The Finnish Graduate School in Astronomy and Space Physics Summer School 2007

Time Series Analysis

Part III. Methods

Numerical and statistical methods in time series analysis

Definition of the Fourier Transform

Let $f(x)$ be a continuous function of real variable x. The *Fourier Transform* of $f(x)$, denoted $\mathcal{F}\{f(x)\}$ is defined by the equation

$$\mathcal{F}\{f(x)\} = F(u) = \int_{-\infty}^{\infty} f(x) e^{-\mathrm{j}2\pi ux} dx \tag{3.1}$$

where $\mathrm{j} = \sqrt{-1}$.

Given $F(u)$, $f(x)$ can be obtained by using the *inverse Fourier Transform*

$$\mathcal{F}^{-1}\{F(u)\} = f(x) = \int_{-\infty}^{\infty} F(u) e^{\mathrm{j}2\pi ux} du \tag{3.2}$$

The *Fourier transform pair* exists, if $f(x)$ is continuous and integrable and $F(u)$ is integrable, which is almost always satisfied in practice.

In Computer Vision we are mainly concerned with real functions. The Fourier Transform of a real function, however, is generally complex, thus

$$F(u) = \Re(u) + \mathrm{j}\Im(u)$$

where $\Re(u)$ and $\Im(u)$ are the real and imaginary components of $F(u)$, respectively. Often it is convenient to express it in exponential form

$$F(u) = |F(u)|e^{\mathrm{j}\phi(u)}$$

where

$$|F(u)| = \sqrt{\Re^2(u) + \Im^2(u)} \qquad (3.3)$$

and

$$\phi(u) = tan^{-1}\frac{\Im(u)}{\Re(u)} \qquad (3.4)$$

The magnitude function $|F(u)|$ is called the ***Fourier Spectrum*** of $f(x)$ and $\phi(u)$ its ***Phase Angle***

The square of the spectrum

$$P(u) = |F(u)|^2 = \Re^2(u) + \Im^2(u) \tag{3.5}$$

is commonly referred to as ***Power Spectrum*** or ***Spectral Density***.

The variable u appearing in the Fourier Transform is often called the ***Frequency Variable***.
The name arises from the exponential term, that can be rewritten using Euler's Formula

$$e^{-\mathrm{j}2\pi ux} = \cos(2\pi ux) + \mathrm{j}\sin(2\pi ux)$$

Fourier Transform Example

Consider the simple function shown in Fig 3.2. Its Fourier transform is obtained from Eq 3.1 as follows:

$$
\begin{aligned}
F(u) &= \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx \\
&= \int_{0}^{X} A e^{-j2\pi ux} dx \\
&= \frac{-A}{j2\pi u} [e^{-j2\pi ux}]_0^X \\
&= \frac{-A}{j2\pi u} [e^{-j2\pi uX} - 1] \\
&= \frac{A}{j2\pi u} [e^{j\pi uX} - e^{-j\pi uX}] e^{-j\pi uX} \\
&= \frac{A}{\pi u} \sin(\pi uX) e^{-j\pi uX}
\end{aligned}
$$

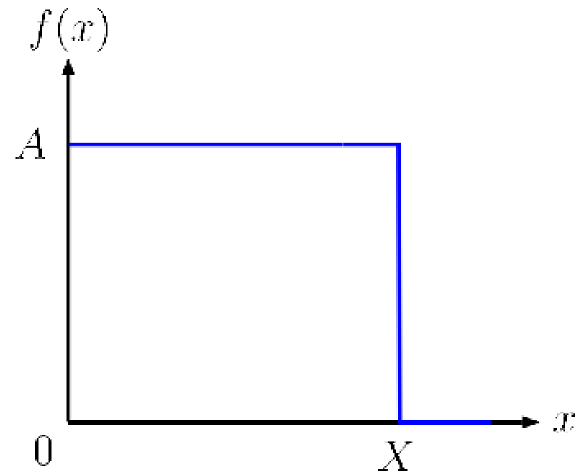


Fig 3.2: A simple function $f(x)$

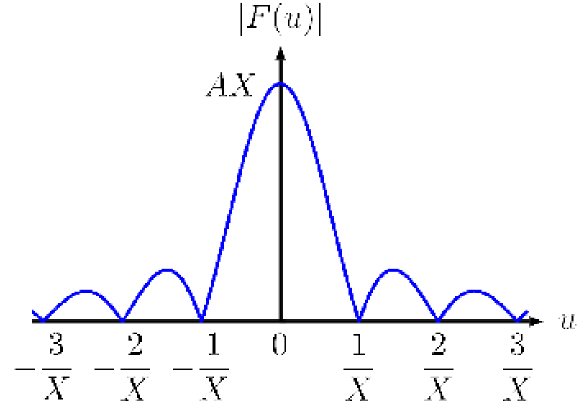


Fig 3.3: Fourier spectrum $|F(u)|$

$$
\begin{aligned}
|F(u)| &= \left|\frac{A}{\pi u}\right| |\sin(\pi uX)| |e^{-j\pi uX}| \\
&= AX \left|\frac{\sin(\pi uX)}{\pi uX}\right|
\end{aligned}
$$

Extension of the Fourier Transform to 2D Functions

The *Fourier Transformation* can be easily extended to 2D functions $f(x,y)$. If the function is continuous and integrable and $F(u,v)$ is integrable, the Fourier Transform pair exists

$$\mathcal{F}\{f(x,y)\} = F(u,v) = \int_{x=-\infty}^{\infty}\int_{y=-\infty}^{\infty} f(x,y)e^{-\mathrm{j}2\pi(ux+vy)}dxdy \qquad (3.7)$$

and the *inverse Fourier Transform*

$$\mathcal{F}^{-1}\{F(u,v)\} = f(x,y) = \int_{u=-\infty}^{\infty}\int_{v=-\infty}^{\infty} F(u,v)e^{\mathrm{j}2\pi(ux+vy)}dudv \qquad (3.8)$$

Similar to the 1D case, the *Fourier Spectrum*, *Phase*, and *Power Spectrum* can be defined as follows

$$|F(u,v)| = \sqrt{\Re^2(u,v) + \Im^2(u,v)} \qquad (3.9)$$

$$\phi(u,v) = tan^{-1}\left(\frac{\Im(u,v)}{\Re(u,v)}\right) \qquad (3.10)$$

$$P(u,v) = |F(u,v)|^2 = \Re^2(u,v) + \Im^2(u,v) \qquad (3.11)$$

Fourier Transform Example of a 2D Function

Consider the simple function shown in Fig 3.6. Its Fourier transform is obtained from Eq 3.7 as follows:

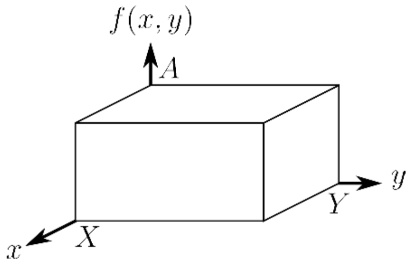


Fig 3.6: A simple function $f(x,y)$

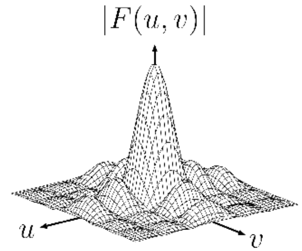


Fig 3.7: Fourier spectrum $|F(u,v)|$

$$|F(u,v)| = AXY\left|\frac{\sin(\pi uX)}{\pi uX}\right|\left|\frac{\sin(\pi vY)}{\pi vY}\right|$$

$$\begin{aligned}
F(u) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-\mathrm{j}2\pi(ux+vy)}dxdy \\
&= A\int_{0}^{X} e^{-\mathrm{j}2\pi ux}dx\int_{0}^{Y} e^{-\mathrm{j}2\pi vy}dy \\
&= A\left[\frac{e^{-\mathrm{j}2\pi ux}}{-\mathrm{j}2\pi u}\right]_0^X\left[\frac{e^{-\mathrm{j}2\pi vy}}{-\mathrm{j}2\pi v}\right]_0^Y \\
&= \frac{A}{-\mathrm{j}2\pi u}[e^{-\mathrm{j}2\pi uX}-1]\frac{1}{-\mathrm{j}2\pi v}[e^{-\mathrm{j}2\pi vY}-1] \\
&= AXY\left[\frac{\sin(\pi uX)e^{-\mathrm{j}\pi uX}}{\pi uX}\right]\left[\frac{\sin(\pi vY)e^{-\mathrm{j}\pi vY}}{\pi vY}\right]
\end{aligned}$$

Definition of the 1D Discrete Fourier Transform

In Computer Vision the continuous functions $f(x)$ are generally discretised into a sequence

$$\{f(x_0), f(x_0+\Delta x), f(x_0+2\Delta x), ..., f(x_0+[N-1]\Delta x)\}$$

by taking N samples Δx units apart. The function $f(x)$ can be redefined

$$f(x) = f(x_0 + x\Delta x)$$

where x now assumes the discrete values $0, 1, 2, ..., N-1$.

