xi$$

with $y \coloneqq x - \xi$ ($dy=dx$):

$$= \int_{\xi=-\infty}^{\infty} g(\xi) \int_{y=-\infty}^{\infty} h(y) \, e^{-2\pi i u (y+\xi)} \, dy \, d\xi$$
$$= \underbrace{\int_{\xi=-\infty}^{\infty} g(\xi) e^{-2\pi i u \xi} \, d\xi}_{=:G(u)} \underbrace{\int_{y=-\infty}^{\infty} h(y) \, e^{-2\pi i u y} \, dy}_{=:H(u)}$$

Note: Similarly, $f(x) = h(x) \cdot g(x)$ corresponds to $F(u) = H(u) * G(u)$

Generalizing FT to 2D

- The Fourier Transform generalizes to 2D functions *f(x,y)* in a straightforward manner:
  - Forward:

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{-2\pi i(ux+vy)}\, dx\, dy$$

  - Inverse:

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)\, e^{2\pi i(ux+vy)}\, du\, dv$$

- It is *separable* in the two variables

Generalizing FT to 3D

- Completely analogous, except we use vector notion:
  - Forward:

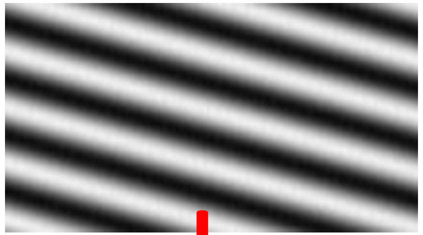
$$F(\mathbf{s}) = \iiint_{-\infty}^{\infty} f(\mathbf{r}) e^{-2\pi i(\mathbf{r}\cdot\mathbf{s})}\, d\mathbf{r}$$

  - Inverse:

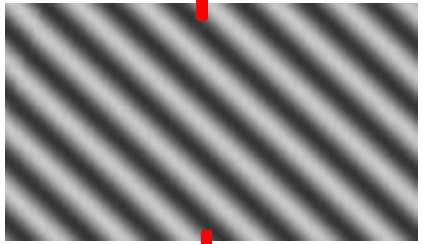
$$f(\mathbf{r}) = \iiint_{-\infty}^{\infty} F(\mathbf{s}) e^{2\pi i(\mathbf{r}\cdot\mathbf{s})}\, d\mathbf{s}$$

