THE UNIVERSITY *of* TEXAS

SCHOOL OF HEALTH INFORMATION SCIENCES AT HOUSTON

Complex Numbers, Convolution, Fourier Transform

For students of HI 6001-125
"Computational Structural Biology"

Willy Wriggers, Ph.D.
School of Health Information Sciences

http://biomachina.org/courses/structures/01.html

Complex Numbers: Review

A complex number is one of the form:

$$a + bi$$

where

$$i = \sqrt{-1}$$

a: real part

b: imaginary part

Complex Arithmetic

When you add two complex numbers, the real and imaginary parts add independently:

$$(a + bi) + (c + di) = (a + c) + (b + d)i$$

When you multiply two complex numbers, you cross-multiply them like you would polynomials:

$$\begin{aligned}(a + bi) \times (c + di) &= ac + a(di) + (bi)c + (bi)(di)\\ &= ac + (ad + bc)i + (bd)(i^2)\\ &= ac + (ad + bc)i - bd\\ &= (ac - bd) + (ad + bc)i\end{aligned}$$

Polynomial Multiplication

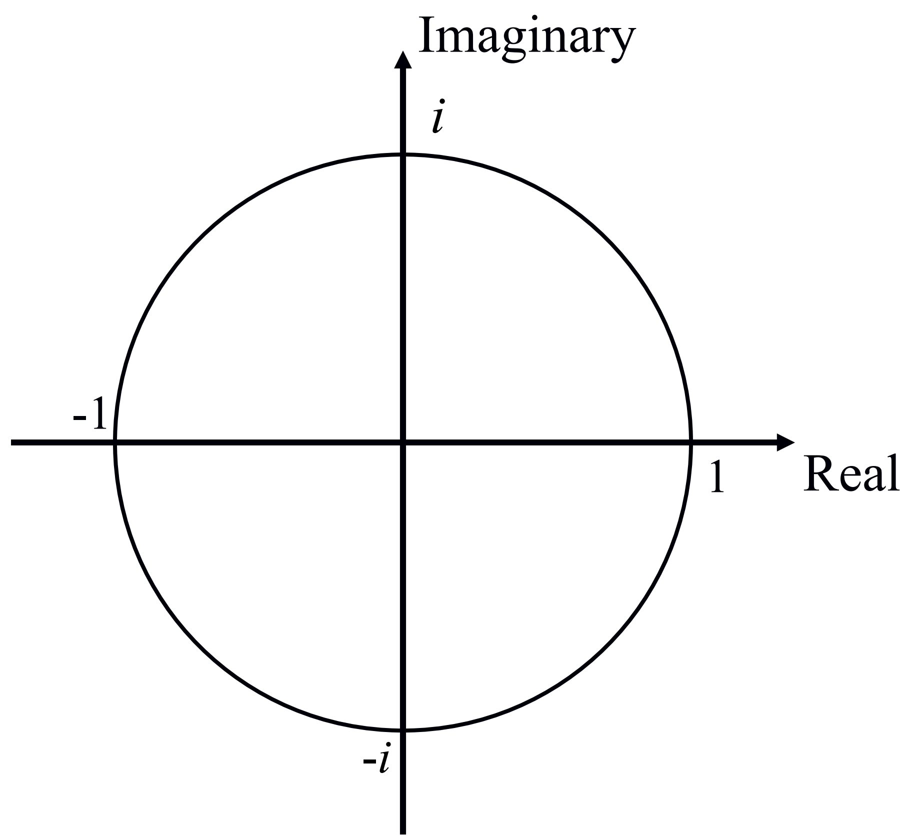
$$p_1(x) \;=\; 3\,x^2 + 2\,x + 4$$

$$p_2(x) \;=\; 2\,x^2 + 5\,x + 1$$

$$p_1(x)\,p_2(x) \;=\; \underline{\qquad}\,x^4 + \underline{\qquad}\,x^3 + \underline{\qquad}\,x^2 + \underline{\qquad}\,x + \underline{\qquad}$$

The Complex Plane

Complex numbers can be thought of as vectors in the complex plane with basis vectors (1, 0) and (0, i):



Magnitude and Phase

The length of a complex number is its *magnitude*:

$$|a + bi| = \sqrt{a^2 + b^2}$$

The angle from the real-number axis is its *phase*:

$$\phi\,(a + bi) = \tan^{-1}(b\,/\,a)$$

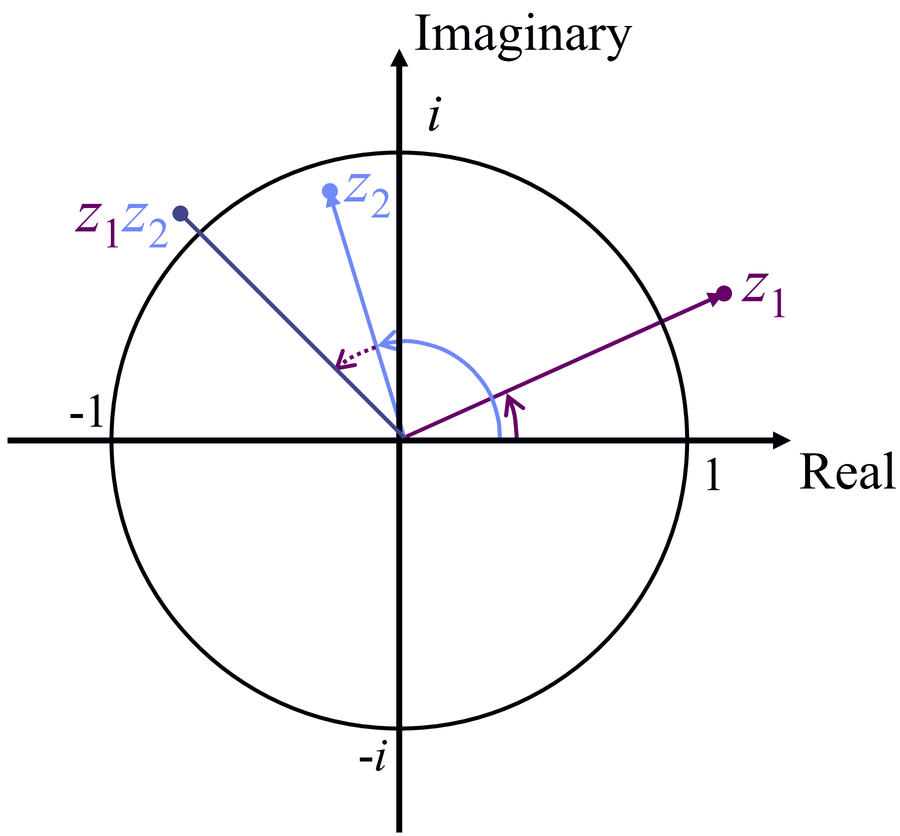
When you multiply two complex numbers, their magnitudes multiply

$$|z_1 z_2| = |z_1||z_2|$$

And their phases add

$$\phi\,(z_1 z_2) = \phi\,(z_1) + \phi\,(z_2)$$

The Complex Plane: Magnitude and Phase



Complex Conjugates

If $z = a + bi$ is a complex number, then its complex conjugate is:

$$z^* = a - bi$$

The complex conjugate z^* has the same magnitude but opposite phase

When you add z to z^*, the imaginary parts cancel and you get a real number:

$$(a + bi) + (a - bi) = 2a$$

When you multiply z to z^*, you get the real number equal to $|z|^2$:

$$(a + bi)(a - bi) = a^2 – (bi)^2 = a^2 + b^2$$

Complex Division

If $z_1 = a + bi,\ z_2 = c + di,\ z = z_1 / z_2,$

the division can be accomplished by multiplying the numerator and denominator by the complex conjugate of the denominator:

$$z = \frac{(a+bi)(c-di)}{(c+di)(c-di)} = \left(\frac{ac+bd}{c^2+d^2}\right) + i\left(\frac{bc-ad}{c^2+d^2}\right)$$

Euler’s Formula

- Remember that under complex multiplication:
  - Magnitudes multiply
  - Phases add
- Under what other quantity/operation does multiplication result in an addition?
  - Exponentiation: $c^a c^b = c^{a+b}$ (for some constant c)
- If we have two numbers of the form $m \cdot c^a$ (where c is some constant), then multiplying we get:

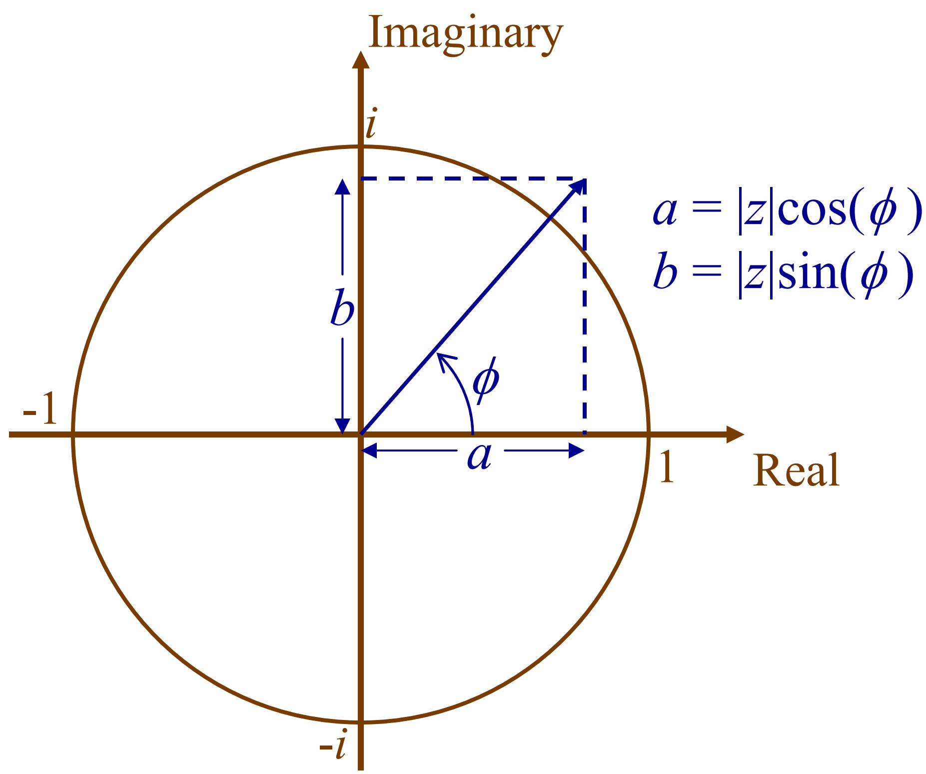
$$(m \cdot c^a)\,(n \cdot c^b) = m \cdot n \cdot c^{a+b}$$

- What constant c can represent complex numbers?

