university of groningen

Frequency Domain Filtering

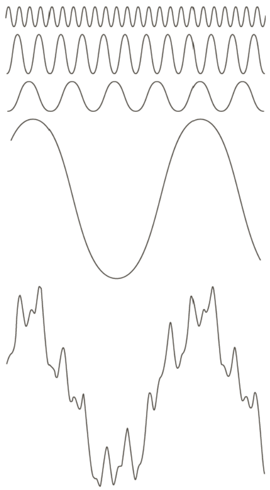
Jean Baptiste Joseph Fourier, 1768 - 1830

Frequency Domain Filtering

SUMMARY:

- Background: Fourier series, Fourier transform
- Convolution Theorem
- Sampling
- Discrete Fourier transform
- Frequency domain filtering

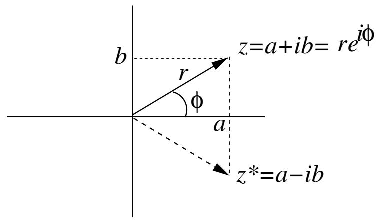
Fourier representation



Representing a signal as a sum of sines and cosines.

Complex numbers

$$\begin{aligned} z &= a + i\,b, \quad i^2 = -1 \\ z^* &= a - i\,b, \quad \text{(conjugate)} \end{aligned}$$



Polar representation:

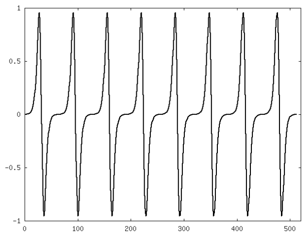
$$z = r\,e^{i\phi}, \qquad r = |z| = \sqrt{a^2+b^2} \text{ (magnitude)}$$

$$\phi = \arctan\left(\frac{b}{a}\right) \text{ (phase)}$$

$$\begin{aligned} e^{ix} &= \cos x + i \sin x \\ \cos x &= (e^{ix} + e^{-ix})/2, \quad \sin x = (e^{ix} - e^{-ix})/2i \\ e^{i2\pi n} &= 1, \quad n \in \mathbb{Z} \end{aligned}$$

Fourier series

A square integrable periodic function f has a Fourier series:

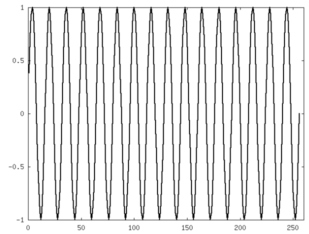


$$\boxed{f(t) = \sum_{n=-\infty}^{\infty} c_n \, e^{i\frac{2\pi n}{T}t}}$$

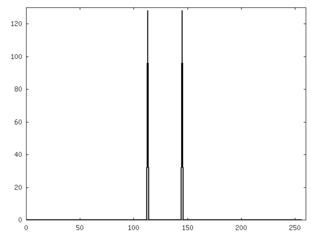
T is the period and the Fourier coefficients are:

$$c_n = \frac{1}{T}\int_{-T/2}^{T/2} f(t)\, e^{-i\frac{2\pi n}{T}t}\, dt, \qquad n = 0, \pm 1, \pm 2, \ldots .$$

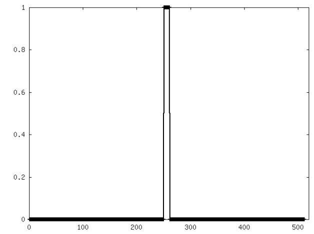
Signals and their Fourier spectra



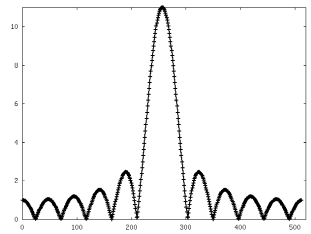
sine



$|c_n|$

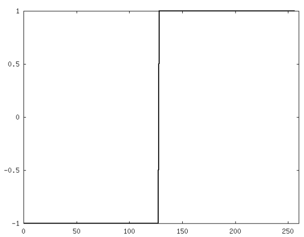


block signal

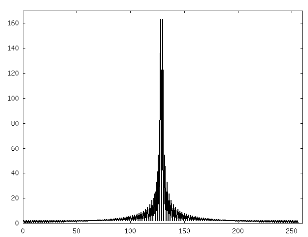


$|c_n|$

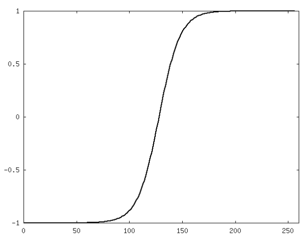
Signals and their Fourier spectra



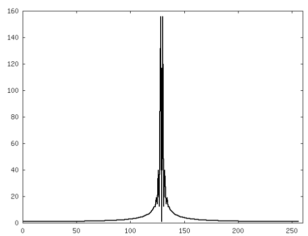
steep step



$|c_n|$



smooth step

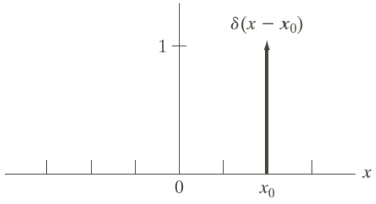


$|c_n|$

Discrete Impulse

Unit discrete impulse function with domain $\mathbb{Z}$ is:

$$\delta(x) = \begin{cases} 1, & x = 0 \\ 0, & x \neq 0. \end{cases}$$



It satisfies $\sum_{x=-\infty}^{\infty} \delta(x) = 1$

Sifting property.