- It remains separable

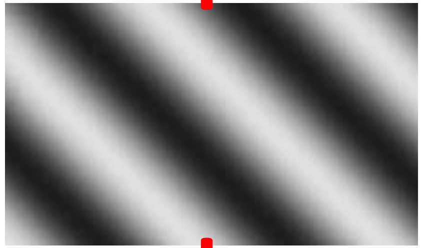
Intuition: Images in Frequency Space

$\alpha_1 \cdot$ 

$+$

$\alpha_2 \cdot$ 

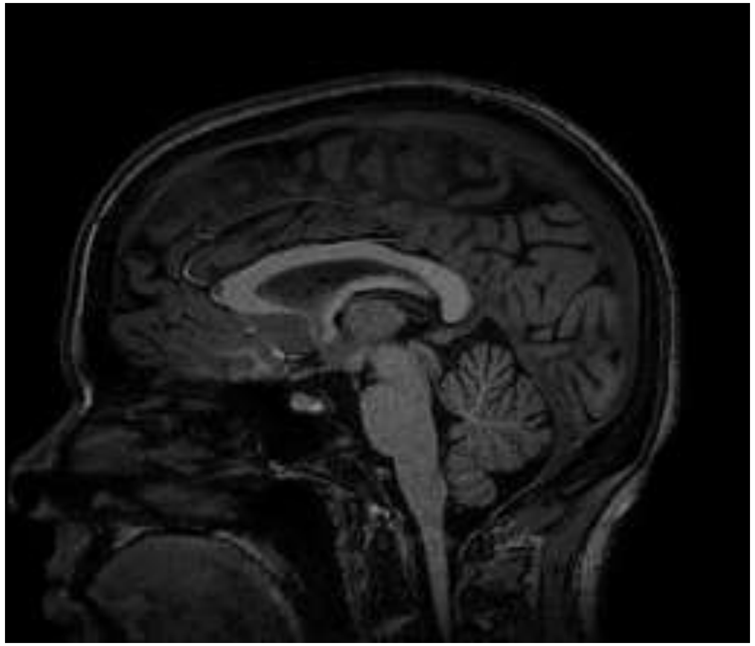
$+$

$\alpha_3 \cdot$ 

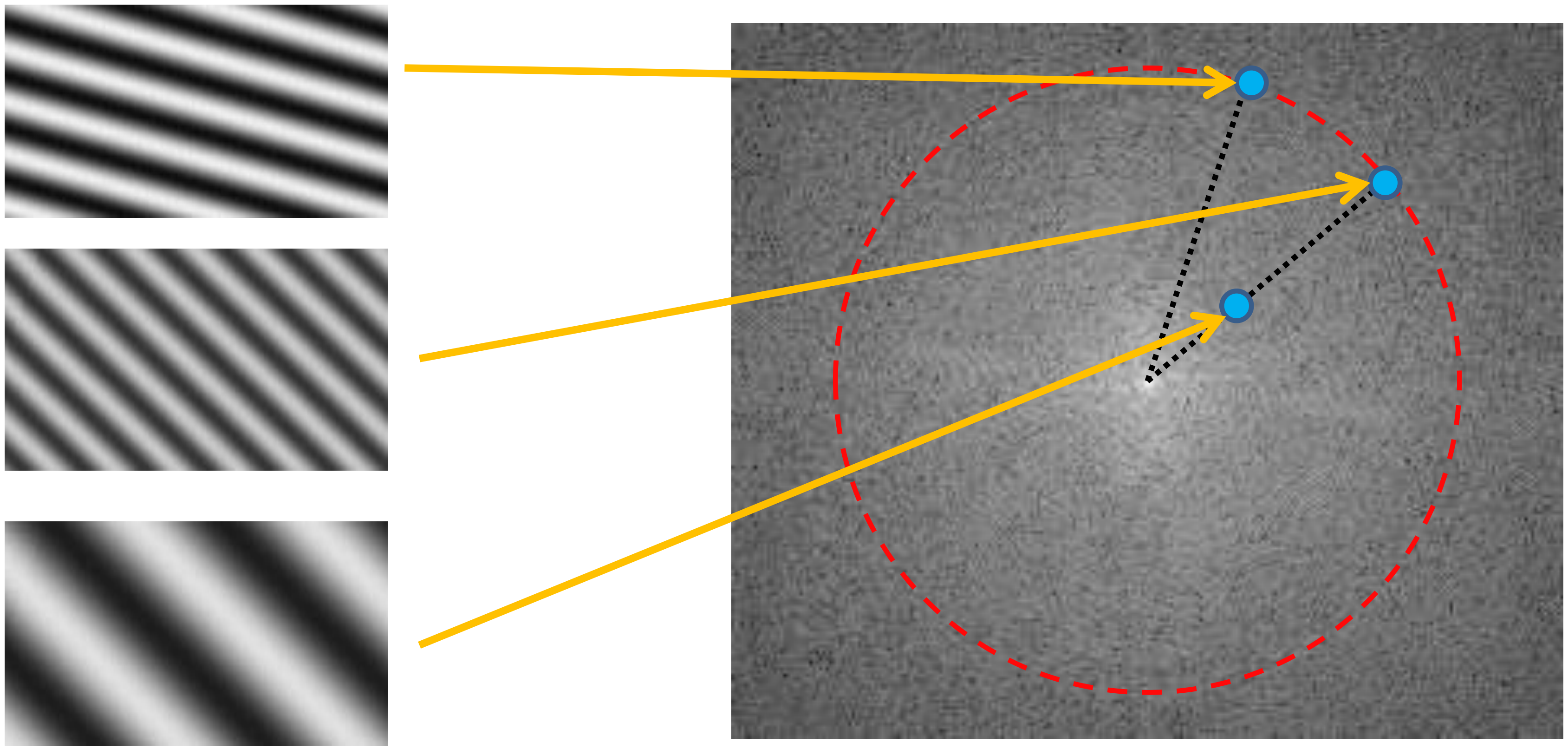
$+$

$\cdots$

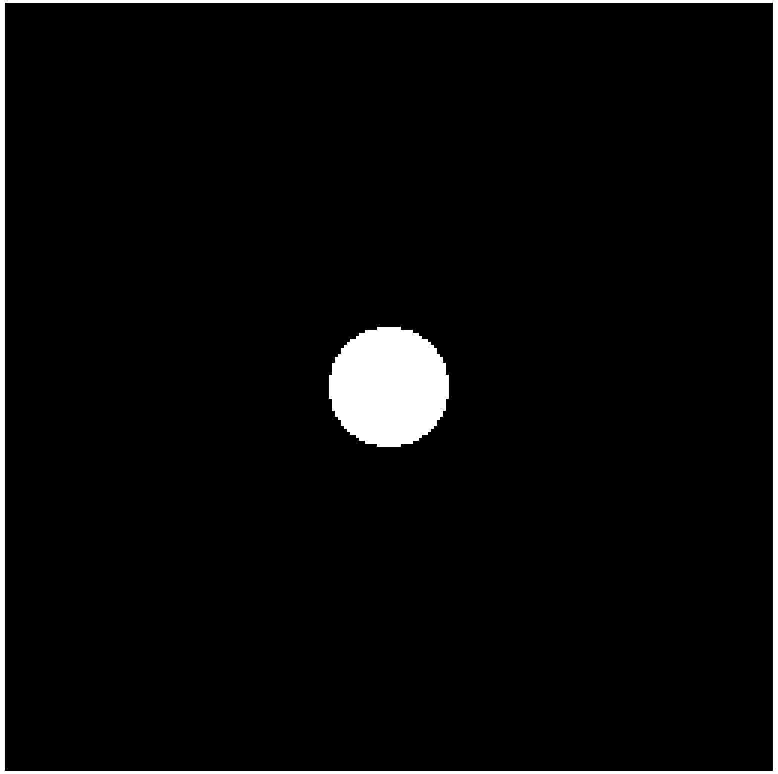
$=$



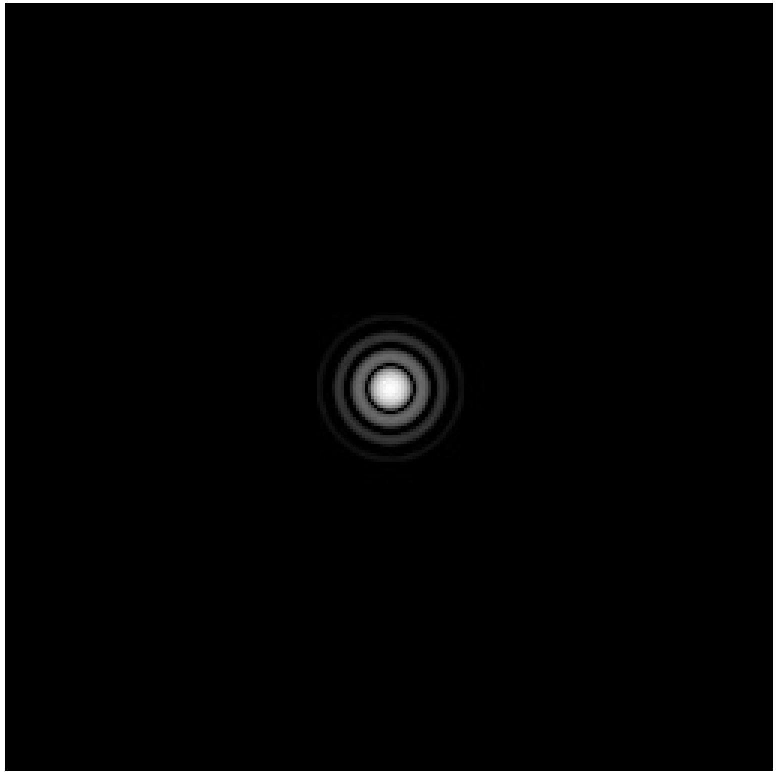
Getting an Intuition for k Space



Power Spectrum of a Disk



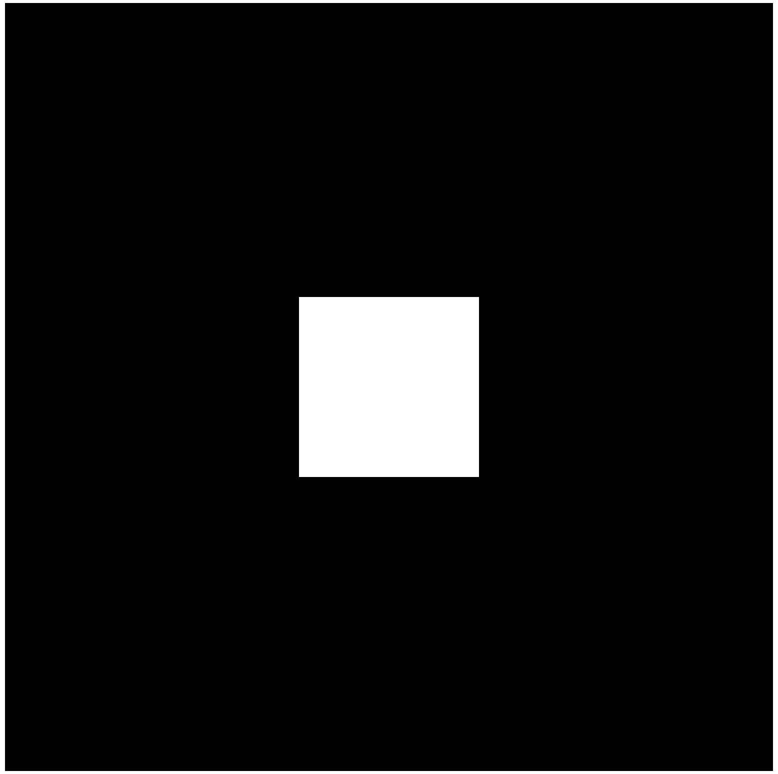
Image



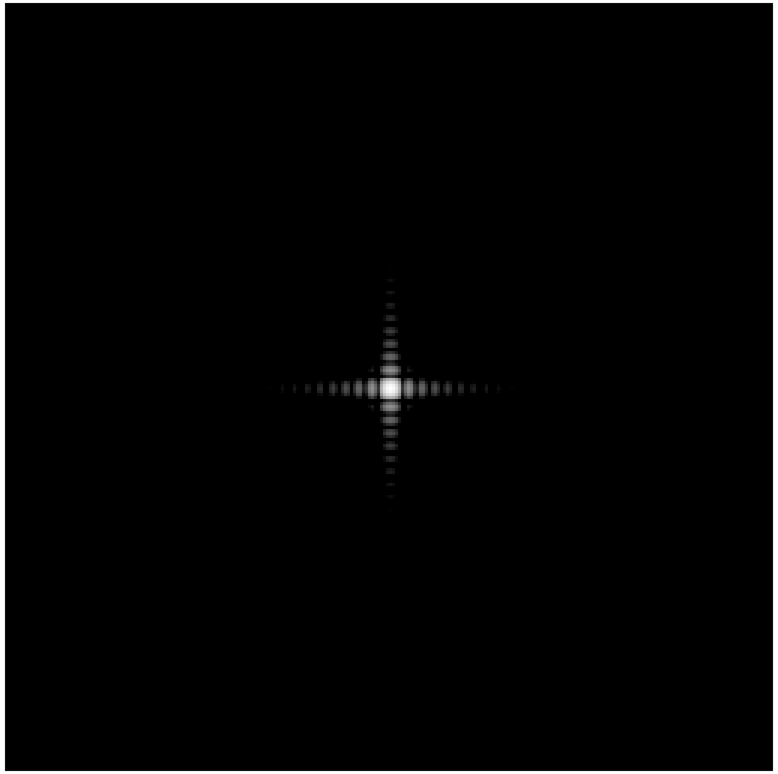
Power Spectrum

Note: Brightness in spectrum reflects logarithmic power, cut off at -30 dB

