TP1 (Image Processing)

Mohammed Akib Iftakher (20215281

Exercise 1 : Interpretation of frequency content

Exercise 1 : Interpretation of frequency content

Picture 1

Two images has been loaded

```
img1 = imread('lenna1.png');
img1;
imshow(img1 );
```

Picture 2

```
img2 = imread('cameraman.tif')
```

```
img2 = 256×256 uint8 matrix
   156   159   158   155   158   156   159   158   157   158   158   159   160 ···
   160   154   157   158   157   159   158   158   158   160   155   156   159
   156   159   158   155   158   156   159   158   157   158   158   159   160
   160   154   157   158   157   159   158   158   158   160   155   156   159
   156   153   155   159   159   155   156   155   155   157   155   154   154
   155   155   155   157   156   159   152   158   156   158   152   153   159
   156   153   157   156   153   155   154   155   157   156   155   156   155
   159   159   156   158   156   159   157   161   162   157   157   159   161
   158   155   158   154   156   160   162   155   159   161   156   161   160
   155   154   157   158   160   160   159   160   158   161   160   160   158
     ⋮
```

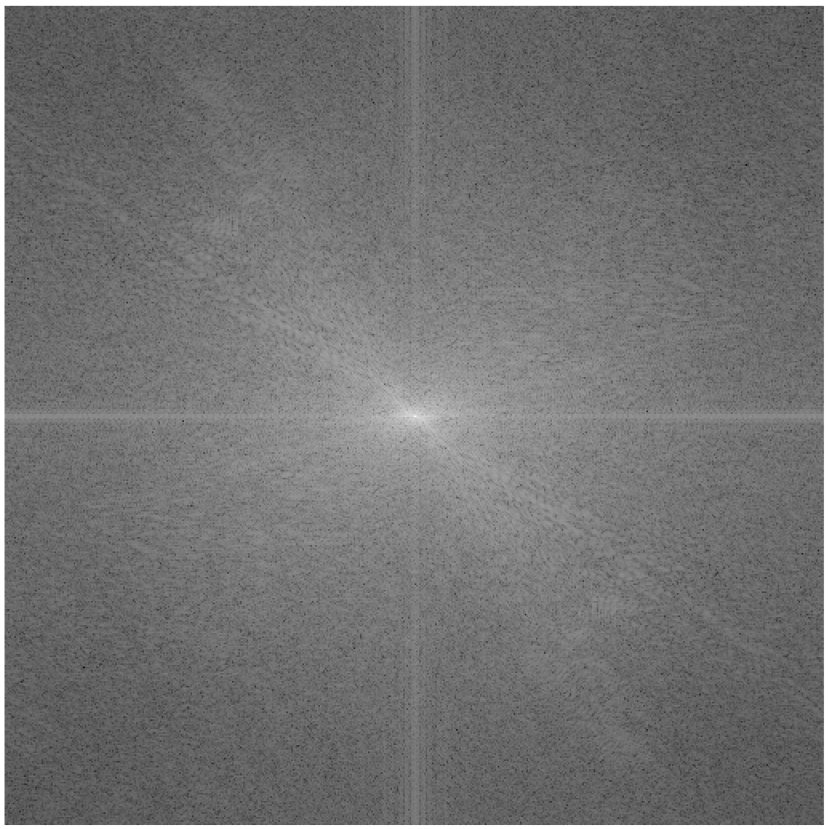
```
imshow(img2);
```

Fourier transform

```
ft1 = fft2(double(img1))
```

```
ft1 = 512×512 complex
10^7 ×
   2.5968 + 0.0000i  -0.0360 + 0.2828i   0.1536 - 0.0999i  -0.0135 + 0.0765i ···
  -0.0148 - 0.1135i  -0.1394 + 0.1872i  -0.1295 - 0.0254i  -0.0584 - 0.1888i
  -0.0641 + 0.0022i  -0.0286 - 0.0276i   0.0804 - 0.0590i  -0.0477 + 0.0770i
   0.0511 + 0.0325i   0.0063 - 0.0575i   0.0071 + 0.0591i   0.1125 - 0.0079i
  -0.0034 - 0.0573i   0.0179 - 0.0119i   0.0356 + 0.0422i  -0.0624 - 0.0096i
  -0.0203 - 0.0538i   0.0059 - 0.0219i   0.0302 - 0.0259i   0.0326 + 0.0363i
   0.0094 + 0.0047i  -0.0258 - 0.0384i   0.0166 + 0.0087i   0.0247 - 0.0221i
   0.0354 - 0.0064i  -0.0007 + 0.0135i   0.0044 + 0.0080i  -0.0010 + 0.0019i
  -0.0032 + 0.0025i   0.0032 + 0.0024i  -0.0087 + 0.0278i  -0.0076 - 0.0011i
   0.0106 - 0.0200i  -0.0168 - 0.0015i   0.0086 - 0.0010i  -0.0041 + 0.0001i
     ⋮
```

```
FS1 = fftshift(ft1);
S1 = log(1+abs(FS1));
imshow(S1, []);
```