$$\boxed{\sum_{x=-\infty}^{\infty} f(x)\, \delta(x - x_0) = f(x_0)}$$

Sieve with one hole!

Impulse function

- Unit impulse function (delta function) with domain $\mathbb{R}$ is:

$$\delta(t) = \begin{cases} \infty, & t = 0 \\ 0, & t \neq 0. \end{cases}$$

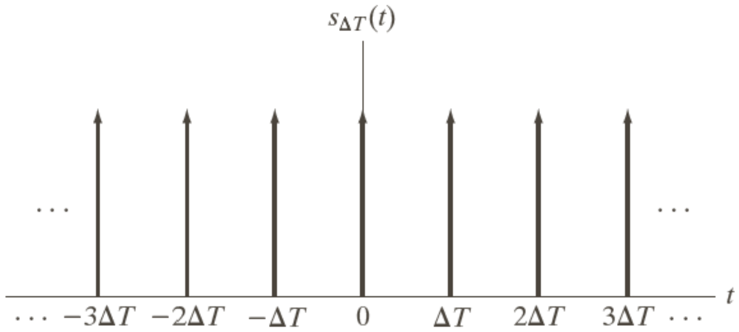
- It satisfies

$$\int_{-\infty}^{\infty} \delta(t)\, dt = 1$$

- Sifting property. For continuous f:

$$\boxed{\int_{-\infty}^{\infty} f(t)\, \delta(t - t_0)\, dt = f(t_0)}$$

Impulse train



Train of impulses (discrete or continuous):

$$s_{\Delta T}(t) = \sum_{-\infty}^{\infty} \delta(t - n\Delta T)$$

Fourier transform

- A square integrable function $f(t)$ on the interval $(-\infty, \infty)$ has a Fourier transform, denoted by $F(\mu)$, defined by

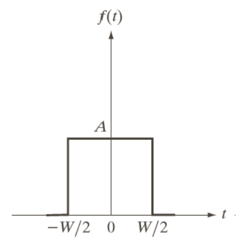
$$\boxed{F(\mu) = \int_{-\infty}^{\infty} f(t)\, e^{-i2\pi\mu t}\, dt}$$

with inverse transform:

$$\boxed{f(t) = \int_{-\infty}^{\infty} F(\mu)\, e^{i2\pi\mu t}\, d\mu}$$

- The quantity $|F(\mu)|$ is called the spectrum of f.

Fourier transform of box signal



$$\begin{aligned} F(\mu) &= \int_{-\infty}^{\infty} f(t)\, e^{-i2\pi\mu t}\, dt = \int_{-W/2}^{W/2} A\, e^{-i2\pi\mu t}\, dt \\ &= \frac{-A}{i2\pi\mu}\big[e^{-i2\pi\mu t}\big]_{-W/2}^{W/2} = \frac{-A}{i2\pi\mu}\big[e^{-i\pi\mu W} - e^{i\pi\mu W}\big] \\ &= \frac{A}{i2\pi\mu}\big[e^{i\pi\mu W} - e^{-i\pi\mu W}\big] = A\, W\, \frac{\sin(\pi\mu W)}{\pi\mu W} \end{aligned}$$

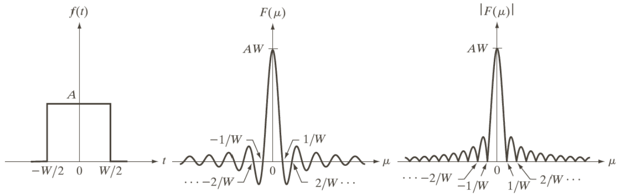
The sinc function

- The sinc-function is defined by

$$\operatorname{sinc}(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x}, & x \neq 0 \\ 1, & x = 0. \end{cases}$$

- So the Fourier transform of the box function on $[-W/2, W/2]$ can be written as:

$$F(\mu) = A\,W\,\frac{\sin(\pi\mu W)}{\pi\mu W} = A\,W\,\operatorname{sinc}(\mu W)$$



Fourier transform of impulse

$$F(\mu) = \int_{-\infty}^{\infty} \delta(t - t_0)\, e^{-i2\pi\mu t}\, dt$$
$$= e^{-i2\pi\mu t_0}$$