Power Spectrum of a Square

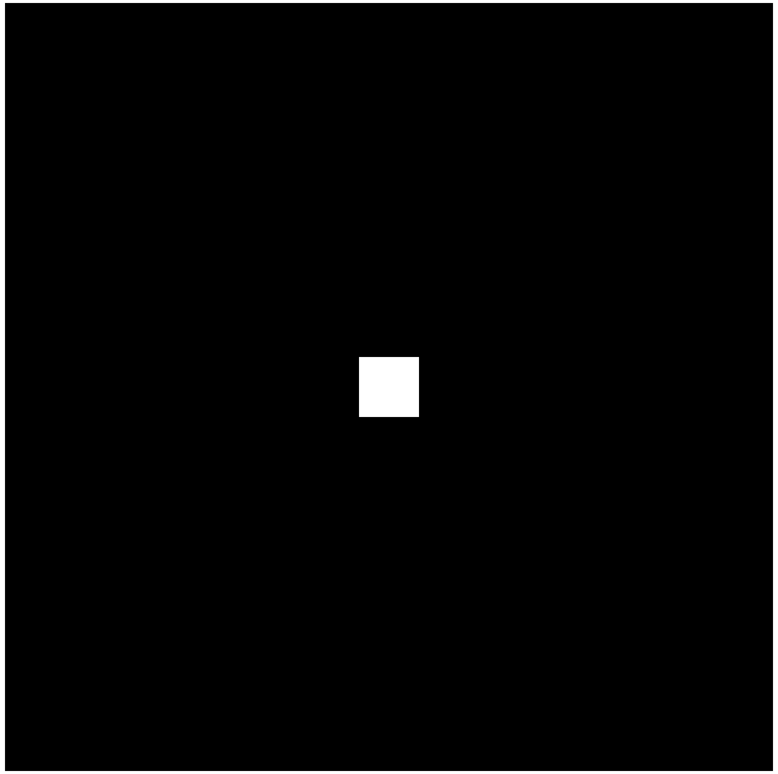


Image

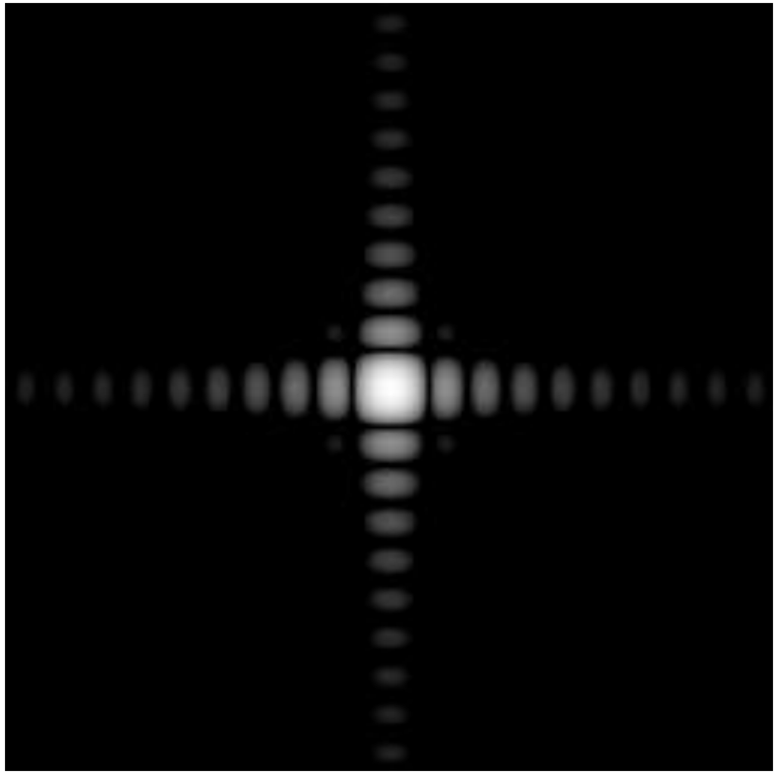


Power Spectrum

Effect of Scaling

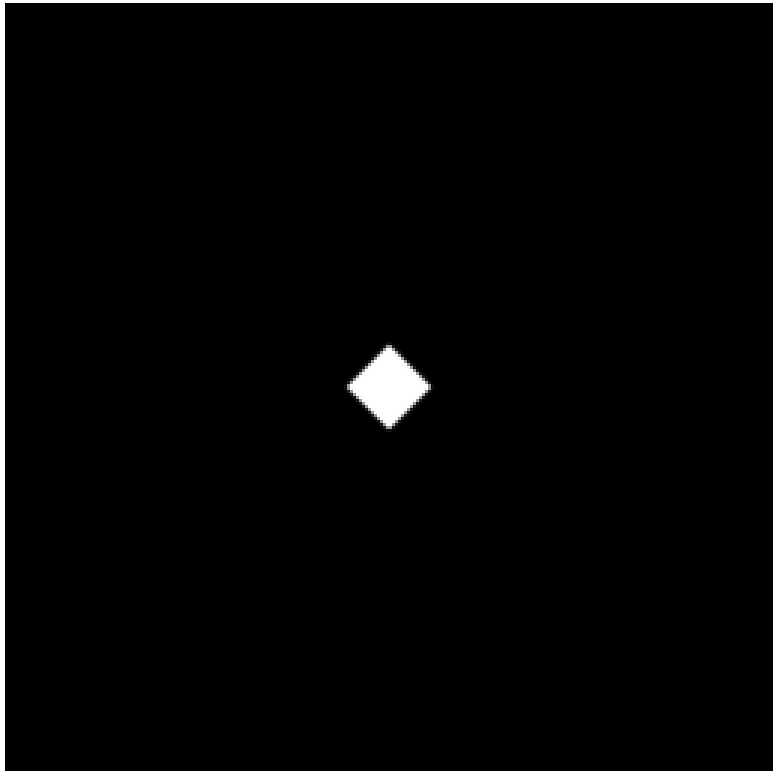


Image

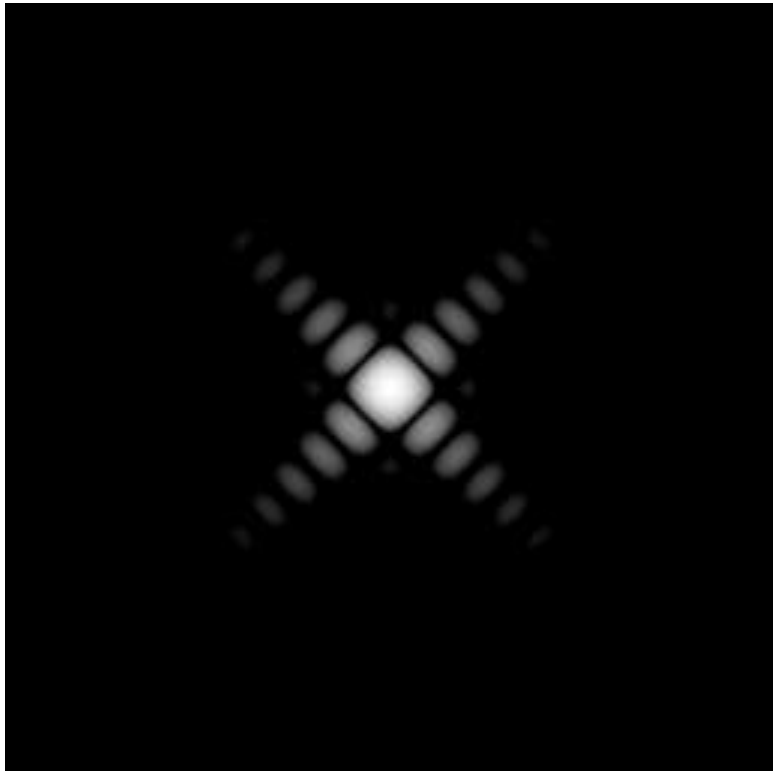


Power Spectrum

Effect of Rotation

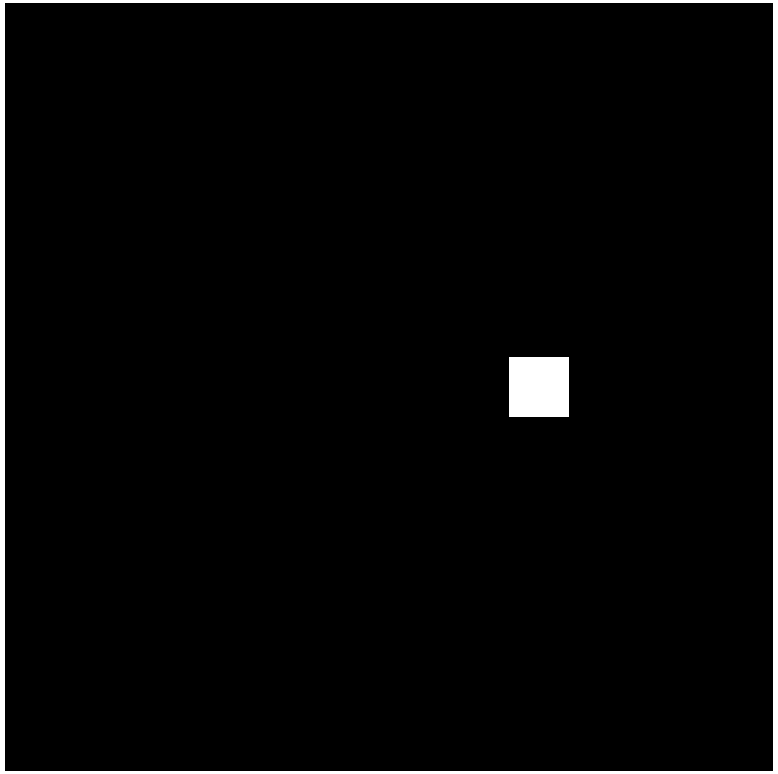


Image

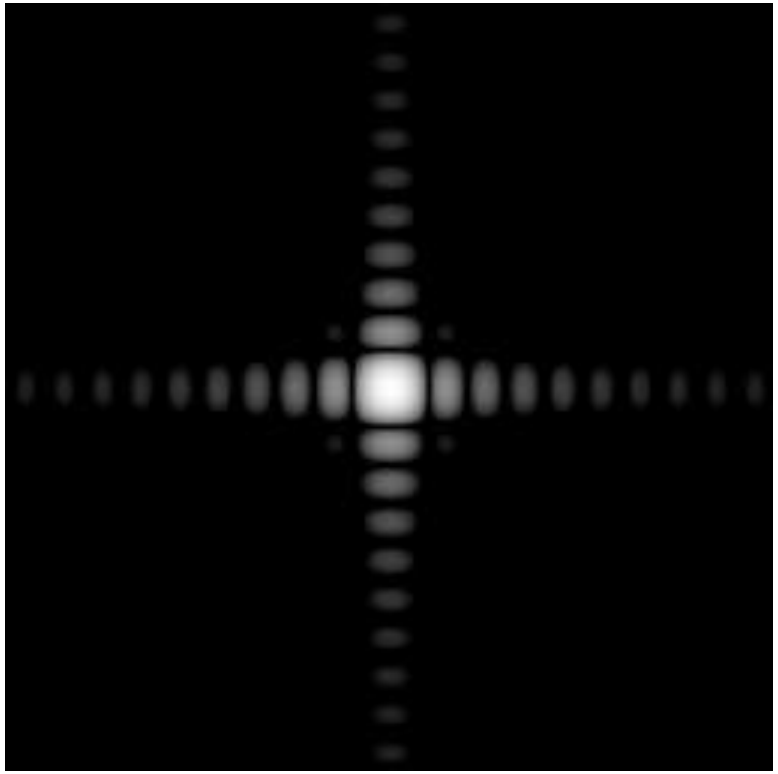


Power Spectrum

Effect of Translation



Image

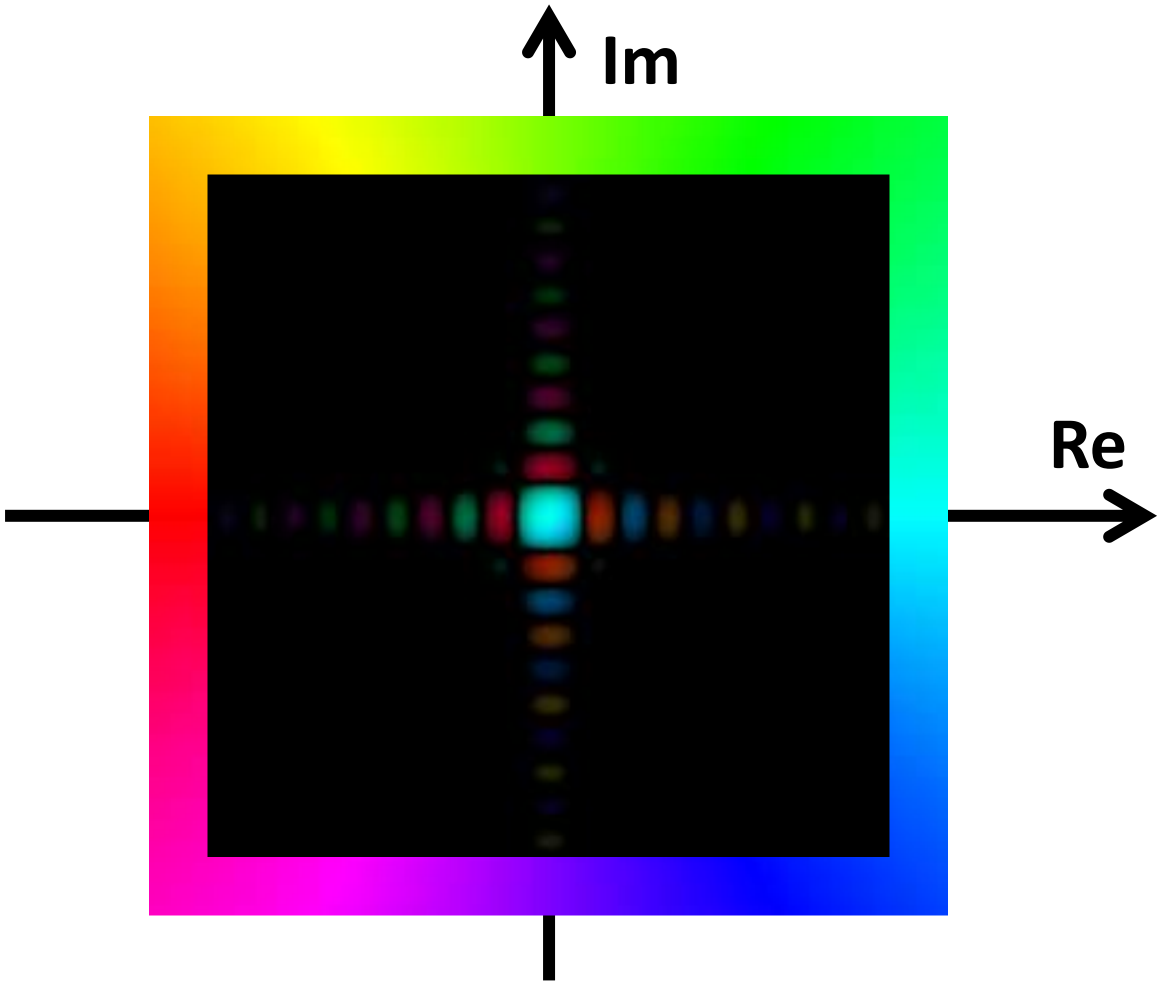


Power Spectrum

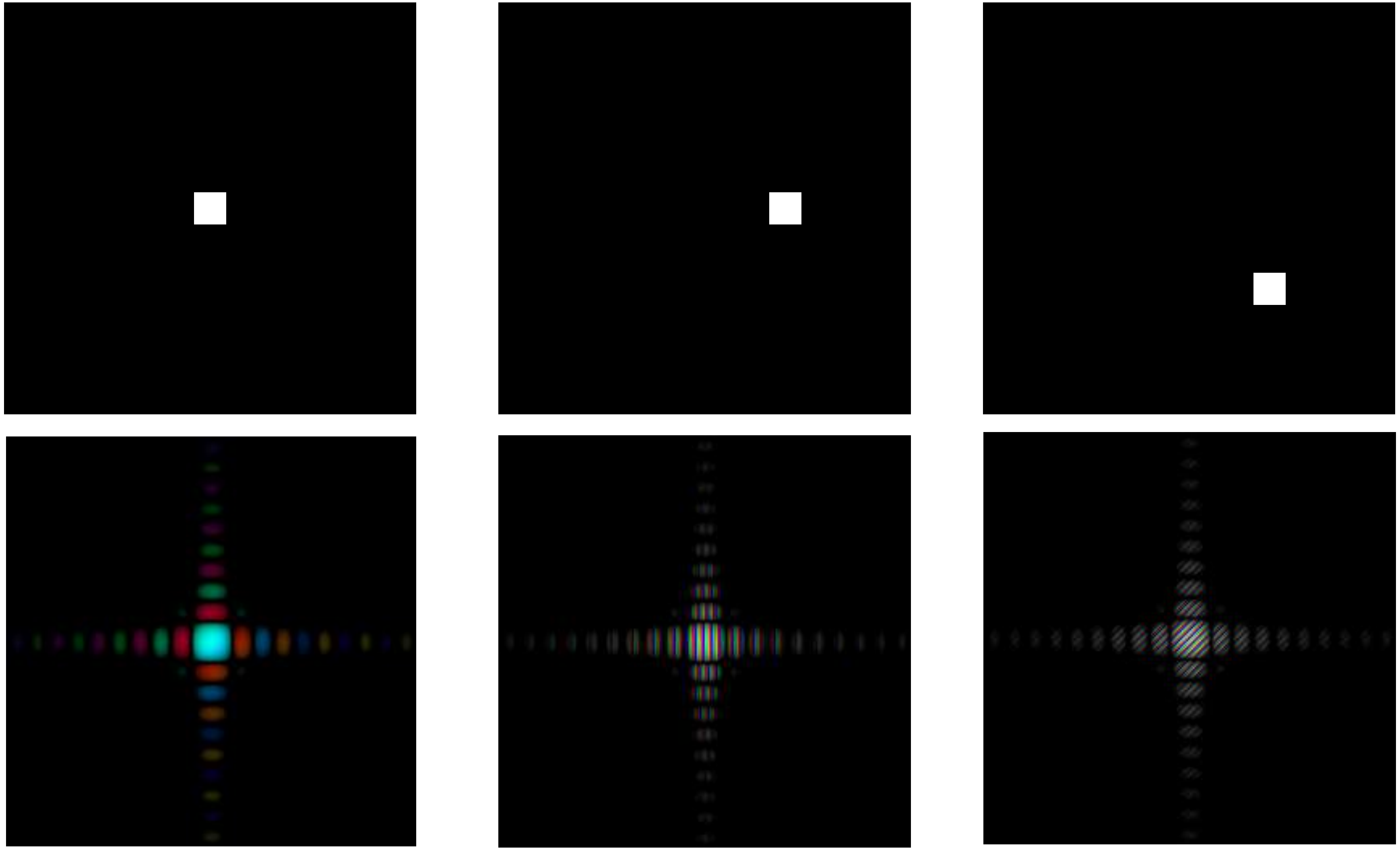
Note: Translation does not affect the power spectrum.