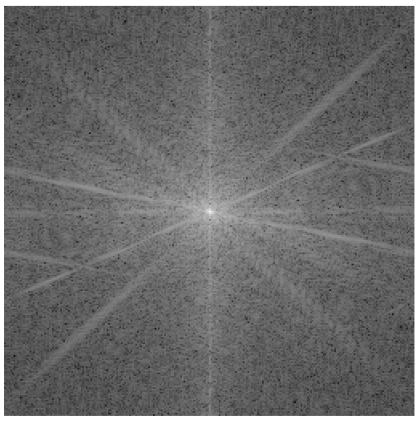


Inverse fourier transform

```
IFS1 = ifftshift(FS1);
ift1 = ifft2(IFS1);
```

image 2

```
ft2 = (fft2(double(img2)));
FS2 = fftshift(ft2);
S2 = log(1+abs(FS2));
imshow(S2, []);
```



Inverse fourier transform

```
IFS2 = ifftshift(FS2);
ift2 = ifft2(IFS2);
```

Where are the low frequencies, the high frequencies, the coefficients having the greatest amplitude?

From the image, it can be seen that the low frequencies are in the center where the high frequencies are outside. The coefficients having the greatest amplitude are in the middle. Low frequencies make up the bulk of the information (areas of low variation in intensity). High frequencies make up the edges and fine detail (areas of high variation in intensity). So, it can be stated that Low frequencies are near the origin and High frequencies are away from the origin.

```
[n,m]=size(f);
x = sum(sum(f))/(m*n);
y = ft1(1,1)/(m*n);
```

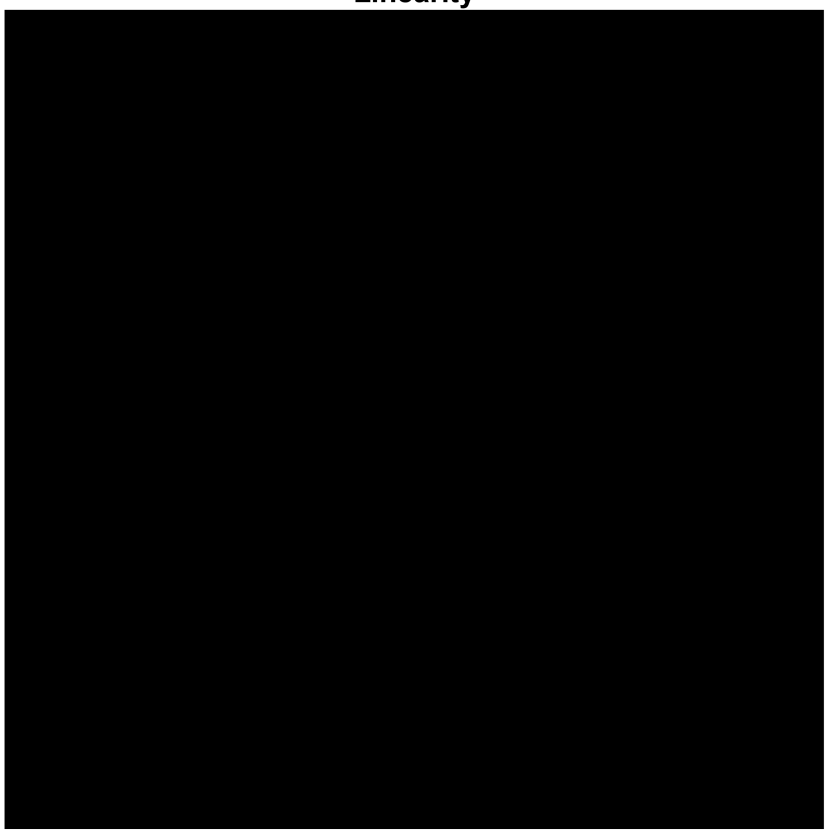
Exercise 2 : Properties of the Fourier transform