- Note that $|F(\mu)| = 1$: the spectrum of an impulse is flat, i.e., it all frequencies are equally represented in it.
- We write

$$\mathcal{F}\{\delta(t - t_0)\} = e^{-i2\pi\mu t_0}$$
$$\mathcal{F}\{e^{i2\pi\mu t_0}\} = \delta(t - t_0)$$

Fourier transform of impulse train

A continuous impulse train $s_{\Delta T}(t)$ is periodic with period ΔT, so has a Fourier series representation:

$$s_{\Delta T}(t) = \sum_{-\infty}^{\infty} \delta(t - n\Delta T) = \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} e^{i\frac{2\pi n}{\Delta T}t}$$

The Fourier transform of $s_{\Delta T}$ is:

$$S(\mu) = \mathcal{F}\{s_{\Delta T}\} = \mathcal{F}\{\frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} e^{i\frac{2\pi n}{\Delta T}t}\}$$
$$= \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} \mathcal{F}\{e^{i\frac{2\pi n}{\Delta T}t}\} = \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} \delta(\mu - \frac{n}{\Delta T})$$

The Fourier transform of an impulse train with period ΔT is an impulse train with period $1/\Delta T$.

Convolution Theorem

- The expression

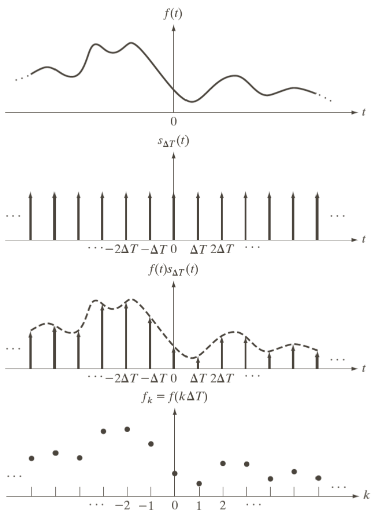
$$(f \star h)(t) = \int_{-\infty}^{\infty} f(\tau)\, h(t - \tau)\, d\tau$$

is called the convolution of the two time signals f and h.
- The convolution theorem says that:

$$\boxed{\mathcal{F}\{f \star h\}(\mu) = F(\mu)\, H(\mu)}$$

- The Fourier transform of the convolution of two signals is the product of the Fourier transforms of the two signals.

Sampling



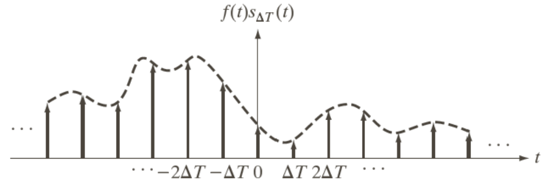
(a) Continuous function.

(b) Impulse train to model the sampling process.

(c) Sampled function as product of (a) and (b).

(d) Sample values obtained by integration of the impulses.

Sampling



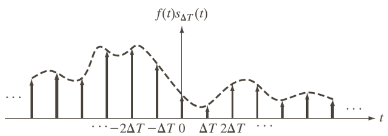
- Sampled function

$$\tilde{f}(t) = f(t)\, s_{\Delta T}(t) = \sum_{n=-\infty}^{\infty} f(t)\, \delta(t - n\Delta T)$$

- Obtain the value of the n^{th} sample by integration:

$$f_n = \int_{-\infty}^{\infty} f(t)\, \delta(t - n\Delta T)\, dt = f(n\Delta T)$$

Fourier transform of sampled function

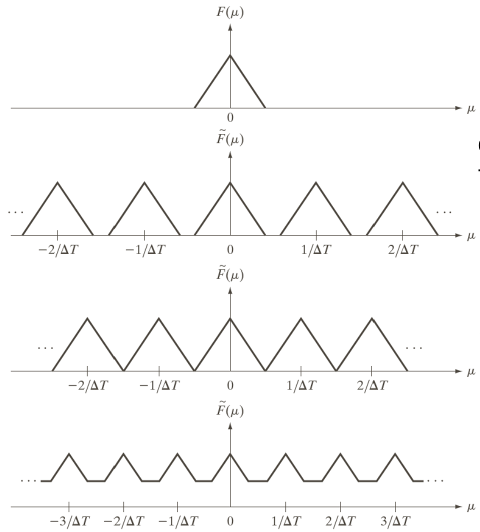


- Fourier transform of $\tilde{f}$:

$$\tilde{F}(\mu) = \mathfrak{F}\{f(t)\, s_{\Delta T}(t)\} = (F \star S)(\mu) = \cdots$$
$$= \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} F(\mu - \frac{n}{\Delta T})$$

- The Fourier transform of the sampled function is an infinite, periodic sequence of copies of the Fourier transform $F(\mu)$ of the original continuous function.