Color Coding Phase

- Use hue to indicate phase of Fourier Transform
- Brightness continues to encode log power

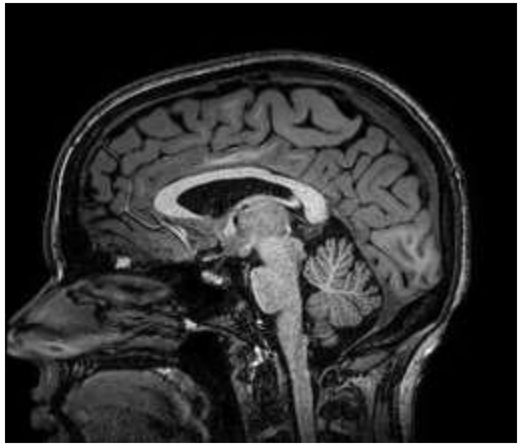


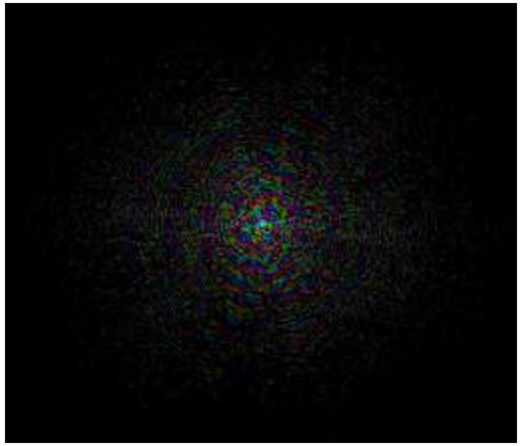
Effect of Translation on Phase

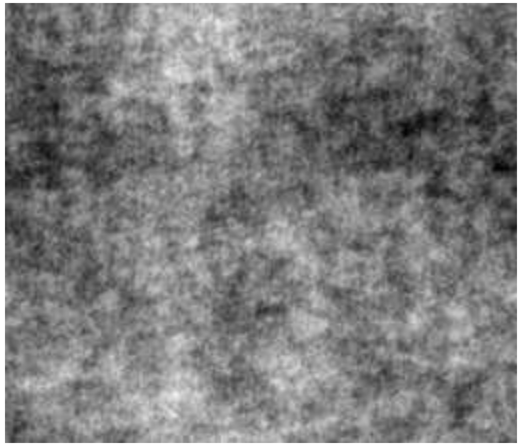


Amplitude vs. Phase Information

Head

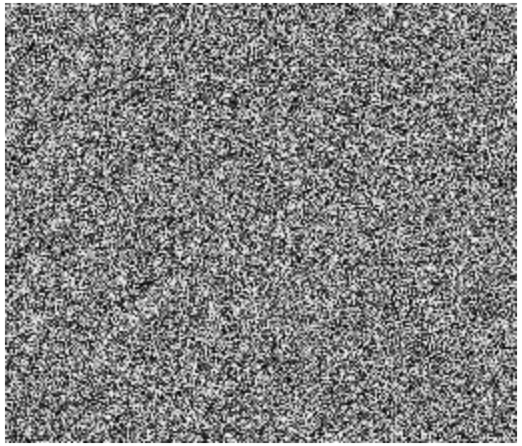


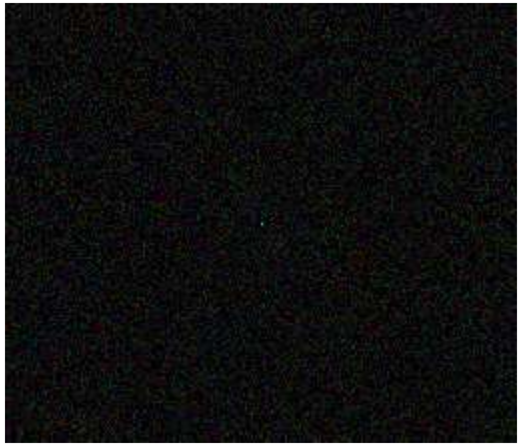


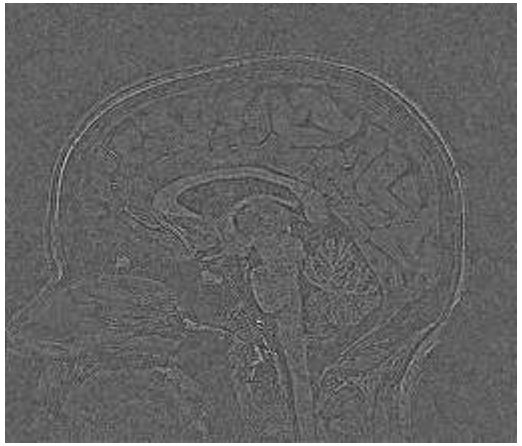


Amplitude: Head *Phase:* Random

Noise





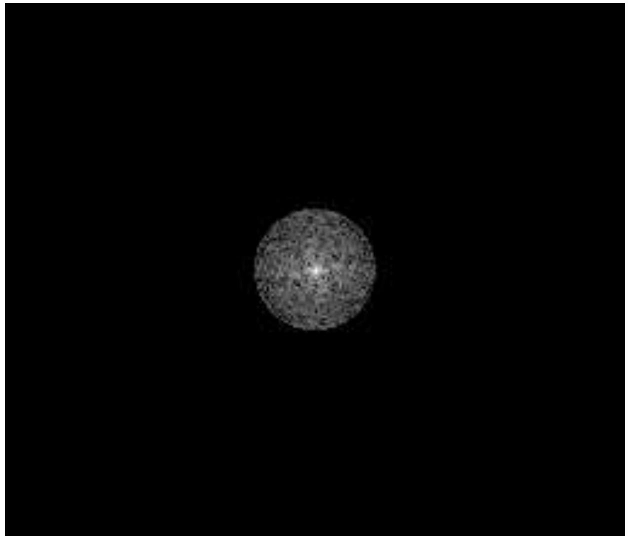


Amplitude: Random *Phase:* Head

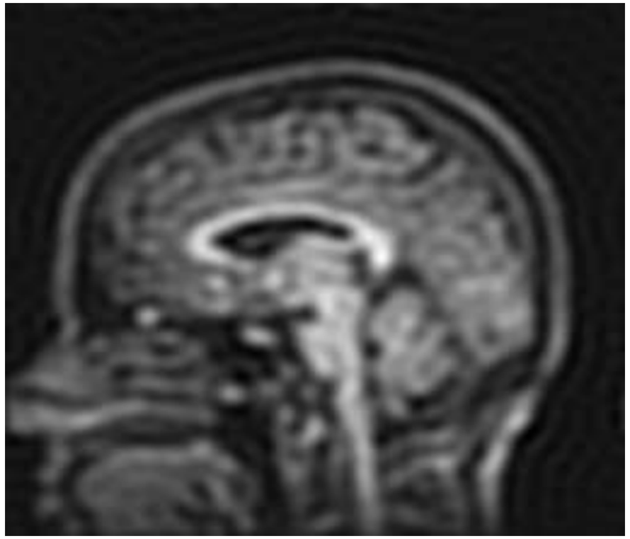
Note: Power spectrum now cut off at -60 dB

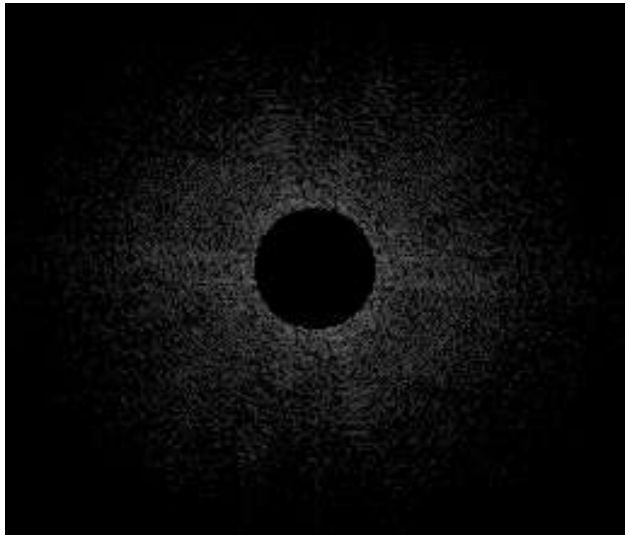
Note: Images are renormalized.

Center vs. Periphery of Fourier Space



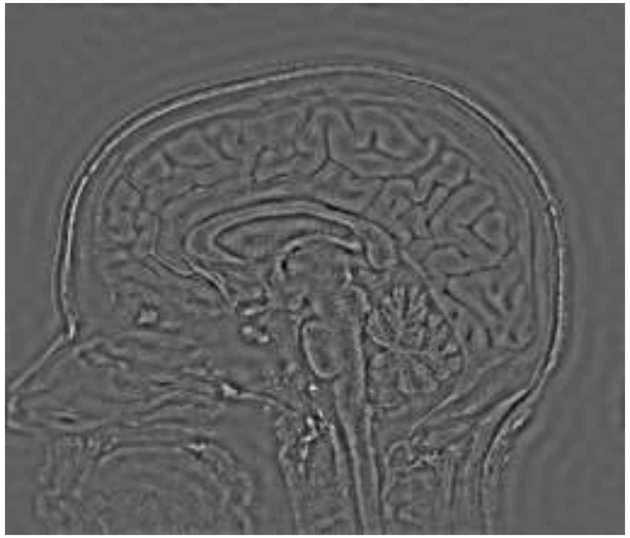
Reconstruction from center:
Low-pass filter





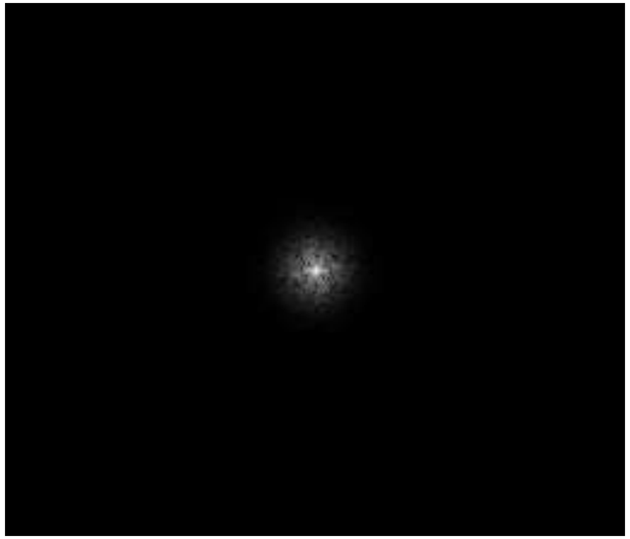
Reconstruction from periphery:
High-pass filter

Note: Hard cutoff introduces ringing

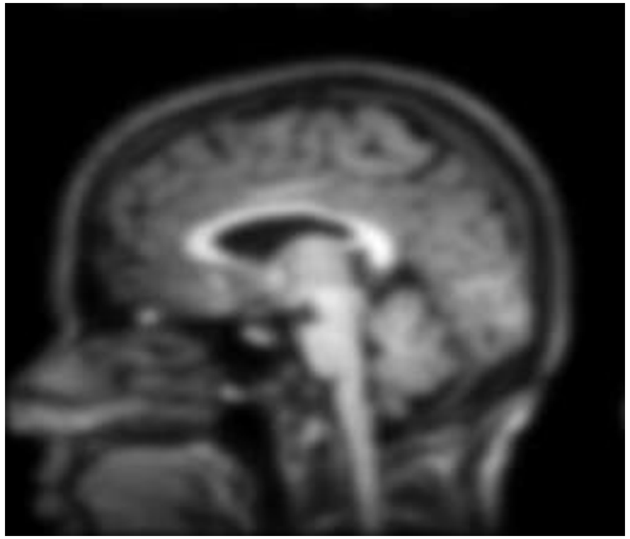


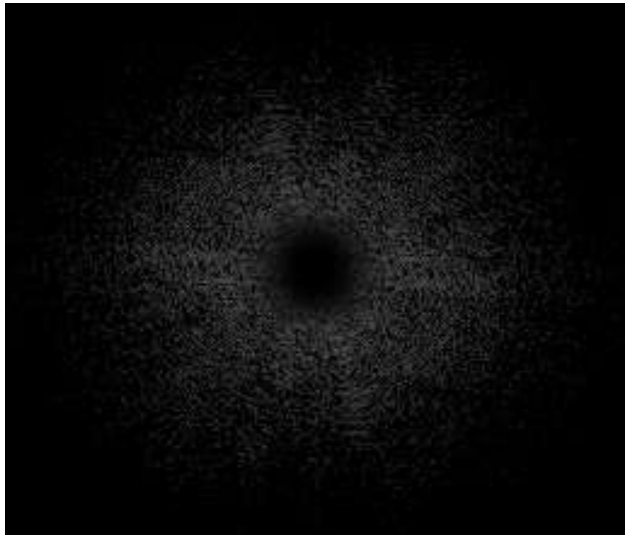
Note: Images are renormalized.

Filtering Without the Ringing

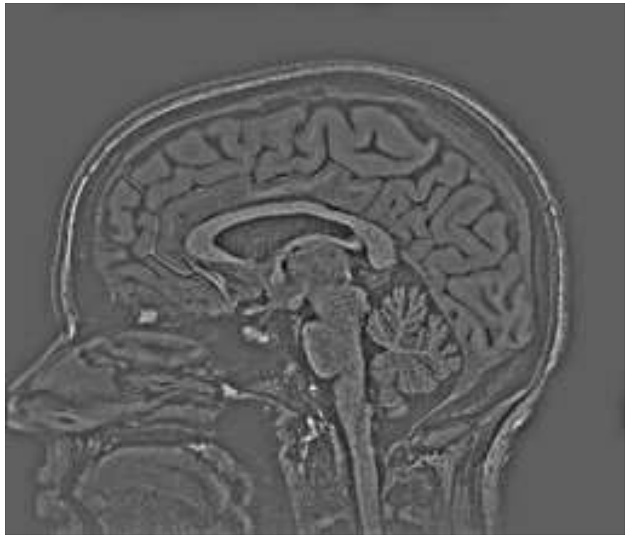


Gaussian low-pass filter





Gaussian high-pass filter



Note: Images are renormalized.