1. Linearity. Warning: please handle typing correctly before each arithmetic operation.

```
img3 = imread('image1.gif');
img4 = imread('image2.gif');

linear1 =fft2((img3))+fft2((img4));
total_image = double(img3)+double(img4);
linear2 =fft2(total_image);

diff = linear2-linear1;
imshow(diff);
```

```
Warning: Displaying real part of complex input.
```

```
title("Linearity")
```



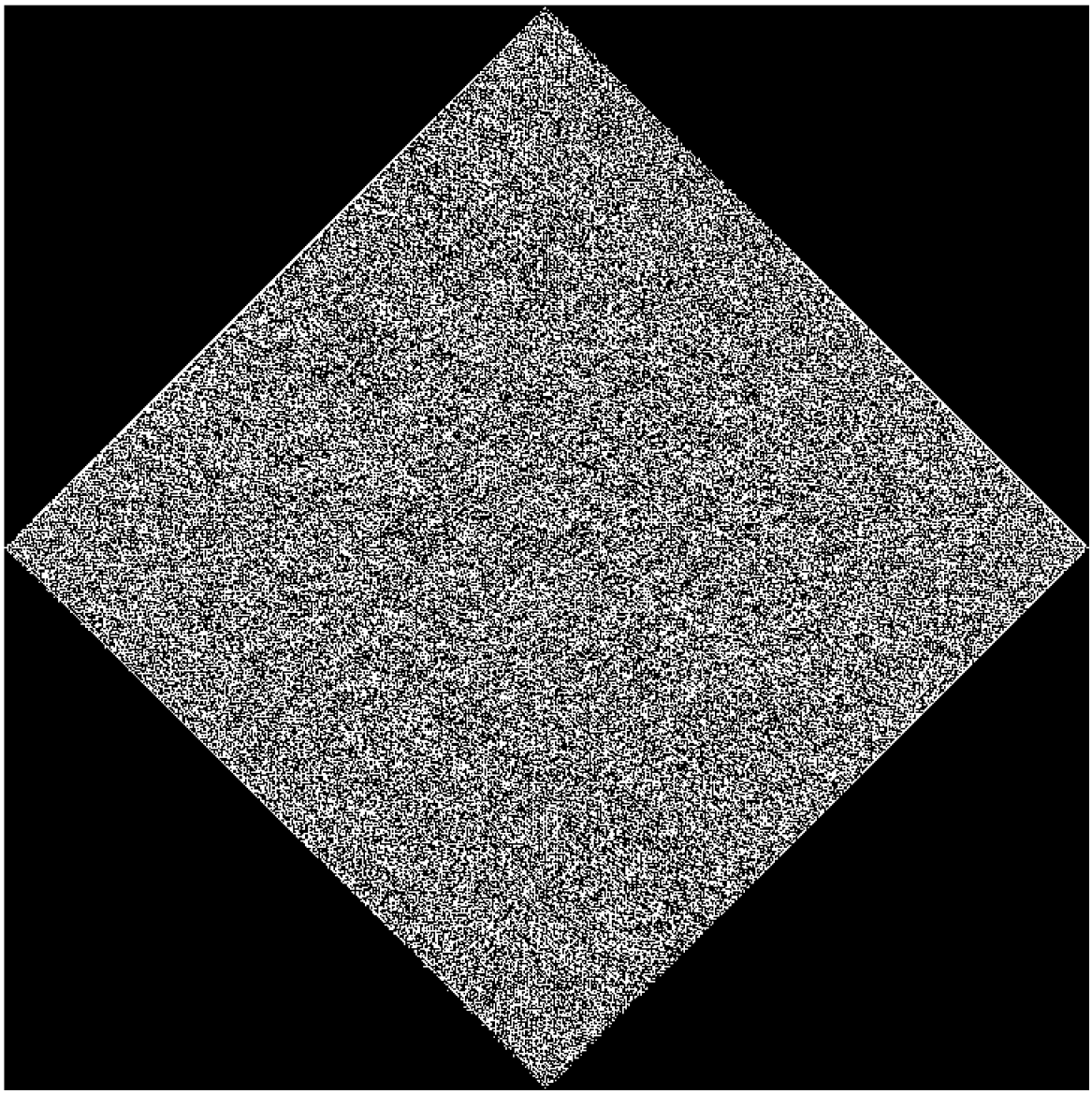
Since the difference is almost zero, the picture is almost black.