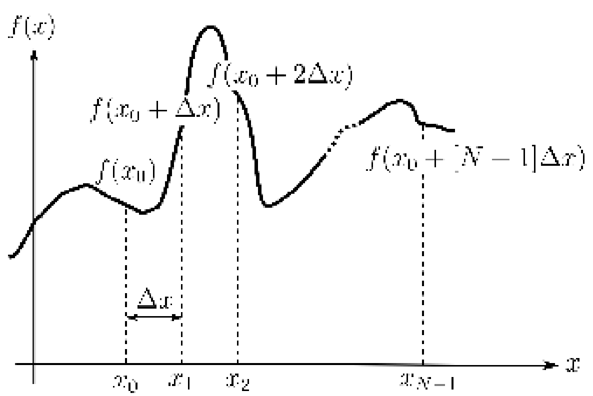


Fig 3.9: Sampling a continuous function

With this notation, the *Discrete Fourier Transform* (*DFT*) can be defined as

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{-\mathrm{j}2\pi ux/N} \quad \forall u = 0, 1, .., N-1$$

and the *Inverse Discrete Fourier Transform*

$$f(x) = \sum_{u=0}^{N-1} F(u) e^{\mathrm{j}2\pi ux/N} \quad \forall x = 0, 1, ..., N-1$$

The terms Δu and Δx are related by

$$\boxed{\Delta u = 1/(N\Delta x)}.$$

1D Discrete Fourier Transformation Example

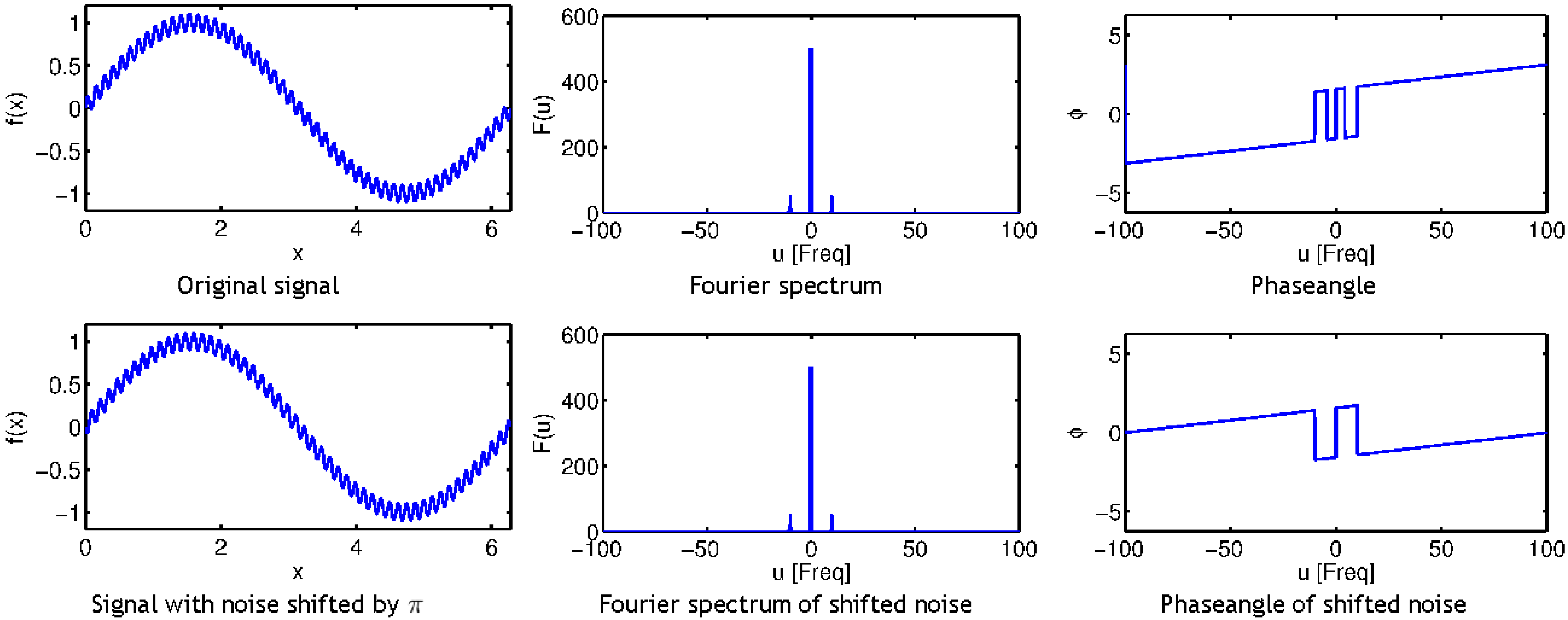


Fig 3.10: One-dimensional discrete Fourier transformation example. Lower example shows the effect of a phaseshift in the high frequency signal to the phase angle.

The 2D Discrete Fourier Transform

The 2D Discrete Fourier Transform

The definition of the (in Computer Vision more common) *2D Discrete Fourier Transform (DFT)* is then given by

$$\mathcal{F}\{f(x,y)\} = F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-\mathrm{j}2\pi(ux/M+vy/N)} \tag{3.13}$$

for $u = 0, 1, 2, ..., M-1$ and $v = 0, 1, 2, ..., N-1$, and the *Inverse DFT*

$$\mathcal{F}^{-1}\{F(u,v)\} = f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{\mathrm{j}2\pi(ux/M+vy/N)} \tag{3.14}$$

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

Sampling of the continuous function $f(x,y)$ is in a 2D grid of width Δx and height Δy in the x, y axis, respectively.

The 2D Discrete Fourier Transform (2)

As in the 1D case, the discrete function $f(x, y)$ represents samples of the function

$$f(x_0 + x\Delta x, y_0 + y\Delta y)$$

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

The sampling increments in the spatial and frequency domain are related by

$$\Delta u = \frac{1}{M\Delta x} \tag{3.15}$$

and

$$\Delta v = \frac{1}{N\Delta y} \tag{3.16}$$

The Scaling Terms

Remark 1: The Scaling Terms

Because $f(x,y)$ and $F(u,v)$ are a ***Fourier Transform pair*** the multiplicative ***scaling*** terms can be ***chosen arbitrary***. As images are often digitised in square arrays, thus $M = N$, the following scaling is often chosen

$$f(x,y) \quad \circ\!\!-\!\!\bullet \quad F(u,v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(ux+vy/N)}$$

and

$$F(u,v) \quad \bullet\!\!-\!\!\circ \quad f(x,y) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u,v) e^{j2\pi(ux+vy/N)}$$

Beware, that the scaling term in Matlab is with the inverse rather than the transform!

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \ldots, \qquad f(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \ldots$$

Existence of the DFT

Remark 2: Existence of the DFT

Claim

In contrast to the continuous case, existence of the discrete Fourier Transform is of no concern, because both $F(u)$ and $F(u,v)$ always exist.

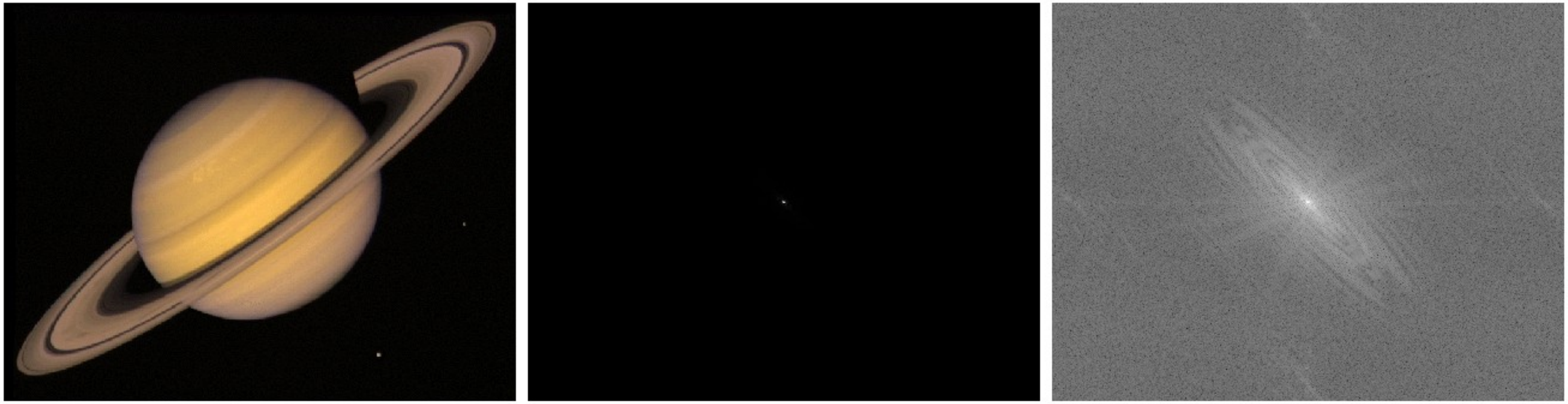
Proof $F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)e^{-j2\pi ux/N}$ and $f(x) = \sum_{u=0}^{N-1} F(u)e^{j2\pi ux/N}$

$$
\begin{aligned}
F(u) &= \frac{1}{N}\sum_{x=0}^{N-1}\left[\sum_{r=0}^{N-1} F(r)e^{j2\pi rx/N}\right] e^{-j2\pi ux/N} \\
&= \frac{1}{N}\sum_{r=0}^{N-1} F(r)\left[\sum_{x=0}^{N-1} e^{j2\pi rx/N} e^{-j2\pi ux/N}\right] \\
&= F(u)
\end{aligned}
$$

The identify follows from the orthogonality condition

$$
\sum_{x=0}^{N-1} e^{j2\pi rx/N} e^{-j2\pi ux/N} = \begin{cases} N & \text{if } r = u \\ 0 & \text{otherwise} \end{cases}
$$

2D Discrete Fourier Transform Example



(a) Original image (b) The Fourier spectrum (c) The log Fourier spectrum

Fig 3.11: Example Voyager 2 image of saturn with its Fourier spectrum and the log Fourier spectrum

The ***dynamic range*** of ***Fourier spectra*** usually is much ***higher*** than the typical ***display*** device can reliably reproduce. The consequence is that ***only*** the ***brightest parts*** are ***shown***, see Fig 3.11(b). A useful technique that compensates for this difficulty is of displaying the function, see Fig 3.11(c)

$$D(u,v) = c \cdot \log[1 + |F(u,v)|] \tag{3.17}$$

Fourier Transform of Even, Odd Functions

Fourier Transform of Even, Odd Functions

The fact that the *Fourier Transform* of a *real-valued* image yields a *complex* output might give the impression that information has somehow doubled. This is of course not the case. In fact, for real input (such as images) a number of important properties hold for the *Fourier Transform*:

Spatial Domain	Frequency Domain
real	real part even, imaginary part odd
real, even	real, even
real, odd	imaginary, odd

Hint:

- A function is ***even*** if it holds for all real x: $f(-x) = f(x)$ thus symmetric to the y-axis.
- A real-valued function is ***odd*** if for all real x it holds: $f(-x) = -f(x)$ thus symmetric to the origin.

