CS663 Assignment 5 Question 6 Report

Shreya Laddha, Archishman Biswas, Shreyan Jabade, Rishabh Arya

November 14, 2020

1 Overview

In this question, we have implemented a MATLAB code in order to compute the N,N-point DFT of two given Laplacian kernels k_1 and k_2. As the kernels are simple, we have implemented the code using simple computation and not by using the in-built fft() function. The zero padding in this question was done for N = 201.

2 Algorithm implementation

2.1 FT of kernel k_1

The first kernel given is as $k_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$

Thus for the above k_1, we will appropriately zero-pad to get a new k_1 which is a $N \times N$ matrix. Hence, we will have the expression for DFT for a particular N is given as:

$$K_1(u,v) = 2exp(\frac{-j2\pi(u+v)(N+1)}{2N})\{cos(\frac{2\pi u}{N}) + cos(\frac{2\pi v}{N}) - 2\}$$

The above equation can be obtained by summing up the terms opposite to the centre of the Laplacian kernel in pairs. As we have two such pairs in this case, we get two cos() terms. The computation in the code is carried as per this formula.

2.2 FT of kernel k_2

The second kernel given is as $k_2 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$

Similarly, for k_2 also, we will first zero-pad it to get a new k_2 of size $N \times N$. Then the DFT is given as:

$$K_2(u,v) = 2exp(\frac{-j2\pi(u+v)(N+1)}{2N})\{4 - cos(\frac{2\pi u}{N}) - cos(\frac{2\pi v}{N}) - cos(\frac{2\pi(u+v)}{N}) - cos(\frac{2\pi(u-v)}{N})\}$$

In the k_2 case, we have a total of 4 pairs. Hence, in the equation, we have received 4 cos() terms. Also the sign is inverted as k_2 kernel has inverted nature as compared to k_1 kernel. In the above equations shown, both the variables u,v range from [-100,100] for N = 201.

3 MATLAB codes

```matlab
clear all; close all;
N = 201; centre = (N+1)/2; fac = (2*pi)/N;
X = -(N-1)/2:(N-1)/2; Y = -(N-1)/2:(N-1)/2;
lap1 = [0 1 0; 1 -4 1; 0 1 0];
lap2 = [-1 -1 -1; -1 8 -1; -1 -1 -1];
%% Computation for Laplacian kernel 1
my_ft_lap1 = zeros(N,N);
for u = 1:N
    for v = 1:N
        my_ft_lap1(u,v) = 2*exp(-1i*fac*centre*(u+v))*(cos(fac*u)+cos(fac*v)-2);
    end
end
my_ft_lap1 = fftshift(my_ft_lap1);
figure();
imagesc(X,Y,mat2gray(log(1+abs(my_ft_lap1)))); colormap jet; colorbar; axis on;
title('Log magnitude plot of Laplacian kernel k_{1}'); pbaspect([1 1 1]);
figure();
surf(X,Y,log(1+abs(my_ft_lap1))); shading interp; colormap jet; colorbar;
title('Log magnitude surface plot of Laplacian kernel k_{1}'); pbaspect([1 1 1]);
%% Computation for Laplacian kernel 2
my_ft_lap2 = zeros(N,N);
for u = 1:N
    for v = 1:N
        my_ft_lap2(u,v) = 2*exp(-1i*fac*centre*(u+v))*(4-cos(fac*u)-cos(fac*v)-cos(fac*(u+v))-cos(fac*(u-v)));
    end
end
my_ft_lap2 = fftshift(my_ft_lap2);
figure();
imagesc(X,Y,mat2gray(log(1+abs(my_ft_lap2)))); colormap jet; colorbar; axis on;
title('Log magnitude plot of Laplacian kernel k_{2}'); pbaspect([1 1 1]);
figure();
surf(X,Y,log(1+abs(my_ft_lap2))); shading interp; colormap jet; colorbar;
title('Log magnitude surface plot of Laplacian kernel k_{2}'); pbaspect([1 1 1]);
```

4 Results

4.1 FT of kernel k_1

The log magnitude contour and surface plot for FT of k_1 is shown in the below figures:

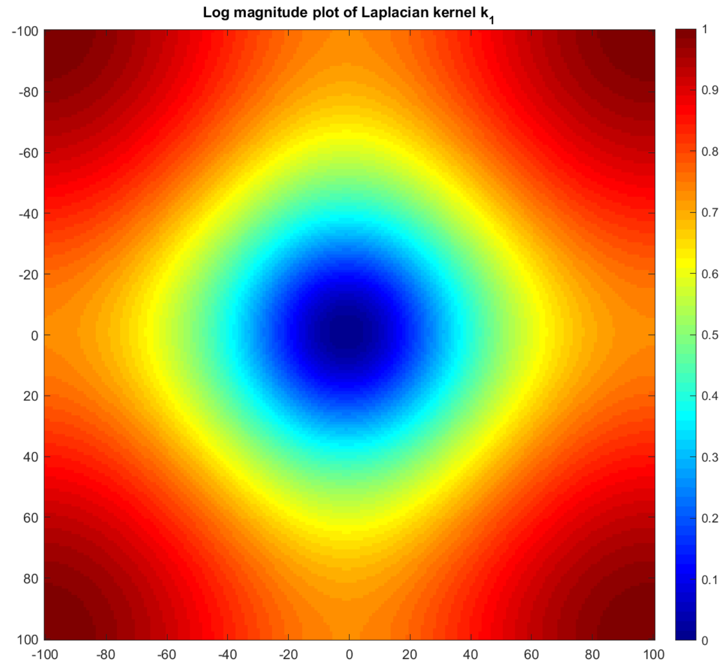

Figure 1: Log magnitude plot of $K_1(u, v)$

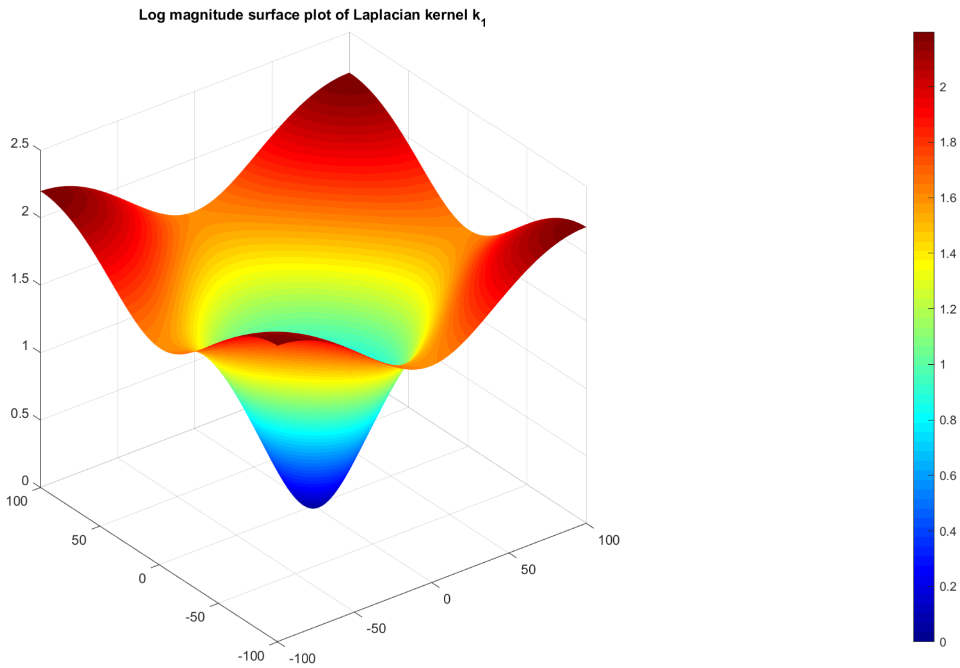

Figure 2: Log magnitude surface plot of $K_1(u, v)$

4.2 FT of kernel k_2

The log magnitude contour and surface plot for FT of k_2 is shown in the below figures:

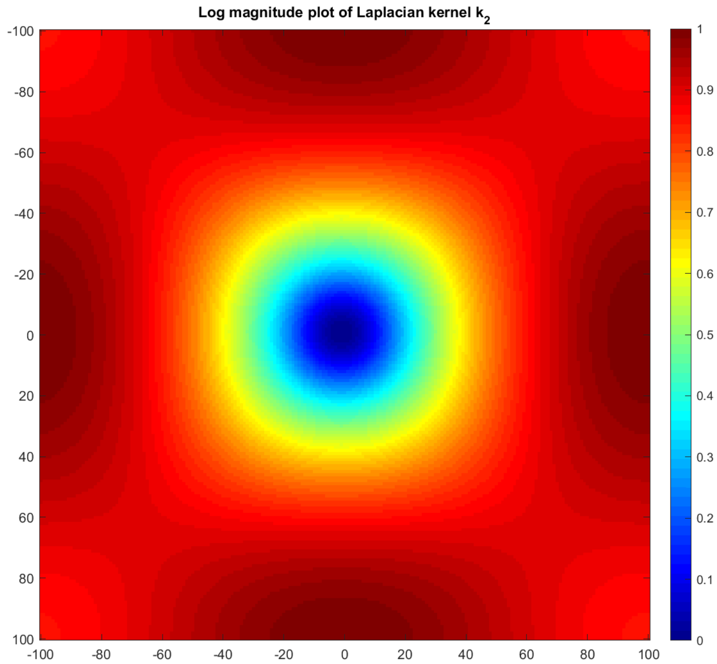

Figure 3: Log magnitude plot of $K_2(u,v)$

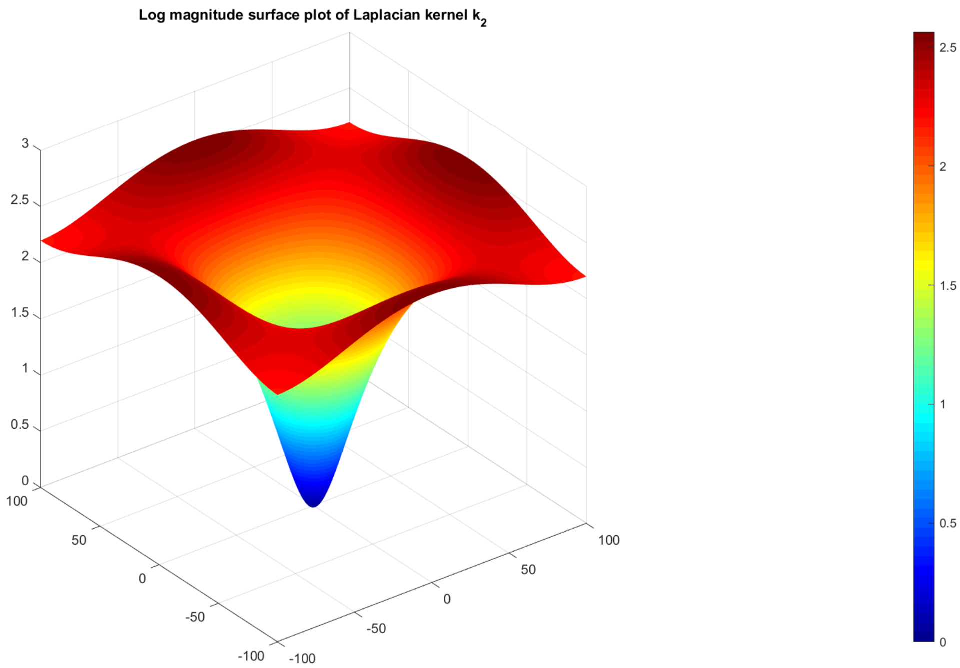

Figure 4: Log magnitude surface plot of $K_1(u,v)$

For an ideal differentiator system, we would want the frequency response to be a cone with centre located at the the zero frequencies (u,v) = (0,0). From the above figures, we can observe that the k_2 approximates the differentiator system better than k_1. The frequency response for the k_1 has some parabolic behaviour.
From the images/contour plots, we can observe that K_2,K_1 both has circular contours within the small frequency ranges, thus it acts ideally in this frequency range. Beyond this range, the contours become square-like, the square contours have different orientations in the two kernels. At the edges/corners, $K_1(u,v)$ has a high rate of change. Whereas the $K_2(u,v)$ is tapered off and is almost flat in the log scale.