2. Rotation. Hint: we can use imrotate.

```
angle = 45;
g = imrotate(fft2(img3), angle);
imshow(g);
```

```
Warning: Displaying real part of complex input.
```

```
title("Rotation")
```



As it can be seen from the picture that there are four triangle made of black color appeared in the four corner.

3. What is the result of the Fourier transform of I(x, y) × $(-1)^{x+y}$? What property of the Fourier transform is revealed?

```
[n,m] = size(img1)
```

```
n = 512
m = 512
```

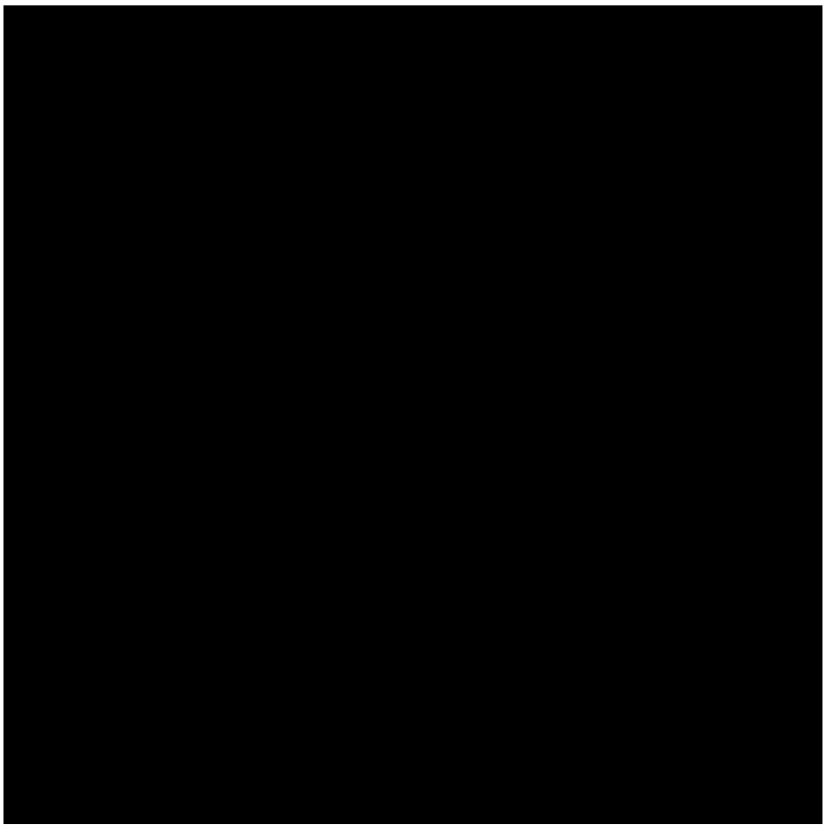
```
result = zeros(n,m)
```

```
result = 512×512
     0     0     0     0     0     0     0     0     0     0     0     0     0 ···
```

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

⋮

```
imshow(result)
```



```
for c = 1:n
    for r = 1:m
        result(r,c) =img1(r,c) *((-1).^(r+c));
    end
end

imshow(fft2((result)))
```