As can be seen in the above table, the relationships between Fourier coefficients are such that the total number of independent variables remains the same.

Separability

Separability

The 2D discrete Fourier Transform pair, Eq 3.13 & 3.14, can be expressed in its separable form

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\left[e^{-j2\pi ux/N}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi vy/N}\right] \tag{3.20}$$

and the Inverse Transformation respectively

$$f(x,y) = \frac{1}{N}\sum_{u=0}^{N-1}\left[e^{j2\pi ux/N}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi vy/N}\right] \tag{3.21}$$

Advantage:

- The Transformations $F(u,v)$ and $f(x,y)$ can be obtained in two successive applications of 1D Fourier transforms → computationally very efficient.

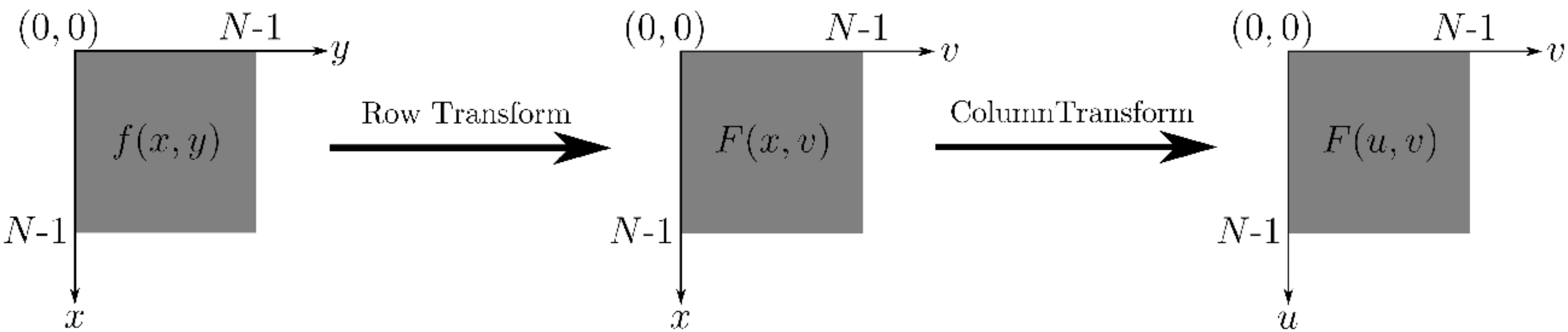
This becomes evident, when we rewrite the separable discrete Fourier Transform in Eq 3.20 in the form

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1} F(x,v)e^{-j2\pi ux/N} \tag{3.23}$$

where

$$F(x,v) = N\left[\frac{1}{N}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi vy/N}\right] \tag{3.24}$$

we get two 1D Fourier Transforms. The same principle applies for the *Inverse Fourier Transform*. The following figure illustrates this process:



Translation

Translation

We have to differentiate between two cases

1. Translations in the ***Fourier (Frequency) Domain***
2. Translations in the ***Image Domain***

Let's assume we know

$$f(x,y) \quad \Longleftrightarrow \quad F(u,v)$$

we want to know how a translation in the Fourier Domain by (u_0, v_0) can by expressed in $f(x,y)\cdots$

$$F(u-u_0, v-v_0) \quad \Longleftrightarrow \quad f(x,y)\cdots$$

and similarly for translations in the Image Domain

$$f(x-x_0, y-y_0) \quad \Longleftrightarrow \quad F(u,v)\cdots$$

Translation in the Fourier Domain

Translation in the Fourier Domain

A *Translation* of (u_0, v_0) in the *Fourier Domain* results in

$$F(u - u_0, v - v_0) \quad \Longleftrightarrow \quad f(x,y)e^{j2\pi(u_0x+v_0y)/N}$$

A multiplication of $f(x,y)$ with the exponential term and taking the transform of the product results in a shift of the origin of the frequency plane to the point (u_0, v_0)

The special case were $u_0 = v_0 = N/2$ is often used and yields

$$\begin{aligned} e^{j2\pi(u_0x+v_0y)/N} &= e^{j\pi(x+y)} \\ &= -1^{x+y} \end{aligned}$$

and

$$f(x,y)(-1)^{x+y} \Longleftrightarrow F(u - N/2, v - N/2).$$

Thus the origin of the Fourier Transform of $f(x,y)$ can be moved to the centre of its corresponding $N \times N$ frequency square by multiplying $f(x,y)$ by the term $(-1)^{x+y}$.

For the 1D case this shift reduces to the term $(-1)^x$.

Translation in the Image Domain

Translation in the Image Domain

A ***Translation*** of (x_0, y_0) in the ***Image Domain*** results in

$$f(x - x_0, y - y_0) \quad \Longleftrightarrow \quad F(u, v)e^{-j2\pi(ux_0+vy_0)/N}$$

Multiplying $F(u, v)$ with the exponential term $e^{-j2\pi(ux_0+vy_0)/N}$ and taking the inverse Fourier Transform moves the origin of the Image to (x_0, y_0).

Note, that a shift in $f(x, y)$ does not affect the magnitude of the Fourier Transform (but only its phase), as

$$|F(u, v)e^{-j2\pi(ux_0+vy_0)/N}| = |F(u, v)|$$

This is important to know, as the magnitude is often taken to visualise the Fourier Transform.

Distributivity and Scaling

Distributivity and Scaling

Form the definition of the discrete Fourier Transform pair follows that

$$\mathcal{F}\{f_1(x,y) + f_2(x,y)\} = \mathcal{F}\{f_1(x,y)\} + \mathcal{F}\{f_2(x,y)\}$$

However, in general

$$\mathcal{F}\{f_1(x,y) \cdot f_2(x,y)\} \neq \mathcal{F}\{f_1(x,y)\} \cdot \mathcal{F}\{f_2(x,y)\}$$

In other words, the Fourier Transform and its inverse is ***distributive*** over ***addition*** but not over ***multiplication***.

For two scalars a and b

$$af(x,y) \Leftrightarrow aF(u,v)$$

and

$$f(ax,by) \Leftrightarrow \frac{1}{|ab|}F(u/a,v/b)$$

Convolution

Convolution (1)

The *convolution* of two 1-dimensional functions $f(x)$ and $g(x)$ is generally denoted by $f(x) * g(x)$ and defined by the integral

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

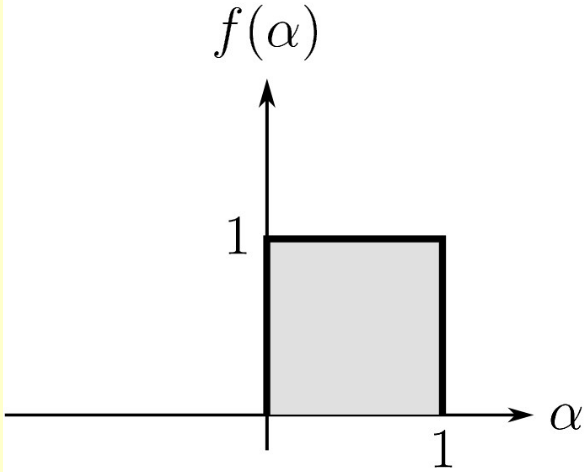
where α is a dummy variable.

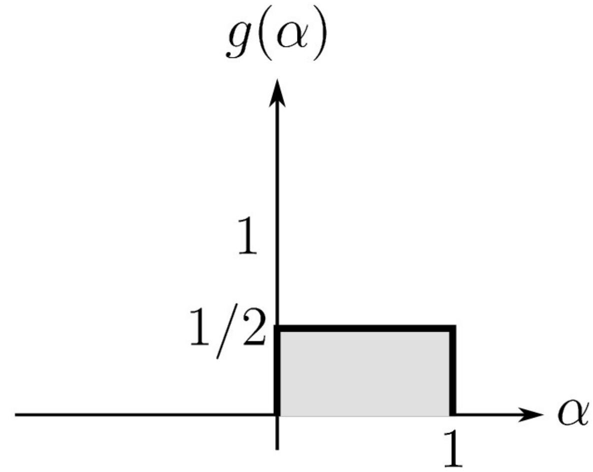
The importance of convolution in the frequency domain analysis lies in the fact that $f(x) * g(x)$ and $F(u)G(u)$ constitute a Fourier Transformation pair thus

$$f(x) * g(x) \quad \Leftrightarrow \quad F(u)G(u)$$

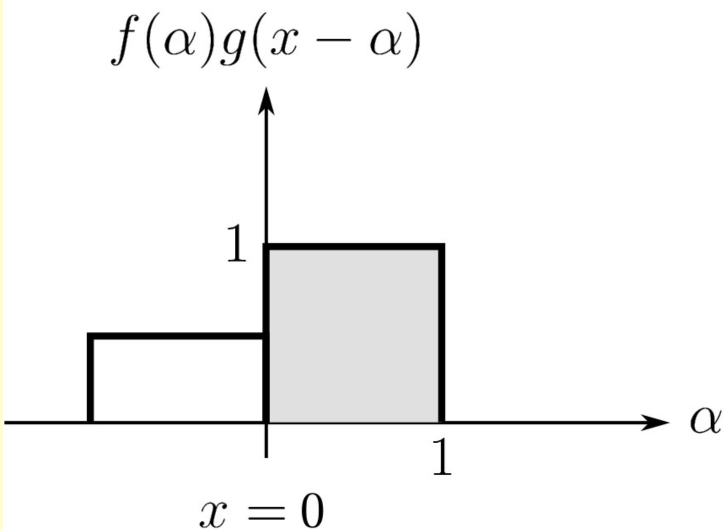
a convolution in the Image domain results in a multiplication in the Frequency domain, and vice versa