Euler's Formula

- Any complex number can be represented using Euler's formula:

$$z = |z|e^{i\phi(z)} = |z|\cos(\phi) + |z|\sin(\phi)i = a + bi$$



Powers of Complex Numbers

Suppose that we take a complex number

$$z = |z| e^{i\,\phi(z)}$$

and raise it to some power

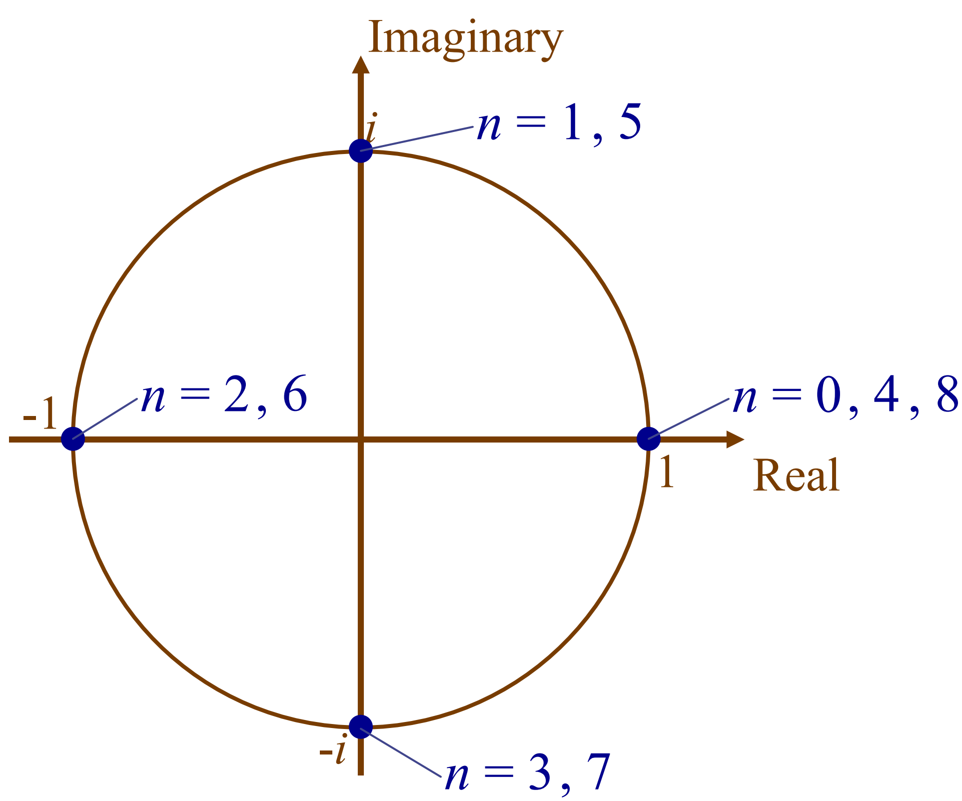
$$z^n = \left[|z| e^{i\,\phi(z)} \right]^n$$

$$= |z|^n \, e^{i\,n\,\phi(z)}$$

z^n has magnitude $|z|^n$ and phase $n\,\phi(z)$

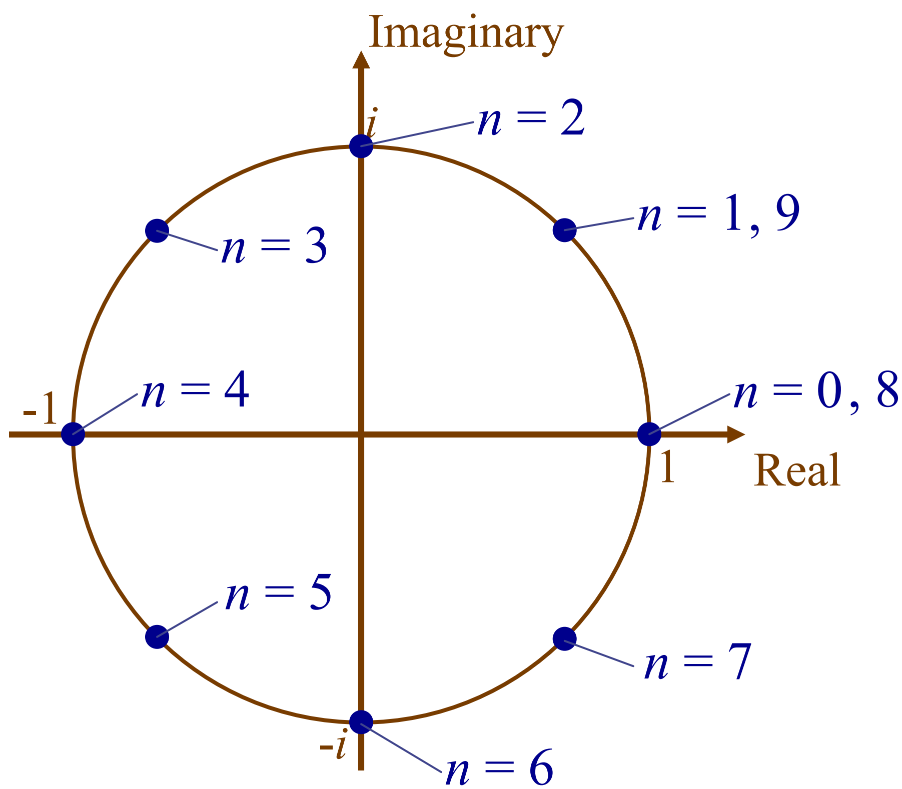
Powers of Complex Numbers: Example

- What is i^n for various n?



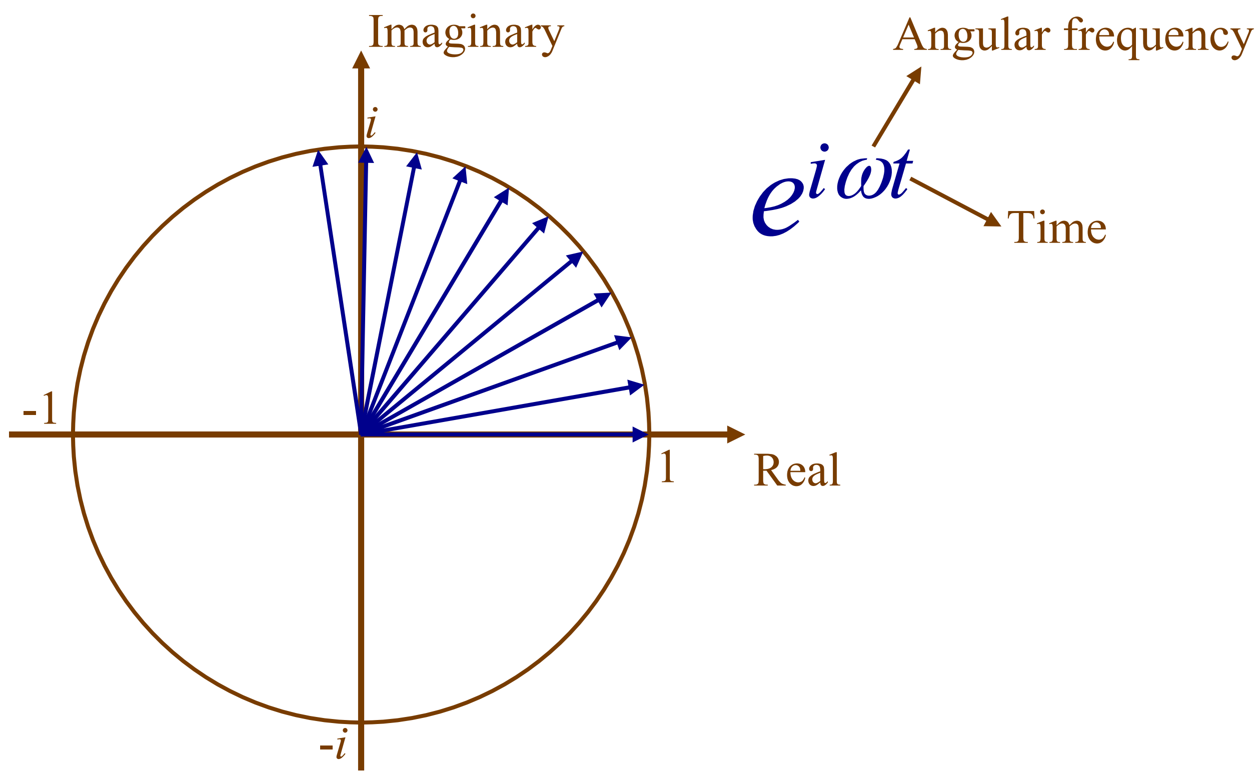
Powers of Complex Numbers: Example

- What is $(e^{i\pi/4})^n$ for various n?



Harmonic Functions

- What does $x(t) = e^{i\omega t}$ look like?
- $x(t)$ is a harmonic function (a building block for later analysis)



Harmonic Functions as Sinusoids

Real Part	Imaginary Part
$\Re(e^{i\omega t})$	$\Im(e^{i\omega t})$
$\cos(\omega t)$	$\sin(\omega t)$

Questions: Complex Numbers

Convolution

Convolution of an input $x(t)$ with the impulse response $h(t)$ is written as

$$x(t) * h(t)$$

That is to say,

$$x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau) h(t - \tau) d\tau$$

Convolution of Discrete Functions

For a discrete function $x[j]$ and impulse response $h[j]$:

$$x[j] * h[j] = \sum_{k} x[k] \cdot h[j-k]$$

One Way to Think of Convolution

$$x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau) h(t - \tau) d\tau$$

$$x[j] * h[j] = \sum_{k} x[k] \cdot h[j - k]$$

Think of it this way:

- Shift a copy of *h* to each position *t* (or discrete position *k*)
- Multiply by the value at that position $x(t)$ (or discrete sample $x[k]$)
- Add shifted, multiplied copies for all *t* (or discrete *k*)

Example: Convolution – One way

$$x[j] = [\ 1 \quad 4 \quad 3 \quad 1 \quad 2\ ]$$

$$h[j] = [\ 1 \quad 2 \quad 3 \quad 4 \quad 5\ ]$$

$$x[0]\ h[j-0] = [\ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_\ ]$$

$$x[1]\ h[j-1] = [\ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_\ ]$$

$$x[2]\ h[j-2] = [\ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_\ ]$$

$$x[3]\ h[j-3] = [\ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_\ ]$$

$$x[4]\ h[j-4] = [\ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_\ ]$$

$$x[j] * h[j] = \sum_k x[k]\ h[j-k]$$

$$= [\ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_ \ \_\_\ ]$$

Example: Convolution – One way

$$x[j] = [\ 1 \quad 4 \quad 3 \quad 1 \quad 2\ ]$$

$$h[j] = [\ 1 \quad 2 \quad 3 \quad 4 \quad 5\ ]$$

$$x[0]\,h[j-0] = [\ 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_\ ]$$

$$x[1]\,h[j-1] = [\ \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_\ ]$$

$$x[2]\,h[j-2] = [\ \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_\ ]$$

$$x[3]\,h[j-3] = [\ \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_\ ]$$

$$x[4]\,h[j-4] = [\ \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_\ ]$$

$$x[j] * h[j] = \sum_k x[k]\,h[j-k]$$

$$= [\ \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_\ ]$$

Example: Convolution – One way

$$x[j] = [\ 1\quad 4\quad 3\quad 1\quad 2\ ]$$

$$h[j] = [\ 1\quad 2\quad 3\quad 4\quad 5\ ]$$

$$x[0]\,h[j-0] = [\ 1\quad 2\quad 3\quad 4\quad 5\quad \_\_\quad \_\_\quad \_\_\quad \_\_\ ]$$

$$x[1]\,h[j-1] = [\ \_\_\quad 4\quad 8\quad 12\quad 16\quad 20\quad \_\_\quad \_\_\quad \_\_\ ]$$