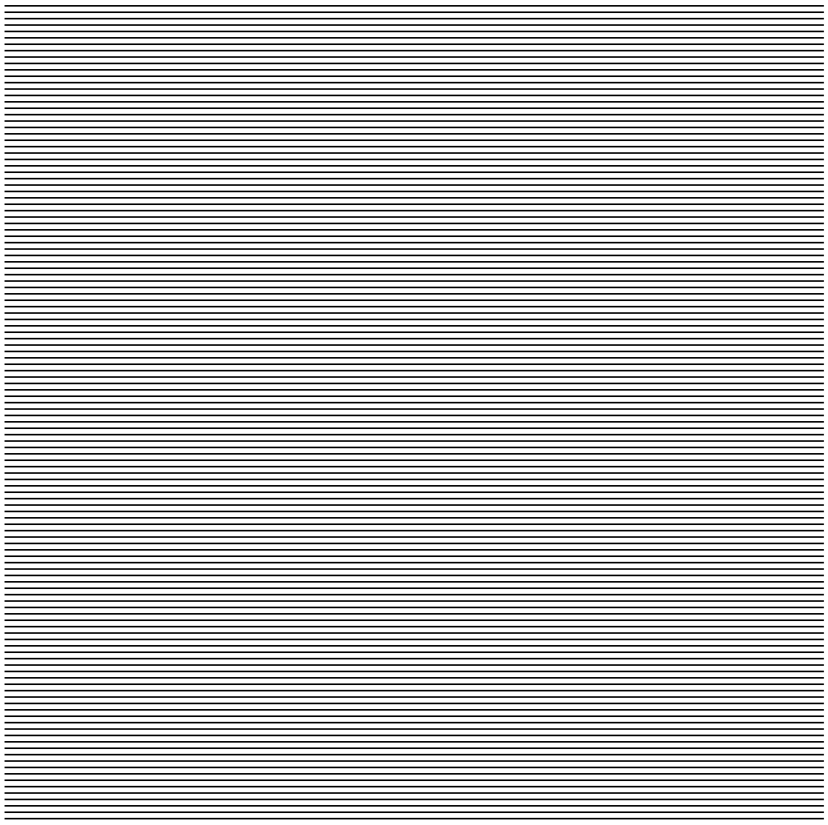
Warning: Displaying real part of complex input.

Its showing the property of translation.

Exercise 3 : Fourier transform of basic images

1. **A sinusoid function**

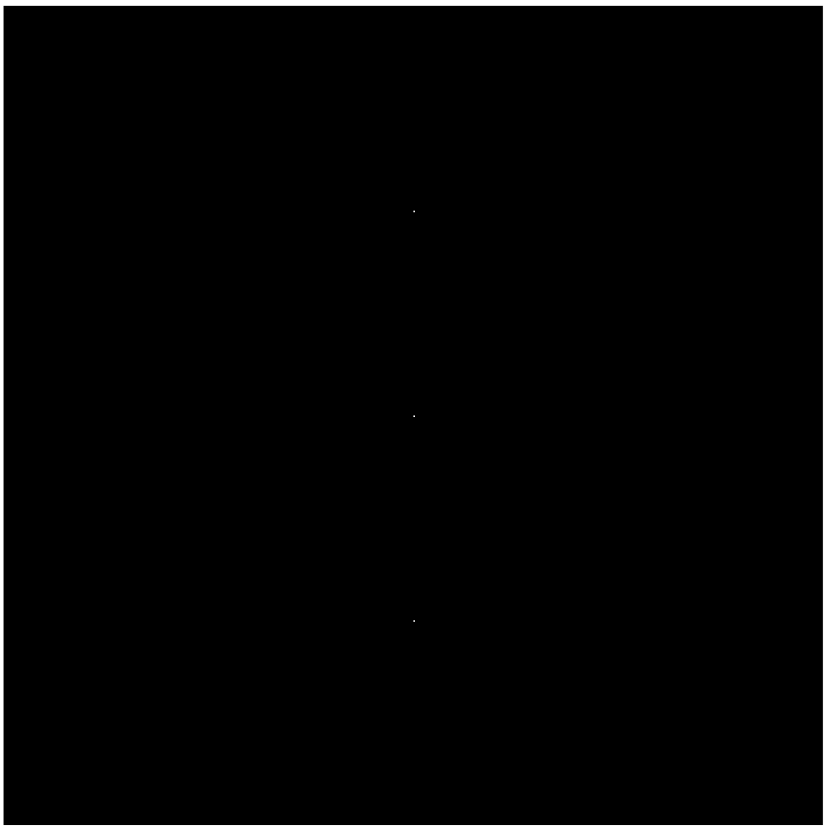
```
u0=0.25;
v0= 0;
M=512;
N=512;
f = zeros(M, N);
for c = 1:N
for r = 1:M
f(r, c) = 255*(sin(2*pi*(u0*r + v0*c))+1)/2;
end
end

imshow(f)
```



```
ft1 = fft2(double(f))
```