Summary: Fourier Transform

- (Almost) any function can be transformed from space/time to **frequency domain**
  - It is possible to transfer back using the inverse transformation
- You should have learned:
  - How to compute the **Fourier transformation**
  - Some widely used **Fourier pairs** (box, sinc, Dirac delta, Gaussian, comb)
  - Some **properties** of the Fourier transform, in particular the **convolution theorem**

2.2 Basics of Digital Signal Processing

What is a Signal?

- A signal is a function that conveys information about the behavior of a system or attributes of some phenomenon. Signals occur naturally and they are also synthesized. […] We will consider a signal to be some function of an independent variable such as time.

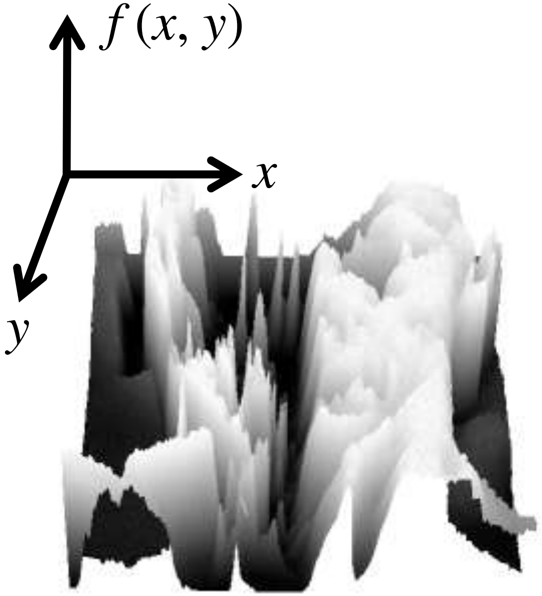
  – *Roland Priemer, Introductory Signal Processing, World Scientific Publishing, 1991*

- The term "signal" includes, among others, audio, video, speech, image, communication, geophysical, sonar, radar, medical and musical signals.

  – *IEEE Trans. on Signal Processing, Aims & Scope*

Image as Function

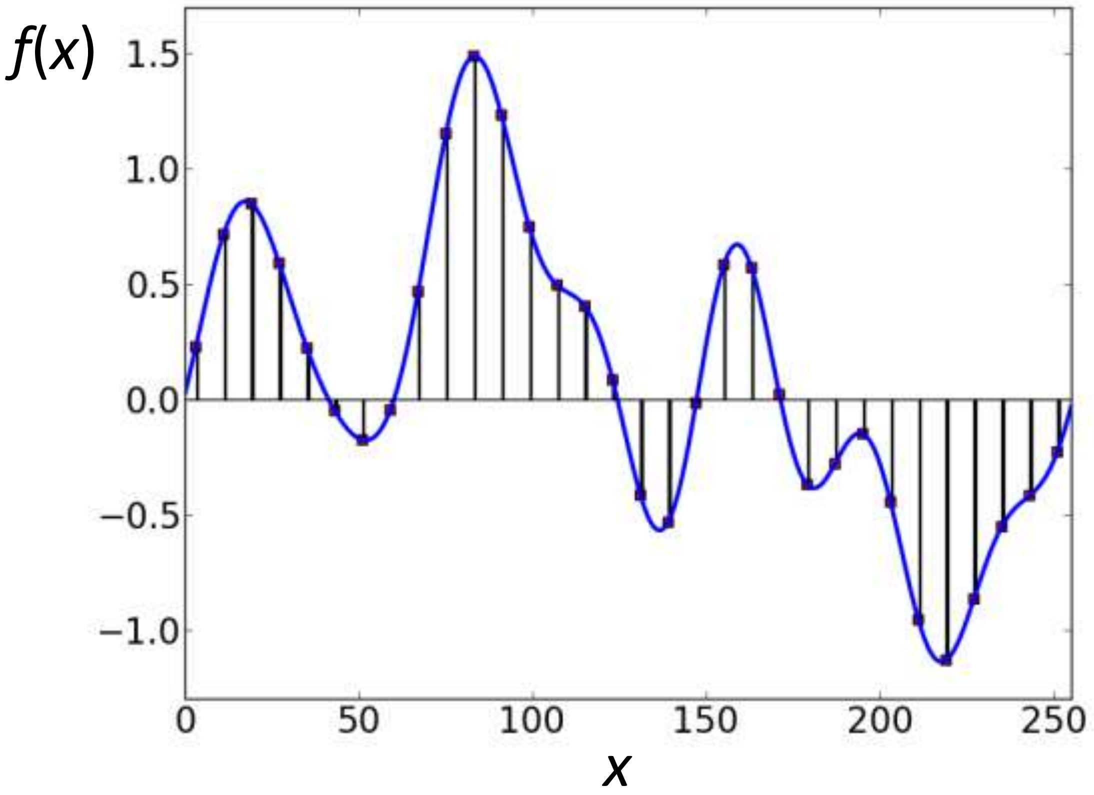
- We can think of a (grayscale) image as a **function** (or 2D *signal*) $f: \mathbb{R}^2 \rightarrow \mathbb{R}$
  - $f(x,y)$ gives the **intensity** at position (x,y)



  - A **digital** image is a discrete (**sampled**, **quantized**) version of this function

Digital Signals: Sampling

- To represent and process a continuous function $f(x)$ in a computer, we reduce it to a finite sequence of values at discrete sample points x_i

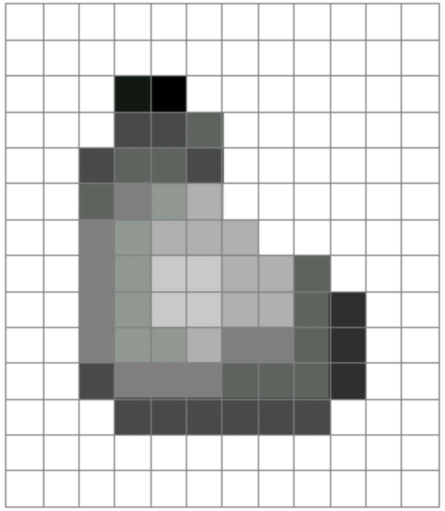


Digital Signals: Quantization

- **Sampling** involves discretization of the
  - Domain of *f*: see previous slide
  - Range of *f*: If *f* is continuous (e.g., real), we round its values to the closest number from some finite set (e.g., `float`, `int`)
    - "Quantization"
    - Introduces error ("noise") into the representation
    - In MRI, other sources of noise are much stronger than quantization
    - We will focus on errors from discretizing the domain

Image as Matrix

- Due to quantization w.r.t. domain and range, digital images can be seen as integer matrices



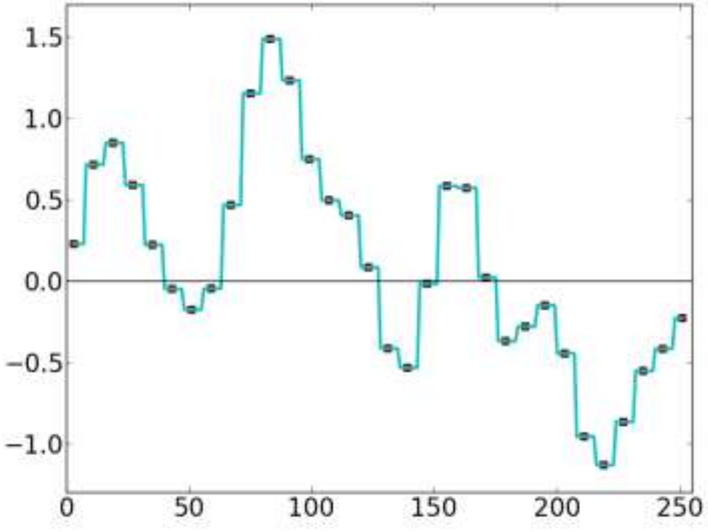
=

255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	20	0	255	255	255	255	255	255	255
255	255	255	75	75	75	255	255	255	255	255	255
255	255	75	95	95	75	255	255	255	255	255	255
255	255	96	127	145	175	255	255	255	255	255	255
255	255	127	145	175	175	175	255	255	255	255	255
255	255	127	145	200	200	175	175	95	255	255	255
255	255	127	145	200	200	175	175	95	47	255	255
255	255	127	145	145	175	127	127	95	47	255	255
255	255	74	127	127	127	95	95	95	47	255	255
255	255	255	74	74	74	74	74	74	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255

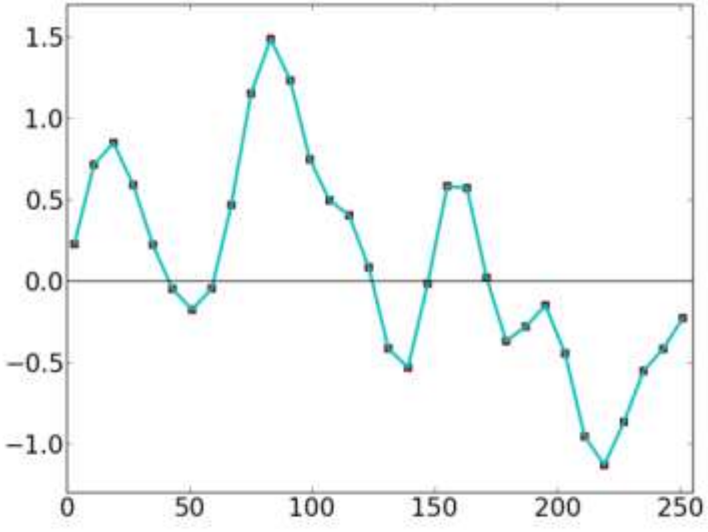
MR scanners often use up to twelve bits per value, range [0,4095]

Digital Signals: Reconstruction

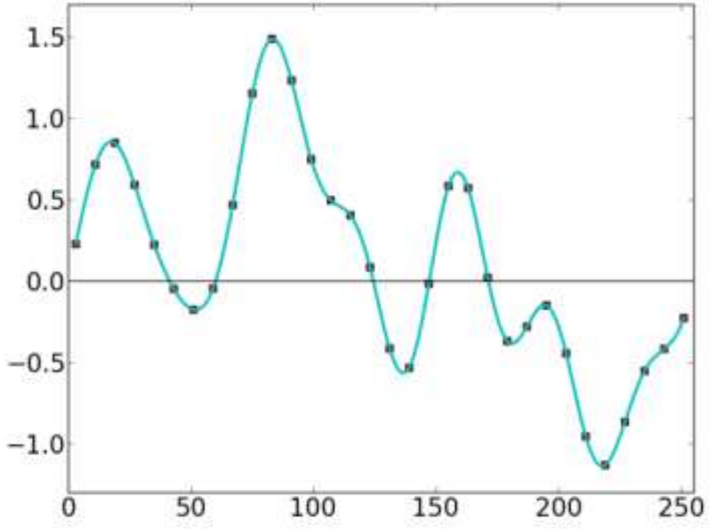
- **Interpolation** is used to reconstruct a function $f(\mathbf{x})$ on $\mathbf{x} \subset \mathbb{R}^n$ from discrete samples $\mathbf{x}_i$
- Some examples of interpolation schemes:



Nearest Neighbor



Linear



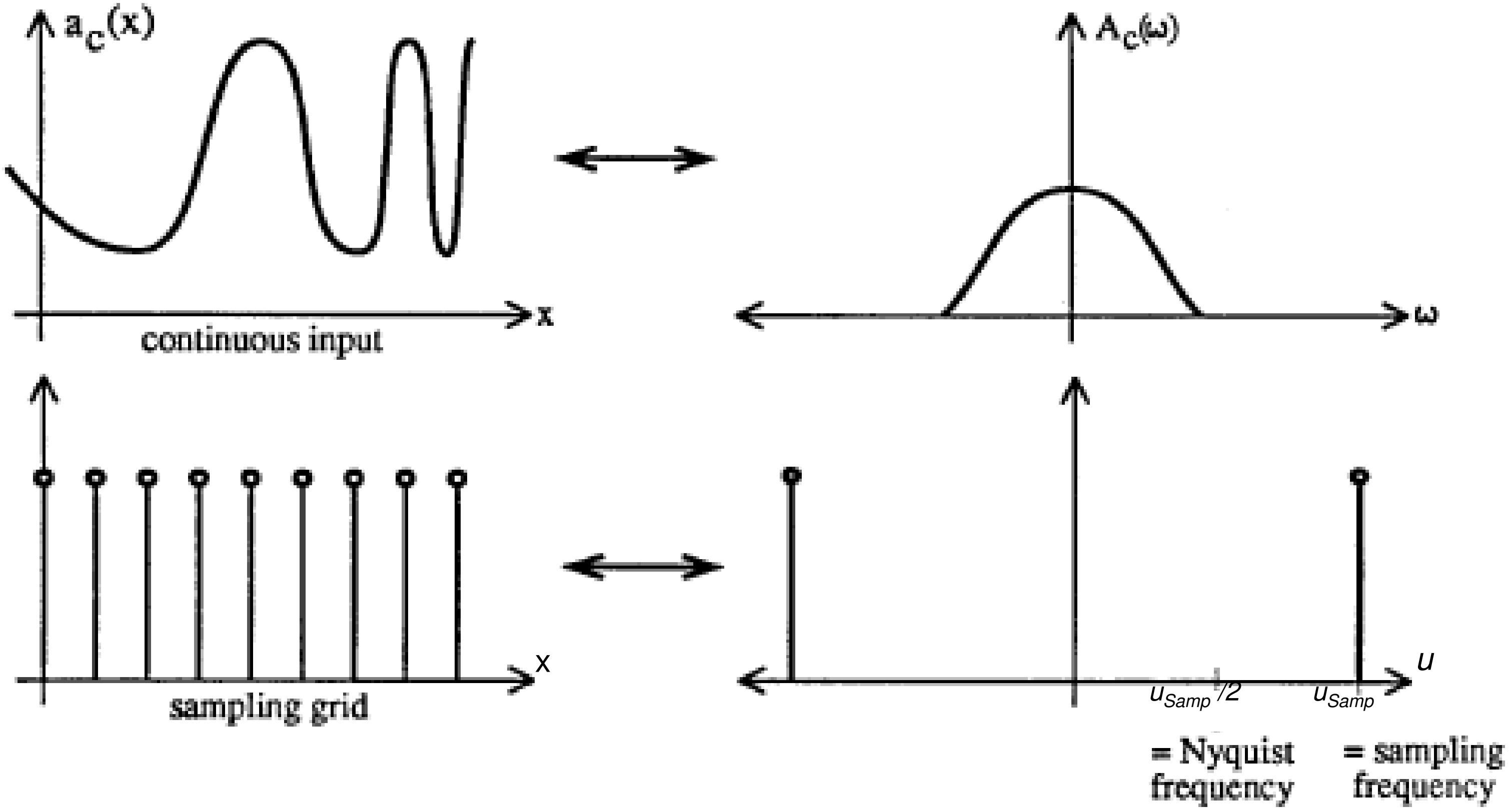
Polynomial Spline

Signal Processing: Fundamental Questions

- **How densely** do we need to sample?
  - What types of artifacts are introduced by not sampling densely enough?
- **Which interpolation scheme** should we use for reconstruction?
  - Often a pragmatic decision (more details in Chapter 2.3)
- Under which conditions can we **recover the original signal exactly?**

A Mathematical Model of Sampling

- Sampling a continuous function *f*(*x*) at regular intervals, $x_k=f(k\cdot\Delta x)$, can be expressed mathematically as multiplication with the comb $Ш_{\Delta x}$
- According to the convolution theorem, this corresponds to convolving the spectrum *F*(*u*) with $\frac{1}{\Delta x} Ш_{\frac{1}{\Delta x}}$.



Convolution with the Comb

1. Convolving $f(x)$ with $\delta(x-a)$ yields $f(x-a)$:

$$f(x) * \delta(x-a) = \int_{-\infty}^{\infty} f(\xi)\,\delta(x-a-\xi)d\xi$$

$\delta(\mathrm{x}) = \delta(-x)$

Basic property of δ (slide 35)

$$= \int_{-\infty}^{\infty} f(\xi)\,\delta\big(\xi-(x-a)\big)d\xi = f(x-a)$$

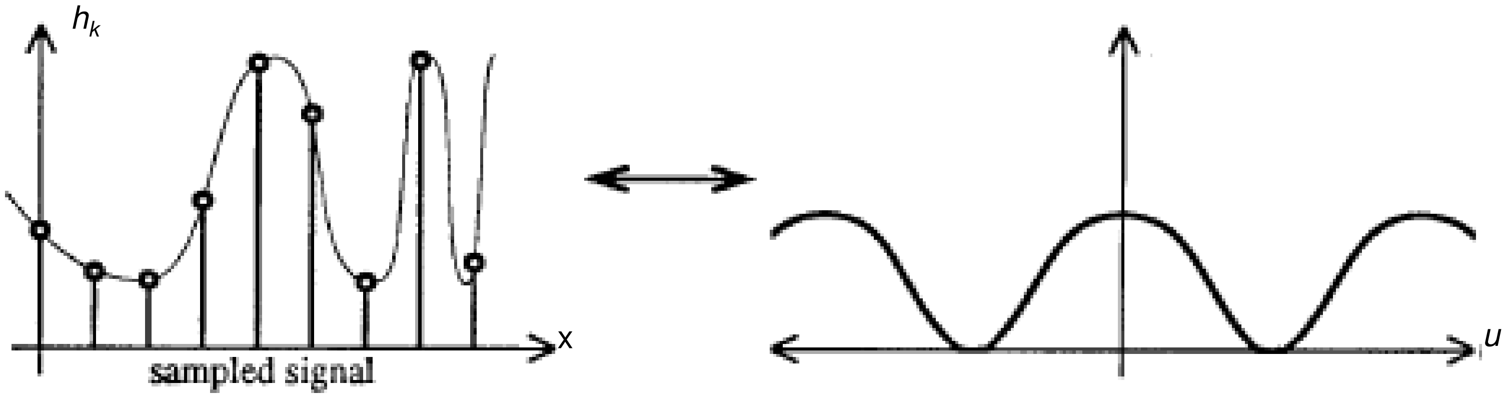
2. Due to linearity of convolution, for $Ш_{\Delta x}$, we obtain

$$f(x) * Ш_{\Delta x} = f(x) * \sum_{k=-\infty}^{\infty} \delta(x-k\Delta x) = \sum_{k=-\infty}^{\infty} f(x-k\Delta x)$$

i.e., a sum of copies of f that are shifted by $k\Delta x$

Aliasing

- Convolution with a comb in frequency space places copies of the spectrum of *f* at distance $\Delta u = 1/\Delta x$ apart



  - **Aliasing:** Frequency components of *f* are duplicated at other frequencies (Latin: *alias,* “elsewhere”)
- If the spectrum of *f* does not contain any frequencies larger than $\Delta u/2$, their copies will not overlap, and the original spectrum can be recovered by multiplying with a box of width Δu in the frequency domain

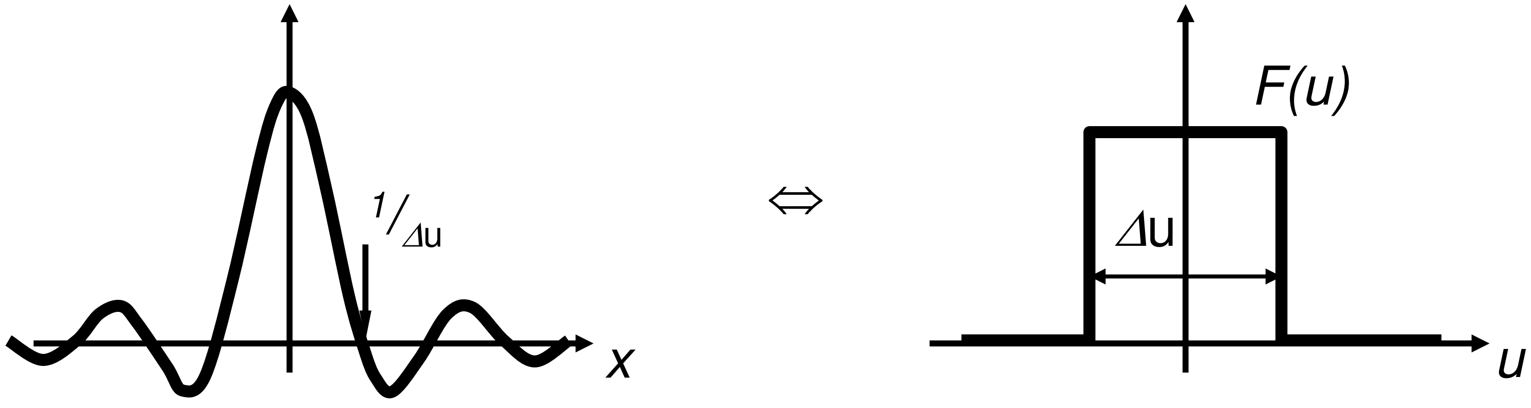
Nyquist Frequency

- If f is band-limited with bandwidth B, sampling it above the **Nyquist frequency** $\omega=2B$ will prevent aliasing.

- Real-world signals not always band limited
  - Pre-filter to remove large frequencies
  - Pre-filtering has to occur *before* sampling

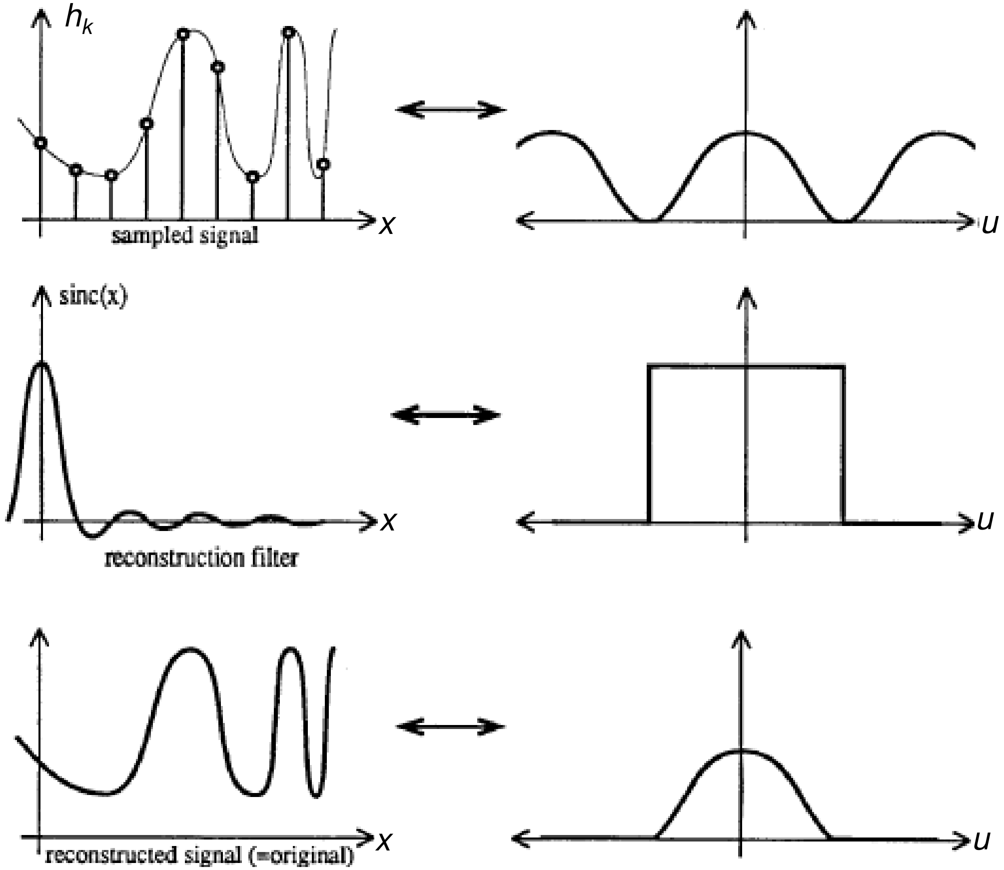
Reconstructing the Signal

- Multiplying with a box in the frequency domain (as required to get rid of duplicate copies of the spectrum) amounts to convolution with sinc function in the spatial domain
  - Reconstructs a continuous signal from samples
  - Provides a natural choice for interpolation kernel



Sampling Theorem

- **Theorem** (Nyquist/Shannon): If $f(x)$ is band-limited and sampled above its Nyquist frequency, it is completely determined by the sequence of sample values $x_k = f(k \cdot \Delta x)$.
  - *In other words:* Sampling and reconstruction reproduce the exact original signal
  - Ignores the effect of quantization
  - Implicitly assumes periodicity for finite sample size

