Statistics for Chemical Engineers: From Data to Models to Decisions

Victor M. Zavala

Department of Chemical and Biological Engineering
University of Wisconsin-Madison

victor.zavala@wisc.edu

Chapter 6: Statistical Data Analysis and Learning (Part II)

This material supports "Statistics for Chemical Engineers" by Victor M. Zavala, ©, 2025

Signal Processing

- Signal processing (SP) is field of mathematics that provides tools for analyzing, modifying, and synthesizing data.
- Data often appears as temporal signals (e.g., temperature, vibration, sound) or spatial signals (e.g., images, tomography, micrographs).
- We look into a SP task known as filtering (removes unwanted features from data) with goal of extracting dominant features (patterns).
- Filtering is a powerful way to preprocess raw data and can be beneficial in performing estimation, modeling, data analysis.
- *Convolution* is fundamental operation behind filtering. This is a *weighted sum of data points* over local subdomain (window) of signal.
- *Fourier transform* is another fundamental operation of SP that transforms data to frequency domain; this facilitates convolution and allows us to characterize features in terms of frequency content.
- Features extracted from convolutions can be used to classify images; this is basic principle behind machine learning techniques such as convolutional neural nets.

1D Convolution Operations

- Consider continuous signal represented by scalar (1D) function $f : \mathbb{R} \to \mathbb{R}$.
- Consider a weight (filter) denoted by scalar function $g : \mathbb{R} \to \mathbb{R}$.
- Convolution of signal and filter is a function $\varphi : \mathbb{R} \to \mathbb{R}$ given by:

$$\varphi(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau, \;\; t \in (-\infty, \infty).$$

- Convolution function is expressed as $\varphi = f * g$.
- Domains of signal and filter is $(-\infty, \infty)$; this is symmetric and centered around zero.
- One can also define convolutions over asymmetric and finite domains $t \in [0, N]$.

1D Convolution Operations

- Consider *discrete* signal given by sequence $f_j,\ j = -\infty, ..., \infty$.
- Consider filter represented by sequence $g_j,\ j = -\infty, ..., \infty$.
- Convolution of signal and filter is sequence $\varphi_k,\ k = -\infty, ..., \infty$ given by:

$$\varphi_k = \sum_{j=-\infty}^{\infty} f_j g_{k-j},\ \ k = -\infty, ..., \infty.$$

- Convolution function is expressed as $\varphi = f * g$.
- Continuous representation used to establish analytical results.
- Discrete representation used to enable computations.

1D Convolution Operations

- Convolution is *matching* operation between signal and filter.
- Operation seeks to extract patterns (features) from signal and such patterns are defined by filter.
- Extraction can also be interpreted as a process that filters out features that are not present in the filter (hence origin of term "filter").
- Filters are strategically defined to extract specific features from a signal.
- Moving average filter is function of form:

$$g(t) = \begin{cases} 0 & -\infty \leq t < a \\ w(t) & a \leq t \leq b \\ 0 & b < t \leq \infty \end{cases}$$

- Here, $w(t) = 1/(b-a)$, $t \in [a,b]$ is weight function that satisfies $\int_a^b w(t) = 1$.
- Moving average filter is pdf of uniform RV.

1D Convolution Operations

- Consider discrete version of moving average filter:

$$g_j = \begin{cases} 0 & -\infty \le j < a \\ w_j & a \le j \le b \\ 0 & b < j \le \infty \end{cases}$$

- Here, $w_j = 1/(b-a+1)\, j = a, ..., b$ are weights that satisfy $\sum_{j=a}^{b} w_j = 1$.
- Define $a = -1$ and $b = +1$ so that $b - a + 1 = 3$.
- Consider weighted average of f over window $k = -1, 0, 1$:

$$\begin{aligned}\varphi_k &= w_{-1} \cdot f_{k-1} + w_0 \cdot f_k + w_{+1} \cdot f_{k+1} \\ &= \sum_{j=-1}^{+1} w_j f_{k+j}.\end{aligned}$$

- Window is also known as *scanning* window.