$$x[2]\,h[j-2] = [\ \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\ ]$$

$$x[3]\,h[j-3] = [\ \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\ ]$$

$$x[4]\,h[j-4] = [\ \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\ ]$$

$$x[j] * h[j] = \sum_k x[k]\,h[j-k]$$

$$= [\ \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\ ]$$

Example: Convolution – One way

$$x[j] = [\ 1\quad 4\quad 3\quad 1\quad 2\ ]$$

$$h[j] = [\ 1\quad 2\quad 3\quad 4\quad 5\ ]$$

$$x[0]\, h[j-0] = [\ 1\quad 2\quad 3\quad 4\quad 5\quad \_\_\quad \_\_\quad \_\_\quad \_\_\ ]$$

$$x[1]\, h[j-1] = [\ \_\_\quad 4\quad 8\quad 12\quad 16\quad 20\quad \_\_\quad \_\_\quad \_\_\ ]$$

$$x[2]\, h[j-2] = [\ \_\_\quad \_\_\quad 3\quad 6\quad 9\quad 12\quad 15\quad \_\_\quad \_\_\ ]$$

$$x[3]\, h[j-3] = [\ \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\ ]$$

$$x[4]\, h[j-4] = [\ \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\ ]$$

$$x[j] * h[j] = \sum_k x[k]\, h[j-k]$$

$$= [\ \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\ ]$$

Example: Convolution – One way

$$x[j] = [\ 1 \quad 4 \quad 3 \quad 1 \quad 2\ ]$$

$$h[j] = [\ 1 \quad 2 \quad 3 \quad 4 \quad 5\ ]$$

$$x[0]\, h[j-0] = [\ 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_\ ]$$

$$x[1]\, h[j-1] = [\ \_\_ \quad 4 \quad 8 \quad 12 \quad 16 \quad 20 \quad \_\_ \quad \_\_ \quad \_\_\ ]$$

$$x[2]\, h[j-2] = [\ \_\_ \quad \_\_ \quad 3 \quad 6 \quad 9 \quad 12 \quad 15 \quad \_\_ \quad \_\_\ ]$$

$$x[3]\, h[j-3] = [\ \_\_ \quad \_\_ \quad \_\_ \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad \_\_\ ]$$

$$x[4]\, h[j-4] = [\ \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_\ ]$$

$$x[j] * h[j] = \sum_k x[k]\, h[j-k]$$

$$= [\ \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_\ ]$$

Example: Convolution – One way

$$x[j] = [\ 1\ \ 4\ \ 3\ \ 1\ \ 2\ ]$$

$$h[j] = [\ 1\ \ 2\ \ 3\ \ 4\ \ 5\ ]$$

$$x[0]\,h[j-0] = [\ 1\ \ 2\ \ 3\ \ 4\ \ 5\ \ \_\_\ \ \_\_\ \ \_\_\ \ \_\_\ ]$$

$$x[1]\,h[j-1] = [\ \_\_\ \ 4\ \ 8\ \ 12\ \ 16\ \ 20\ \ \_\_\ \ \_\_\ \ \_\_\ ]$$

$$x[2]\,h[j-2] = [\ \_\_\ \ \_\_\ \ 3\ \ 6\ \ 9\ \ 12\ \ 15\ \ \_\_\ \ \_\_\ ]$$

$$x[3]\,h[j-3] = [\ \_\_\ \ \_\_\ \ \_\_\ \ 1\ \ 2\ \ 3\ \ 4\ \ 5\ \ \_\_\ ]$$

$$x[4]\,h[j-4] = [\ \_\_\ \ \_\_\ \ \_\_\ \ \_\_\ \ 2\ \ 4\ \ 6\ \ 8\ \ 10\ ]$$

$$x[j] * h[j] = \sum_k x[k]\,h[j-k]$$

$$= [\ \_\_\ \ \_\_\ \ \_\_\ \ \_\_\ \ \_\_\ \ \_\_\ \ \_\_\ \ \_\_\ \ \_\_\ ]$$

Example: Convolution – One way

$$x[j] = [\ 1\ \ 4\ \ 3\ \ 1\ \ 2\ ]$$

$$h[j] = [\ 1\ \ 2\ \ 3\ \ 4\ \ 5\ ]$$

$$x[0]\,h[j-0] = [\ 1\ \ 2\ \ 3\ \ 4\ \ 5\ \ \_\_\ \ \_\_\ \ \_\_\ \ \_\_\ ]$$

$$x[1]\,h[j-1] = [\ \_\_\ \ 4\ \ 8\ \ 12\ \ 16\ \ 20\ \ \_\_\ \ \_\_\ \ \_\_\ ]$$

$$x[2]\,h[j-2] = [\ \_\_\ \ \_\_\ \ 3\ \ 6\ \ 9\ \ 12\ \ 15\ \ \_\_\ \ \_\_\ ]$$

$$x[3]\,h[j-3] = [\ \_\_\ \ \_\_\ \ \_\_\ \ 1\ \ 2\ \ 3\ \ 4\ \ 5\ \ \_\_\ ]$$

$$x[4]\,h[j-4] = [\ \_\_\ \ \_\_\ \ \_\_\ \ \_\_\ \ 2\ \ 4\ \ 6\ \ 8\ \ 10\ ]$$

$$x[j] * h[j] = \sum_k x[k]\,h[j-k]$$

$$= [\ 1\ \ 6\ \ 14\ \ 23\ \ 34\ \ 39\ \ 25\ \ 13\ \ 10\ ]$$

Another Way to Look at Convolution

$$x[j]*h[j]=\sum_k x[k]\cdot h[j-k]$$

Think of it this way:

- Flip the function h around zero
- Shift a copy to output position j
- Point-wise multiply for each position k the value of the function x and the flipped and shifted copy of h
- Add for all k and write that value at position j

Convolution in Higher Dimensions

In one dimension:

$$x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau$$

In two dimensions:

$$I(x,y) * h(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(\tau_x,\tau_y)h(x-\tau_x, y-\tau_y)d\tau_x d\tau_y$$

Or, in discrete form:

$$I[x,y] * h[x,y] = \sum_k \sum_j I[j,k]h[x-j, y-k]$$

Example: Two-Dimensional Convolution

$$\begin{matrix} 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \end{matrix} \quad * \quad \begin{matrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{matrix} \quad =$$

____	____	____	____	____	____
____	____	____	____	____	____
____	____	____	____	____	____
____	____	____	____	____	____
____	____	____	____	____	____
____	____	____	____	____	____

see homework assignment!

Properties of Convolution

- Commutative: $f * g = g * f$
- Associative: $f * (g * h) = (f * g) * h$
- Distributive over addition: $f * (g + h) = f * g + f * h$
- Derivative:

$$\frac{d}{dt}(f * g) = f' * g + f * g'$$

Convolution has the same mathematical properties as multiplication

(This is no coincidence)

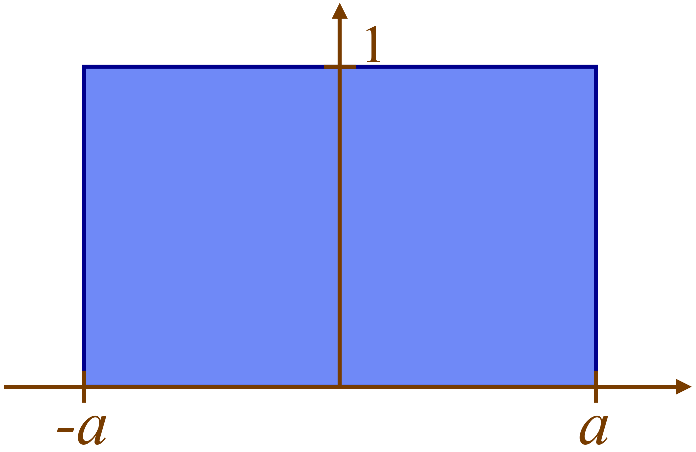
Useful Functions

- Square: $\Pi_a(t)$
- Triangle: $\Lambda_a(t)$
- Gaussian: $G(t, s)$
- Step: $u(t)$
- Impulse/Delta: $\delta(t)$
- Comb (Shah Function): $\mathrm{comb}_h(t)$

Each has their two- or three-dimensional equivalent.

Square

$$\Pi_a(t) = \begin{cases} 1 & \text{if } |t| \le a \\ 0 & \text{otherwise} \end{cases}$$

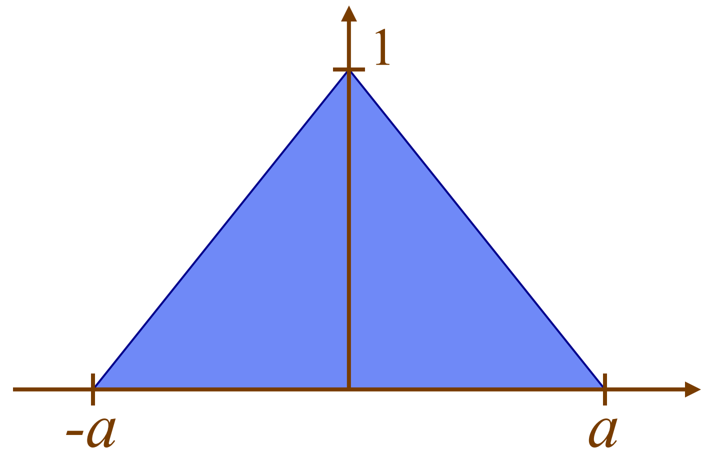


What does $f(t) * \Pi_a(t)$ do to a signal $f(t)$?

What is $\Pi_a(t) * \Pi_a(t)$?

Triangle

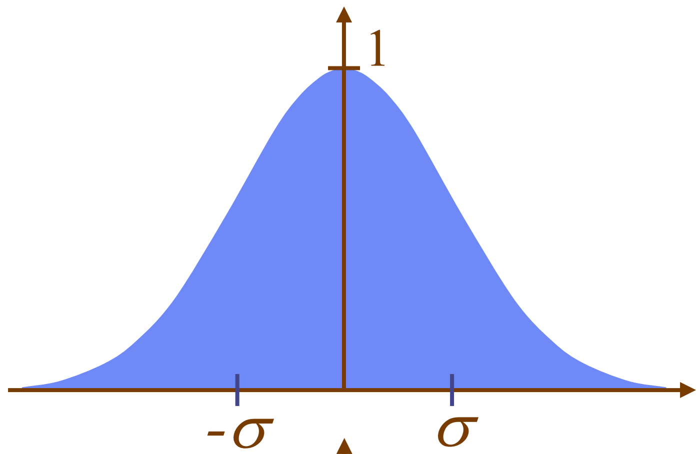
$$\Lambda_a(t) = \begin{cases} 1 - \left| t/a \right| & \text{if } |t| \le a \\ 0 & \text{otherwise} \end{cases}$$



