

# Frequency Domain

نسیم فانی

اطلاعات گزارش	چکیده
تاریخ: 99/8/19	در این تمرین، به بررسی توابع، تبدیلات و فیلترینگ در حوزه فرکانس می‌پردازیم.
واژگان کلیدی:	
تبدیل فوریه	
فیلترینگ	
طیف	
DFT	
IDFT	

## 1-مقدمه

در دامنه فرکانس، یک تصویر دیجیتال از دامنه spatial به دامنه فرکانس تبدیل می‌شود. در دامنه فرکانس، از فیلتر کردن تصویر برای بهبود تصویر برای یک برنامه خاص استفاده می‌شود. Transformation سریع فوریه ابزاری است که برای تبدیل دامنه spatial به دامنه فرکانس استفاده می‌شود. برای smooth کردن تصویر، low filter و برای وضوح تصویر، فیلتر عبور بالا اجرا می‌شود. وقتی هر دو فیلتر اجرا می‌شوند، از نظر فیلتر ایده آل، فیلتر Butterworth و فیلتر Gaussian مورد تجزیه و تحلیل قرار می‌گیرند.

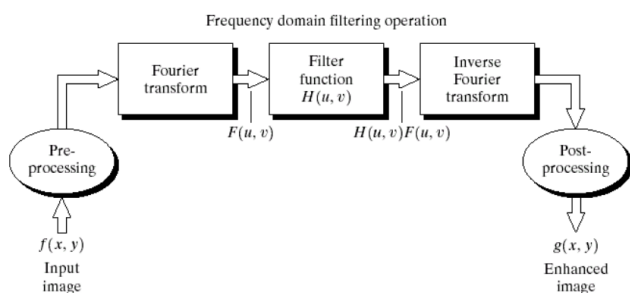
دامنه فرکانس فضایی است که با تبدیل فوریه تعریف می‌شود. تبدیل فوریه کاربرد بسیار گسترده‌ای در پردازش تصویر دارد. تجزیه و تحلیل دامنه فرکانس برای نشان دادن

چگونگی توزیع انرژی سیگنال در یک دامنه فرکانس استفاده می‌شود.

## 2-توضیحات تکنیکال

4.1.1 - اساس فیلتر کردن در حوزه فرکانس به صورت بلوک دیاگرام زیر است:

تبدیل فوریه تصویر ورودی بعد از پیش پردازش، در حوزه فرکانس در پاسخ فرکانسی فیلتر مورد نظر ضرب شده و با تبدیل فوریه معکوس، تصویر نهایی به دست می‌آید.



فرمول DFT دوبعدی به صورت زیر است:

$$F(u) = \sum_{x=0}^{M-1} f(x) e^{-j2\pi ux/M} \quad u = 0, 1, 2, \dots, M-1$$

این عمل در متلب به کمک تابع fft2 انجام می‌شود.

- اعمال فیلتر حقیقی با مبدا مرکز (P/2) و (Q/2) با ضرب آرایه ای
- بدست آوردن IDFT حاصل

فرمول IDFT دوبعدی به صورت زیر است:

$$f(x) = \frac{1}{M} \sum_{u=0}^{M-1} F(u) e^{j2\pi ux/M} \quad x = 0, 1, 2, \dots, M-1$$

این عمل در متلب به کمک تابع ifft2 انجام می‌شود.

- 2-1 • انتخاب بخش حقیقی نتیجه (چون با تغییر ضرایب ممکن است بخش موهومی هم به تصویر اضافه شده باشد که قابل صرف نظر کردن است)
- 2-2 • ضرب کردن نتیجه در  $(-1)^{x+y}$  برای بدست آوردن  $g_p$
- 2-3 • برداشتن بخشی به اندازه  $M \times N$  از گوشه ی بالای سمت چپ  $g_p$

#### فیلتر a:

این فیلتر جداپذیر است:

$$\frac{1}{4} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

این فیلتر برای smoothing به کار می‌رود و فیلتر گاوسین میانگین گیر می‌باشد.

#### فیلتر b:

این فیلتر جدایی‌ناپذیر است.

فیلترهای مکانی که قبلاً استفاده شد، می‌تواند در حوزه فرکانس نیز انجام گیرد.

در حوزه مکانی عمل فیلترینگ با کانولوشن تصویر و ماسک فیلتر به دست می‌آید. در حوزه فرکانس، این کار با ضرب تبدیل فوریه فیلتر و تصویر به دست می‌آید.