Convolution Operations

- Consider symmetric window with N entries: $a = -(N-1)/2$ and $b = (N-1)/2$.
- Weighted average over window centered at f_k is:

$$\begin{aligned}\varphi_k &= \sum_{j=-(N-1)/2}^{(N-1)/2} w_j \cdot f_{k+j} \\ &= \sum_{j=-\infty}^{\infty} g_j \cdot f_{k+j}\end{aligned}$$

- Weighted average is convolution of signal and filter.
- Convolution value φ_k is obtained by centering symmetric window at f_k.
- Next value φ_{k+1} is obtained by shifting (sliding) scanning window forward and centering it at f_{k+1} (hence term moving average filter).
- Moving average filter g smooths out noisy signals f and this helps reveal patterns.
- Filters are expressed as weight vectors over scanning window; e.g., for $N = 3$ we have $g = (1/3, 1/3, 1/3)$.

Convolution Operations

$$1 \quad 1 \quad 0 \quad 0 \quad 0 \quad \overset{k}{\boxed{1 \quad 1 \quad 1}} \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1$$

$$f_k = (1,1,1) \implies \varphi_k = (1/3)\cdot 1 + (1/3)\cdot 1 + (1/3)\cdot 1 = 1$$

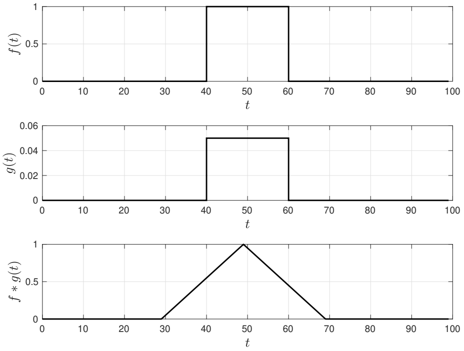
$$1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad \overset{k+1}{\boxed{1 \quad 1 \quad 0}} \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1$$

$$f_{k+1} = (1,1,0) \implies \varphi_{k+1} = (1/3)\cdot 1 + (1/3)\cdot 1 + (1/3)\cdot 0 = 2/3$$

Figure: Illustration of the application of moving average filter to a signal.

Example: Moving Average Filter for Pulse Signal `ch6_example_convolutions.m`

- Rectangular signal f, moving average filter g, and convolution $\phi = f * g$.
- Signal and filter are vectors with 100 entries.
- Filter has window with $N = 20$ entries and each weight is $1/N = 1/20$.
- Note convolution peaks at location in which filter and signal match.



Convolution Operations

- Another popular filter is the Gaussian filter:

$$g(t) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{t^2}{\sigma^2}},\ t \in (-\infty, \infty).$$

- Here, σ^2 is param of kernel function (pdf of $\mathcal{N}(0, \sigma^2)$).

- Gaussian filter also induces averaging effect but effect is smoother.

- Weights of moving average filter are held constant over averaging window and then are decreased abruptly to zero outside the window.

- For Gaussian filter, weights are progressively reduced inside averaging window and this creates a smoother transition outside the window.

1D Convolution Operations

- One often expresses discrete Gaussian filter over symmetric window.

- Consider a window of length N, this creates symmetric domain

$$-(N-1)/2, ..., 0, ..., (N-1)/2$$

- With this, we construct discrete Gaussian filter:

$$g_j = e^{-j^2/\sigma^2},\ j = -(N-1)/2, ..., (N-1)/2.$$

  with $\sigma = (N-1)/(2\alpha)$.

- Here, α is called width factor; for window length N, increasing α decreases σ.

- Gaussian filters also expressed as vectors; e.g., for $N = 5$ with $\alpha = 2$ ($\sigma = 1$) we have $g = (0.13, 0.6, 1, 0.6, 0.13)$.

Example: Gaussian Filter for Pulse Signal `ch6_example_convolutions.m`

- Rectangular pulse signal f, Gaussian g, and convolution $\phi = f * g$.
- Filter constructed using window with $N = 20$ and setting $\alpha = 4$ ($\sigma = 2.4$).
- Applying Gaussian filter has effect of smoothing sharp edges.

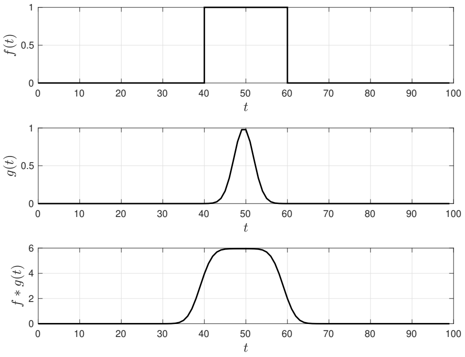


Figure: Signal (top), filter (middle), and convolution (bottom) using Gaussian filter.