Gaussian

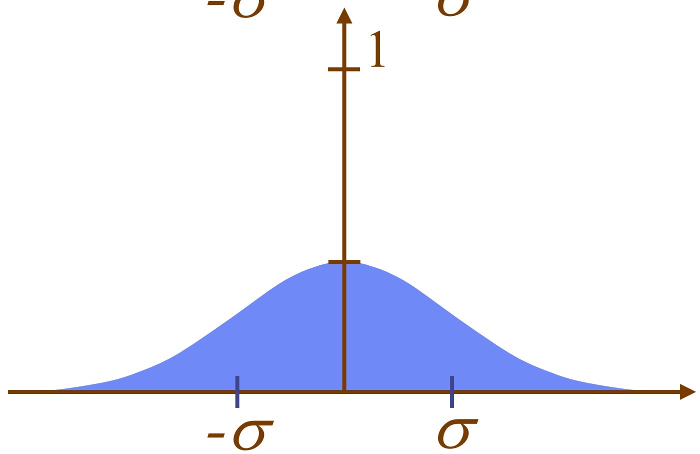
Gaussian: maximum value = 1

$$G(t,\sigma) = e^{-t^2/2\sigma^2}$$



Normalized Gaussian: area = 1

$$G(t,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-t^2/2\sigma^2}$$

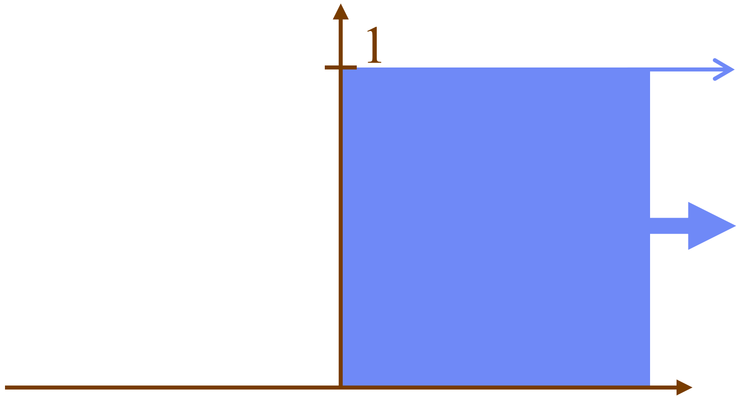


Convolving a Gaussian with another:

$$G(t,\sigma_1) * G(t,\sigma_2) = G(t,\sqrt{\sigma_1^2 + \sigma_2^2})$$

Step Function

$$u(t) = \begin{cases} 1 & \text{if } t \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

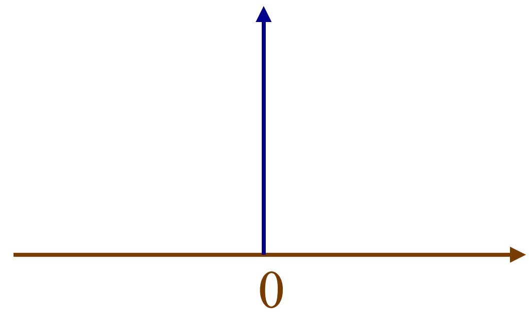


What is the derivative of a step function?

Impulse/Delta Function

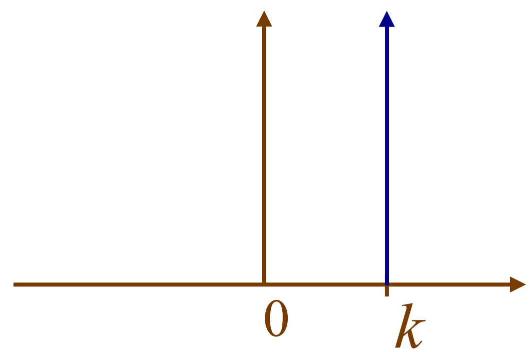
- We've seen the delta function before:

$$\delta(t) = \begin{cases} \infty & \text{if } t = 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \int_{-\infty}^{\infty} \delta(t)\,dt = 1$$



- Shifted Delta function: impulse at t = k

$$\delta(t-k) = \begin{cases} \infty & \text{if } t = k \\ 0 & \text{otherwise} \end{cases}$$



- What is a function $f(t)$ convolved with $\delta(t)$?
- What is a function $f(t)$ convolved with $\delta(t - k)$?

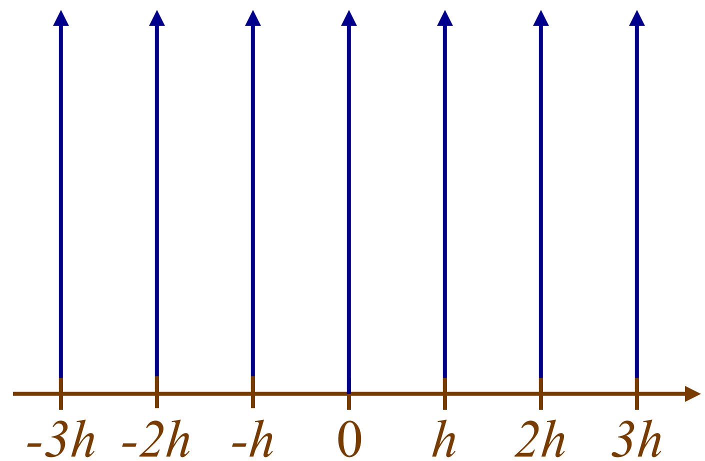
Comb (Shah) Function

A set of equally-spaced impulses: also called an impulse train

$$comb_h(t) = \sum_k \delta(t - hk)$$

h is the spacing

What is $f(t) * \text{comb}_h(t)$?



Convolution Filtering

- Convolution is useful for modeling the behavior of filters
- It is also useful to do ourselves to produce a desired effect
- When we do it ourselves, we get to choose the function that the input will be convolved with
- This function that is convolved with the input is called the *convolution kernel*

Convolution Filtering: Averaging

Can use a square function ("box filter") or Gaussian to locally average the signal/image

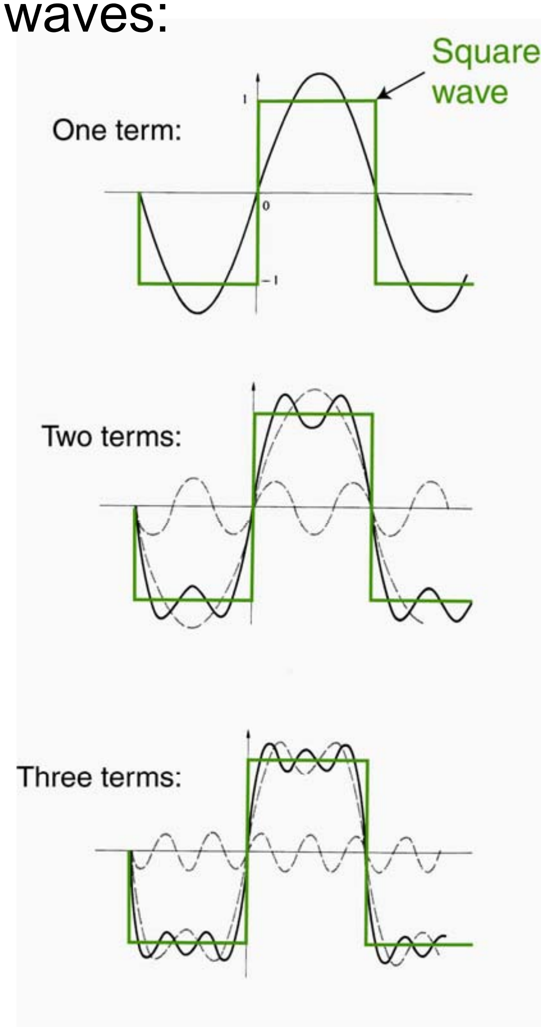
- Square (box) function: uniform averaging
- Gaussian: center-weighted averaging

Both of these blur the signal or image

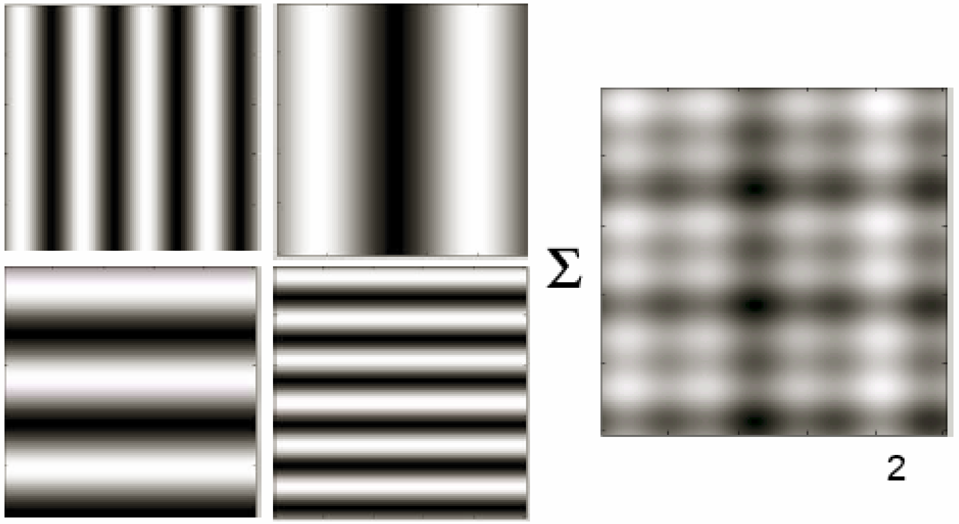
Questions: Convolution

Frequency Analysis

Here, we write a square wave as a sum of sine waves:



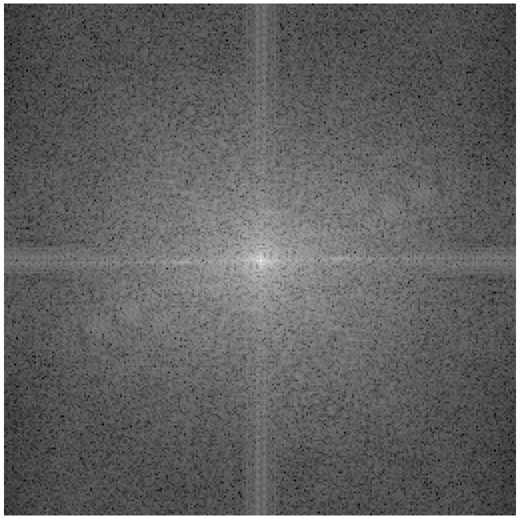
- Fourier Domain
- Signals (1D, 2D, …) decomposed into sum of signals with different frequencies



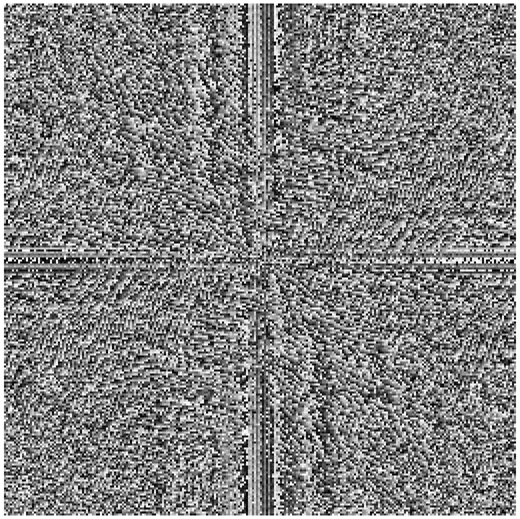
Frequency Analysis

- To use transfer functions, we must first decompose a signal into its component frequencies
- Basic idea: any signal can be written as the sum of phase-shifted sines and cosines of different frequencies
- The mathematical tool for doing this is the ***Fourier Transform***

image



wave magnitudes



wave phases

General Idea of Transforms

Given an orthonormal (orthogonal, unit length) basis set of vectors $\{\bar{e}_k\}$:

Any vector in the space spanned by this basis set can be represented as a weighted sum of those basis vectors:

$$\bar{v} = \sum_k a_k \bar{e}_k$$

To get a vector's weight relative to a particular basis vector $\bar{e}_k$:

$$a_k = \bar{v} \cdot \bar{e}_k$$

Thus, the vector can be transformed into the weights a_k

Likewise, the transformation can be inverted by turning the weights back into the vector

Linear Algebra with Functions

The inner (dot) product of two vectors is the sum of the point-wise multiplication of each component:

$$\overline{u} \cdot \overline{v} = \sum_j \overline{u}[j] \cdot \overline{v}[j]$$

Can't we do the same thing with functions?