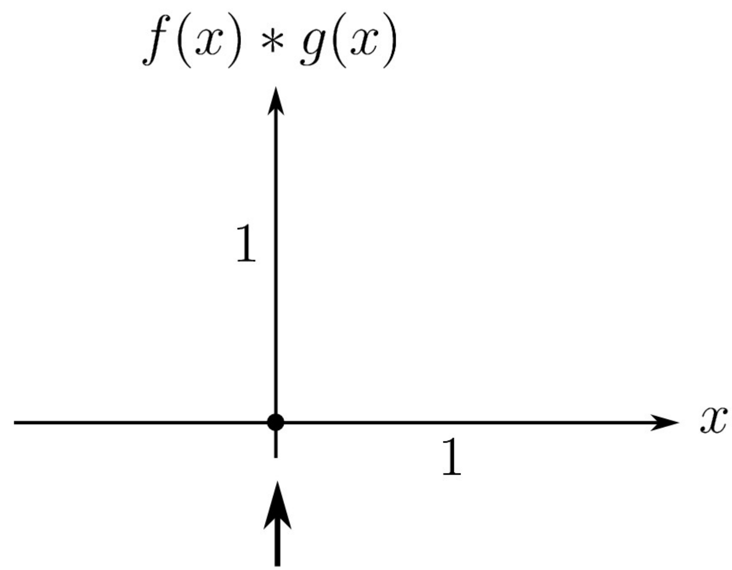
$$f(x)g(x) \quad \Leftrightarrow \quad F(u) * G(u).$$

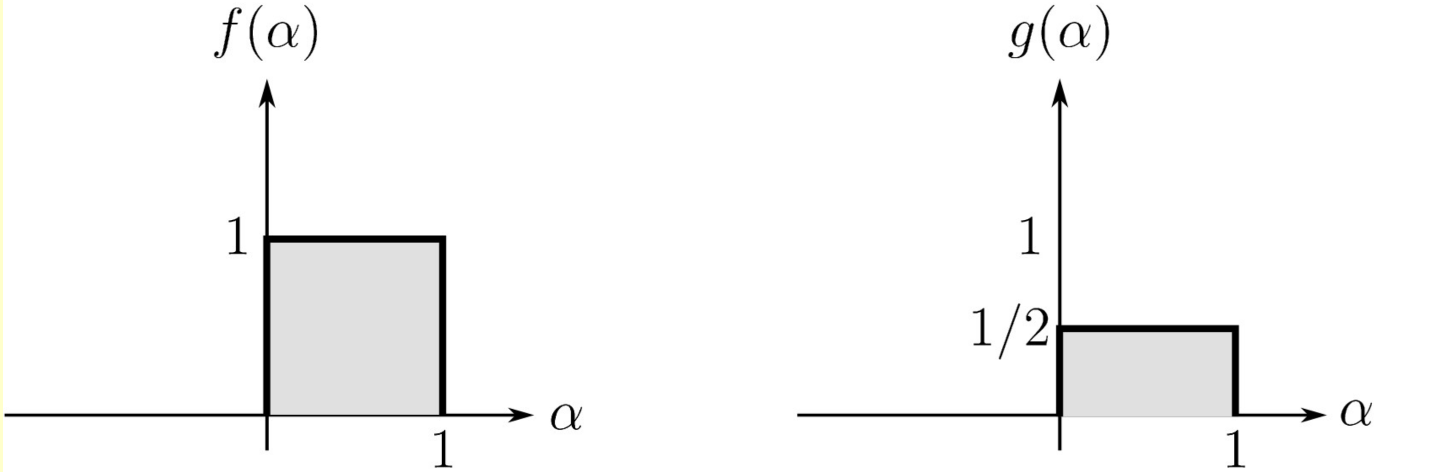




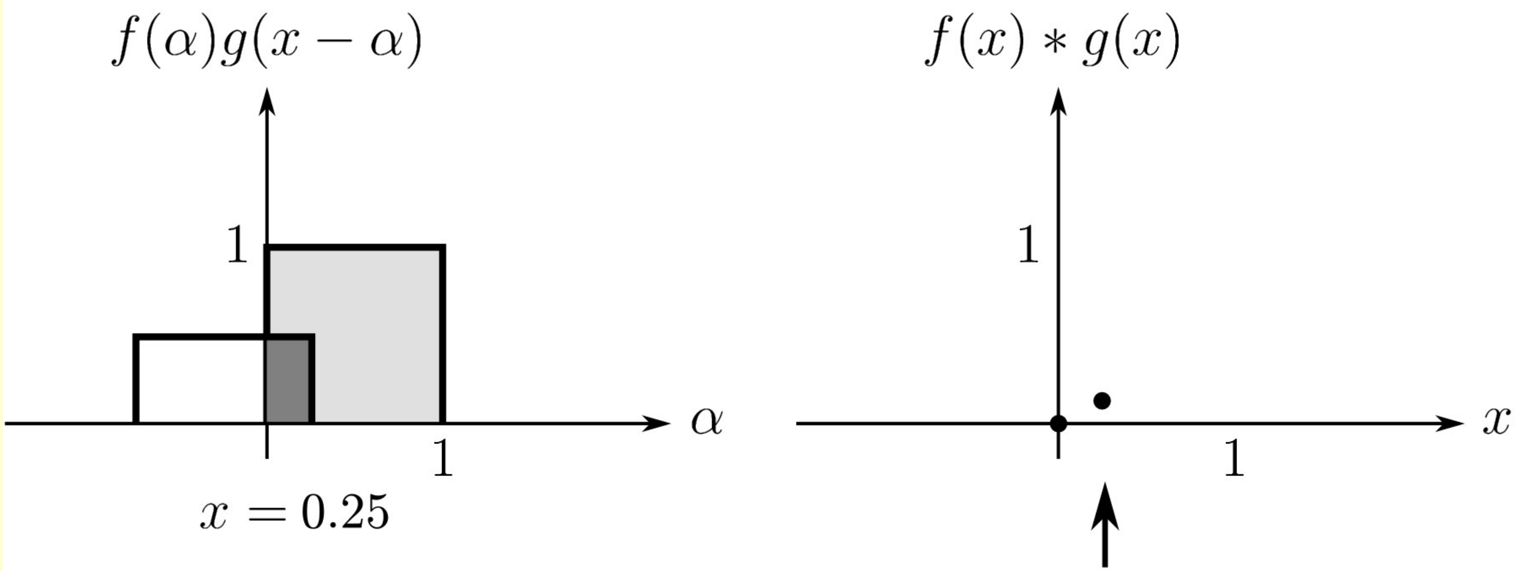
$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\alpha) g(x-\alpha) d\alpha$$

