

## AMATH 482 Homework 1

### Section I. Introduction and Overview

This project involves using the fourier transform to perform signal processing of a data provided. In this assignment, we processed an ultrasound data describing the spatial variations in the intestinal region of a dog (Figure 1) who just swallowed a marble. Frequency domain analysis helps pinpoint the location of the marble in the intestine accurately despite the noisy data and the fact that the marble is moving in the spatial domain at each data point.



**Figure 1. Fluffy Who Swallowed a Marble**

The strategy to process the data is divided into three steps:

1. Adding the data in the frequency domain to find the character frequency that corresponds to the location of the marble.
2. Creating a filter around the character frequency, applying it to the data in the frequency domain.
3. Inverse transforming the filtered frequency data to recover the position of the marble at each data point.

Ultimately, we are able to recover the position of the center of the marble at each data point.

### Section II. Theoretical Background

In this assignment, Fourier Transforms and Inverse Fourier Transforms are shown in Equation 1. Computationally, we are using the Fast Fourier algorithm with runtime complexity of  $O(N\log N)$ .

$$\begin{aligned} F(k) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-ikx} f(x) dx \\ f(x) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{ikx} F(k) dk. \end{aligned} \quad (\text{Equation 1})$$

We are also implementing a 3D Gaussian Filter that is specified by Equation 2. All the  $K_c$  represents the frequency domain coordinate of the center of the filter and  $\tau$  specifies the variance of the filter.

$$filter = \exp(-\tau((Kx - K_{c_x})^2 + (Ky - K_{c_y})^2 + (Kz - K_{c_z})^2)) \quad (\text{Equation 2})$$

Lastly, we rely on the fact that averaging the spectrum of multiple data points will give us the frequency of interest since the rest of the frequencies corresponds to the noise which has 0 mean.

### Section III. Algorithm Implementation and Development

To perform spectral analysis, we need to prepare the variables needed to be passed to the fast fourier transform algorithm. First, we set the range of the spatial domain (L), which is 15 units from the center. Then, we set up the amount of fourier modes (n) of the data which is 64, implied from the size of the data points (64X64X64). Then we can set up the wave numbers which are scaled to span -L to L. In order to plot the frequency domain data, we need to set up a 3D coordinate system that is the result of the fftshift of the prepared wave numbers [Kx, Ky, Kz] (the unshifted frequency 3D coordinate system is [kx, ky, kz]).

We are then ready to compute the average of the frequency domain representation of the data. For each of the data point, the 64X64X64 data point obtained by using the reshape function is Fourier Transformed in 3 dimension to obtain the frequency domain signal. Each of the spectral frequency domain is added (averaged) in each loop in order to cancel out the noise. After 20 data points, we can normalize the summation of all the spectral domain, and we can use ind2sub and the find function to find the highest amplitude location in the frequency domain coordinate.

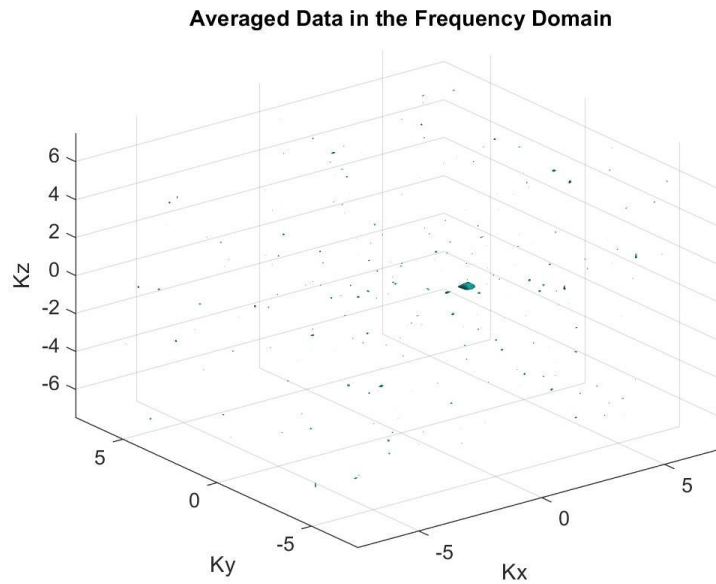
The next step is to construct a gaussian filter that is centered around the frequency location found earlier. The gaussian filter is implemented in a 3D coordinate system described in Equation 2, with  $\tau = 1$ . When plotting the filter, it is important to use fftshift for both the spectral coordinates and the filter value.

The filter is applied to the frequency domain representation of each of the data point. When applying the filter to the frequency domain representation of the data, it is important to un-fftshift the filter value before performing Inverse Fourier Transform. Once the filter is applied, we can recover the filtered spatial domain representation of the data by performing Inverse Fourier Transform in 3 Dimension. This will yield blobs representing the location of the marble.

Lastly, we can use the functions we used to find the location of the character frequency by using ind2sub and find to obtain the highest amplitude location inside Fluffy's intestines. Once all the centers are obtained, the points are plotted using plot3 to represent the trajectory of the marble at each data points.