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

Functions satisfy all of the linear algebraic requirements of vectors

Transforms with Functions

Just as we transformed vectors, we can also transform functions:

	Vectors $\{\bar{e}_k[j]\}$	Functions $\{e_k(t)\}$
Transform	$a_k = \bar{v} \cdot \bar{e}_k = \sum_j \bar{v}[j] \cdot \bar{e}_k[j]$	$a_k = f \cdot e_k = \int_{-\infty}^{\infty} f(t) e_k^*(t) dt$
Inverse	$\bar{v} = \sum_k a_k \bar{e}_k$	$f(t) = \sum_k a_k e_k(t)$

Basis Set: Generalized Harmonics

The set of generalized harmonics we discussed earlier form an orthonormal basis set for functions:

$$\left\{e^{i2\pi st}\right\}$$

where each harmonic has a different frequency s

Remember:

$$e^{i2\pi st} = \cos(2\pi st) + i\,\sin(2\pi st)$$

The real part is a cosine of frequency s

The imaginary part is a sine of frequency s

The Fourier Series

	All Functions $\{e_k(t)\}$	Harmonics $\{e^{i2\pi st}\}$
Transform	$a_k = f \cdot e_k = \int_{-\infty}^{\infty} f(t) e_k^*(t) dt$	$a_k = f \cdot e^{i2\pi s_k t}$ $= \int_{-\infty}^{\infty} f(t) e^{-i2\pi s_k t} dt$
Inverse	$f(t) = \sum_k a_k e_k(t)$	$f(t) = \sum_k a_k e^{i2\pi s_k t}$

The Fourier Transform

Most tasks need an infinite number of basis functions (frequencies), each with their own weight $F(s)$:

	Fourier Series	Fourier Transform
Transform	$a_k = f \cdot e^{i2\pi s_k t}$ $= \int_{-\infty}^{\infty} f(t) e^{-i2\pi s_k t} dt$	$F(s) = f \cdot e^{i2\pi st}$ $= \int_{-\infty}^{\infty} f(t) e^{-i2\pi st} dt$
Inverse	$f(t) = \sum_k a_k e^{i2\pi s_k t}$	$f(t) = \int_{-\infty}^{\infty} F(s) e^{i2\pi s_k t} ds$

The Fourier Transform

To get the weights (amount of each frequency): $\mathcal{F}$

$$F(s) = \int_{-\infty}^{\infty} f(t) e^{-i2\pi st} dt$$

$F(s)$ **is the Fourier Transform of** $f(t)$: $\mathcal{F}(f(t)) = F(s)$

To convert weights back into a signal (invert the transform):

$$f(t) = \int_{-\infty}^{\infty} F(s) e^{i2\pi st} ds$$

$f(t)$ **is the Inverse Fourier Transform of** $F(s)$: $\mathcal{F}^{-1}(F(s)) = f(t)$

Notation

Let $\mathcal{F}$ denote the Fourier Transform:

$$F = \mathcal{F}(f)$$

Let $\mathcal{F}^{-1}$ denote the Inverse Fourier Transform:

$$f = \mathcal{F}^{-1}(F)$$

How to Interpret the Weights F(s)

The weights $F(s)$ are complex numbers:

Real part	How much of a *cosine* of frequency *s* you need
Imaginary part	How much of a *sine* of frequency *s* you need
Magnitude	How *much* of a sinusoid of frequency *s* you need
Phase	What *phase* that sinusoid needs to be

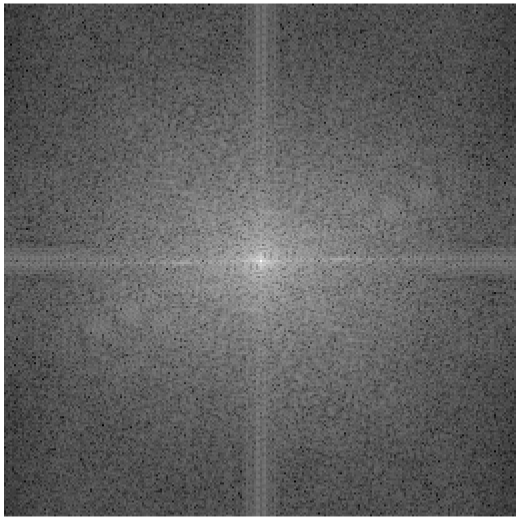
Magnitude and Phase

Remember: complex numbers can be thought of in two ways: (*real*, *imaginary*) or (*magnitude*, *phase*)

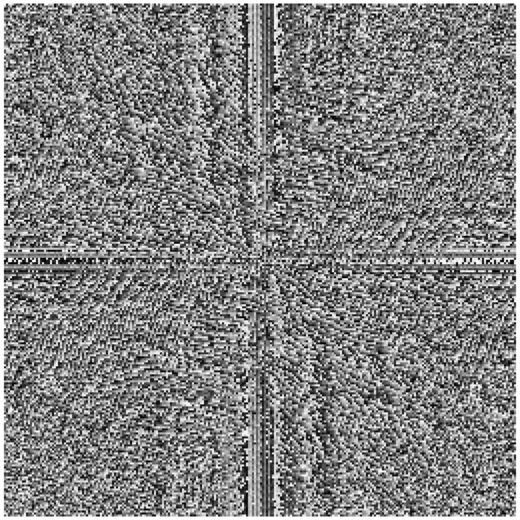
Magnitude: $|F| = \sqrt{\Re(F)^2 + \Im(F)^2}$

Phase: $\phi(F) = \arctan\left(\dfrac{\Re(F)}{\Im(F)}\right)$

image



$|F|$



$\phi(F)$

Periodic Objects on a Grid: Crystals

- Periodic objects with period N:
  - Underlying frequencies must also repeat over the period N
  - Each component frequency must be a multiple of the frequency of the periodic object itself:

$$\frac{1}{N}, \frac{2}{N}, \frac{3}{N}, \cdots$$

- If the signal is discrete:
  - Highest frequency is one unit: period repeats after a single sample
  - No more than N components

$$\frac{1}{N}, \frac{2}{N}, \frac{3}{N}, \cdots \frac{N}{N}$$

Discrete Fourier Transform (DFT)

If we treat a discrete signal with N samples as one period of an infinite periodic signal, then

$$F[s] = \frac{1}{N}\sum_{t=0}^{N-1} f[t]e^{-i2\pi st/N}$$

and

$$f[t] = \sum_{s=0}^{N-1} F[s]e^{i2\pi st/N}$$

Note: For a periodic function, the discrete Fourier transform is the same as the continuous transform

- We give up nothing in going from a continuous to a discrete transform as long as the function is periodic

Normalizing DFTs: Conventions

Basis Function	Transform	Inverse
$e^{i2\pi st/N}$	$F[s]=\frac{1}{N}\sum_{t=0}^{N-1} f[t]e^{-i2\pi st/N}$	$f[t]=\sum_{s=0}^{N-1} F[s]e^{i2\pi st/N}$
$\frac{1}{\sqrt{N}}e^{i2\pi st/N}$	$F[s]=\frac{1}{\sqrt{N}}\sum_{t=0}^{N-1} f[t]e^{-i2\pi st/N}$	$f[t]=\frac{1}{\sqrt{N}}\sum_{s=0}^{N-1} F[s]e^{i2\pi st/N}$
$\frac{1}{N}e^{i2\pi st/N}$	$F[s]=\sum_{t=0}^{N-1} f[t]e^{-i2\pi st/N}$	$f[t]=\frac{1}{N}\sum_{s=0}^{N-1} F[s]e^{i2\pi st/N}$

Discrete Fourier Transform (DFT)

$$F[s] = \frac{1}{N} \sum_{t=0}^{N-1} f[t] e^{-i2\pi st/N}$$

$$f[t] = \sum_{s=0}^{N-1} F[s] e^{i2\pi st/N}$$

Questions:

- What would the code for the discrete Fourier transform look like?
- What would its computational complexity be?

Fast Fourier Transform

developed by Tukey and Cooley in 1965

If we let

$$W_N = e^{-i2\pi/N}$$

the Discrete Fourier Transform can be written

$$F[s] = \frac{1}{N}\sum_{t=0}^{N-1} f[t]\cdot W_N^{st}$$

If N is a multiple of 2, $N = 2M$ for some positive integer M, substituting $2M$ for N gives

$$F[s] = \frac{1}{2M}\sum_{t=0}^{2M-1} f[t]\cdot W_{2M}^{st}$$

Fast Fourier Transform

Separating out the M even and M odd terms,

$$F[s] = \frac{1}{2}\left\{\frac{1}{M}\sum_{t=0}^{M-1} f[2t]\cdot W_{2M}^{s(2t)} + \frac{1}{M}\sum_{t=0}^{M-1} f[2t+1]\cdot W_{2M}^{s(2t+1)}\right\}$$

Notice that

$$W_{2M}^{s(2t)} = e^{-i2\pi s(2t)/2M} = e^{-i2\pi st/M} = W_M^{st}$$

and

$$W_{2M}^{s(2t+1)} = e^{-i2\pi s(2t+1)/2M} = e^{-i2\pi st/M} e^{-i2\pi s/2M} = W_M^{st} W_{2M}^{s}$$

So,

$$F[s] = \frac{1}{2}\left\{\frac{1}{M}\sum_{t=0}^{M-1} f[2t]\cdot W_M^{st} + \frac{1}{M}\sum_{t=0}^{M-1} f[2t+1]\cdot W_M^{st} W_{2M}^{s}\right\}$$

Fast Fourier Transform

$$F[s] = \frac{1}{2}\left\{\frac{1}{M}\sum_{t=0}^{M-1} f[2t]\cdot W_M^{st} + \frac{1}{M}\sum_{t=0}^{M-1} f[2t+1]\cdot W_M^{st} W_{2M}^{s}\right\}$$

Can be written as

$$F[s] = \frac{1}{2}\left\{F_{even}(s) + F_{odd}(s) W_{2M}^{s}\right\}$$

We can use this for the first M terms of the Fourier transform of $2M$ items, then we can re-use these values to compute the last M terms as follows:

$$F[s+M] = \frac{1}{2}\left\{F_{even}(s) - F_{odd}(s) W_{2M}^{s}\right\}$$

Fast Fourier Transform

If *M* is itself a multiple of 2, do it again!