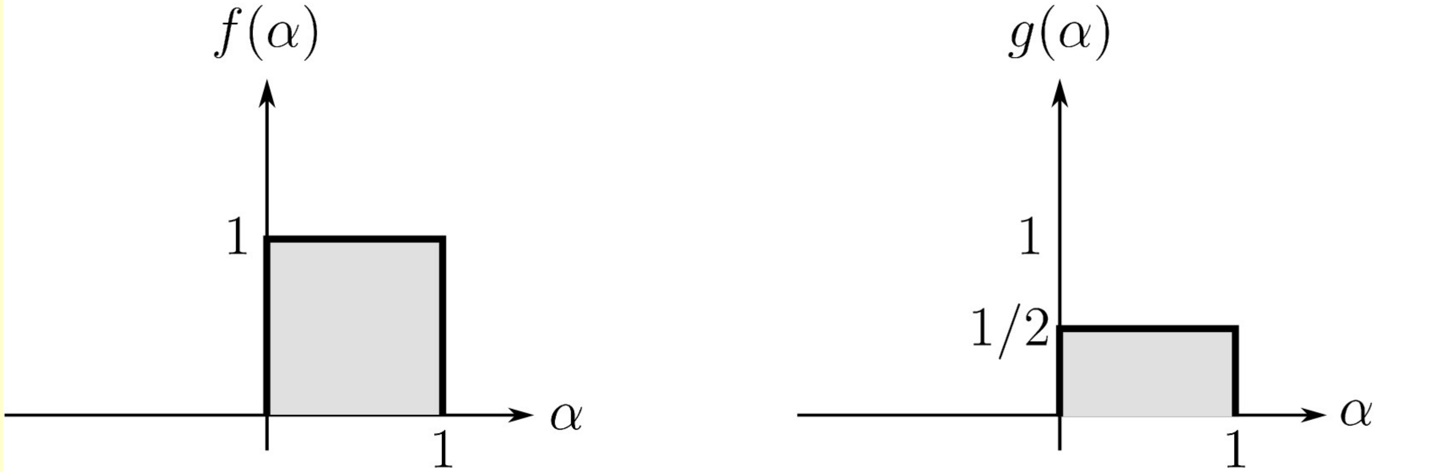




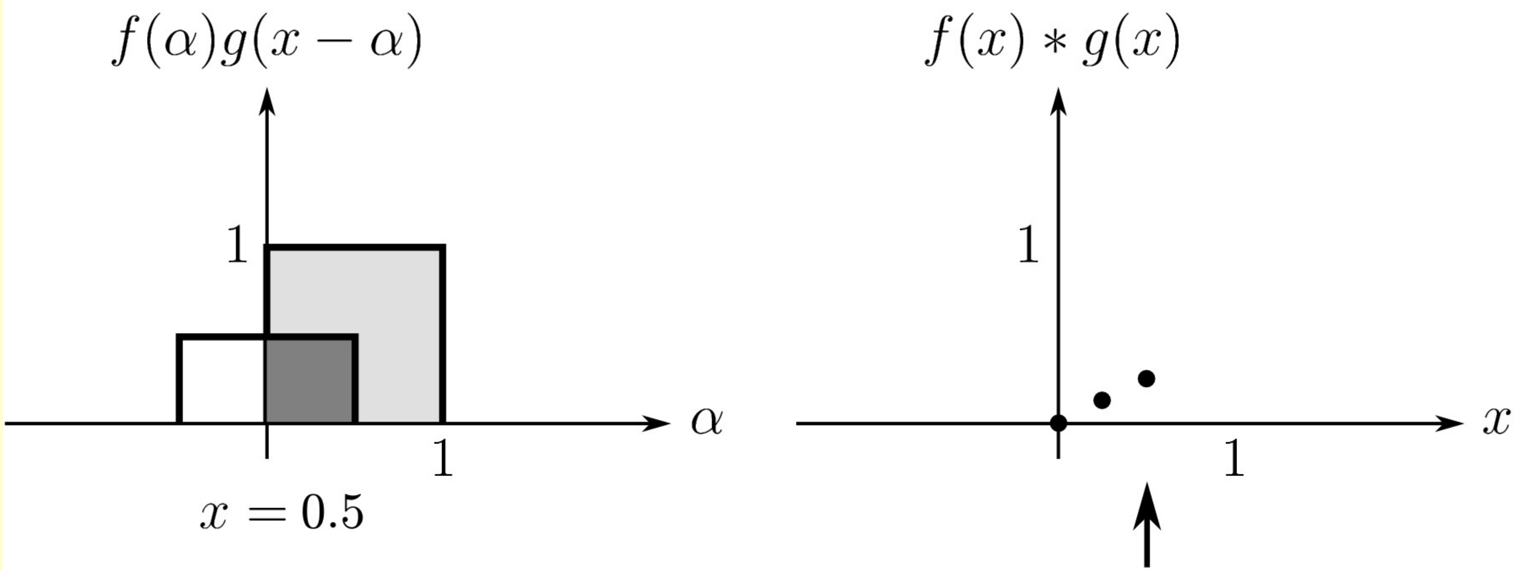


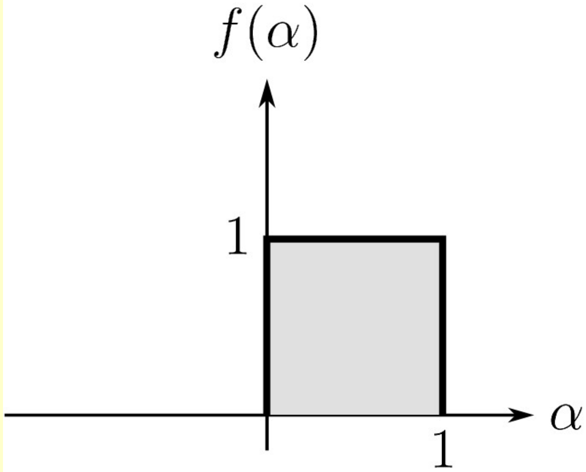
$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\alpha) g(x-\alpha) d\alpha$$

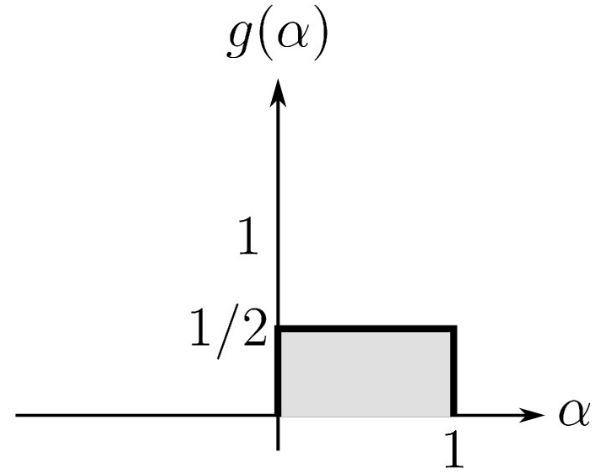




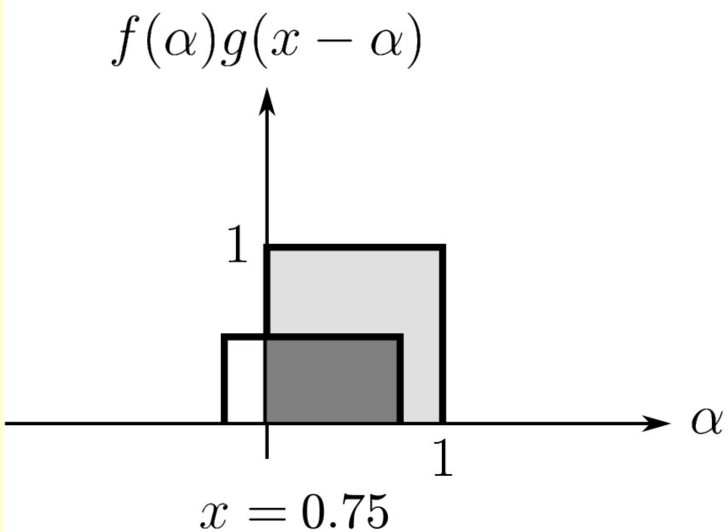
$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\alpha) g(x-\alpha) d\alpha$$

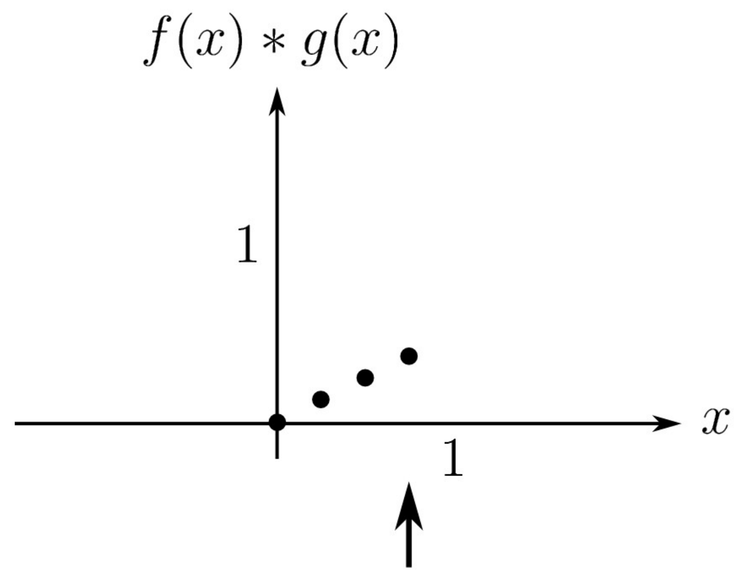


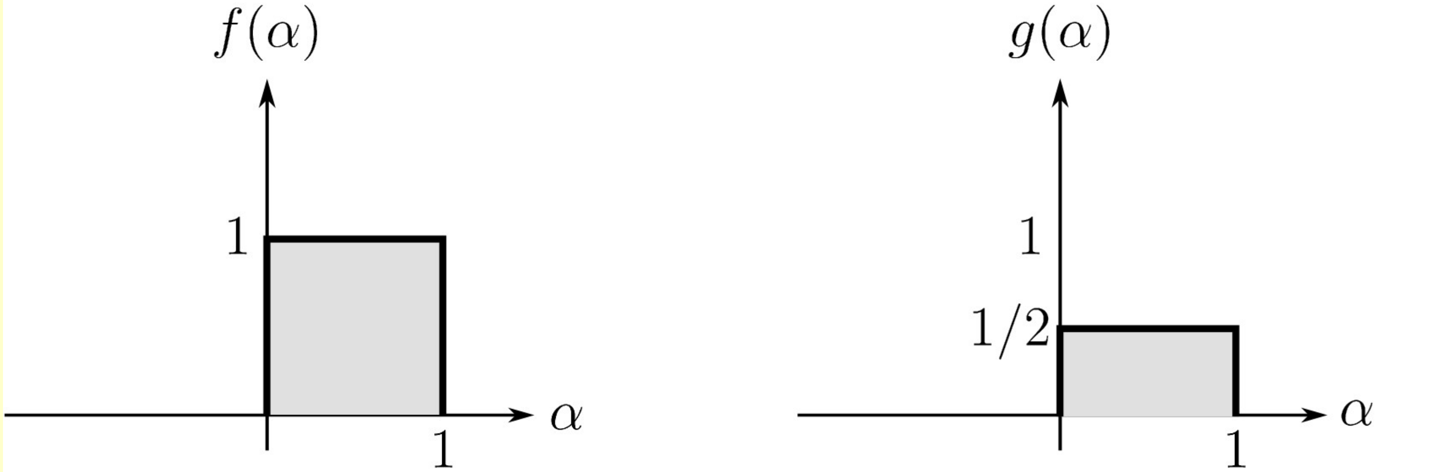




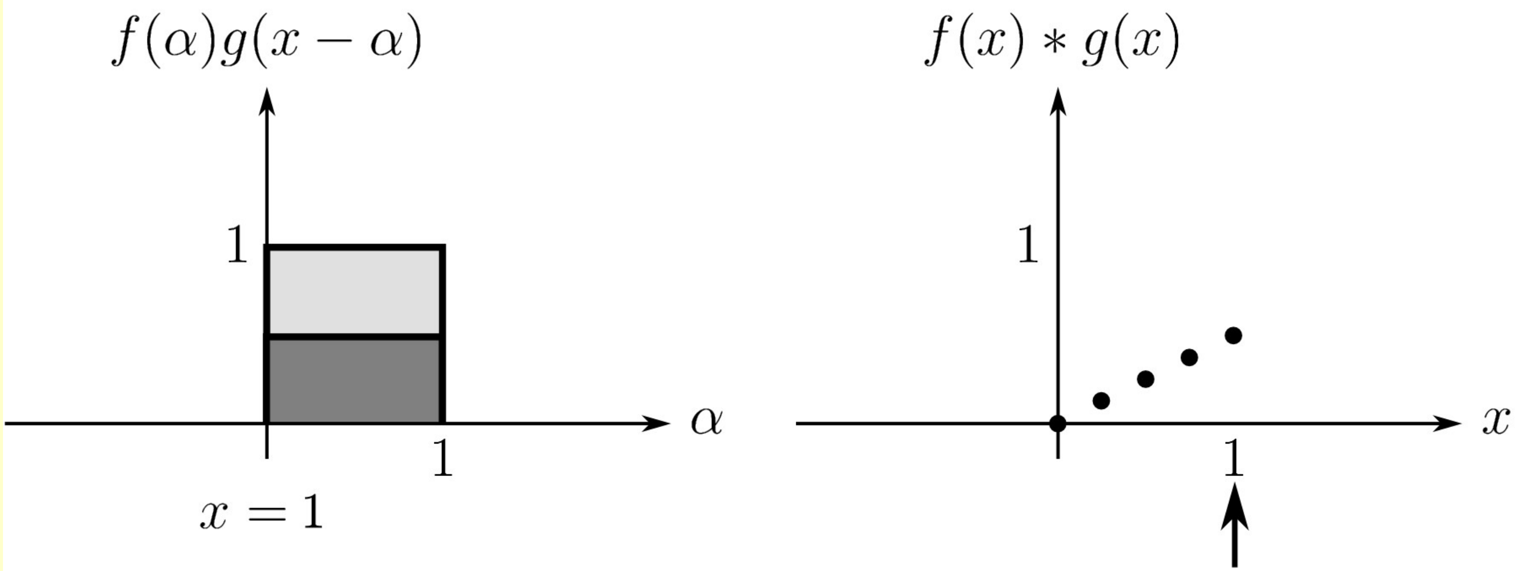
$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\alpha) g(x - \alpha) d\alpha$$