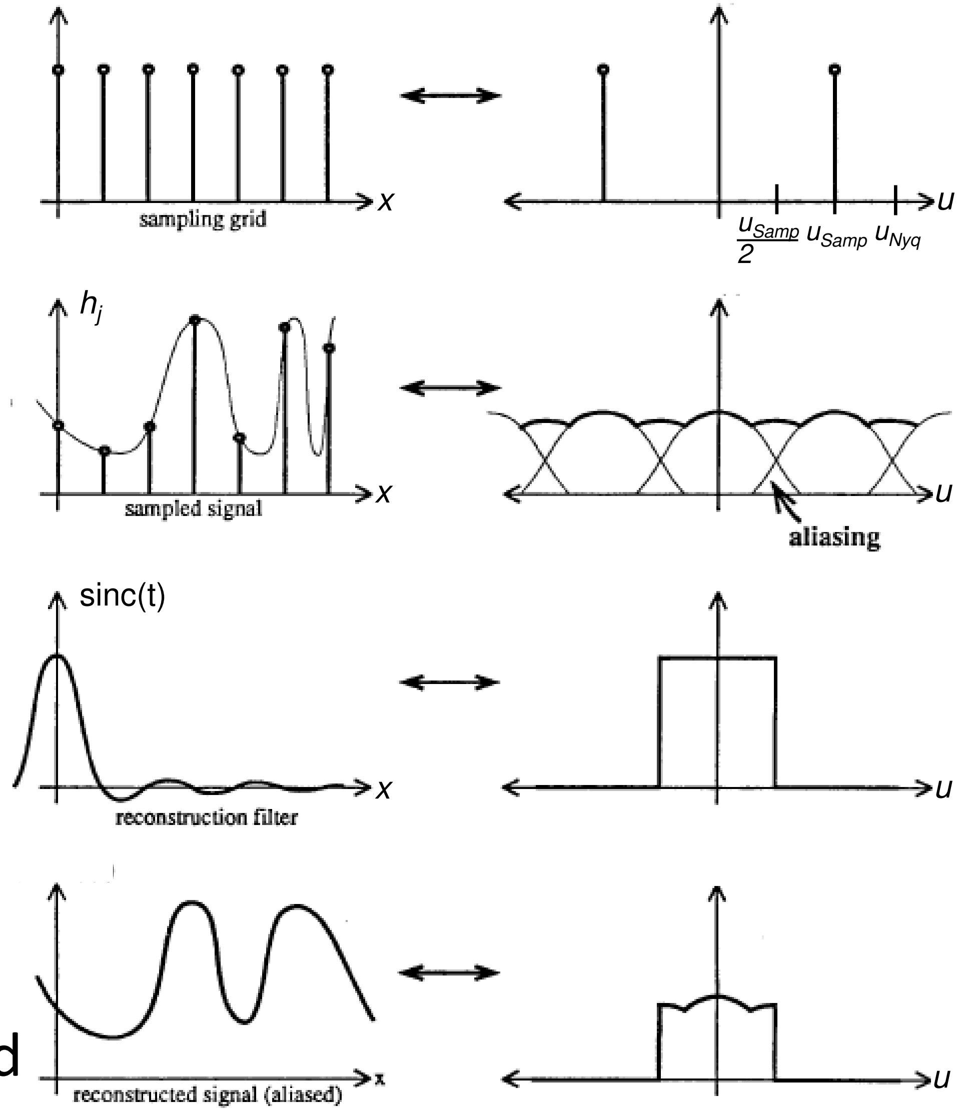


Counter-Example: Undersampling

Illustration:
In an undersampled 1D signal, aliasing prevents exact reproduction

true signal

reconstructed signal



Some Real-World Caveats

- Real measurement devices may be poorly approximated by the comb function
  - Point spread function (PSF) $p(x\text{-}x_0)$ might differ substantially from $\delta(x\text{-}x_0)$
  - Convolution in spatial domain, can try to remove it computationally if PSF is known
- Real measurements are subject to noise
  - Aim to reconstruct an unknown, ideal underlying signal

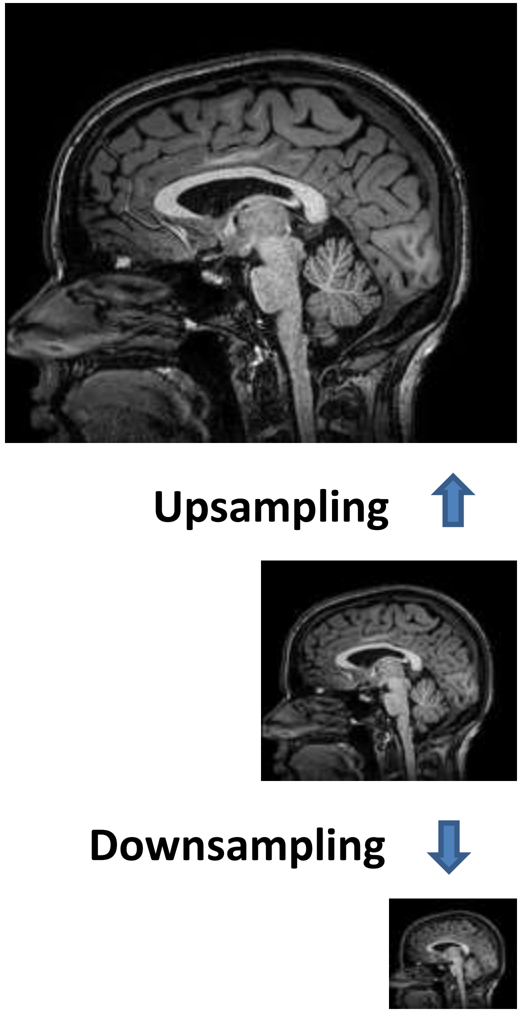
Summary: Signal Processing

- The **sampling theorem** asserts that sampling a continuous signal and convolving the result with a sinc reconstructs the original signal
  – Assumes that original signal was band limited with bandwidth B and sampling used at least the **Nyquist frequency** $\omega=2B$
- Sampling below the Nyquist frequency will lead to **aliasing**. Some MR image artifacts in Chapter 3 will be explained by this.

2.3 Image Resampling

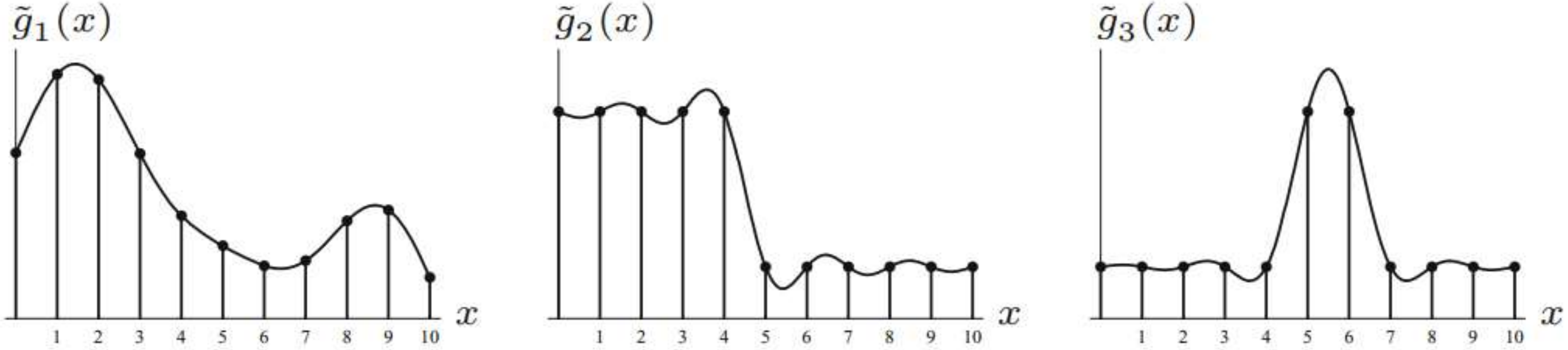
Motivation

- **Upsampling** increases the image resolution
  – For example, for display on a large screen
- **Downsampling** reduces the image resolution
  – For example, to make further processing more efficient
- **Resampling** becomes necessary when applying image transformations such as rotations or (non-integer) shifts
  – For example, as part of image registration



Convolution-Based Interpolation

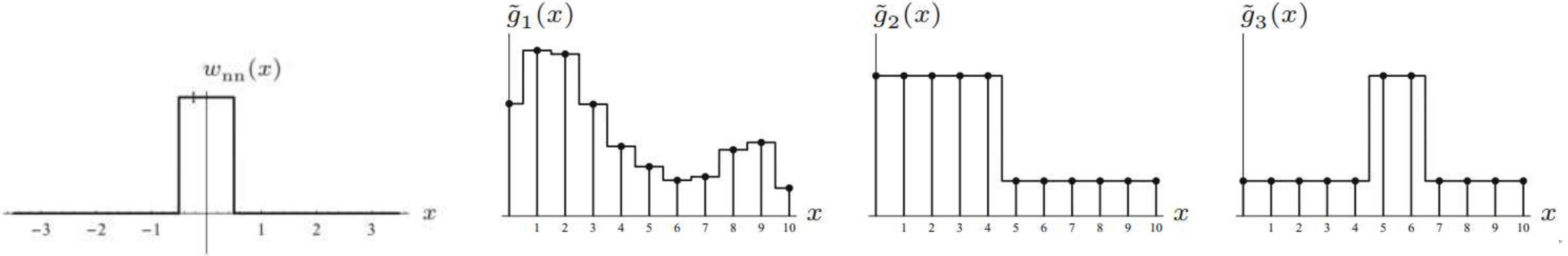
- Many interpolation schemes can be expressed as convolving the discrete signal $(x_i, y_i), i = 1, \dots, n$, described by a weighted sum of Dirac peaks $g(x) = \sum_{i=1}^{n} y_i \delta(x - x_i)$, with a continuous interpolation kernel $w(x)$
  - Sampling theorem suggests $w(x) := \mathrm{sinc}(x)$, but that leads to
    - high computational effort
    - Gibbs ringing around discontinuities



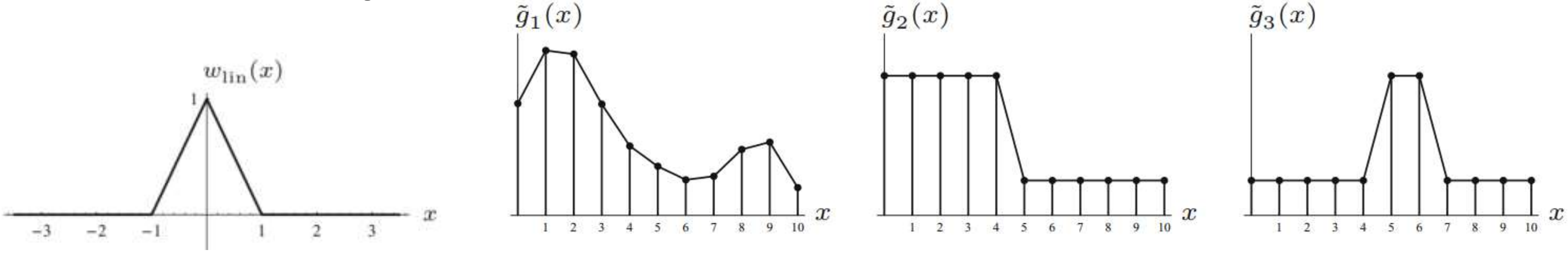
Plots from Burger/Burge, Digital Image Processing, Springer 2016

Nearest Neighbor and Linear Interpolation

- Two computationally efficient alternatives are
  - **nearest neighbor interpolation** (box kernel)



  - **linear interpolation** (tent kernel)



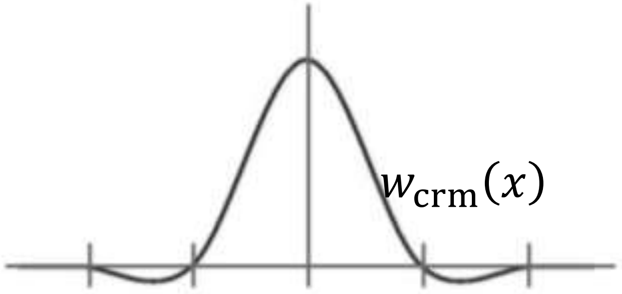
- Handle discontinuities well, but do not produce smooth results