If *N* is a power of 2, recursively subdivide until you have one element, which is its own Fourier Transform

```
ComplexSignal FFT(ComplexSignal f) {
  if (length(f) == 1) return f;

  M = length(f) / 2;
  W_2M = e^(-I * 2 * Pi / M)   // A complex value.

  even = FFT(EvenTerms(f));
  odd  = FFT( OddTerms(f));

  for (s = 0; s < M; s++) {
    result[s  ] = even[s] + W_2M^s * odd[s];
    result[s+M] = even[s] – W_2M^s * odd[s];
  }
}
```

Fast Fourier Transform

Computational Complexity:

Discrete Fourier Transform	→	$O(N^2)$
Fast Fourier Transform	→	$O(N \log N)$

Remember: The FFT is just a faster algorithm for computing the DFT — it does not produce a different result

Fourier Pairs

Use the Fourier Transform, denoted $\mathcal{F}$, to get the weights for each harmonic component in a signal:

$$F(s) = \mathcal{F}(f(t)) = \int_{-\infty}^{\infty} f(t) e^{-i2\pi st} dt$$

And use the Inverse Fourier Transform, denoted $\mathcal{F}^{-1}$, to recombine the weighted harmonics into the original signal:

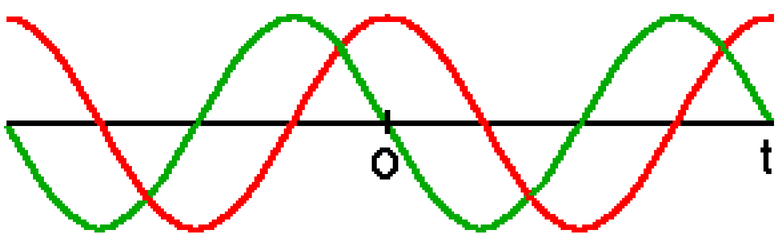
$$f(t) = \mathcal{F}^{-1}(F(s)) = \int_{-\infty}^{\infty} F(s) e^{i2\pi st} ds$$

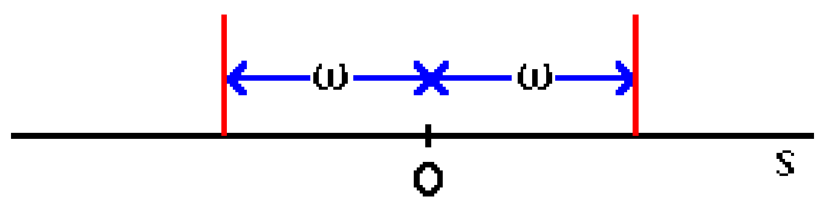
We write a signal and its transform as a Fourier Transform pair:

$$f(t) \leftrightarrow F(s)$$

Sinusoids

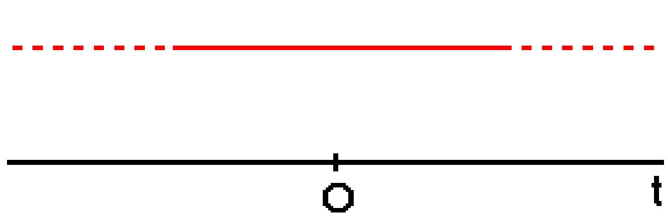
Spatial Domain $f(t)$	Frequency Domain $F(s)$
$\cos(2\pi\omega t)$	$\frac{1}{2}[\delta(s+\omega) + \delta(s-\omega)]$
$\sin(2\pi\omega t)$	$\frac{1}{2}[\delta(s+\omega) - \delta(s-\omega)]i$

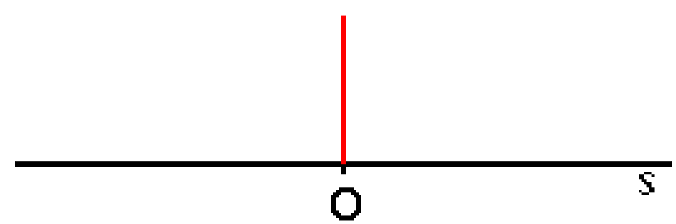




Constant Functions

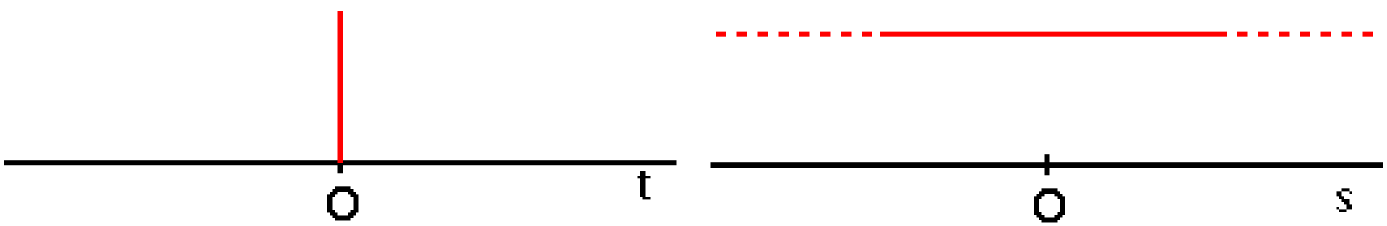
Spatial Domain $f(t)$	Frequency Domain $F(s)$
1	$\delta(s)$
a	$a\,\delta(s)$





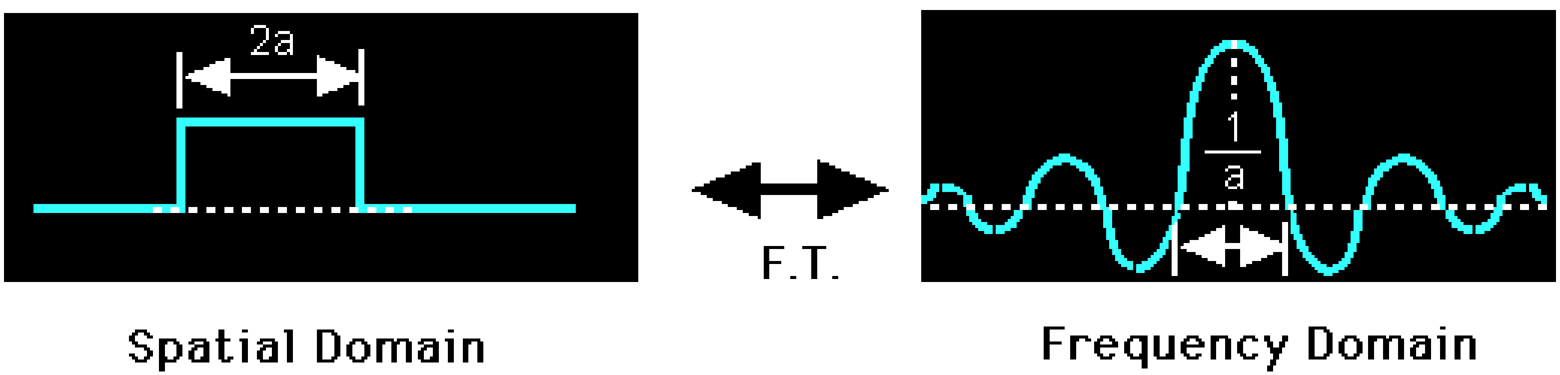
Delta (Impulse) Function

Spatial Domain $f(t)$	Frequency Domain $F(s)$
$\delta(t)$	1



Square Pulse

Spatial Domain	Frequency Domain
$f(t)$	$F(s)$
$\Pi_a(t)$	$2a\ \mathrm{sinc}(2as) = \dfrac{\sin(2\pi as)}{\pi s}$



Adapted from http://www.med.harvard.edu/JPNM/physics/didactics/improc/intro/fourier3.html

Sinc Function

- The Fourier transform of a square function, $\Pi_a(t)$ is the (normalized) sinc function:

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

- To show this, we substitute the value of $\Pi_a(t) = 1$ for $-a < t < a$ into the equation for the continuous FT, i.e.

$$F(s) = \int_{-a}^{a} e^{-i2\pi st}\,dt$$

- We use a substitution. Let u = -i2πst, du = -i2πs dt and then dt = du / -i2πst

$$F(s) = \frac{1}{-i2\pi s}\int_{i2\pi sa}^{-i2\pi sa} e^{u}\,du = \frac{1}{-i2\pi s}\left[e^{-i2\pi as} - e^{i2\pi as}\right] =$$

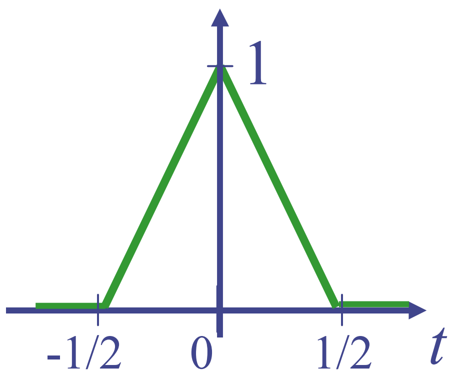
$$\frac{1}{-i2\pi s}\left[\cos(-2\pi as) + i\sin(-2\pi as) - \cos(2\pi as) - i\sin(2\pi as)\right] =$$

$$\frac{1}{-i2\pi s}[-2i\sin(2\pi as)] = \frac{1}{\pi s}\sin(2\pi as) = 2a\ \mathrm{sinc}(2as).$$

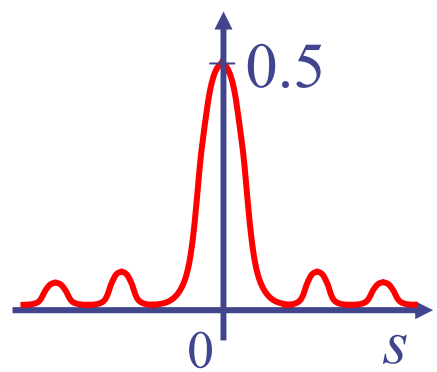
Triangle

Spatial Domain $f(t)$	Frequency Domain $F(s)$
$\Lambda_a(t)$	$a \text{ sinc}^2(as)$

$\Delta_{1/2}(t)$

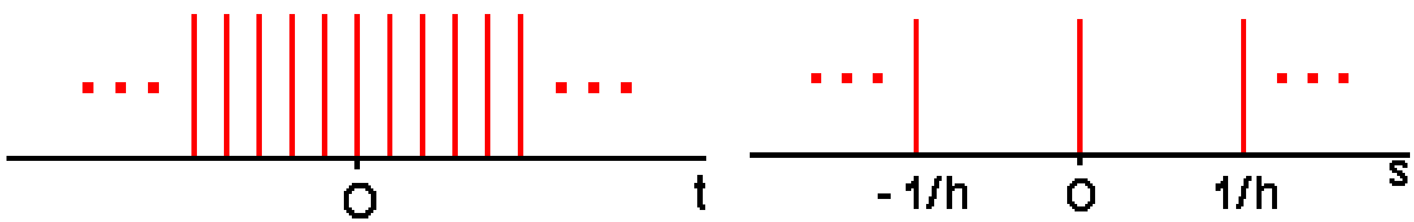


$1/2 \text{ sinc}^2(s/2)$



Comb (Shah) Function

Spatial Domain $f(t)$	Frequency Domain $F(s)$
$\text{comb}_h(t) = \delta(t \bmod h)$	$\delta(t \bmod 1/h)$