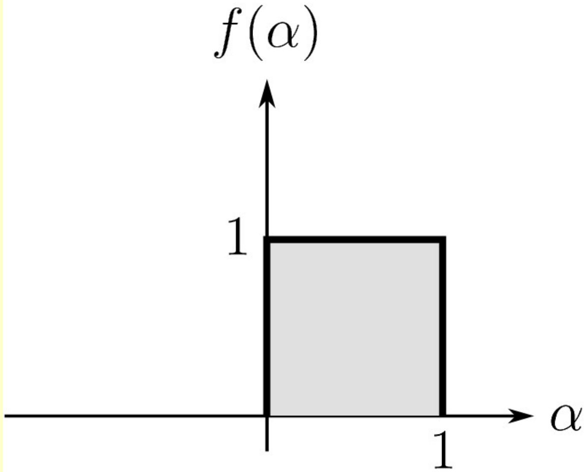


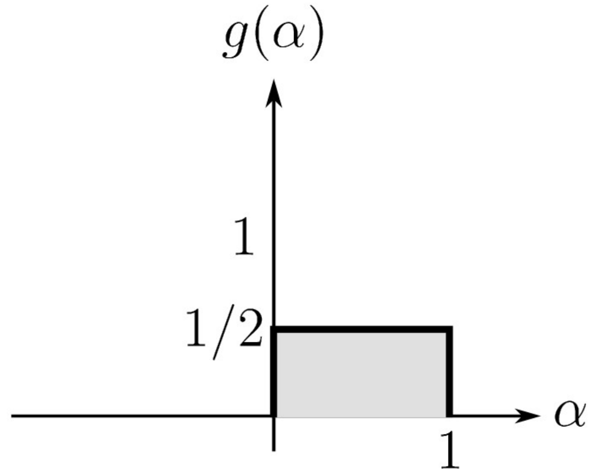




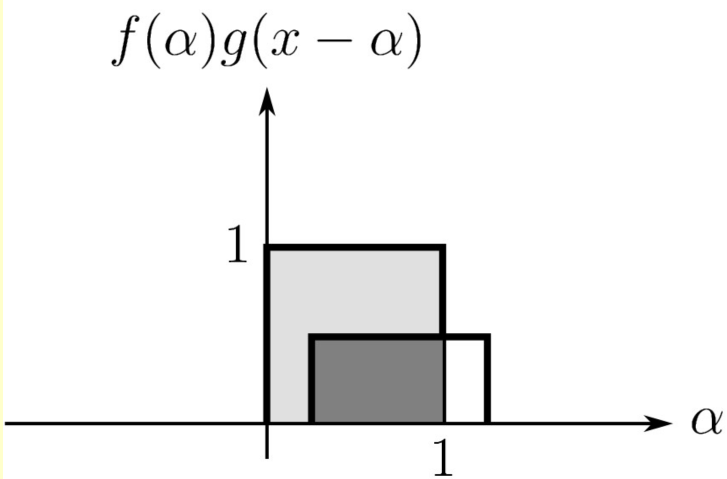
$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\alpha) g(x-\alpha) d\alpha$$



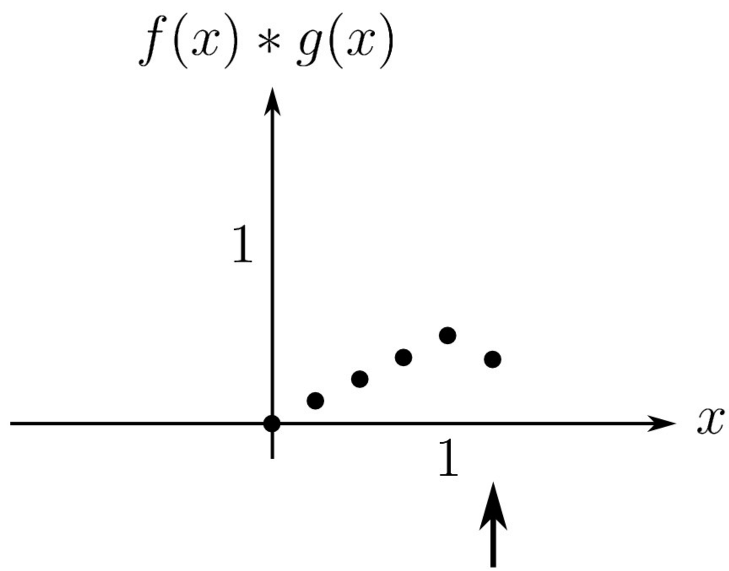


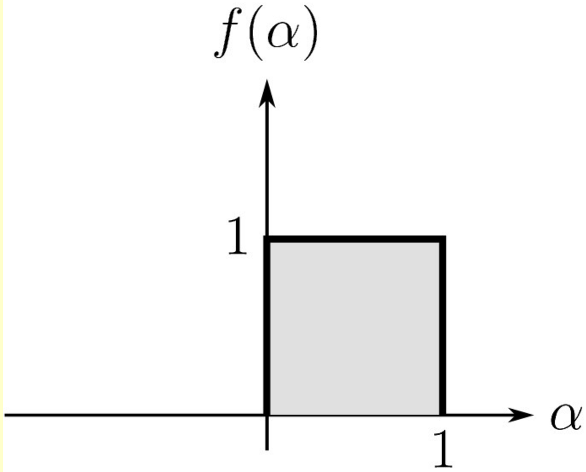


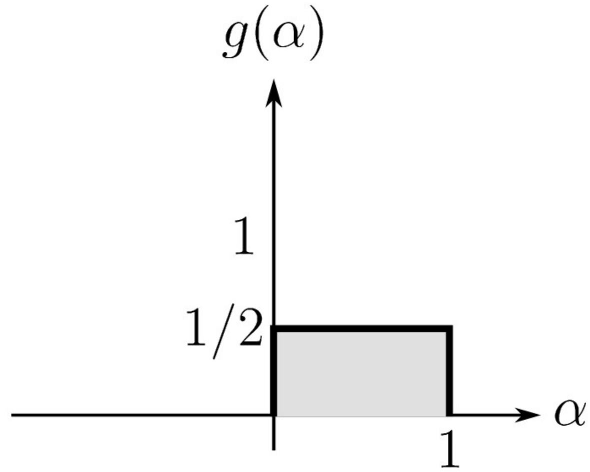
$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\alpha) g(x-\alpha) d\alpha$$