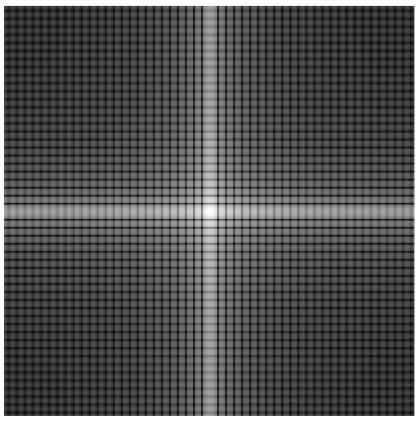
```
ft1 = 512×512 complex
10^7 ×
   3.3423 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i ···
  -0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
  -0.0000 - 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
  -0.0000 - 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
  -0.0000 - 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 - 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
  -0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
  -0.0000 - 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 - 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
     ⋮
     ⋮
```

```
FS1 = fftshift(ft1);
S2 = log(1+abs(FS1));
imshow(S2, []);
```



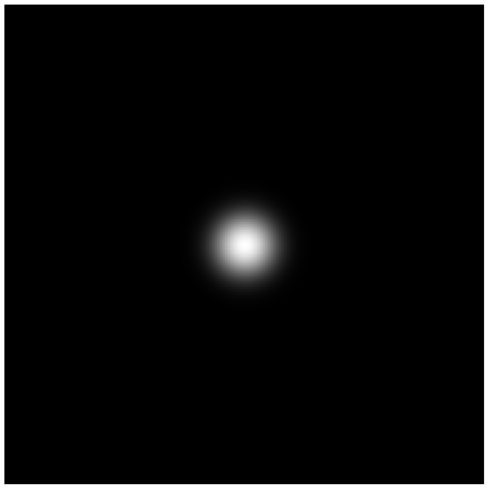
A gate function (image of a rectangle)

```
I = zeros(256,256);
I(100:150,100:150) = 255;
ft1 = fft2(double(I));
FS1 = fftshift(ft1);
S2 = log(1+abs(FS1));
imshow(S2, [])
```

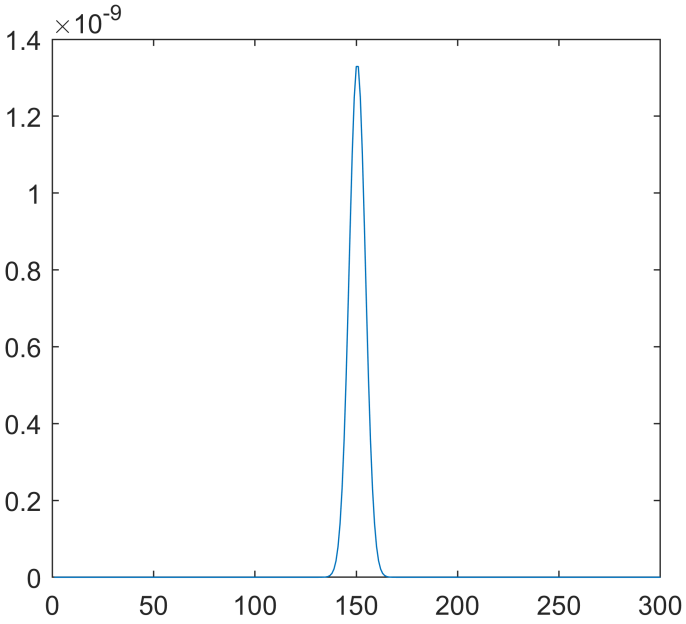


Gaussian

```
h = fspecial('gaussian',300, 4);
x = fftshift(fft2(double(h)));
y1 = abs(x);
im2 = log(1+y1);
figure(15)
imshow(im2, [])
```

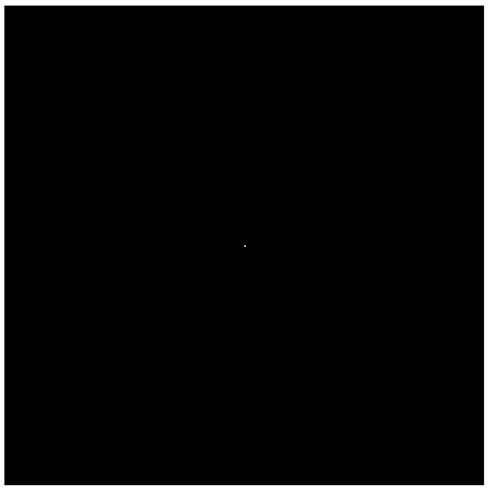


```
plot(h(128, :))
```



Uniform average filter

```
avg = fspecial('average', 300);

x = fftshift(fft2(double(avg)));
y1 = abs(x);
im2 = log(1+y1);
figure(16)
imshow(im2, [])
```