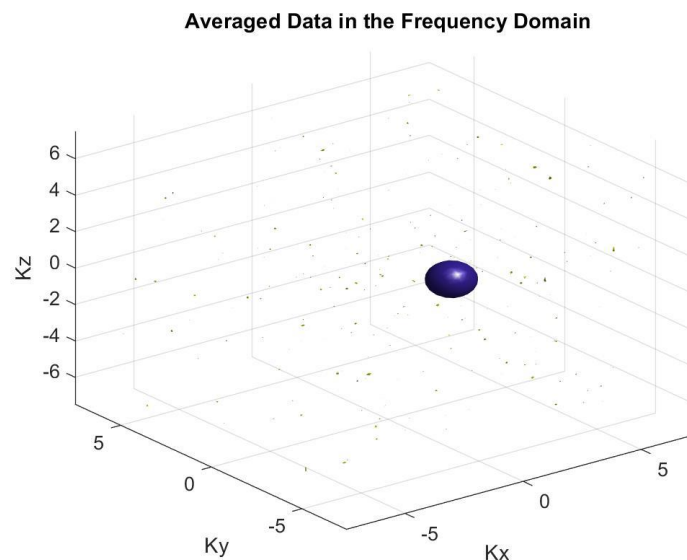
## Section IV. Computational Results

Figure 2 represents the averaged normalized frequency domain representation of the data. We can clearly see that there is a cluster of high amplitude region, which represents the character frequency of interest which is located at  $[1.8850, -1.0472, 0]$  ( $K_x, K_y, K_z$ ).



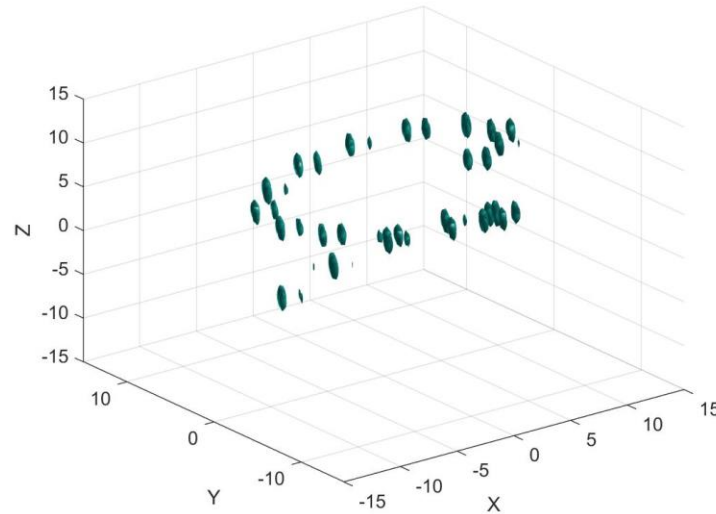
**Figure 2. Averaged Spectral Data**

Figure 3 represents the 3D plot of the filter that we are applying to the frequency domain data.



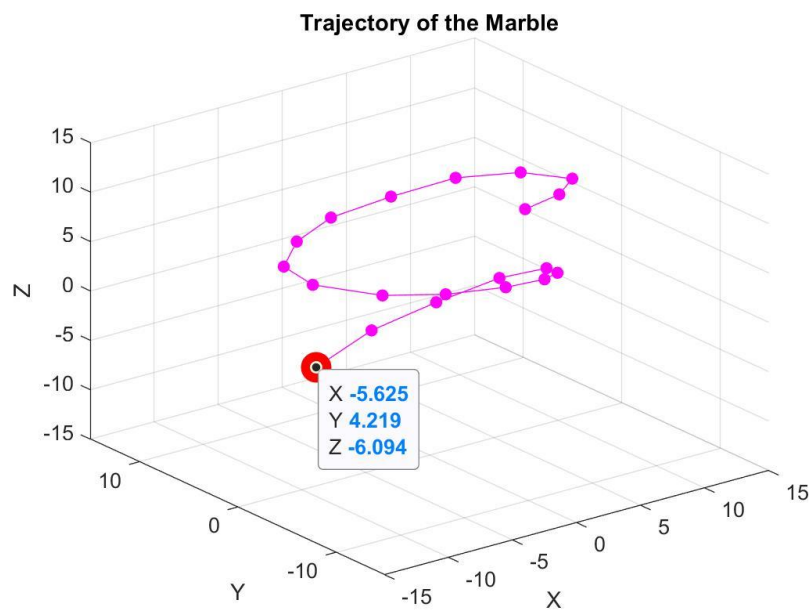
**Figure 3. 3D Gaussian Filter**

Upon applying the filter and inverse fourier transforming the filtered spectrum, plotting the filtered spatial data results in the plot shown in Figure 4. We can see blobs of data that represents the location of the marble. Varying the  $\tau$  value of the filter from 0.05 to 1 will yield the same result.



**Figure 3. Filtered Spatial Domain Data**

Figure 5 conveys the trajectory of the marble in each of the data point. Highlighted in red is the location of the marble at the 20<sup>th</sup> data point, which is where the high energy beam should be focused at inside Fluffy to save him. The coordinate of the location of the marble is specified in Figure 5.