Fourier transform of sampled bandlimited function



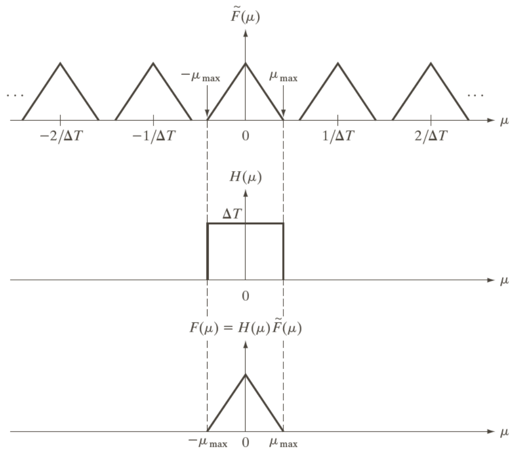
(a) Fourier transform of continuous bandlimited function.

(b) Over-sampling.

(c) Critical sampling.

(d) Under-sampling.

Signal reconstruction



(a) Fourier transform $\tilde{F}(\mu)$ of sampled bandlimited function.

(b) Box filter $H(\mu)$.

(c) Extracting one period (product of $\tilde{F}(\mu)$ and $H(\mu)$).

Extracting one period of the Fourier transform of a sampled bandlimited function using an ideal lowpass filter allows to recover the original continuous signal.

Shannon's sampling theorem

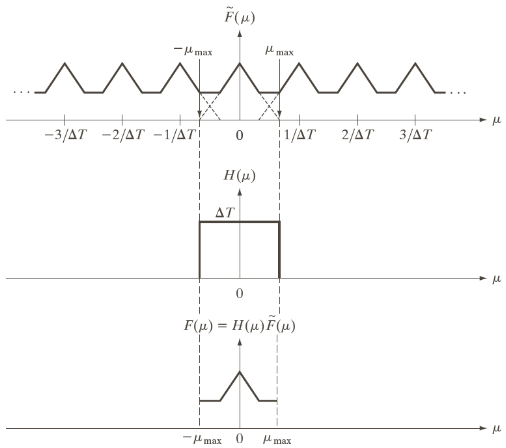
- A signal $f(t)$ with bandwidth $\mu_{\max}$ can be exactly reconstructed from samples at equidistant points $0, \pm\Delta T, \pm 2\Delta T, \cdots$, provided

$$\frac{1}{\Delta T} \geq 2\mu_{\max}$$

- The minimal sampling frequency required is $2\mu_{\max}$: this is called the Nyquist rate (critical sampling).
- The formula for recovering f from its samples is:

$$\boxed{f(t) = \sum_{n=-\infty}^{\infty} f(n\Delta T)\,\mathrm{sinc}\left[\frac{t - n\Delta T}{n\Delta T}\right]}$$

Aliasing



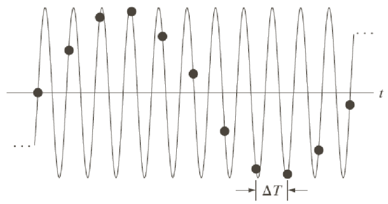
(a) Fourier transform $\tilde{F}(\mu)$ of under-sampled bandlimited function.

(b) Box filter $H(\mu)$.

(c) Product of $\tilde{F}(\mu)$ and $H(\mu)$.

In the undersampled case the standard recovery procedure results in interference from overlapping periods: aliasing.

Aliasing



- Frequency aliasing occurs if the sampling rate is less than the Nyquist rate $2\mu_{\max}$.
- In practice aliasing occurs even for bandlimited functions because signals are always of finite duration $\rightarrow$ infinite frequency components.
- Anti-aliasing: remove or attenuate frequency components higher than the sampling rate $\frac{1}{\Delta T}$.

Finite number of samples

- The Fourier transform of a function sampled with interval ΔT is periodic with period $\frac{1}{\Delta T}$:

$$\tilde{F}(\mu) = \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} F(\mu - \frac{n}{\Delta T}) = \sum_{n=-\infty}^{\infty} f_n \, e^{-i2\pi\mu n\Delta T}$$

- Sampling one period $[0, \frac{1}{\Delta T}]$ is enough to characterize $\tilde{F}(\mu)$. Take M frequency samples:

$$\begin{aligned} F_m &= \tilde{F}(\frac{m}{M\Delta T}), \\ &= \sum_{n=0}^{M-1} f_n \, e^{-i2\pi m\, n/M}, \qquad m = 0, 1, 2, \ldots, M-1 \end{aligned}$$

Discrete Fourier Transform (DFT)