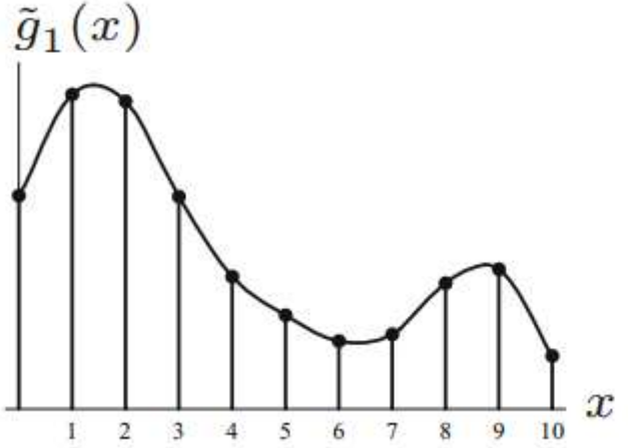
Plots from Burger/Burge, Digital Image Processing, Springer 2016

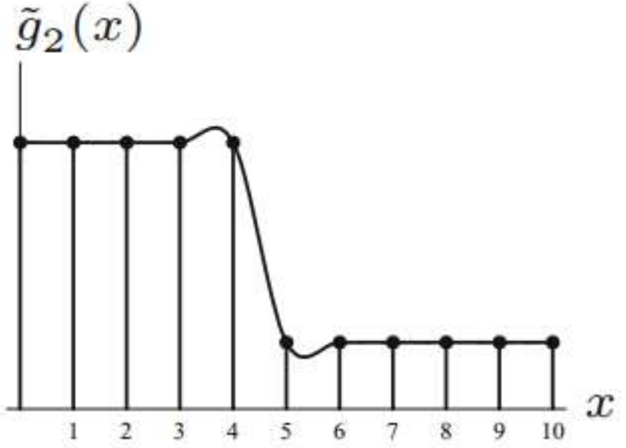
Cubic Spline Interpolation

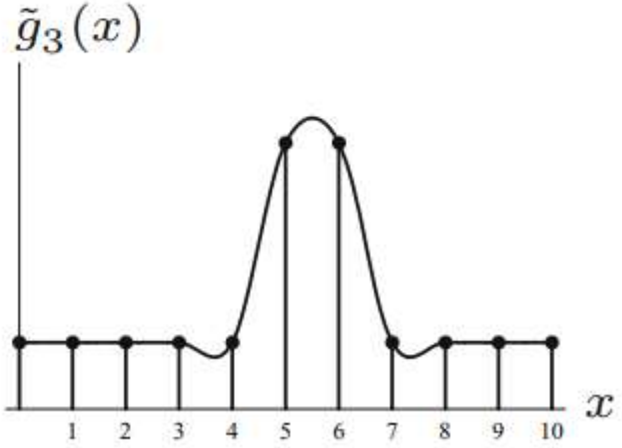
- **Piecewise cubic polynomials** on $(-2,2)$ provide smoothness with limited computational effort (up to four samples)
- **Catmull-Rom spline** is particularly widely used:

$$w_{\mathrm{crm}}(x) = \frac{1}{2}\begin{cases} 3|x|^3 - 5|x|^2 + 2 & \text{for } 0 \le |x| < 1 \\ -|x|^3 + 5|x|^2 - 8|x| + 4 & \text{for } 1 \le |x| < 2 \\ 0 & \text{for } |x| \ge 2 \end{cases}$$





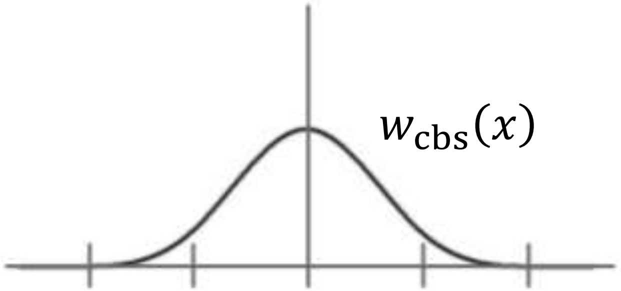




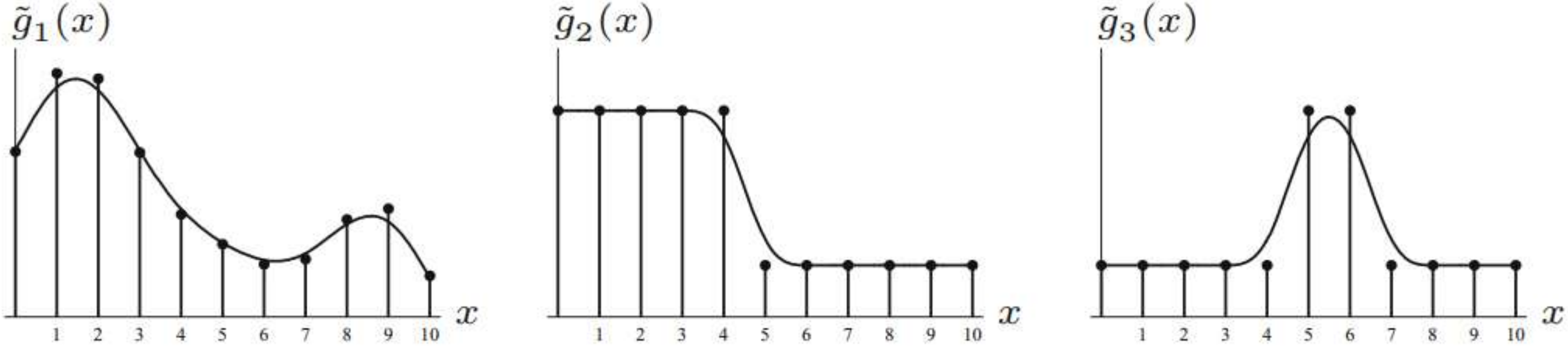
Cubic B-spline Approximation / Interpolation

- A C^2 continuous alternative is the **cubic B-spline**:

$$w_{cbs}(x) = \frac{1}{6}\begin{cases} 3|x|^3 - 6|x|^2 + 4 & \text{for } 0 \le |x| < 1 \\ -|x|^3 + 6|x|^2 - 12|x| + 8 & \text{for } 1 \le |x| < 2 \\ 0 & \text{for } |x| \ge 2 \end{cases}$$



  – No oscillations, no over- or undershoots
  – **Approximates;** use for interpolation requires suitable pre-filtering



Plots from Burger/Burge, Digital Image Processing, Springer 2016

Separable Convolution in 2D/3D

- Interpolation in 2D requires 2D convolution:
  $\mathbf{t}/\boldsymbol{\tau}$ are 2D vectors

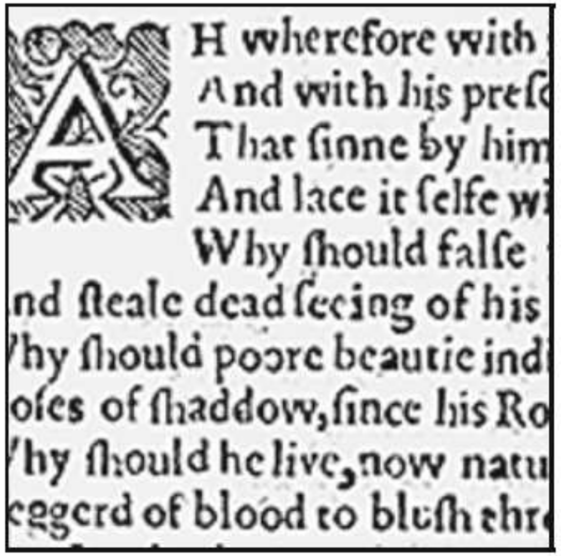
$$(g * h)(\mathbf{t}) = \iint g(\boldsymbol{\tau})h(\mathbf{t} - \boldsymbol{\tau})d\boldsymbol{\tau}$$

- The use of **separable kernels** *h* simplifies this:

$$(g * h)(\mathbf{t}) = \iint g(\tau_1, \tau_2)h_1(t_1 - \tau_1)h_2(t_2 - \tau_2)d\boldsymbol{\tau}$$

- We often use h_1=h_2
- Generalization to 3D straightforward
- 2D/3D interpolation with separable linear/cubic kernels is called “bi/tri-linear/cubic interpolation”

Example: Image Rotation with Different Kernels



Bilinear

Nearest Neighbor

Catmull-Rom

Cubic B-spline Approximation

Images from Burger/Burge, Digital Image Processing, Springer 2016

Downsampling

- A simple way to reduce resolution by factor k is to keep only every kth row and column
  - **Problem**: Violation of Nyquist frequency leads to aliasing
  - **Solutions**: Pooling (averaging over $k \times k$ pixels) or pre-filtering

Original

Restored after naïve reduction with factor 2

Restored after pooling over 2×2 windows

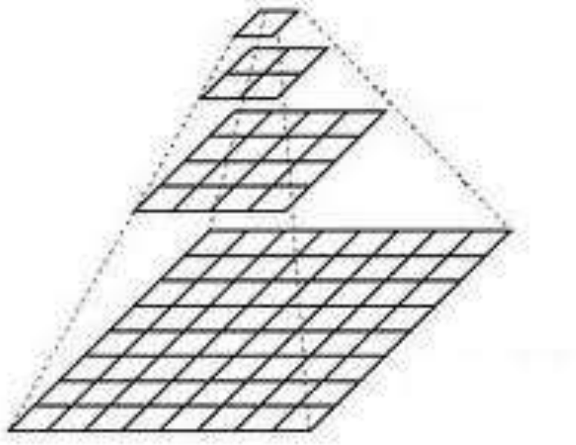
Restored after pre-filtered reduction

Image source: TheFrog001 via Wikimedia commons

Image Pyramids

- Applications that involve structures at different scales often use **image pyramids** as multi-scale representations
  - Obtained by repeated downsampling by factor 2
  - Requires only 33% additional memory:

$$\frac{1}{4}+\frac{1}{16}+\frac{1}{64}+\cdots=\frac{1}{3}$$



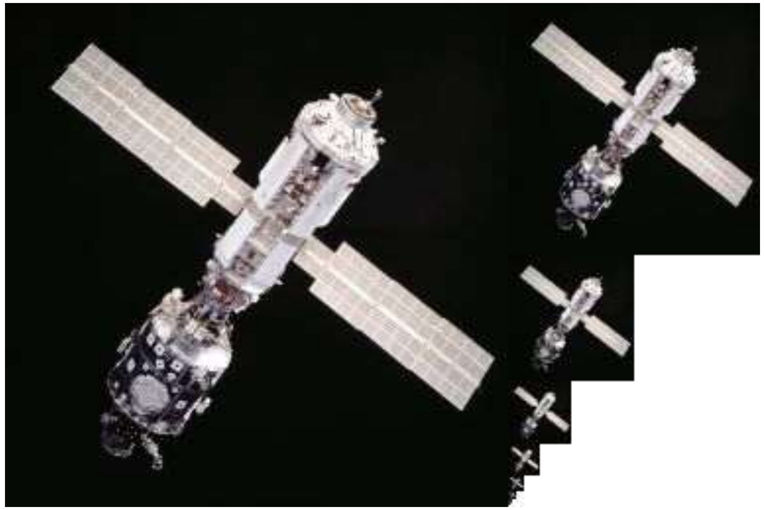


Image from Wikipedia

Summary: Resampling

- **Convolution** is a common framework for all widely used types of image interpolation
  - Different kernels give us large degree of flexibility with respect to
    - smoothness
    - computational effort
    - strength of oscillations
  - **Nearest neighbor** is cheapest, but discontinuous
  - **Linear interpolation** is still quite affordable, but not smooth
  - **Cubic spline interpolation** is a popular trade-off
- **Downsampling** should involve pooling or pre-filtering to prevent aliasing artifacts

Further Reading

- E. Kreyszig: *Advanced Engineering Mathematics.* Wiley, 10th edition, 2011
- W. Burger, M.J. Burge: *Digital Image Processing: An Algorithmic Introduction Using Java.* Springer, 2nd edition, 2016
- W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery: *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 2nd edition, 1992

Exam-like Questions

- In the lecture, we introduced the Dirac comb $Ш_{\Delta x}$ as an infinite series of Dirac deltas with uniform spacing Δx. What is the Fourier Transform of such a Dirac comb? Give an example in which it plays a role in digital signal processing.

- Name two different image interpolation schemes and sketch the corresponding convolution kernels. For each of the two schemes, point out an advantage compared to the other one.