The average filter is utilized for smoothing images. It also reduces the variation of the intensity. So, it can be said that it is used for noise reduction. The average filter simplify the greyscale images.

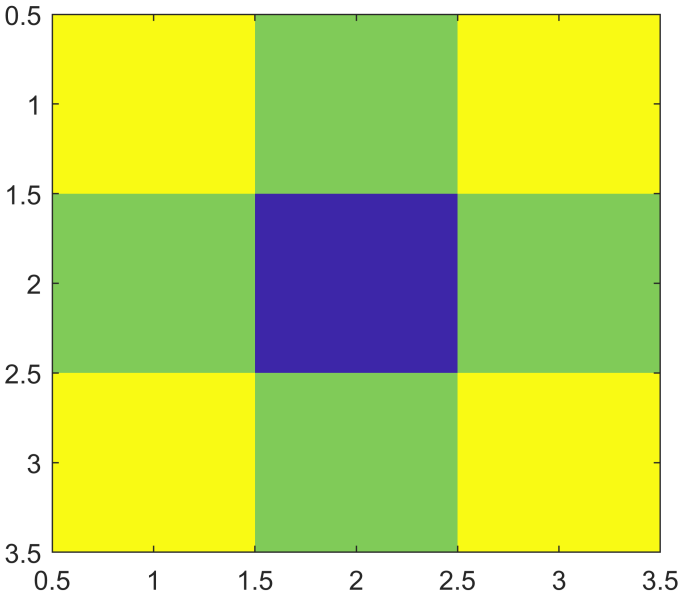
A Laplacian filter

Here the alpha has been changed twice to see the difference between the two different filter.

```
h = fspecial('laplacian',0.2)
```

```
h = 3×3
    0.1667    0.6667    0.1667
    0.6667   -3.3333    0.6667
    0.1667    0.6667    0.1667
```

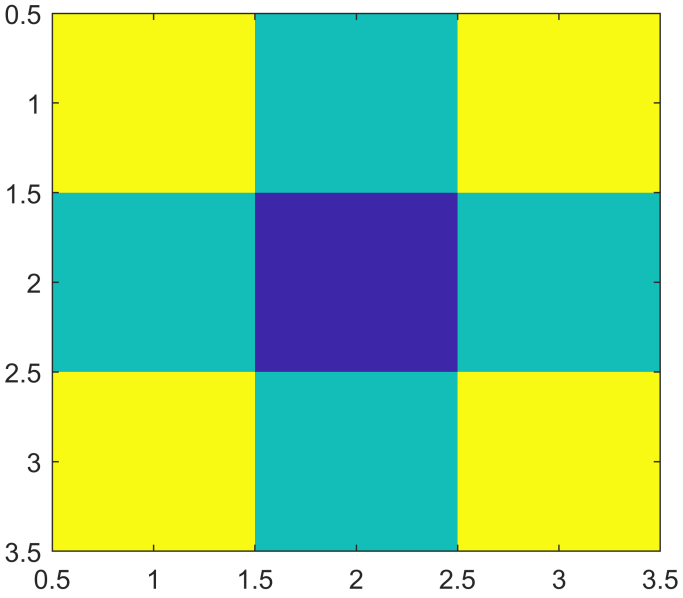
```
Y = fftshift(fft2(h));
imagesc(abs(Y))
```




```
h_L = fspecial('laplacian',0)
```

```
h_L = 3×3
     0     1     0
     1    -4     1
     0     1     0
```

```
Y5 = fftshift(fft2(h_L));
imagesc(abs(Y5))
```



