Assignment 5: Question 6

Q6) Given kernels $k_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$, $k_2 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$

$N = 201$

$\therefore$ After padding the kernels, so that they're at the centre of a 201×201 image of other values zero

$\therefore$ We have $k_1'(x,y)$: $(x, y \in [-100, 100])$

$k_1'(x,y)$ values:

$k_1'(0,0) = -4$; $\cancel{k_1'(1,0)}$

$k_1'(1,0)$, $k_1'(0,1)$, $k_1'(-1,0)$, $k_1'(0,-1) = 1$

$k_1'(x,y) = 0$ otherwise.

$\therefore$ From our working out, we have the Fourier Transform (N-point), given by: $(N = 201)$

$$K_1(u,v) = \sum_{x=-100}^{100} \sum_{y=-100}^{100} k_1'(x,y)\, e^{-2\pi j \left(\frac{ux+vy}{201}\right)}$$

Only 5 terms persist:

$$K_1(u,v) = -4 + e^{-j2\pi\left(\frac{u}{201}\right)} + e^{-j2\pi\left(\frac{v}{201}\right)} + e^{j2\pi\left(\frac{u}{201}\right)} + e^{j2\pi\left(\frac{v}{201}\right)}$$

We can verify these by plugging in values and checking our plot.

Date :

Similarly, we have $k_2'(x,y)$: (padded version of k_2, with indices running from -100 to 100, both inclusive.

Values:

$\therefore k_2'(0,0) = 8$

$k_2'(-1,-1) = k_2'(-1,0) = k_2'(-1,1) = k_2'(0,-1) = k_2'(0,1)$
$= k_2'(1,-1) = k_2'(1,0) = k_2'(1,1) = -1$

$k_2'(x,y) = 0$ otherwise.

$$\therefore K_2(u,v) = \sum_{x=-100}^{100} \sum_{y=-100}^{100} k_2'(x,y)\, e^{-2\pi j\left(\frac{ux+vy}{201}\right)}$$

Only 9 terms persist:

$$K_2(u,v) = 8 - \Big[e^{-j2\pi\left(\frac{u+v}{201}\right)} + e^{j2\pi\left(\frac{u+v}{201}\right)} + e^{-j2\pi\left(\frac{u-v}{201}\right)} + e^{j2\pi\left(\frac{u-v}{201}\right)}$$
$$+ e^{-j2\pi\left(\frac{u}{201}\right)} + e^{-j2\pi\left(\frac{v}{201}\right)} + e^{j2\pi\left(\frac{u}{201}\right)} + e^{j2\pi\left(\frac{v}{201}\right)} \Big]$$

This can be verified by plugging in values and comparing with our plot.

In the formulae derived above, the indices are -100 to 100 and N = 201

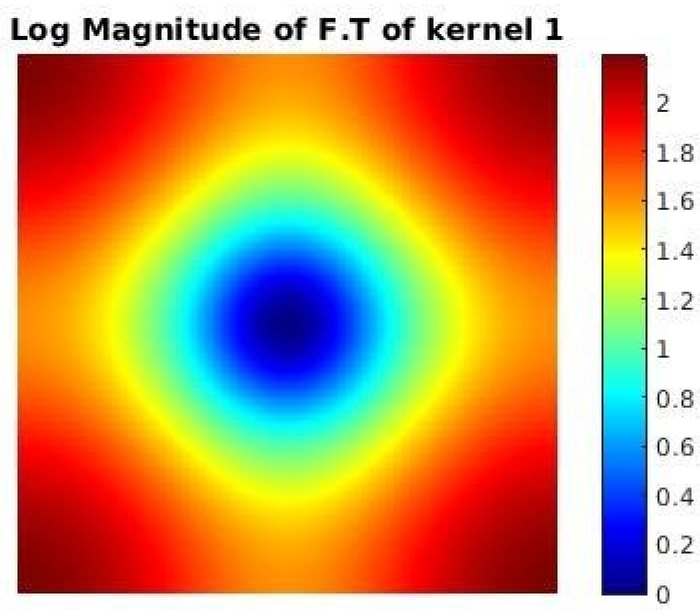
<u>For the 1st kernel:</u>

```
k1 = 0  1   0
       1 -4  1
       0  1  0
```