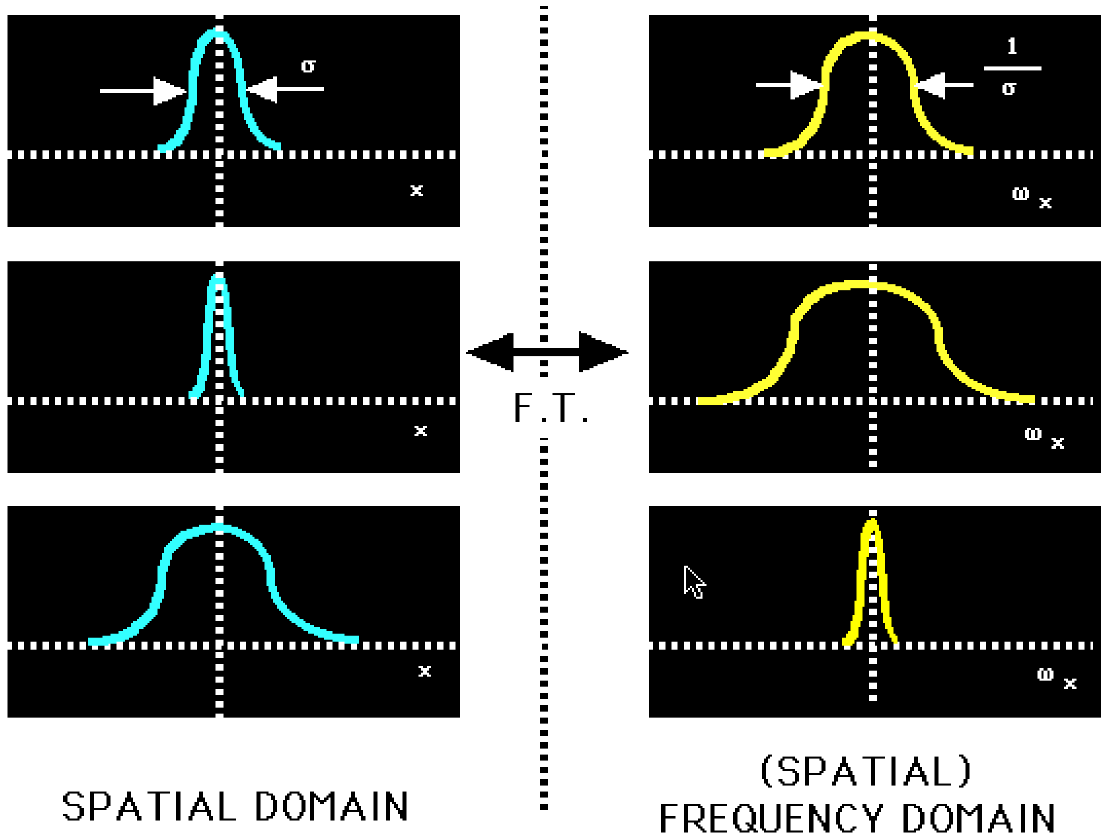
Gaussian

Spatial Domain $f(t)$	Frequency Domain $F(s)$
$e^{-\pi t^2}$	$e^{-\pi s^2}$
$e^{-\pi\left(\frac{t}{\sigma}\right)^2}$	$e^{-\pi(\sigma s)^2}$

see homework assignment!

Graphical Picture

$$e^{-\pi\left(\frac{t}{\sigma}\right)^2} \qquad e^{-\pi(\sigma s)^2}$$



http://www.med.harvard.edu/JPNM/physics/didactics/improc/intro/fourier3.html

Common Fourier Transform Pairs

Spatial Domain: $f(t)$		Frequency Domain: $F(s)$	
Cosine	$\cos(2\pi\omega t)$	Shifted Deltas	$\frac{1}{2}[\delta(s+\omega)+\delta(s-\omega)]$
Sine	$\sin(2\pi\omega t)$	Shifted Deltas	$\frac{1}{2}[\delta(s+\omega) - \delta(s-\omega)]i$
Unit Function	1	Delta Function	$\delta(s)$
Constant	a	Delta Function	$a\,\delta(s)$
Delta Function	$\delta(t)$	Unit Function	1
Comb	$\delta(t \bmod h)$	Comb	$\delta(t \bmod 1/h)$
Square Pulse	$\Pi_a(t)$	Sinc Function	$2a\,\mathrm{sinc}(2as)$
Triangle	$\Lambda_a(t)$	Sinc Squared	$a\,\mathrm{sinc}^2(as)$
Gaussian	$e^{-\pi t^2}$	Gaussian	$e^{-\pi s^2}$

FT Properties: Addition Theorem

Adding two functions together adds their Fourier Transforms:

$$\mathcal{F}(f + g) = \mathcal{F}(f) + \mathcal{F}(g)$$

Multiplying a function by a scalar constant multiplies its Fourier Transform by the same constant:

$$\mathcal{F}(a\, f) = a\, \mathcal{F}(f)$$

Consequence: Fourier Transform is a linear transformation!

FT Properties: Shift Theorem

Translating (shifting) a function leaves the magnitude unchanged and adds a constant to the phase

If

$$f_2(t) = f_1(t - a)$$

$$F_1 = \mathcal{F}(f_1)$$

$$F_2 = \mathcal{F}(f_2)$$

then

$$|F_2| = |F_1|$$

$$\phi(F_2) = \phi(F_1) - 2\pi s a$$

Intuition: magnitude tells you "how much",
phase tells you "where"

FT Properties: Similarity Theorem

Scaling a function's abscissa (domain or horizontal axis) inversely scales the both magnitude and abscissa of the Fourier transform.

If $$f_2(t) = f_1(a\, t)$$

$$F_1 = \mathcal{F}(f_1)$$

$$F_2 = \mathcal{F}(f_2)$$

then

$$F_2(s) = (1/|a|)\, F_1(s\, /\, a)$$

FT Properties: Rayleigh's Theorem

Total sum of squares is the same in either domain:

$$\int_{-\infty}^{\infty} \left| f(t) \right|^2 dt = \int_{-\infty}^{\infty} \left| F(s) \right|^2 ds$$

The Fourier Convolution Theorem

Let F, G, and H denote the Fourier Transforms of signals f, g, and h respectively

$$g = f * h \qquad \text{implies} \qquad G = F\,H$$

$$g = f\,h \qquad \text{implies} \qquad G = F * H$$

Convolution in one domain is multiplication in the other and vice versa

Convolution in the Frequency Domain

One application of the Convolution Theorem is that we can perform time-domain convolution using frequency domain multiplication:

$$f * g = \mathcal{F}^{-1}(\mathcal{F}(f)\,\mathcal{F}(g))$$

How does the computational complexity of doing convolution compare to the forward and inverse Fourier transform?

Deconvolution

If $G = FH$, can't you reverse the process by $F = G / H$?

This is called *deconvolution*: the "undoing" of convolution

Problem: most systems have noise, which limits deconvolution, especially when H is small.

2-D Continuous Fourier Transform

Basic functions are sinusoids with frequency u in one direction times sinusoids with frequency v in the other:

$$F(u,v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) e^{-i2\pi(ux+vy)} dx\, dy$$

Same process for the inverse transform:

$$f(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u,v) e^{i2\pi(ux+vy)} dx\, dy$$

2-D Discrete Fourier Transform

For an $N \times M$ image, the basis functions are:

$$h_{u,v}[x,y] = e^{i2\pi ux/N} e^{i2\pi vy/M}$$

$$= e^{-i2\pi(ux/N + vy/M)}$$

$$F[u,v] = \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f[x,y] e^{-i2\pi(ux/N + vy/M)}$$

Same process for the inverse transform:

$$f[x,y] = \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} F[u,v] e^{i2\pi(ux/N + vy/M)}$$

2D and 3D Fourier Transforms

The point (u, v) in the frequency domain corresponds to the basis function with:

Frequency u in x		Frequency $\|(u, v)\|$
and	OR	in the
Frequency v in y		Direction $\phi(u, v)$

This follows from rotational invariance

Properties

All other properties of 1D FTs apply to 2D and 3D:

- Linearity
- Shift
- Scaling
- Rayleigh's Theorem
- Convolution Theorem

Rotation

Rotating a 2D function rotates it's Fourier Transform

If

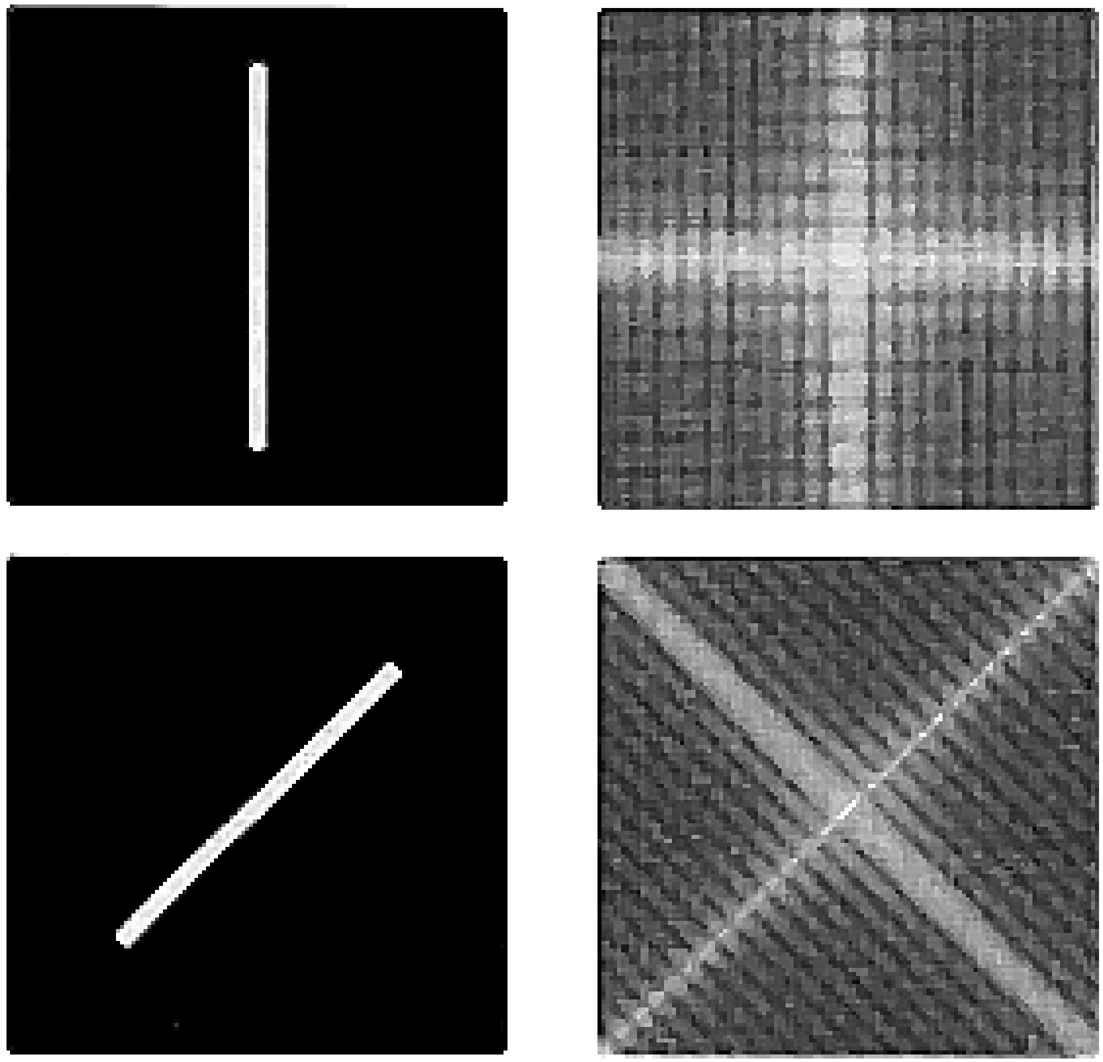
$$\begin{aligned} f_2 &= \text{rotate}_\theta(f_1) \\ &= f_1(x\cos(\theta) - y\sin(\theta),\, x\sin(\theta) + y\cos(\theta)) \\ F_1 &= \mathcal{F}(f_1) \\ F_2 &= \mathcal{F}(f_2) \end{aligned}$$

then

$$F_2(s) = F_1(x\cos(\theta) - y\sin(\theta),\, x\sin(\theta) + y\cos(\theta))$$

i.e., the Fourier Transform is rotationally invariant.

Rotation Invariance (sort of)



needs more boundary padding!