مراحل فیلترینگ در حوزه فرکانس به شرح زیر است:

- بدست آوردن P و Q با توجه به اندازه  $M \times N$  تصویر f. عموماً  $M=P2$  و  $N=Q2$  انتخاب می‌شود.

- ساختن تصویر  $f_p$  با عمل افزودن صفر به تصویر f.

نیاز به Padding: ضرب در حوزه فرکانس معادل کانولوشن در حوزه زمان است. همانطور که گفته شد، می‌توان تصویر و تبدیل فوریه آن را متناوب فرض کرد که دوره تناوب به اندازه ابعاد تصویر است. این مسئله می‌تواند مشکل ایجاد کند.

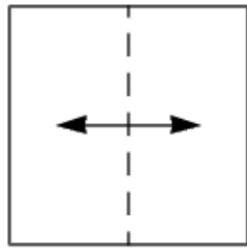
برای برطرف کردن این مشکل می‌توان با استفاده از Padding طول سیگنال و در نتیجه دوره تناوب را تغییر داد به نحوی که کانولوشن سیگنالهای متوالی روی هم تاثیر نداشته و بتوان نتیجه کانولوشن را بدون تداخل استخراج کرد.

- ضرب کردن  $f_p$  در  $(-1)^{x+y}$  برای انتقال مبدا به مرکز

در متلب تابع fftshift فیلتر حوزه فرکانسی را از حالت متمرکز خارج ساخته و به فرم مناسب برای فیلترینگ تبدیل می‌کند.

- بدست آوردن DFT حاصل

For dim = 2:



برای مثال در یک آرایه ی یک بعدی داریم:

$$x = [5 \quad 4 \quad 3 \quad 2 \quad 1]$$

$$y = \text{fftshift}(x)$$

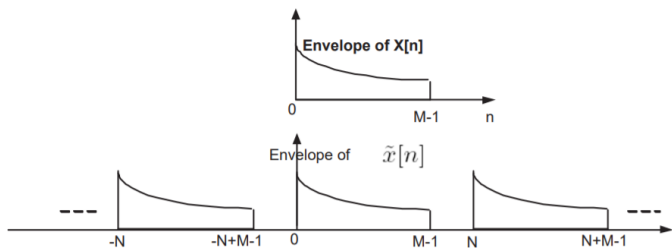
$$y = [4 \quad 5 \quad 1 \quad 2 \quad 3]$$

این کار برای برای visualizing تبدیل فوریه با مولفه فرکانس صفر در وسط طیف مفید است. در تبدیل فوریه تصویر دیجیتالی، پیکسل های با فرکانس پایین در مرکز تبدیل فوریه و پیکسل های با فرکانس بالا در کناره های تبدیل فوریه ظاهر می شوند.

اگر از تصویری تبدیل فوریه بگیریم و آن را شیفته دهیم، مرکز تصویر نشان دهنده فرکانس های بالا (لبه ها) هستند و بالعکس با دورتر شدن، فرکانس ها کمتر می شوند.

#### -4.2.1

(1) محاسبه linear convolution با DFT:



باید طوری طول هر ورودی را گسترش دهیم که داشته باشیم:

$$N \geq M = (L + P - 1)$$

پس:

این فیلتر برای edge detection به کار می رود و لبه ها را در تمام جهات با مشتق دوم بدست می آورد پس فیلتر لاپلاسین در 8 جهت می باشد.

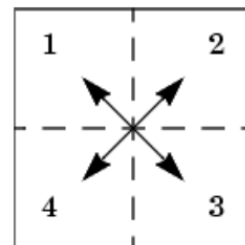
#### فیلتر c:

این فیلتر جدایی ناپذیر است.

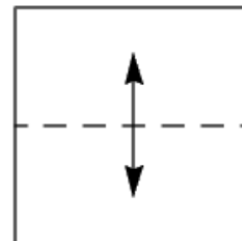
این فیلتر برای edge enhancement به کار می رود و مشتق سوم در 4 جهت است.