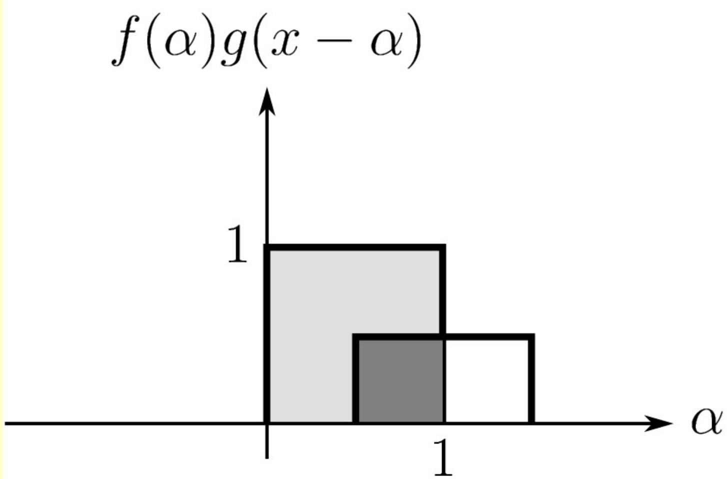
$x = 1.25$



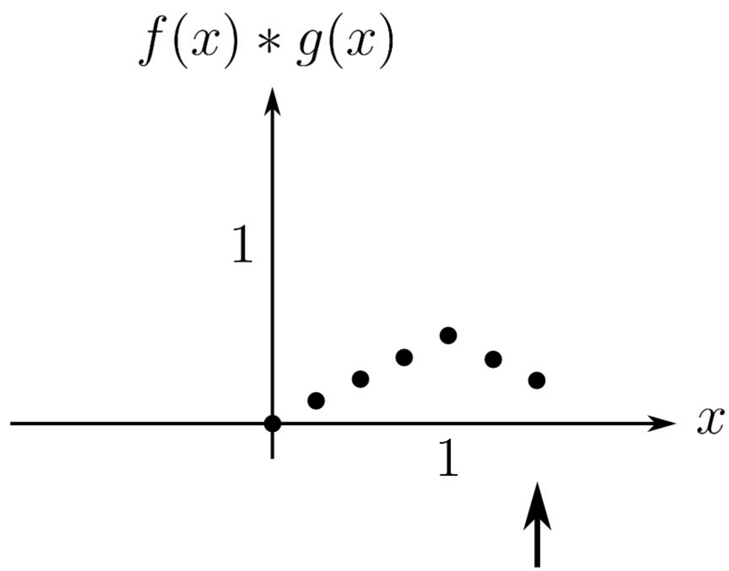


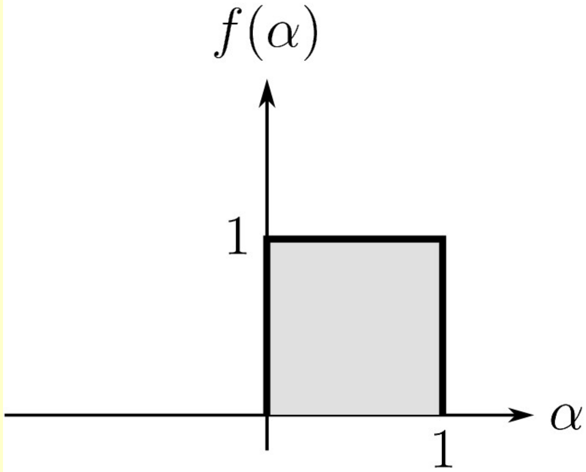


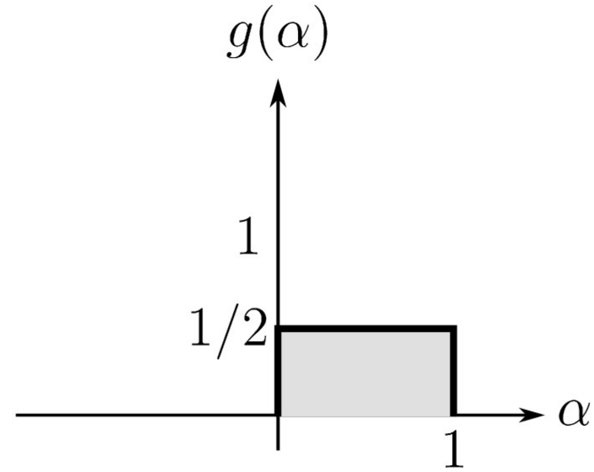
$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\alpha) g(x-\alpha) d\alpha$$



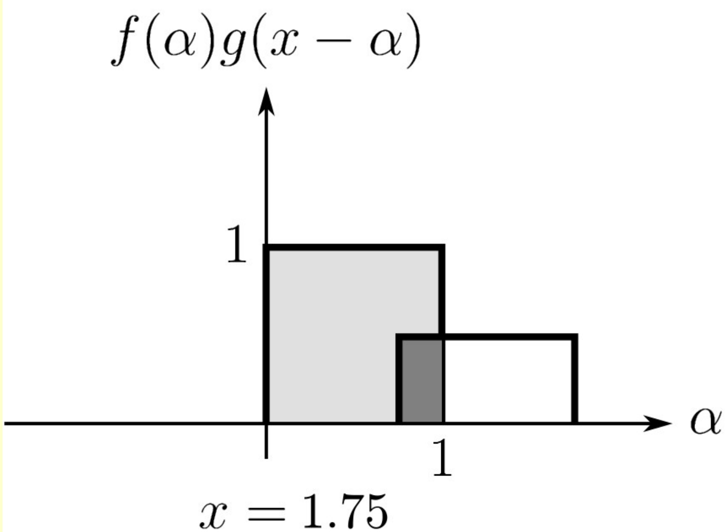
$x = 1.5$

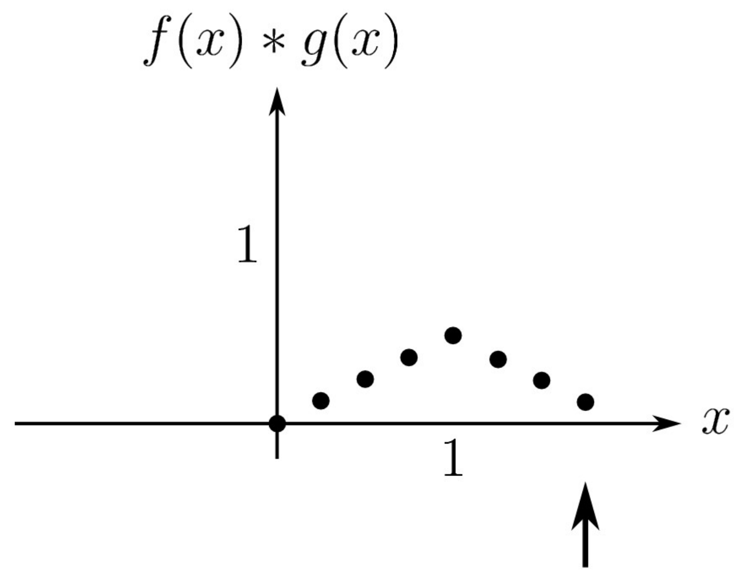


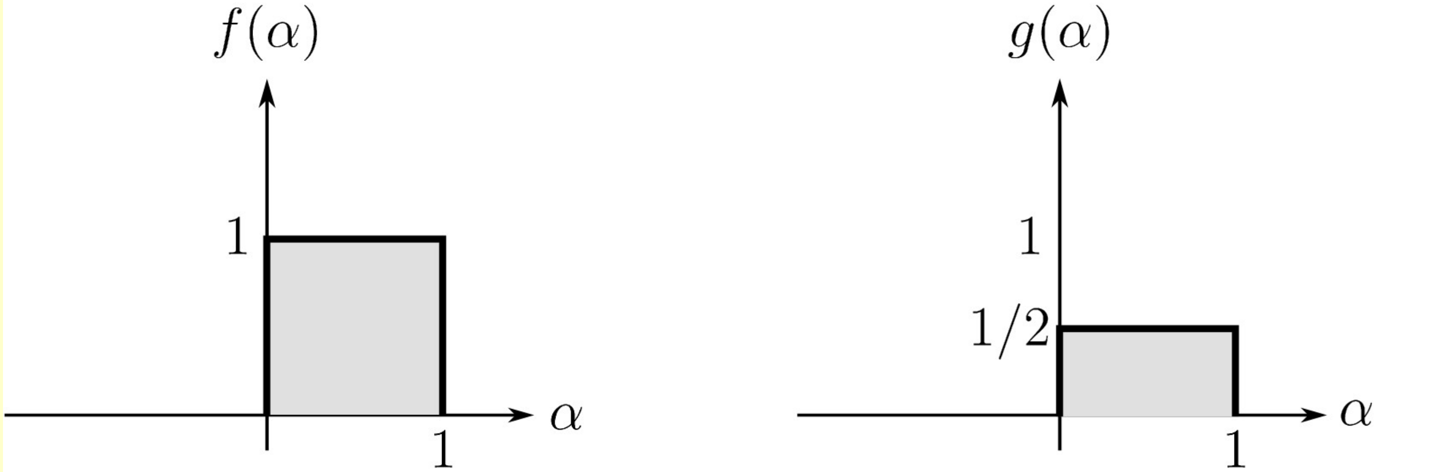




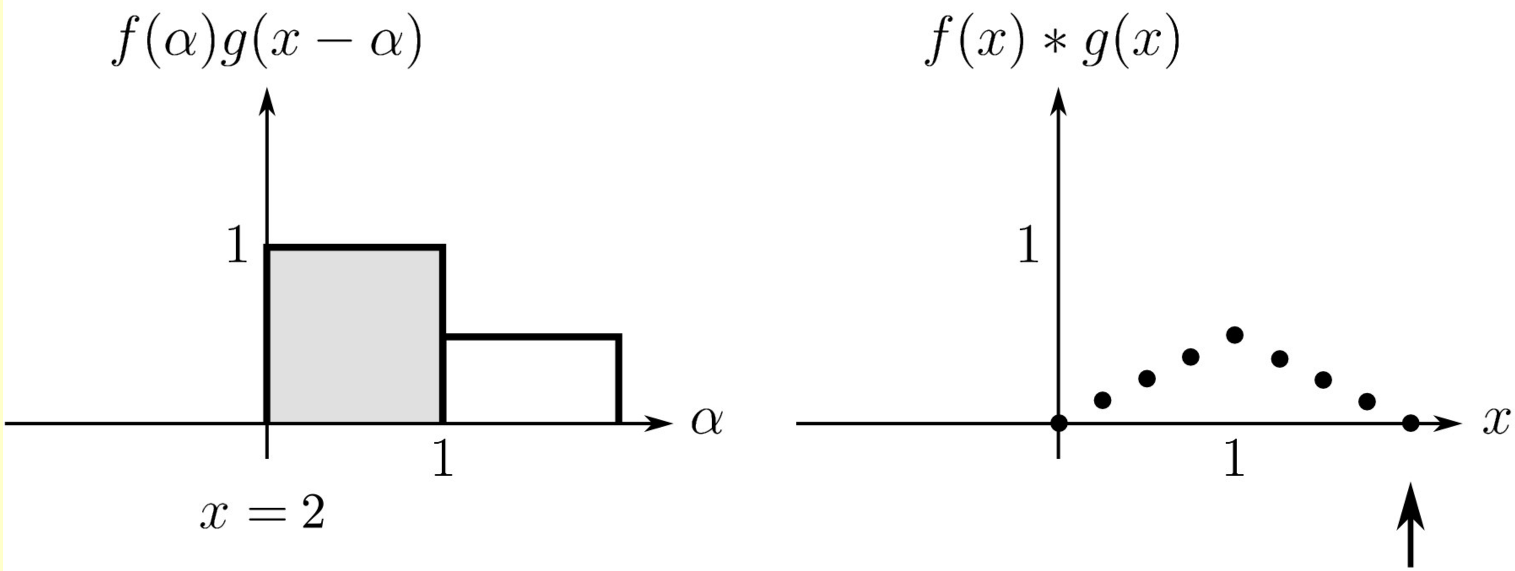
$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\alpha) g(x-\alpha) d\alpha$$







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

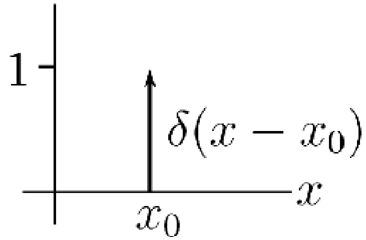


Convolution with an Impulse Function

Convolution with an Impulse Function

The special case of convoluting a function $f(x)$ with an *Impulse Function* $\delta(x - x_0)$ is of particular interest as will be shown later.