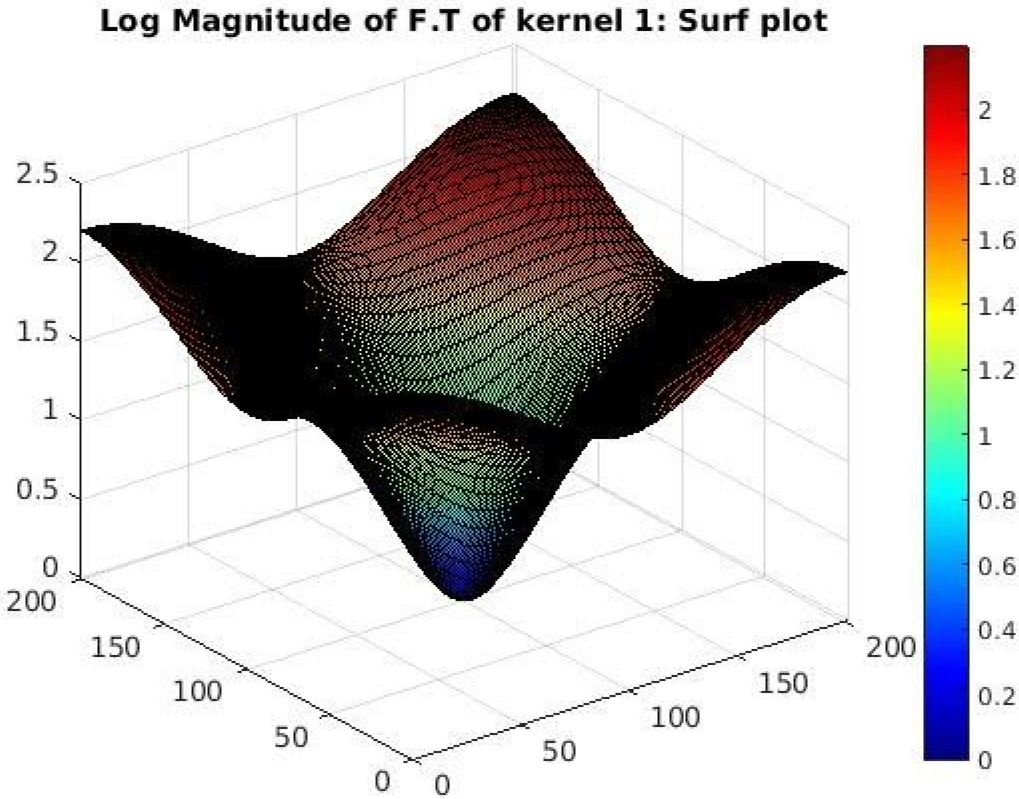
Code snippet to generate the plot: (MATLAB)

```
%%
N = 201;
k1 = [0,1,0;1,-4,1;0,1,0];
k1_padded = zeros(N);
k1_padded(100:102,100:102)=k1;
K1 = fftshift(fft2(k1_padded));
logK1 = log(abs(K1)+1);
figure, imshow(logK1, [min(logK1(:)) max(logK1(:))]),
colormap('jet'), colorbar,
title("Log Magnitude of F.T of kernel 1");
figure, surf(logK1), colormap('jet'), colorbar,
title("Log Magnitude of F.T of kernel 1: Surf plot");
%%
```

The centre of the following plot is coordinate (0,0). The indices extend from -100 to 100 for both u and v. Though, for the image itself, its spatial coordinates range from 0 to 200.



This is the surface plot for the same kernel. It is the perspective view if we consider the previous plot as a sort of ‘top view’. Here, the same observations about indices are made.



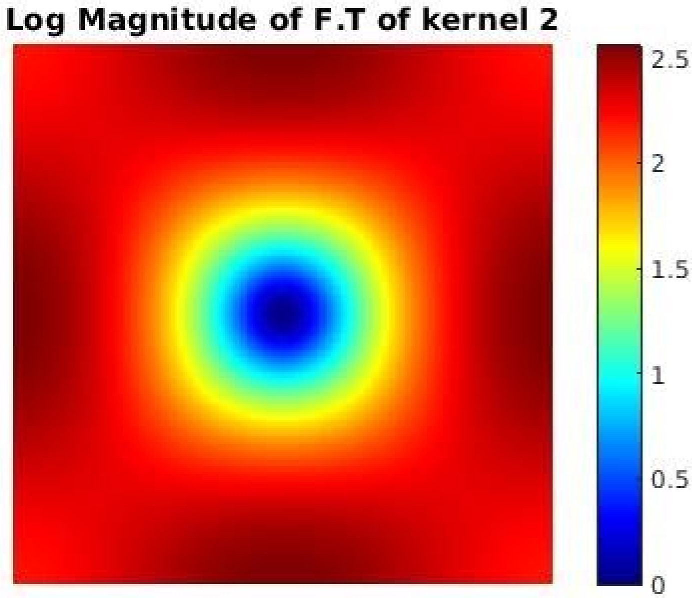
<u>For the 2nd kernel:</u>

```
k2 = -1  -1  -1
       -1  8   -1
       -1  -1  -1
```