#### 4.1.2 - MATLAB fftshift, fftshift با Y

انتقال جز component فرکانس صفر به مرکز آرایه، خروجی های fft، fft2 و fftn را rearrange می کند. اگر X یک بردار باشد، پس fftshift نیمه چپ و راست X را مبادله می کند. اگر X ماتریس است، fftshift ربع اول X را با سوم و ربع دوم را با چهارم عوض می کند. اگر X یک آرایه چند بعدی است، پس fftshift نیم فاصله X را در امتداد هر بعد عوض می کند.



For dim = 1:



پس طبق آنچه گفته شد تصویر و فیلتر  $H$  و  $F$  هردو باید به اندازه‌ای zero pad شوند که  $266 \times 266$  شوند:

$$(256+11-1) \times (256+11-1)$$

سپس عمل ضرب را انجام می‌دهیم:

$$G = \text{DFT}(F) \times \text{DFT}(H)$$

در نهایت با کمک IDFT حاصل را به دست می‌آوریم:

$$G = \text{DFT}(F) \times \text{DFT}(H)$$

2) مطابق آنچه گفته شد  $Y$  خروجی حاصل از اعمال فیلتر بر روی تصویر در حوزه spatial است و مطابق فرمول زیر محاسبه می‌شود:

$$Y(m, n) = \sum_{k=0}^{255} \sum_{l=0}^{255} F(k, l) H(m - k, n - l)$$

همچنین  $Z$  خروجی اعمال فیلتر بر روی تصویر در حوزه-ی فرکانس است و مطابق فرمول زیر محاسبه می‌شود:

$$Z(m, n) = \sum_{k=0}^{255} \sum_{l=0}^{255} F(k, l) H(((m - k))_{256}, ((n - l))_{256})$$

در بازه‌ی  $[0, 255] \times [0, 255]$  ،  $Y(m, n)$  و  $Z(m, n)$

باهم برابر می‌شوند اگر شروط زیر باشد:

$$10 \leq m \leq 255$$

$$10 \leq n \leq 255$$

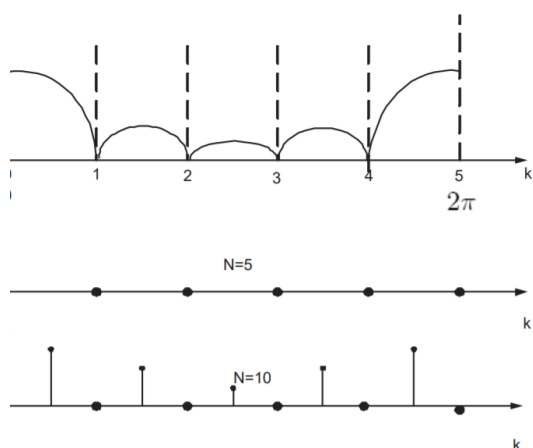
$$H(((m - k))_{256}, ((n - l))_{256}) = H(m - k, n - l)$$

4.2.2- به بررسی هر فیلتر به صورت جدا می‌پردازیم:

در قسمت a فیلتر فرکانس‌های پایین را عبور می‌دهد و فرکانس‌های بالا را تضعیف می‌کند پس یک فیلتر پایین گذر است.

فیلترهای پایین گذر (Low Pass Filter) با عبور از پیکسل‌های با فرکانس کم، بر روی پیکسل‌های با فرکانس بالا تغییرات ایجاد می‌کنند. نتیجه اعمال

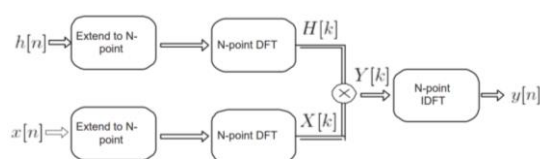
$$h[n] \otimes x[n] = h[n] * x[n]$$



ملاحظات در رابطه با zero padding:

- تصویر DTFT را بهبود می‌بخشد
- وضوح طیفی (spectral resolution) را افزایش نمی‌دهد و leakage را کاهش نمی‌دهد.

مراحل Linear Convolution با DFT:



- انتخاب  $N$  به طوری که حداقل

$(L + P - 1)$  باشد.

- Pad کردن ورودی‌ها با صفر تا به اندازه  $N$  برسد.

- محاسبه  $N$ -point DFT برای  $H[k]$  و  $X[k]$  دست آوردن

- بدست آوردن حاصل:

$$Y[k] = H[k]X[k] \quad k = 0, \dots, (N - 1)$$

- محاسبه‌ی  $y[n]$  با  $N$ -point IDFT از  $Y[k]$  به روش زیر:

- محاسبه‌ی DFT برای  $Y^*[k]$

- محاسبه‌ی مزدوج عدد مختلط

- ضرب آن در  $1/N$

- ذخیره اولین  $(L + P - 1)$  از  $y[n]$ .