- The DFT transforms a finite set of samples $\{f_n\}$ into a set $\{F_u\}$ by

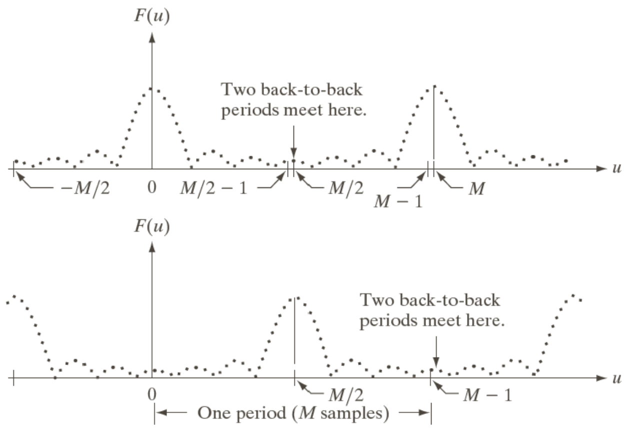
$$\boxed{F_u = \sum_{n=0}^{M-1} f_n \, e^{-i2\pi u\, n/M}, \qquad u = 0, 1, 2, \ldots, M-1}$$

- The inverse DFT (IDFT) transforms $\{F_u\}$ back to $\{f_n\}$ by

$$\boxed{f_n = \frac{1}{M} \sum_{u=0}^{M-1} F_u \, e^{i2\pi u\, n/M}, \qquad n = 0, 1, 2, \ldots, M-1}$$

- DFT and IDFT are implemented by the Fast Fourier Transform (FFT).

Centering the DFT



The DFT data in one period cover two back-to-back half periods. Reordering needed to obtain samples from lowest to highest frequency.

Discrete convolution

- The forward and inverse DFTs are discrete functions with period M: $f_n = f_{n+M}$, $F_u = F_{u+M}$.
- For discrete functions f_n and h_n with period M the discrete circular convolution is defined by

$$(f \star h)_n = \sum_{m=0}^{M-1} f_m \, h_{n-m}$$

- Discrete convolution theorem for periodic sequences states that the DFT of $f \star h$ equals the product of the DFTs of f and h:

$$\boxed{\mathcal{F}\{f \star h\}_u = F_u \, H_u, \qquad u = 0, 1, 2, \ldots, M-1}$$

Sampling and frequency intervals

- If $f_n = f(n\Delta T)$, $n = 0, 1, \ldots, M$ are discrete samples of a continuous function $f(t)$ taken ΔT units apart, the total duration is

$$T = M\Delta T$$

- The spacing in discrete frequency domain equals

$$\Delta u = \frac{1}{M\Delta T} = \frac{1}{T}$$

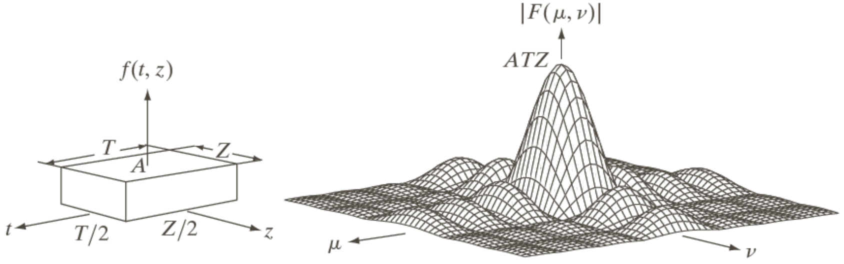
and the entire frequency range spanned is

$$\Omega = M\Delta u = \frac{1}{\Delta T}$$

Fourier transforms and convolution

$\boxed{\text{Now in 2D}}$

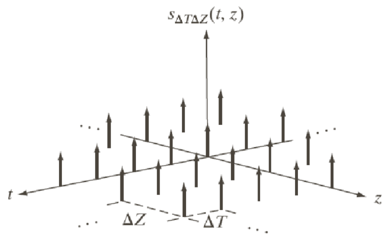
Fourier transform of 2D box signal



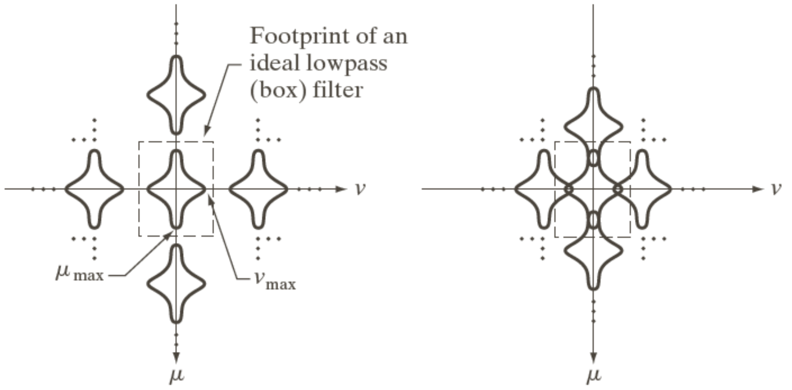
Sampling in 2D

A signal $f(t, z)$ whose 2D Fourier transforms satisfies $F(\mu, \nu) = 0$ for $|\mu| \geq \mu_{\max}$ and $|\nu| \geq \nu_{\max}$ can be exactly reconstructed from its equidistant samples if the sampling intervals ΔT and Δz in t- and z-direction satisfy:

$$\Delta T \leq \frac{1}{2\mu_{\max}}, \qquad \Delta z \leq \frac{1}{2\nu_{\max}}$$



Aliasing in images



Bandlimited function. Left: oversampled. Right: undersampled.