Example: Gaussian Filter for Noisy Signal `ch6_example_convolutions.m`

- Show signal f and convolution $\phi = f * g$ using Gaussian filter $g = (0.60, 0.94, 0.94, 0.60)$.
- We also perform convolution using Gaussian filter $g = (0.0003, 0.41, 0.41, 0.003)$. Smoothing is not as effective; why?

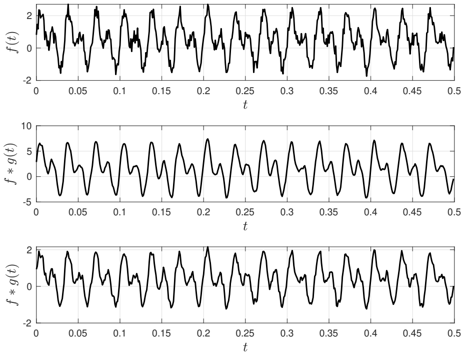


Figure: Signal (top), convolution with Gaussian filter with $\sigma = 1.5$ (middle), and convolution with Gaussian filter with $\sigma = 0.37$ (bottom).

1D Convolution Operations

- One can design filters that extract interesting features from signal.
- e.g., imagine you want to extract derivative (gradient) of $f(t)$:

$$\frac{df(t)}{dt} = \frac{f(t+\Delta t) - f(t-\Delta t)}{2 \cdot \Delta t}.$$

- Define $\frac{df(t)}{dt} = \varphi_k$, $f(t) = f_k$, $f(t+\Delta t) = f_{k+1}$, and $f(t-\Delta t) = f_{k-1}$.
- Define weights $g_{-1} = 1/2\Delta t$, $g_0 = 0$, and $g_1 = -1/2\Delta t$.
- Derivative can be written as weighted sum:

$$\varphi_k = g_{-1} f_{k-1} + g_0 f_k + g_1 f_{k+1}.$$

- This is a convolution with filter $g = (-1, 0, 1)$.
- Known as gradient filter and helps identify location of min/max/flat points of signal.

1D Convolution Operations

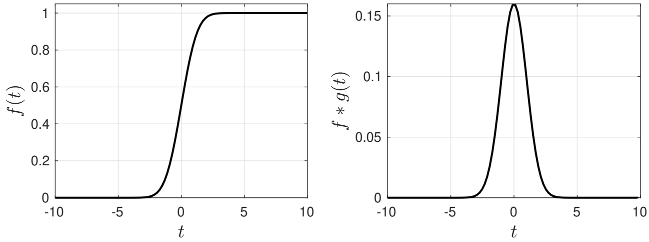
- Weight has positive/negative values (Gaussian and uniform have positive).
- Weights correspond to coefficients of central finite difference approx.
- Another popular finite difference approx has weights $g = (1, -2, 1)$:

$$\frac{d^2 f(t)}{dt^2} = \frac{f(t + \Delta t) - 2f(t) + f(t - \Delta t)}{(\Delta t)^2}.$$

- This computes second derivative of signal f.
- This helps identify min/max/flat points of signal.

Example: Applying Derivative Filter to Signal `ch6_derivative_filter.m`

- We have previously observed that pdf is derivative of cdf.
- To verify, define f as cdf for $\mathcal{N}(0,1)$ and use derivative filter $g = (1, 0, -1)$.
- Cdf f and convolution $f * g$ are shown below.
- Convolution exactly matches pdf of $\mathcal{N}(0,1)$.
- Convolution tells us where cdf grows faster/slower.