فیلترهای پایین گذر ، تصویری آرام خواهد بود ( لبه ها و نویزها تا حدودی آرام می شوند )

در قسمت b فیلتر فرکانس های بالا را عبور داده و فرکانس های پایین را تضعیف می کند پس یک فیلتر بالاگذر است.

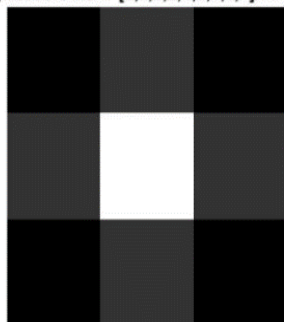
فیلترهای بالاگذر ( High Pass Filter ) با عبور از پیکسل های با فرکانس بالا ، بر روی پیکسل های با فرکانس پایین تغییرات ایجاد می کنند. اعمال فیلتر بالاگذر نیز تصویری با جزئیات بیشتر به دست می آید. به این فیلترها آشکارکننده لبه ها نیز میگویند.

### 3-شکل ها، جدول ها و روابط (فرمول ها)

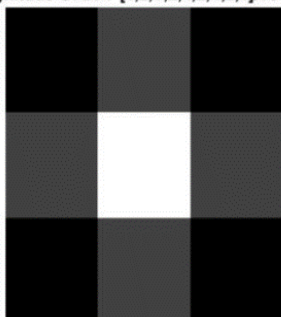
#### 4.1.1

magnitude هر فیلتر به شرح زیر است:

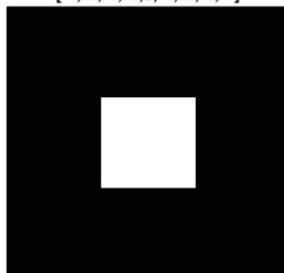
Magnitude Of DFT [1,2,1;2,4,2;1,2,1]/16+Shift



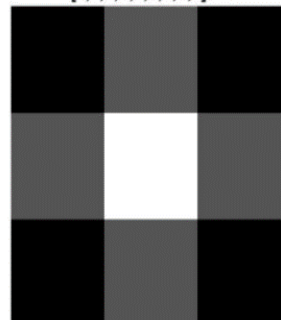
Magnitude Of DFT [1,2,1;2,4,2;1,2,1]/16+Log



[-1,-1,-1;-1,8,-1;-1,-1,-1]



[1,2,1;2,4,2;1,2,1]/16



Magnitude Of DFT [1,2,1;2,4,2;1,2,1]/16



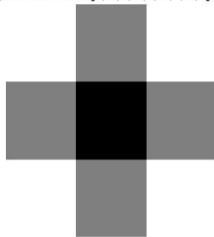
Magnitude Of DFT [0,-1,0;-1,5,-1;0,-1,0]



Magnitude Of DFT [-1,-1,-1;-1,8,-1;-1,-1,-1]



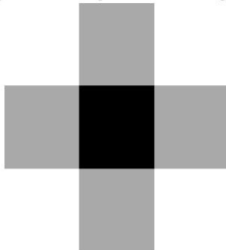
Magnitude Of DFT [0,-1,0;-1,5,-1;0,-1,0]\*Shift



Magnitude Of DFT [-1,-1,-1;-1,8,-1;-1,-1,-1]\*Shift



Magnitude Of DFT [0,-1,0;-1,5,-1;0,-1,0]\*Log



Magnitude Of DFT [-1,-1,-1;-1,8,-1;-1,-1,-1]\*Log



[0,-1,0;-1,5,-1;0,-1,0]