Aliasing in images

(a) Negligible aliasing. (b) Aliased image after resizing to 50% of original by pixel deletion. (c) Image after blurring prior to resizing.

2-D Discrete Fourier Transform

- The 2-D DFT transforms a digital image $f(x, y)$ into a transformed image $F(u, v)$ by

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

for $u = 0, 1, 2, \ldots, M-1, v = 0, 1, 2, \ldots, N-1$.

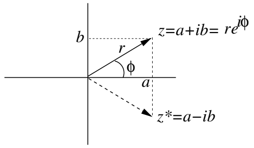
- The inverse discrete Fourier transform (IDFT) is:

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

for $x = 0, 1, 2, \ldots, M-1, y = 0, 1, \ldots, N-1$.

Fourier spectrum and phase angle

- Since the DFT is complex, we have $F(u, v) = R(u, v) + i\,I(u, v)$.



- Polar representation:

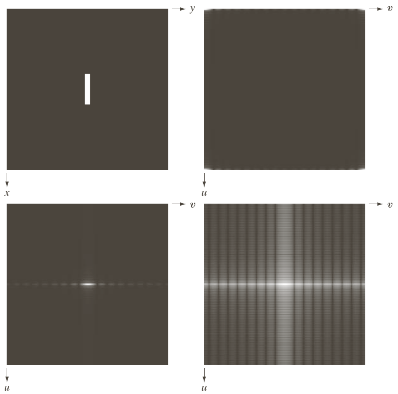
$$F(u,v) = |F(u,v)|\, e^{i\phi(u,v)} = |F(u,v)| \left[\cos(\phi(u,v)) + i\sin(\phi(u,v))\right]$$

$$|F(u,v)| = \sqrt{R^2(u,v) + I^2(u,v)}$$ Fourier spectrum

$$\phi(u,v) = \arctan\left(\frac{I(u,v)}{R(u,v)}\right)$$ phase angle

- If f is real, then the complex conjugate $F^*(u, v)$ is symmetric, i.e., $F^*(u, v) = F(-u, -v)$, so $|F(u, v)| = |F(-u, -v)|$ and $\phi(u, v) = -\phi(-u, -v)$

Centering the 2-D DFT



(top left) Image. (top right) Spectrum.
(bottom left) Centered spectrum. (bottom right) Enhanced version of centered spectrum.

2-D Convolution Theorem

- For $M \times N$ digital images f and h the 2-D discrete circular convolution is defined by

$$(f \star h)(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n)\, h(x - m, y - n)$$

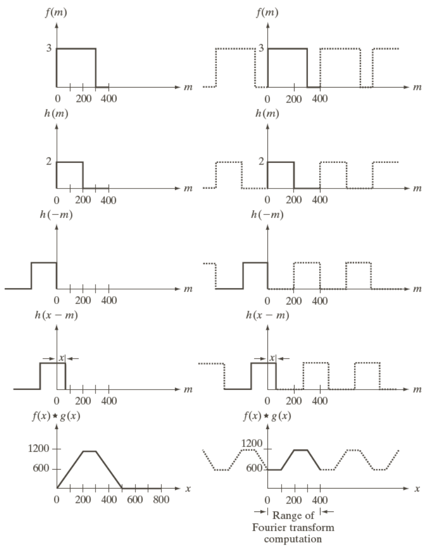
for $x = 0, 1, 2, \ldots, M - 1, y = 0, 1, 2, \ldots, N - 1.$

- 2-D discrete convolution theorem:

$$\boxed{\mathcal{F}\{f \star h\}(u, v) = F(u, v)\, H(u, v)}$$

for $u = 0, 1, 2, \ldots, M - 1, v = 0, 1, 2, \ldots, N - 1.$

Periodic Convolution - Wraparound



Frequency domain filtering

1. For $M \times N$ input $f(x, y)$ and $C \times D$ filter $h(x, y)$ apply zero padding to get image $f_p(x, y)$ of size $P \times Q$ ($P \geq M + C - 1$, $Q \geq N + D - 1$).
2. Center its transform by multiplying $f_p(x, y)$ by $(-1)^{x+y}$ (Matlab: apply `fftshift` to the DFT after step 3).
3. Compute the DFT of f_p: $F(u, v)$.
4. Generate a real symmetric filter $H(u, v)$ of size $P \times Q$ centered at $(P/2, Q/2)$ by zero padding and multiplying $h_p(x, y)$ by $(-1)^{x+y}$.
5. Pointwise multiplication: $G(u, v) = H(u, v)F(u, v)$
6. Do the 2-D inverse DFT, take the real part (to remove small imaginary parts due to computational inaccuracy) and undo the centering:

$$g_p(x, y) = \Big\{\text{real}\big[IDFT[G(u, v)]\big]\Big\}(-1)^{x+y}$$

7. Obtain $g(x, y)$ by extracting the top left $M \times N$ quadrant from $g_p(x, y)$.

Laplace filter in the spatial domain

```
function [g] = laplace (f)
% [G] = LAPLACE(F) performs laplace filtering
% kernel h=[0 1 0; 1 -4 1; 0 1 0]
% Zero padding is applied to avoid boundary overlap effects

f=im2double(f);              % convert f to double
M=size(f,1); N=size(f,2);    % nr of rows/columns of image f
C=3; D=3;                    % nr of rows/columns of kernel h
P=M+C-1; Q=N+D-1;            % nr of rows/columns after padding

fp=zeros(P,Q);               % zero padding: start with zeroes
fp(1:M,1:N)=f;               % insert f into image fp

fp_up=[fp(2:P,:); fp(1,:)];       % shift one row up (cyclic)
fp_down=[fp(P,:); fp(1:P-1,:)];   % shift one row down (cyclic)
fp_left=[fp(:,2:Q) fp(:,1)];      % shift one column left (cyclic)
fp_right=[fp(:,Q) fp(:,1:Q-1)];   % shift one column left (cyclic)

g=(-4*fp+fp_up+fp_down+fp_left+fp_right);  % the laplace filtering
g=g(1:M,1:N);                              % undo zero padding
```


Laplace filter in the frequency domain

```
function [g] = laplace_FD (f)
% [G] = LAPLACE_FD(F) performs laplace filtering
% in the Fourier domain. Kernel h=[0 1 0; 1 -4 1; 0 1 0]
% Zero padding is applied to avoid boundary overlap effects

M=size(f,1); N=size(f,2);    % nr of rows/columns of image f
C=3; D=3;                    % nr of rows/columns of kernel h
P=M+C-1; Q=N+D-1;            % nr of rows/columns after padding
fp=zeros(P,Q);               % zero padding: start with zeroes
fp(1:M,1:N)=f;               % insert f into image fp
hp=zeros(P,Q);     % Construct filter matrix hp, same size as fp.
hp(1,1)=-4; hp(2,1)=1; hp(1,2)=1;     % Center is at (1,1)
hp(P,1)=1; hp(1,Q)=1;                 % Indices modulo P or Q
Fp=fft2(double(fp), P, Q);            % FFT of image fp
Hp=fft2(double(hp), P, Q);            % FFT of kernel hp
Gp=Fp .* Hp;                          % Product of FFTs
gp=ifft2(Gp);                         % Inverse FFT
gp=real(gp);                          % Take real part
g=gp(1:M, 1:N);                       % Undo zero padding
```


Laplace filter in the frequency domain

- Mask: $h = \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$

Center the mask at $(0,0)$:
$h(0,0) = -4$
$h(1,0) = h(-1,0) = 1$
$h(0,1) = h(0,-1) = 1$

- Input image of size $M \times N$. The 2D DFT of $h(x,y)$ is [1]:

$$H(u,v) = \sum_{x=-M/2}^{M/2-1} \sum_{y=-N/2}^{N/2-1} h(x,y)\, e^{-i2\pi(u\,x/M + v\,y/N)}$$
$$= -4 + e^{i2\pi u/M} + e^{-i2\pi u/M} + e^{i2\pi v/N} + e^{-i2\pi v/N}$$
$$= -4 + 2\cos(2\pi u/M) + 2\cos(2\pi v/N)$$

This is centered around $(0,0)$.

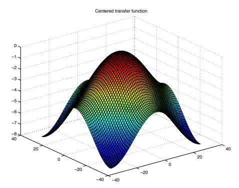
[1] Working with positive and negative indices (compare Fig. 4.23 of the course book)

Laplace filter in the frequency domain: Matlab

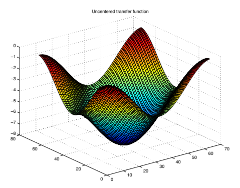
Compute $H(u,v)$ in Matlab on a grid with zero padding:
$P = M + C - 1$ rows and $Q = N + D - 1$ columns ($C = D = 3$):

$$H(u,v) = -4 + 2\cos(2\pi u/P) + 2\cos(2\pi v/Q)$$

```
rfloor = floor(P/2); rceil = ceil(P/2);          % midpoint along rows
cfloor = floor(Q/2); cceil = ceil(Q/2);          % midpoint along columns
u = -rfloor:rceil-1;                             % centered vector along rows
v = -cfloor:cceil-1;                             % centered vector along columns
[V,U] = meshgrid(v,u);                           % create mesh of size P x Q
Hc = -4 + 2*cos(2*pi*U/P) + 2*cos(2*pi*V/Q);     % centered at midpoint of grid
H = ifftshift(Hc);                               % centered at corner (1,1)
```



Hc



H

Filter design

Lowpass filter : suppress high frequencies. Example: box filter ($H(\mu) = 1$ for $0 \leq \mu \leq \mu_M$, μ_M is the cutoff frequency).