1D Fourier Transforms

- Convolution can be performed using Fourier transforms (FT) of signal and filter.
- FT also provide important insights into feature content of signals and filters.
- Recall that FT of function $f : \mathbb{R} \to \mathbb{R}$ is:

$$\begin{aligned}\hat{f}(\omega) &= \int_{-\infty}^{\infty} f(\tau) \cdot e^{-i(2\pi\omega\tau)} d\tau \\ &= \left(\int_{-\infty}^{\infty} f(\tau)\cos(2\pi\omega\tau)d\tau\right) - i\left(\int_{-\infty}^{\infty} f(\tau)\sin(2\pi\omega\tau)d\tau\right), \quad \omega \in (-\infty, \infty)\end{aligned}$$

- FT is expressed as $\hat{f} = \mathcal{F}\{f\}$; $i = \sqrt{-1}$ and ω is frequency.
- FT transforms f from domain of t (time, space) to frequency domain of ω.
- Inverse FT recovers function $f(t)$ from $\hat{f}(\omega)$:

$$f(t) = \int_{-\infty}^{\infty} \hat{f}(\omega) e^{i(2\pi\omega\tau)} d\omega \;\; t \in (-\infty, \infty).$$

- Inverse FT is expressed as $f = \mathcal{F}^{-1}\{\hat{f}\}$.

1D Fourier Transforms

- Discrete FT of signal f_j, $j = 0, ..., N-1$ is:

$$\begin{aligned}\hat{f}_k &= \sum_{j=0}^{N-1} f_j e^{-i(2\pi kj/N)} \\ &= \sum_{j=0}^{N-1} f_j \left(\cos\left(2\pi kj/N\right) - i\sin\left(2\pi kj/N\right)\right), \qquad k = 0, ..., N-1\end{aligned}$$

- Discrete FT is expressed as $\hat{f} = \mathcal{F}\{f\}$ and inverse FT is $f = \mathcal{F}^{-1}\{\hat{f}\}$.
- FT returns complex numbers $\hat{f}_k$, $k = 0, ..., N-1$ (each correspond to freq ω_k).
- Power of each freq is $|\hat{f}_k|$ (captures how strong its presence in the signal is).
- *Any* signal $f(t)$ can be obtained by combining sinusoidal functions of different freqs.
- One often visualizes frequency content by using a plot of ω_k vs. $|\hat{f}_k|$.
- Freq components are symmetric and thus $|\hat{f}_k| = |\hat{f}_{-k}|$ (only consider nonnegative k).

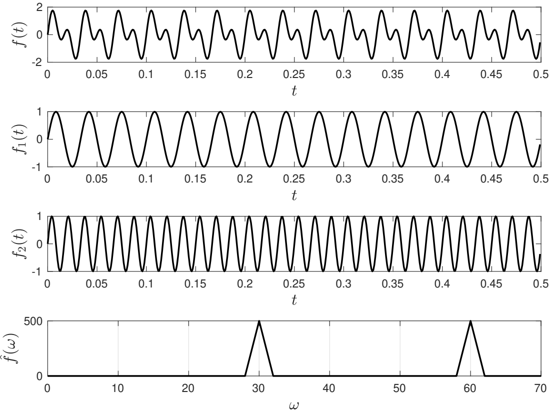
Example: FT of Periodic Signal `ch6_fourier_transform.m`

- Construct a periodic signal:

$$f(t) = \sin(2\pi\omega_1 t) + \sin(2\pi\omega_2 t)$$

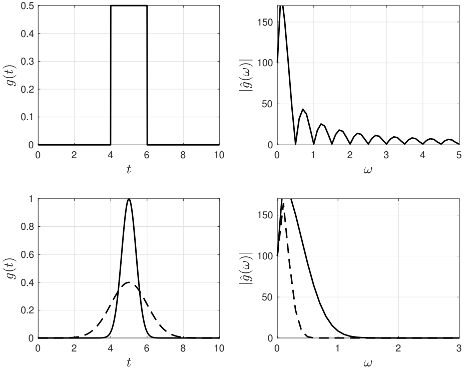
  with freqs $\omega_1 = 30$ and $\omega_2 = 60$.

- Compute FT to obtain $\hat{f}(\omega)$; this reveals dominant freqs at $\omega_1 = 30$ and $\omega_2 = 60$.



Example: FT of Pulse and Gaussian `ch6_fourier_transform_filters.m`

- FT for rect pulse function and for Gaussian function.
- Rect has strong content at low freqs and decays (non-monotonic) at high freqs.
- Gaussian has strong content at low freqs and decays (monotonic) at high freqs.
- Freq content of Gaussian controlled using σ.