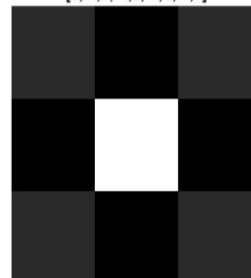


Figure 1 - تصویر اصلی

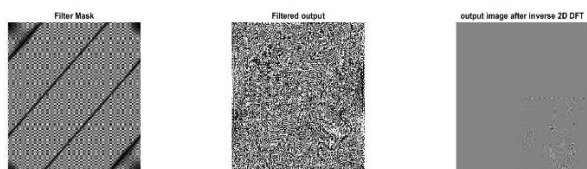


Figure 2 - تصویر فیلتر شده با فیلتر 1



Figure 4 - تصویر فیلتر شده با فیلتر 3

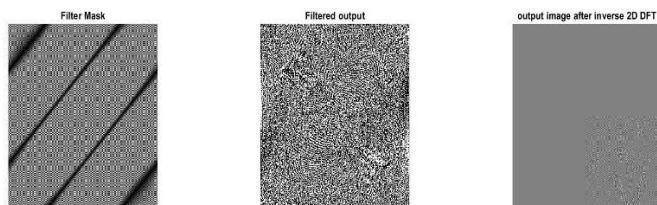
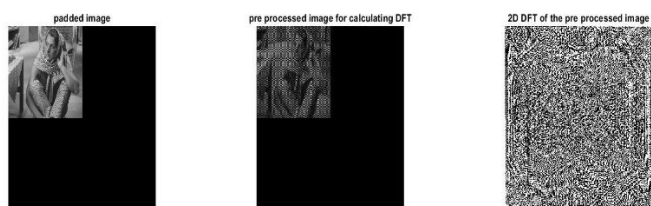
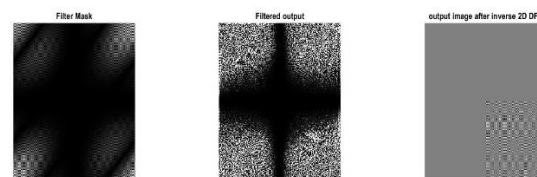
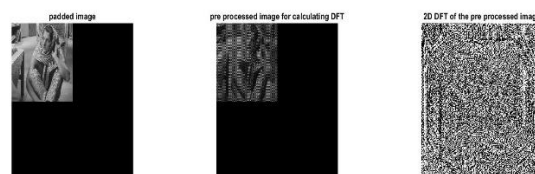
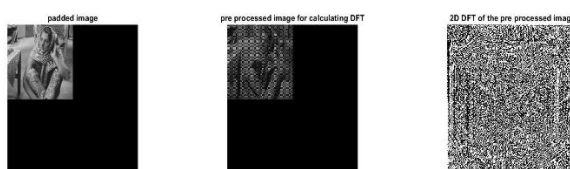


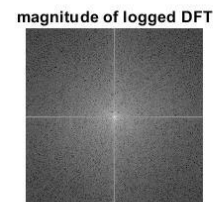
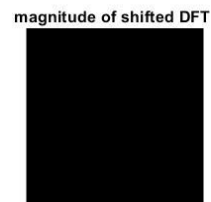
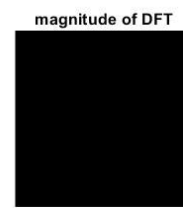
Figure 3 - تصویر فیلتر شده با فیلتر 2



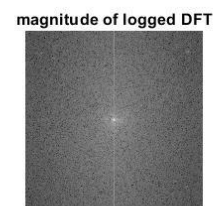
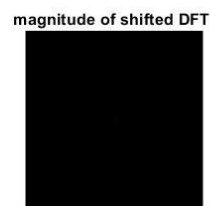
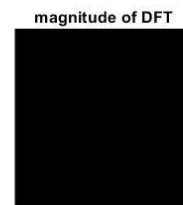
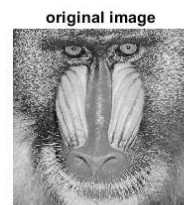
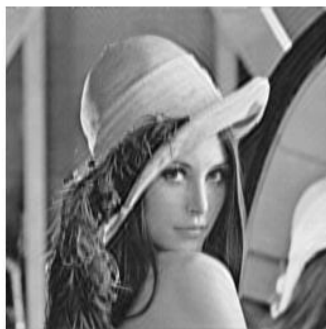
$T=1/4$



-4.1.2

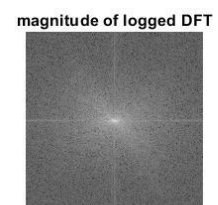
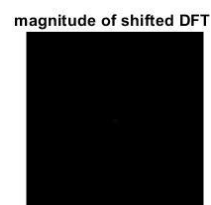
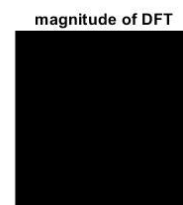