Definition: The *Impulse Function (Dirac delta function)* is often referred to as the unit impulse function introduced by the physicist Paul Dirac. The function $\delta(x - x_0)$ may be viewed as having an area of unity in an infinitesimal small neighbourhood about x_0 and zero elsewhere; that is,



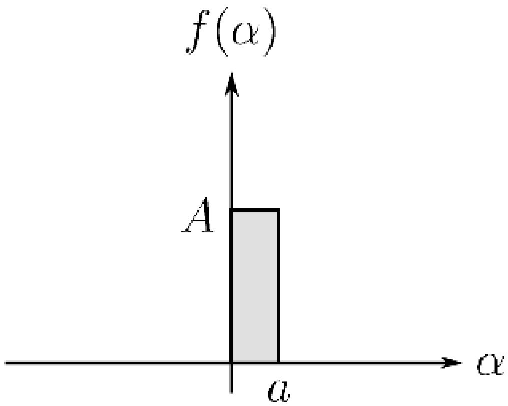
Dirac Impulse

Sifting Theorem

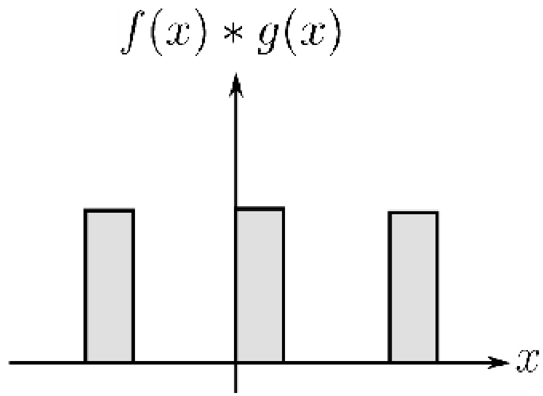
$$\int_{-\infty}^{\infty} \delta(x - x_0)dx = 1 \quad \text{and thus} \quad \int_{-\infty}^{\infty} f(x)\delta(x - x_0)dx = f(x_0)$$

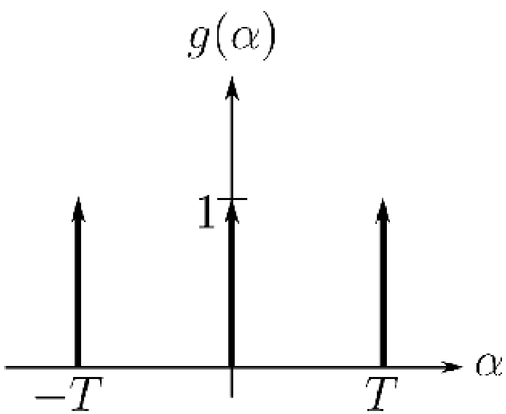
It is common practice to graphically represent the Dirac impulses as arrows at x_0 with a height equal to the impulse strength (area).

Convolution with an Impulse Function (2)



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





This important relationship will be used again in the sampling and quantisation section.

Discrete Convolution

Discrete Convolution

Suppose that, instead of being continuous, $f(x), g(x)$ are discretised into sampled arrays of size A and B

$$f(x) : \{f(0), f(1), f(2), ..., f(A-1)\}$$

$$g(x) : \{g(0), g(1), g(2), ..., g(B-1)\}$$

With $M \geq A + B - 1$ the discrete convolution can be defined as

$$f(x) * g(x) = \frac{1}{M} \sum_{m=0}^{M-1} f(m) g(x-m)$$

Because M is bigger than A and B they must be padded with zeros, so that both are of length M.

Two-Dimensional Continuous Convolution

Two-Dimensional Continuous Convolution

The *2D Convolution* is analogous to the 1D, thus

$$f(x,y) * g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)g(x-\alpha,y-\beta)d\alpha d\beta$$

The *Convolution Theorem* in two dimensions can then be expressed as

$$f(x,y) * g(x,y) \quad \Leftrightarrow \quad F(u,v)G(u,v)$$

and

$$f(x,y)g(x,y) \quad \Leftrightarrow \quad F(u,v) * G(u,v)$$

Two-Dimensional Discretised Convolution

Two-Dimensional Discretised Convolution

The ***discretised 2D Convolution*** is defined by

$$f(x,y) * g(x,y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) g(x-m, y-n)$$

where $A \times B$ and $C \times D$ are the discretised arrays of $f(x,y)$ and $g(x,y)$, respectively.

Wraparound error in the individual convolutions is avoided by choosing

$$M \geq A + C - 1$$

and

$$N \geq B + D - 1$$

Calculating the discrete convolution in the frequency domain is often more efficient than directly using the equation above.

FFT Computational Complexity

Computational Complexity

The number of complex multiplications and additions required to implement the Discrete Fourier Transform

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{-j2\pi ux/N}$$

is proportional to $\mathcal{O}(N^2)$,

- as for each of the N values of u the expansion of $\sum$ requires N complex multiplications of $f(x)$ by $e^{-j2\pi ux/N}$
- as the terms $e^{-j2\pi ux/N}$ can be precalculated and tabulated they are not counted in the complexity analysis

Computational Complexity (2)

Proper decomposition can reduce the number of multiplications and addition proportional to $\mathcal{O}(N \log_2 N)$. This decomposition is called the *fast Fourier Transform (FFT)* algorithm.

N	N^2	$N \log_2 N$	$N / \log_2 N$
	DFT	FFT	
32	1'024	160	6.40
64	4'096	384	10.67
128	16'384	896	18.29
256	65'536	2'048	32.00
512	262'144	4'608	56.89
1024	1'048'576	10'240	102.40
2048	4'194'304	22'528	186.18
4096	16'777'216	49'152	341.33
8192	67'108'864	106'496	630.15

Example:

- Let's assume that an FFT of size 8'192 takes on one particular machine 1s. Using the DFT method the same Fourier Transform would require 10min30s.

Derivation of the DDT Algorithm

Derivation of the FFT Algorithm

The FFT algorithm developed in the next few slides is based on the *successive doubling* method. We start with the general form of the DFT

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)e^{-j2\pi ux/N} \quad \forall u = 0, 1, .., N-1 \tag{3.30}$$

and rewrite it in the form

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)W_N^{ux} \tag{3.31}$$

$$W_N = e^{-j2\pi/N} \tag{3.32}$$

and N is assumed to be of the form $N = 2^n$ where n is a positive integer.

The requirement that N must be a power of 2 **does not** limit generality of the algorithm, as one can always achieve this requirement by zero-padding the data to the next power of 2.

As N is a power of 2 we can express it as

$$N = 2M \tag{3.33}$$

where M is also a positive integer. Substitution into Eq. 3.31 yields

$$\begin{aligned} F(u) &= \frac{1}{2M} \sum_{x=0}^{2M-1} f(x) W_{2M}^{ux} \\ &= \frac{1}{2} \left[\frac{1}{M} \sum_{x=0}^{M-1} f(2x) W_{2M}^{u(2x)} + \frac{1}{M} \sum_{x=0}^{M-1} f(2x+1) W_{2M}^{u(2u+1)} \right] \end{aligned}$$

From Eq. 3.32 we know that $W_{2M}^{2ux} = W_M^{ux}$, so the previous equation can be expressed as

$$F(u) = \frac{1}{2} \left[\frac{1}{M} \sum_{x=0}^{M-1} f(2x) W_M^{ux} + \frac{1}{M} \sum_{x=0}^{M-1} f(2x+1) W_M^{ux} W_{2M}^{u} \right] \tag{3.36}$$

Defining

$$F_{even}(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(2x)W_M^{ux} \quad \forall\ u = 0, 1, 2, ..., M-1 \tag{3.37}$$

and

$$F_{odd}(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(2x+1)W_M^{ux} \quad \forall\ u = 0, 1, 2, ..., M-1 \tag{3.38}$$

Eq. 3.36 reduces to

$$F(u) = \frac{1}{2}[F_{even}(u) + F_{odd}(u)W_{2M}^{u}] \tag{3.40}$$

Also, since $W_M^{u+M} = W_M^u$ and $W_{2M}^{u+M} = -W_{2M}^u$ the above equations get

$$F(u+M) = \frac{1}{2}\left[F_{even}(u) - F_{odd}(u)W_{2M}^u\right] \tag{3.41}$$

Carefull analysis of Eq. 3.37-3.41 reveals some interesting properties of these expressions.

An N-point Fourier Transform can be computed by evaluating two $N/2$-point Fourier Transforms.

The resulting values of $F_{even}(u)$ and $F_{odd}(u)$ are substituted into Eq. 3.40 to obtain $F(u)$ for $u = 0, 1, 2, ..., (N/2-1)$. The other half then follows directly from Eq. 3.41 without additional transform evaluations.

If $F_{even}(u)$ and $F_{odd}(u)$ are recursively split further we finally end up with a computational complexity for the Fast Fourier Transform (FFT) of

complex multiplications: $m(n) = \frac{1}{2}Nn$

complex additions: $a(n) = Nn$

The complexity of FFT is thus in the order of $\mathcal{O}\{Nlog_2(N)\}$.

The Inverse FFT

The Inverse FFT

On the previous slide we developed a fast implementation for the *Fourier Transform*, but what about the *Inverse Fourier Transform?*

The reason is that any method implementing the forward transform can also be used to compute the inverse. To show this let us repeat the equations for the DFT and inverse DFT

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{-j2\pi ux/N} \qquad (3.46)$$

$$f(x) = \sum_{u=0}^{N-1} F(u) e^{j2\pi ux/N} \qquad (3.47)$$

Taking the complex conjugate of Eq. 3.47 and dividing both sides by N yields

$$\frac{1}{N} f^*(x) = \frac{1}{N} \sum_{u=0}^{N-1} F^*(u) e^{-j2\pi ux/N} \qquad (3.49)$$

Comparing the result with Eq. 3.46 shows that the right hand side of Eq. 3.49 is in the form of the forward Fourier Transform. Thus inputting $F^*(u)$ into an algorithm to compute the forward transform gives $f^*(x)/N$ that can be easily converted to $f(x)$.