Properties of Convolution

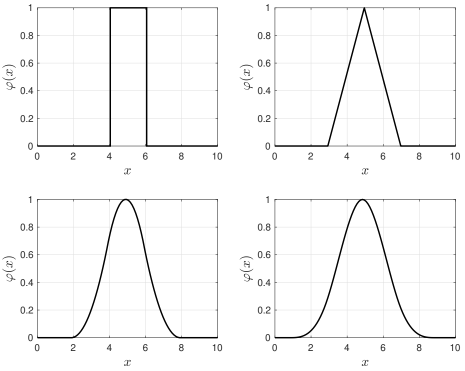
- FT facilitates analysis and implementation of convolution operations.
- Enabled by fundamental result known as *convolution theorem*:

$$\mathcal{F}\{f * g\} = \mathcal{F}\{f\} \cdot \mathcal{F}\{g\}.$$

- Convolution in freq domain obtained via products (instead of integrals).
- Pdf of sum of indep RVs X_1 and X_2 is convolution of pdfs f_{X_1} and f_{X_2}.
- Specifically, pdf of $Y = X_1 + X_2$ is $f_Y = f_{X_1} * f_{X_2}$.

Example: Recursive Application of Convolution `ch6_clt_convolution.m`

- Consider indep RVs $X_1, X_2, X_3, X_4 \sim \mathcal{U}(4,6)$ with same pdf $f(x)$.
- Pdf of X_1 is f, of $X_1 + X_2$ is $\phi = f * f$, of $X_1 + X_2 + X_3$ is $\phi = f * (f * f)$), and of $X_1 + X_2 + X_3 + X_4$ is $\phi = f * (f * (f * f))$.
- Result confirms central limit theorem (sum of i.i.d. RVs is Gaussian).



Convolutions and Fourier Transforms in 2D

- Convolutions can be performed on 2D/3D signals (or even higher-dim).
- This allows analysis and processing of images, spatial fields, and other objects.
- Consider 2D discrete signal:

$$f_{j_1,j_2}, \;\; j_1, j_2 = -\infty, ..., \infty,$$

- Consider 2D filter represented:

$$g_{j_1,j_2}, \;\; j_1, j_2 = -\infty, ..., \infty.$$

- Convolution is given by:

$$\phi_{k_1,k_2} = \sum_{j_1=-\infty}^{\infty} \sum_{j_2=-\infty}^{\infty} g_{j_1,j_2} \cdot f_{k_1-j_1,k_2-j_2}, \;\; k_1, k_2 = -\infty, ..., \infty.$$

- As in 1D, one can think of a 2D conv ϕ_{k_1,k_2} as weighted sum in a given neighborhood of signal f that is centered at f_{k_1,k_2}.

Convolutions and Fourier Transforms in 2D

- Consider $N \times N$ neighb of f (centered at f_{k_1,k_2} and $N = 3$). This window is a matrix with $N^2 = 9$ entries.

- Goal is to compute a weighted avg of all the entries in window; to do so, we associate a weight to each signal value in window.

f_{k_1-1,k_2+1}	f_{k_1,k_2+1}	f_{k_1+1,k_2+1}
f_{k_1-1,k_2}	f_{k_1,k_2}	f_{k_1+1,k_2}
f_{k_1-1,k_2-1}	f_{k_1,k_2-1}	f_{k_1-1,k_2+1}

$g_{-1,+1}$	$g_{0,+1}$	$g_{+1,+1}$
$g_{-1,0}$	$g_{0,0}$	$g_{+1,0}$
$g_{-1,-1}$	$g_{0,-1}$	$g_{+1,-1}$

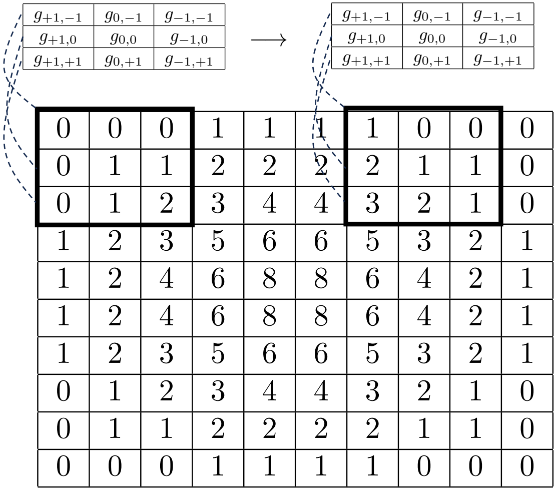
- Convolution is given by:

$$\begin{aligned}\varphi_{k_1,k_2} = \; & f_{k_1-1,k_2+1} \cdot g_{-1,1} + f_{k_1,k_2+1} \cdot g_{0,1} + f_{k_1+1,k_2+1} \cdot g_{1,1} \\ & f_{k_1-1,k_2} \cdot g_{-1,0} + f_{k_1,k_2} \cdot g_{0,0} + f_{k_1+1,k_2} \cdot g_{1,0} \\ & f_{k_1-1,k_2-1} \cdot g_{-1,-1} + f_{k_1,k_2-1} \cdot g_{0,-1} + f_{k_1+1,k_2-1} \cdot g_{1,-1}.\end{aligned}$$

- In compact form:

$$\varphi_{k_1,k_2} = \sum_{j_1=-(N-1)/2}^{(N-1)/2} \sum_{j_2=-(N-1)/2}^{(N-1)/2} g_{j_1,j_2} \cdot f_{k_1-j_1,k_2-j_2}.$$

Convolutions and Fourier Transforms in 2D



$g_{+1,-1}$	$g_{0,-1}$	$g_{-1,-1}$
$g_{+1,0}$	$g_{0,0}$	$g_{-1,0}$
$g_{+1,+1}$	$g_{0,+1}$	$g_{-1,+1}$

$\longrightarrow$

$g_{+1,-1}$	$g_{0,-1}$	$g_{-1,-1}$
$g_{+1,0}$	$g_{0,0}$	$g_{-1,0}$
$g_{+1,+1}$	$g_{0,+1}$	$g_{-1,+1}$

0	0	0	1	1	1	1	0	0	0
0	1	1	2	2	2	2	1	1	0
0	1	2	3	4	4	3	2	1	0
1	2	3	5	6	6	5	3	2	1
1	2	4	6	8	8	6	4	2	1
1	2	4	6	8	8	6	4	2	1
1	2	3	5	6	6	5	3	2	1
0	1	2	3	4	4	3	2	1	0
0	1	1	2	2	2	2	1	1	0
0	0	0	1	1	1	1	0	0	0

Figure: Example of convolution of a 2D field by using a moving window.

Example: Convolution of Matrix with Filter

- Consider conv of f given by 5×5 matrix and filter given by 3×3 matrix g:

$$\underbrace{\begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}}_{f} * \underbrace{\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}}_{g} = \underbrace{\begin{bmatrix} 5 & 1 & 4 \\ 1 & 4 & 2 \\ 3 & 3 & 4 \end{bmatrix}}_{\phi}.$$

- There are two ways of performing the convolution.

- In 1st approach we center filter at every entry of signal matrix but in such a way that the window is always inside signal matrix (we avoid boundary entries); this will return a convolution $f * g$ that is a 3×3 matrix.

- In 2nd approach, we center filter at every entry of signal matrix and allow for some entries of window to be outside signal matrix (we simply ignore any filter entries that are outside); this will return a convolution $f * g$ that is a 5×5 matrix.

- Note that convolution matrix has high values where pattern matches.

2D Convolutional Filters

- Define 2D kernel functions to generate filters; e.g., kernel of 2D Gaussian filter is:

$$g(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}}.$$

- This is pdf of $\mathcal{N}(0, \Sigma)$ with $\Sigma_{1,1} = \Sigma_{2,2} = \sigma^2$ and $\Sigma_{1,2} = \Sigma_{2,1} = 0$.

- Recall pdf projected on 2D field defines ellipses.

Example: Weight Matrix of 2D Gaussian Filter `ch6_gauss_filter_2D.m`

- Below we see 2D Gaussian filter with 10×10 window and $\sigma = \sqrt{2}$.
- Here, we visualize filter matrix and image (a 2D field).
- Filter is symmetric around center point; center point has highest value and values decay quickly as one moves away from the center point.

0	0	0	1	1	1	1	0	0	0
0	1	1	2	2	2	2	1	1	0
0	1	2	3	4	4	3	2	1	0
1	2	3	5	6	6	5	3	2	1
1	2	4	6	8	8	6	4	2	1
1	2	4	6	8	8	6	4	2	1
1	2	3	5	6	6	5	3	2	1
0	1	2	3	4	4	3	2	1	0
0	1	1	2	2	2	2	1	1	0
0	0	0	1	1	1	1	0	0	0

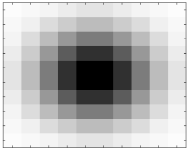


Figure: Weight matrix for 2D Gaussian filter (top) and visualization as a 2D field (bottom).