$T=1/8$



قسمت (b)  
-1

$T=1/4$



-4.2.2

قسمت (a)



T=1/4



T=1/8



T=1/8



T=1/4



-2

-4

T=1/4



T=1/8



-3

T=1/8



## 4.2.2

```
h1 = eshift.*zshift;
hh = ifftshift(h1);
h2 = ifft2(hh);
rh2 = real(h2);
h3 = zeros(p,q);
for i = 1:p
    for j = 1:q
        h3(i,j) = rh2(i,j).*((-1).^(i+j));
    end
end

out = zeros(m,n);
for i = 1:m
    for j = 1:n
        out(i,j) = h3(p-1,q-j);
    end
end
imshow(im2double(out));
end
```

```
function customFilter(img,t)
[m,n]=size(img);
N = n;
fimg = fft2(img);
filter = ones(m,n);
%a
for i=1:m
    for j=1:n
        if(i>t*N && j>t*N && i<(1-t)*N && i<(1-t)*N)
            filter(i,j)=0;
        end
    end
end

%b.1
for i=1:m
    for j=1:n
        if(i<=t*N && j<=t*N && i>0 && j>=0)
            filter(i,j)=0;
        end
    end
end

%b.2
for i=1:m
    for j=1:n
        if(i<=t*N && j<=N-1 && j>=(1-t)*N)
            filter(i,j)=0;
        end
    end
end

%b.3
for i=1:m
    for j=1:n
        if(i<=N && i>=(1-t)*N && j<=t*N)
            filter(i,j)=0;
        end
    end
end

%b.4
for i=1:m
    for j=1:n
        if(i>=(1-t)*N && j<=N-1)
            filter(i,j)=0;
        end
    end
end

fimg = fimg.*filter;
image = ifft2(fimg);
image = real(uint8(image));
figure;
imshow(image, []);
end
```

4- کدها:

4.1.2 و Main:

```
clear all;
img = imread('lena.bmp');
img = rgb2gray(img);
img = imresize(img,0.5);
f = customFilter(img,1/4);

img1 = imread('Barbara.bmp');
img1 = rgb2gray(img1);
img1 = im2double(img1);
mask1=[1,2,1;2,4,2;1,2,1]/16;
mask2 =[-1,-1,-1;-1,8,-1;-1,-1,-1];
mask3 =[0,-1,0;-1,5,-1;0,-1,0];
out=filtering(mask2,img1);

img2 = imread('Baboon.bmp');
img2 = rgb2gray(img2);
subplot(2,2,2),imshow(abs(fft2(img2)),[]),title("magnitude of DFT");
subplot(2,2,3),imshow(abs(fftshift(fft2(img2))),[]),title("magnitude of shifted DFT ");
subplot(2,2,4),imshow(log(abs(fftshift(fft2(img2)))).,[]),title("magnitude of logged DFT");
subplot(2,2,1),imshow(img2,[]),title("original image");
```

## 4.1.1

```
function out = filtering(filter,input)
[m,n] = size(input);
c = zeros(2*m,2*n);
[p,q] = size(c);
for i = 1:p
    for j = 1:q
        if i <= m && j<= n
            c(i,j) = input(i,j);
        else
            c(i,j) = 0;
        end
    end
end

d = zeros(p,q);
for i = 1:p
    for j = 1:q
        d(i,j) = c(i,j).*(-1).^(i + j);
    end
end

e = fft2(d);
eshift =fftshift(e);
z = zeros(p,q);
z(floor(p/2-1):floor(p/2+1),floor(q/2-1):floor(q/2+1))= filter;
zshift=fft2(z);
```

- <http://haranarasaratnam.com/docs/UQ4.pdf>
- [https://spinlab.wpi.edu/courses/ece503\\_2014/12-1linear\\_convolution\\_with\\_DFT.pdf](https://spinlab.wpi.edu/courses/ece503_2014/12-1linear_convolution_with_DFT.pdf)
- <https://uk.mathworks.com/help/signal/ug/linear-and-circular-convolution.html>
- <https://uk.mathworks.com/matlabcentral/fileexchange/53250-filtering-of-an-image-in-frequency-domain>
- <https://kgut.ac.ir/useruploads/1554342785847rqc.pdf>
- <http://matlab.izmiran.ru/help/techdoc/ref/fftshift.html>