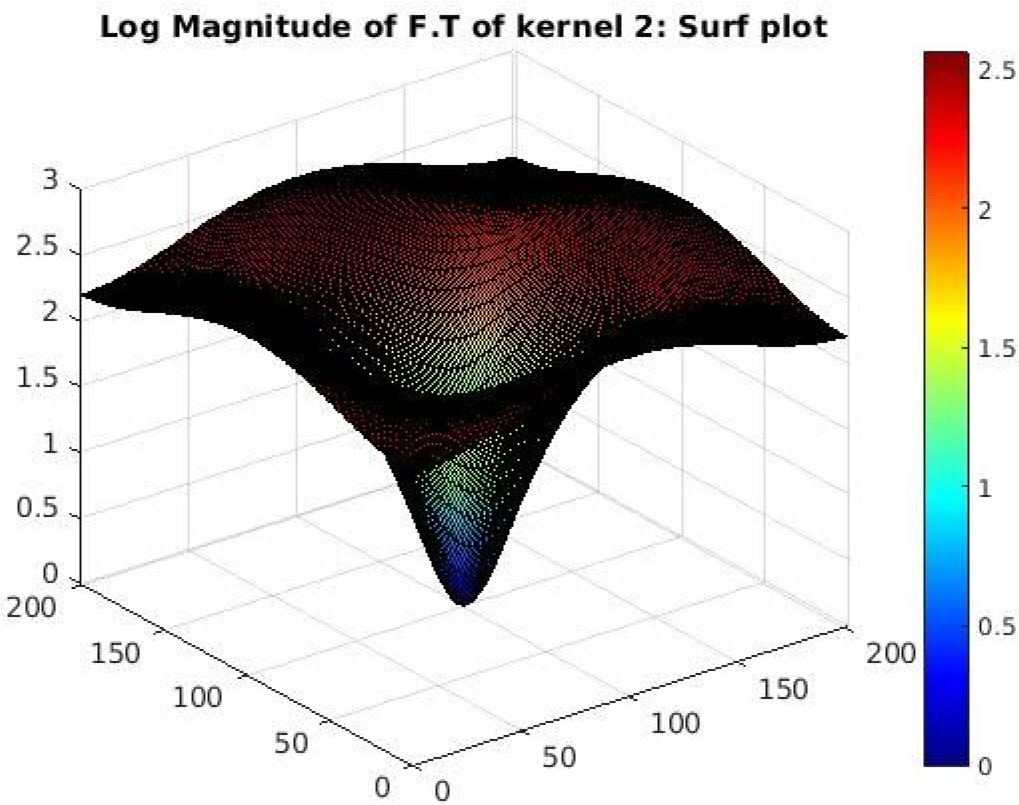
Code snippet to generate the plot: (MATLAB)

```
%%
N = 201;
k2 = [-1,-1,-1;-1,8,-1;-1,-1,-1];
k2_padded = zeros(N);
k2_padded(100:102,100:102)=k2;
K2 = fftshift(fft2(k2_padded));
logK2 = log(abs(K2)+1);
figure, imshow(logK2, [min(logK2(:)) max(logK2(:))]),
colormap('jet'), colorbar,
title("Log Magnitude of F.T of kernel 2");
figure, surf(logK2), colormap('jet'), colorbar,
title("Log Magnitude of F.T of kernel 2: Surf plot");
%%
```

The centre of the following plot is coordinate (0,0). The indices extend from -100 to 100 for both u and v. Though, for the image itself, its spatial coordinates range from 0 to 200.



This is the surface plot for the same kernel. It is the perspective view if we consider the previous plot as a sort of 'top view'. Here, the same observations about indices are made.



Comparison between the Fourier Transforms:

We observe that both the plots seem to be radially symmetrical for low frequencies (close to (u,v)=(0,0)). The values here are very close to zero, so we know both from the plot as well as our knowledge of these Laplacian kernels that they are high-pass filters.

This symmetry is a good feature for such a filter. However, at higher frequencies (higher magnitudes of u and v), we observe that the second kernel (k2) maintains this symmetry longr than the first kernel (k1). This is because of the presence of more terms in the Fourier transform of the 2nd kernel. The added waves provide a certain stability to the filter and allow it to behave consistently for a larger range of values.
i.e, we see that the first kernel (k1) has a large difference in how much it boosts some frequencies and weakens others, despite being of the same magnitude.

Another point here is that the second kernel has a much steeper jump from the low values to the high than the first (evidenced by the size of the blue and green contours in the image). This is a better approximation of the ideal high pass filter than the slow, diffused first kernel. Similarly, after this steep jump to higher frequencies, the second kernel flattens out radially (not exactly, but definitely better than the first kernel, while the first undulates and produces undesirable effects).

Hence we may conclude that for the purpose of high-pass filtering, the second kernel does a better job than the first.