2D Convolutional Filters

- Deriv filters are used in image analysis to extract gradients and detect boundaries.
- Consider you want to compute derivative:

$$\nabla^2 f(x,u) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}.$$

- This derivative is known as Laplace operator and can be approx as:

$$\begin{aligned}\nabla^2 f(x,u) &= \frac{f(x-\Delta,y) - 2f(x,y) + f(x+\Delta,y)}{\Delta^2} + \frac{f(x,y-\Delta) - 2f(x,y) + f(x,y+\Delta)}{\Delta^2} \\ &= \frac{f(x-\Delta,y) + f(x+\Delta,y) - 4f(x,y) + f(x,y-\Delta) + f(x,y+\Delta)}{\Delta^2}\end{aligned}$$

- Laplace operator can be written as convolution:

$$f * g = \begin{bmatrix} f_{k_1-1,k_2+1} & f_{k_1,k_2+1} & f_{k_1+1,k_2+1} \\ f_{k_1-1,k_2} & f_{k_1,k_2} & f_{k_1+1,k_2} \\ f_{k_1-1,k_2-1} & f_{k_1,k_2-1} & f_{k_1+1,k_2-1} \end{bmatrix} * \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

- Another set of popular derivative filters are Sobel filters:

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}.$$

Example: Applying 2D Filters to an Image `ch6_micrograph_filters_2D.m`

- Left panel is micrograph of liquid crystal droplet when exposed to polarized light.
- Middle panel shows micrograph after applying a horizontal Sobel filter; this reveals regions in which the the brightness exhibits strong changes in intensity (gradients).
- Right panel we see that application of Gaussian filter smooths out image and this reveals coarse features of micrograph.

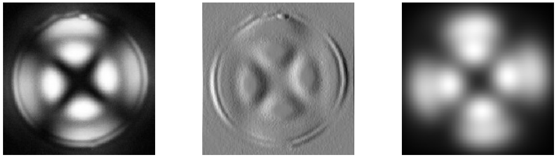


Figure: Micrograph image (left), image filtered with Sobel filter (middle), and image filtered with Gaussian filter (right).

2D Fourier Transforms

- FTs for 2D signals are used to analyze and perform conv operations.
- Discrete FT of 2D field is:

$$\hat{f}_{k_1,k_2} = \frac{1}{N}\sum_{j_1=0}^{N-1}\sum_{j_2=0}^{N-1} f_{j_1,j_2} e^{-i2\pi\left(\frac{j_1 k_1}{N}+\frac{j_2 k_2}{N}\right)},\ k_1 = k_2 = 0, ..., N-1.$$

- 2D FT returns a 2D field in the frequency domain.
- Each freq comp contains a wave (sheet); original field is a superposition of sheets.
- As in 1D, can use convolution theorem to obtain $f * g = \mathcal{F}^{-1}\{\mathcal{F}\{f\} \cdot \mathcal{F}\{g\}\}$.

Example: 2D Signals as Superposition of 2D Waves `ch6_fourier_transform_2D.m`

- 2D field (e.g., an image) is superposition of 2D fields (each with a different freq).
- 2D field with a given freq is sheet with wave pattern (higher freqs have finer waves).
- Consider 2D function:

$$f(x_1, x_2) = \sin\ 2\pi\omega_1(x_1 + x_2) - \sin\ 2\pi\omega_2(x_1 + x_2)$$

  with frequencies $\omega_1 = 20$ and $\omega_2 = 60$.

- Below we plot this function in 3D and we can see that this forms a wavy field (sheet) and show two waves that form function.
- We also visualize function as 2D projection. This is is what you would see if you observed wavy sheet from the top.