Bandpass filter : pass energy within a given frequency window.

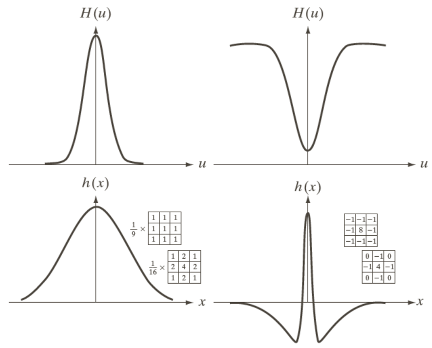
Bandstop filter : suppress energy within a given frequency window.

Highpass filter : suppress low frequencies. (Useful for edge detection).

Gaussian filters

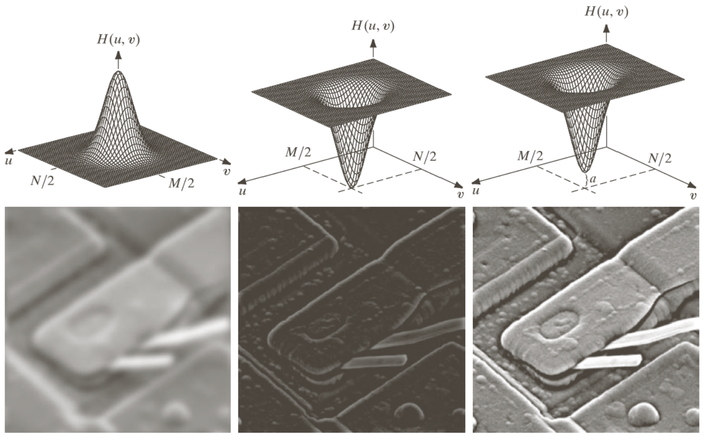
Low pass : $H(u) = Ae^{-u^2/2\sigma^2}$, $h(x) = \sqrt{2\pi}\sigma\, A\, e^{-2\pi^2\sigma^2 x^2}$

High pass : $H(u) = Ae^{-u^2/2\sigma_1^2} - Be^{-u^2/2\sigma_2^2}$ Difference of Gaussians

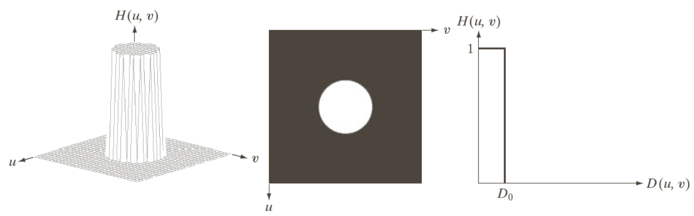


Left: low pass. Right: high pass.

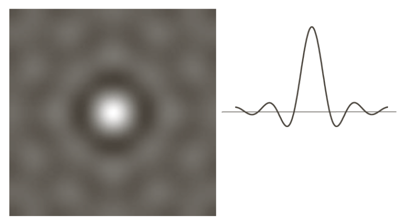
Gaussian filters



Ideal lowpass filter (ILPF)

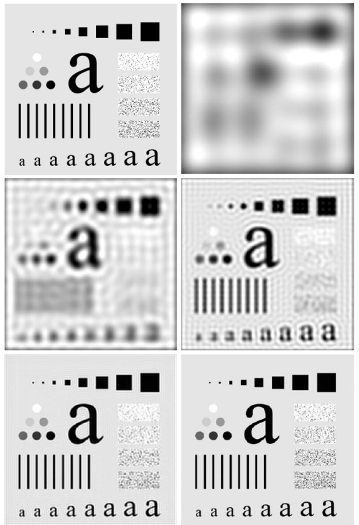


Frequency domain



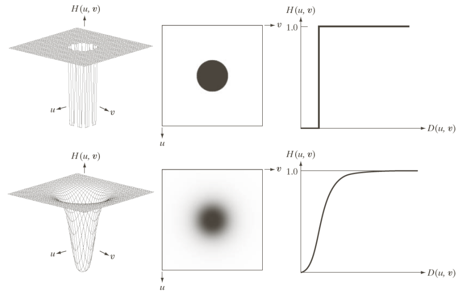
Spatial domain

Ideal lowpass filter (ILPF)

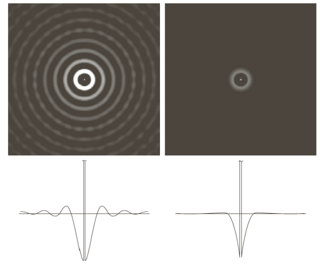


ILPF with increasing frequency cutoff radius.

Highpass filter



Frequency domain.
Top: ideal; bottom: Butterworth.



Spatial domain.
Left: ideal;
right: Butterworth.

Ideal highpass filter (IHPF)

IHPF with increasing frequency radius.