Transforms of Separable Functions

If

$$f(x, y) = f_1(x)\, f_2(y)$$

the function f is separable and its Fourier Transform is also separable:

$$F(u,v) = F_1(u)\, F_2(v)$$

Linear Separability of the 2D FT

The 2D Fourier Transform is linearly separable: the Fourier Transform of a two-dimensional image is the 1D Fourier Transform of the rows followed by the 1D Fourier Transforms of the resulting columns (or vice versa)

$$F[u,v] = \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f[x,y] e^{-i2\pi(ux/N + vy/M)}$$

$$= \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f[x,y] e^{-i2\pi ux/N} e^{-i2\pi vy/M}$$

$$\frac{1}{M} \sum_{y=0}^{M-1} \left[\frac{1}{N} \sum_{x=0}^{N-1} f[x,y] e^{-i2\pi ux/N} \right] e^{-i2\pi vy/M}$$

Likewise for higher dimensions!

Convolution using FFT

Convolution theorem says

$$f * g = \mathcal{F}^{-1}(\mathcal{F}(f)\ \mathcal{F}(g))$$

Can do either:

- Direct Space Convolution
- FFT, multiplication, and inverse FFT

Computational breakeven point: about 9×9 kernel in 2D

Correlation

Convolution is

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

Correlation is

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

Correlation in the Frequency Domain

Convolution

$$f(t) * g(t) \leftrightarrow F(s)\,G(s)$$

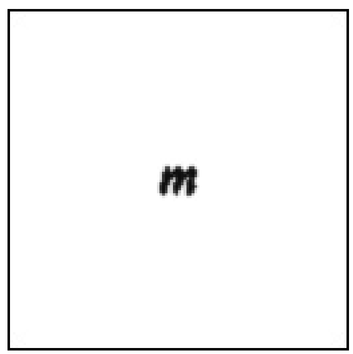
Correlation

$$f(t) * g(-t) \leftrightarrow F(s)\,G^{*}(s)$$

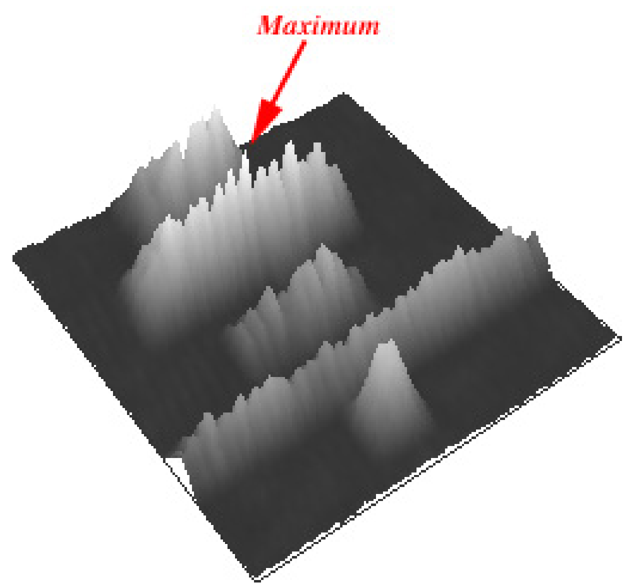
Template “Convolution”

- Actually, is a correlation method
- Goal: maximize correlation between target and probe image
- Here: only translations allowed but rotations also possible

target



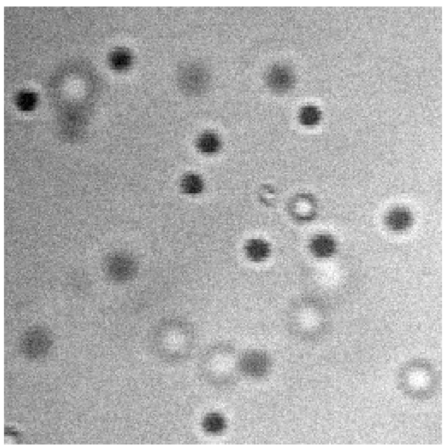
probe



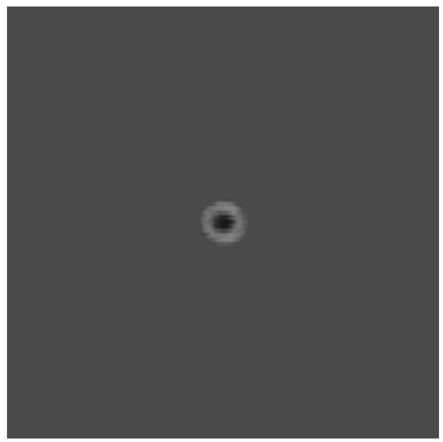
Particle Picking

- Use spherical, or rotationally averaged probes
- Goal: maximize correlation between target and probe image

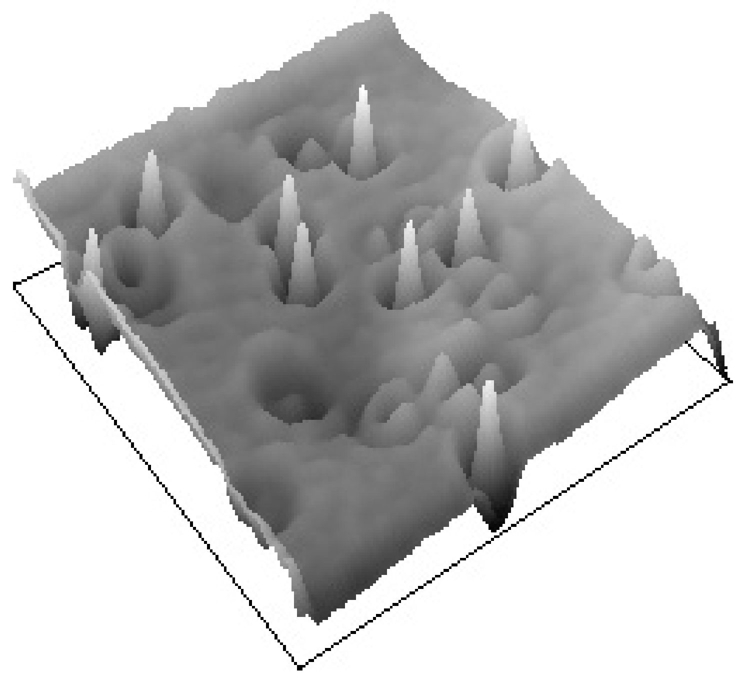
microscope image of latex spheres



target



probe



Autocorrelation

Autocorrelation is the correlation of a function with itself:

$$f(t) * f(-t)$$

Useful to detect self-similarities or repetitions / symmetry within one image!

Power Spectrum

The power spectrum of a signal is the Fourier Transform of its autocorrelation function:

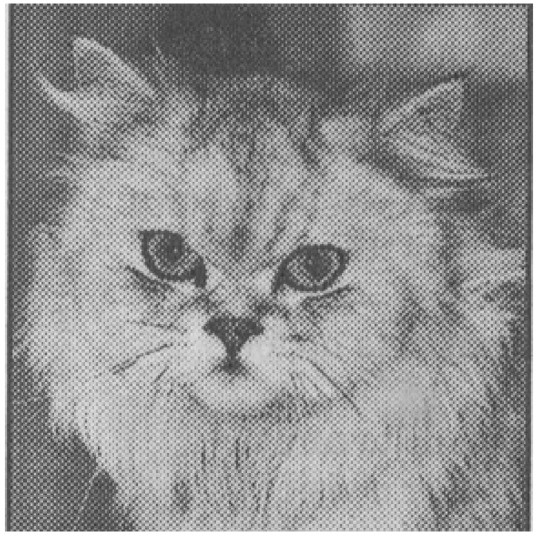
$$
\begin{aligned}
P(s) &= \mathcal{F}(f(t) * f(-t)) \\
&= F(s)\, F^*(s) \\
&= |F(s)|^2
\end{aligned}
$$

It is also the squared magnitude of the Fourier transform of the function

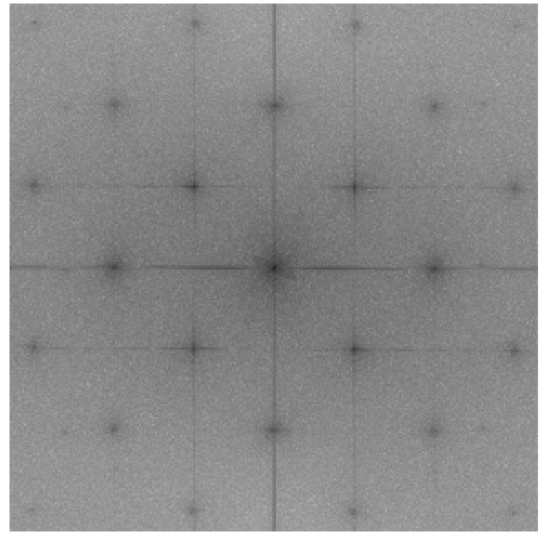
It is entirely real (no imaginary part).

Useful for detecting periodic patterns / texture in the image.

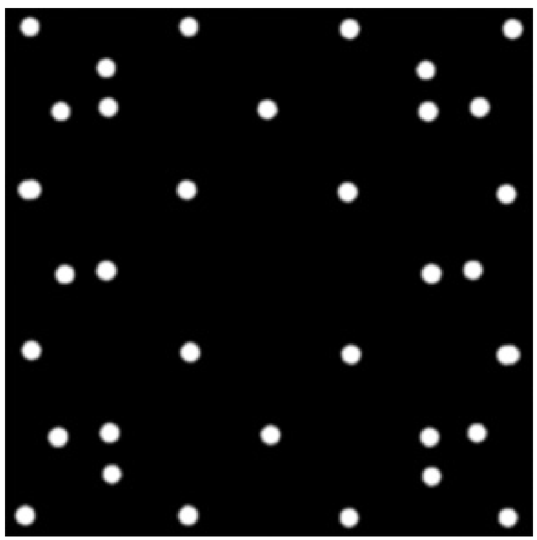
Use of Power Spectrum in Image Filtering



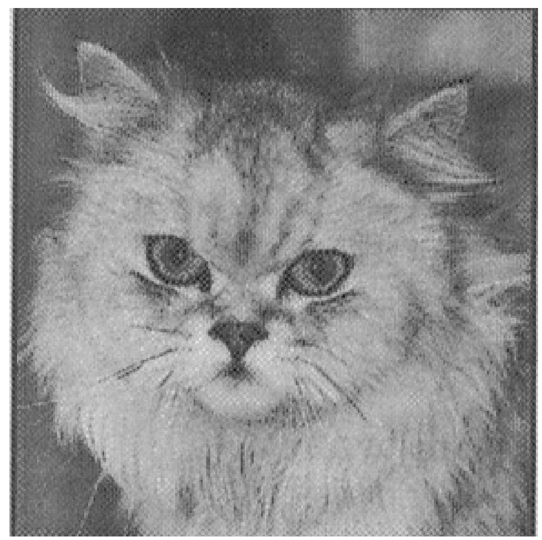
Original with noise patterns



Power spectrum showing noise spikes



Mask to remove periodic noise



Inverse FT with periodic noise removed

Figure and Text Credits

Text and figures for this lecture were adapted in part from the following source, in agreement with the listed copyright statements:

http://web.engr.oregonstate.edu/~enm/cs519

Resources

Textbooks:
Kenneth R. Castleman, Digital Image Processing, Chapters 9,10
John C. Russ, The Image Processing Handbook, Chapter 5