Example: 2D Signals as Superposition of 2D Waves `ch6_fourier_transform_2D.m`

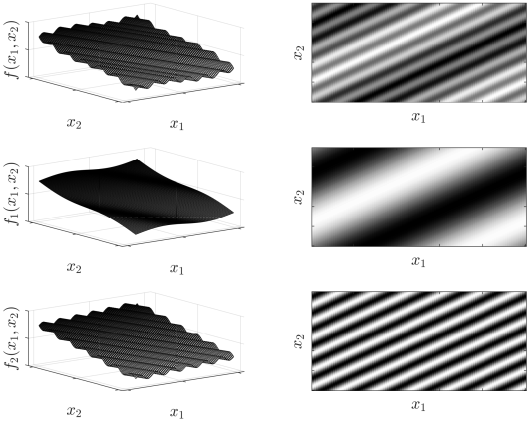


Figure: 2D function $f(x_1, x_2)$ shown in 3D and as a projection in 2D (top panels). Low-frequency component (middle panels) and high-frequency component (bottom panels).

Example: 2D FT Applied to Filtering `ch6_fourier_transform_filters_2D.m`

- We analyze frequency content of a 2D uniform average filter (defined by a 2D step function) and of a 2D Gaussian filter (defined a by 2D Gaussian kernel).

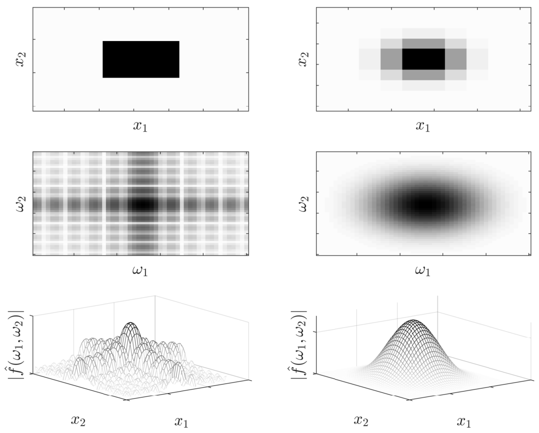


Figure: Left panels: Uniform filter, Fourier transform in 2D, and Fourier transform in 3D. Right panels: Gaussian filter, Fourier transform in 2D, and Fourier transform in 3D.

Example: 2D FT Applied to Filtering `ch6_image_filters_2D.m`

- We now illustrate how to use FTs to perform convolutions.
- We take original image f and obtain its 2D FT $\hat{f}$.
- Construct Gaussian filter with $\sigma = 1$ (g_1). We obtain its FT $\hat{g}_1 = \mathcal{F}\{g_1\}$.
- Construct Gaussian filter with $\sigma = 10$ (g_2). We obtain its FT $\hat{g}_2 = \mathcal{F}\{g_2\}$.
- Value of σ for first filter is smaller (filter is designed to only eliminate high freqs).
- Obtain filtered images in freq domain: $\hat{f}_1 = \hat{f} \cdot \hat{g}_1$ and $\hat{f}_2 = \hat{f} \cdot \hat{g}_2$.
- Obtain filtered images in orig domain via inverse FT: $f_1 = \mathcal{F}^{-1}\{\hat{f}_1\}$, $f_2 = \mathcal{F}^{-1}\{\hat{f}_2\}$.

Example: 2D FT Applied to Filtering `ch6_image_filters_2D.m`

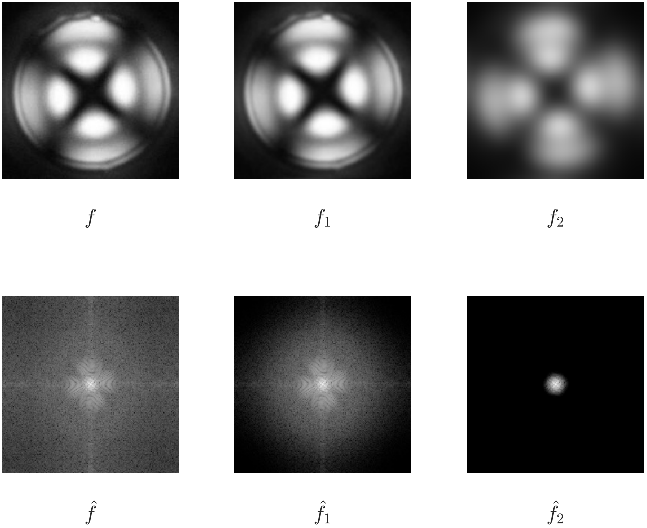


Figure: Top panels: Original and filtered images. Bottom panels: Frequency spectrum of original and filtered images.