Exercise 4 : Fourier transform and convolution

```
L = fspecial('laplacian')
```

```
L = 3×3
    0.1667    0.6667    0.1667
    0.6667   -3.3333    0.6667
    0.1667    0.6667    0.1667
```

```
img8 =conv2(img1,L,'same')
```

```
img8 = 512×512
 -248.5000 -137.1667 -140.5000 -139.5000 -132.5000 -131.3333 -134.0000 -131.0000 ···
 -135.0000   -0.8333   -0.1667    0.0000    2.0000    2.8333    0.1667    4.3333
 -134.3333   -0.8333   -0.8333    0.6667    0.1667    1.1667    1.6667    5.0000
 -136.8333    0.6667   -0.1667   -2.5000   -1.0000    0.6667   -0.8333    5.0000
 -139.1667   -3.5000   -1.8333    0.5000   -1.0000   -2.6667   -1.0000    6.3333
 -141.3333    0.6667    0.8333   -0.6667   -0.1667   -1.1667   -0.8333    3.5000
 -137.8333    1.6667    1.1667    0.8333   -1.8333   -2.6667    1.5000    4.5000
 -138.5000   -5.0000    2.0000    5.1667    3.5000    1.3333    1.0000   11.8333
 -121.6667   10.3333    5.6667   -7.0000   -8.5000   -1.5000   -1.6667   -5.0000
 -133.5000    3.6667    2.8333   -2.8333   -4.0000    2.1667    3.8333    6.3333
     ⋮
```

```
imshow(img8)
```

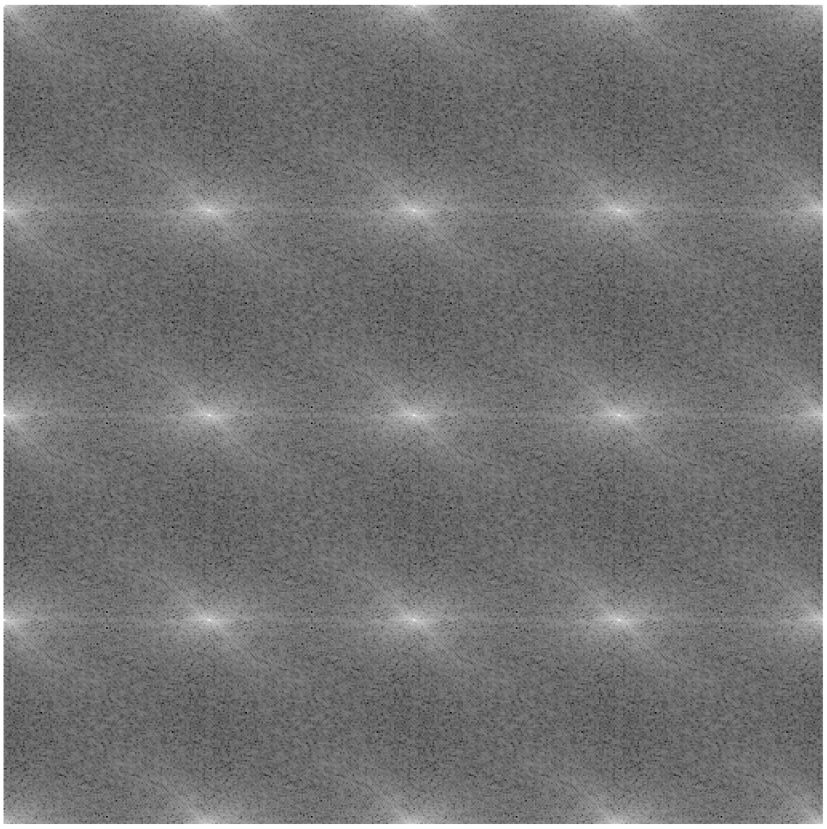
Exercise 5 : Sampling

```
A = zeros (4,4);
A (1,1) = 1;
B = repmat (A, 512/4,512/4);
img6 = B.*double(img1)
```

```
img6 = 512×512
   136     0     0     0   134     0     0     0   141     0     0     0   133 ···
     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0
   138     0     0     0   136     0     0     0   140     0     0     0   131
     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0
   128     0     0     0   135     0     0     0   138     0     0     0   134
     0     0     0     0     0     0     0     0     0     0     0     0     0
     ⋮
```

```
x = fftshift(fft2(img6));
y1 = abs(x);
im2 = log(1+y1);
```

```
figure(15)
imshow(im2, [])
```



Exercise 6 : Filtering in the frequency domain

```

```