**Figure 4. Trajectory of the Marble**

## Section V. Summary and Conclusions

There are 3 very powerful tools implemented in this assignment: frequency domain analysis, averaging data spectrum, and filtering around a frequency of interest. Frequency domain analysis lets us observe the frequencies occurring in the data. Averaging the data spectrum helps us cancel the noise to find the characteristic frequency of interest. Lastly, Filtering around a frequency of interest lets us recover the signal of interest in the spatial domain.

The coordinate where a high frequency beam should be focused at inside Fluffy's stomach is at  $[5.6250 \quad 4.2188 \quad -6.0938]$  (x, y, z), shown in Figure 4.

There are some design parameters required to be tuned in order to yield reliable results. The filter width ( $\tau$ ) is crucial to be tuned to filter out unnecessary frequencies. The isosurface value should also be tuned to obtain presentable isosurface plot.

This assignment is applicable to analyzing radar or wave data to find the location or the time of occurrence of our subject. Another application example would be to find the location of airplanes from RADAR data.

## Appendix A MATLAB Functions used an brief implementation explanation

### reshape(X,n,n,n)

Reshapes X into a matrix to become n x n x n dimension

### fft(x)

Returns the frequency domain transform of x (n-dimension) with fast fourier transform algorithm.

### fftshift(X)

Shifts the fourier transform X such that it is placed in the correct order where the zero-frequency components are shifted to the center of the array.

### [ind1 ind2 ind3] = ind2sub([n,n,n], index) find

Returns the corresponding indices in 3 dimension of the value specified in index. In this assignment, the index passed refers to the maximum value in the 3-D data provided.

### find(X == x)

Returns the indices of matrix X that corresponds to the value in the vector x

### fv = isosurface(X,Y,Z,V,isovalue)

Computes the points that intersects the specified value isovalue from the 3D contour map given.

### [X,Y,Z] = meshgrid(x,y,z)

Returns a 3-D grid coordinates based on the coordinates specified in x, y, and z.

### plot3(x,y,z)

Plots the points specified in x, y, z.

## Appendix B MATLAB codes

```
% AMATH 482 - Homework 11
% Khrisna Kamarga
clear all; close all; clc;
load Testdata % load the ultrasound data

% set up the fourier coefficients
L = 15; % spatial domain
n = 64; % Fourier modes
x2 = linspace(-L,L,n+1); x = x2(1:n); y = x; z = x;
k = (2*pi/(2*L))*[0:(n/2-1) -n/2:-1]; ks = fftshift(k);
% set up the 3D coordinate points
[X,Y,Z] = meshgrid(x,y,z); % spatial coordinates
[Kx,Ky,Kz] = meshgrid(ks,ks,ks); % wave numbers
[kx, ky, kz] = meshgrid(k,k,k); % unshifted wave numbers
Klim = max(Kx, [], 'all') + 1; % limits of the frequency domain axes
realize = 20; % amount of data points

%% 1) Through averaging of the spectrum, determine the frequency signature
% center frequency) generated by the marble.
UtnAve = zeros(n,n,n); % kernel for the averaged frequency domain signal
for j = 1:realize
```

```

    Un(:,:,:) = reshape(Undata(j,:),n,n,n); % gets the 3D coordinate representation of
the sample
    Utn = fftn(Utn); % fourier transform of the data
    UtnAve = UtnAve + Utn; % cummulative sum of the frequency domain signal
end

% plot the resulting normalized averaged data in the frequency domain
isosurface(Kx, Ky, Kz, abs(fftshift(UtnAve))/max(abs(UtnAve), [], 'all'), 0.6);
xlabel("Kx"); ylabel("Ky"); zlabel("Kz");
title("Averaged Data in the Frequency Domain");
axis([-Klim Klim -Klim Klim -Klim Klim]), grid on, drawnow
% find the indices of the max magnitude in the frequency domain
[ind1 ind2 ind3] = ind2sub([n,n,n], find(fftshift(UtnAve) == max(fftshift(UtnAve), [],
'all')));
% look up the frequency domain coordinate of the strongest signal
Kc = [Kx(ind1, ind2, ind3), Ky(ind1, ind2, ind3), Kz(ind1, ind2, ind3)];

% 3D gaussian filter
tau = 1; % bandwidth of the filter (good: 0.2)
filter = exp(-tau*((kx - Kc(1)).^2+(ky - Kc(2)).^2+(kz - Kc(3)).^2));
% plot the filter
isosurface(Kx,Ky,Kz,fftshift(abs(filter))/max(abs(filter), [], 'all'),0.2)

%% 2) Filter the data around the center frequency determined above in order
%     to denoise the data and determinethe path of the marble.
%     (use plot3 to plot the path once you have it)

close all;

marble = zeros(realize, 3); % kernel for the coordinates of the marble
for j = 1:realize
    Un(:,:,:) = reshape(Undata(j,:),n,n,n);
    Utn = fftn(Utn); %Utn = fftshift(Utn);
    UtnFilter = Utn.*filter; % filtered frequency domain signal
    UnFilter = real(ifftn(UtnFilter)); % obtain the spatial filtered data

    % draw the resulting spatial filtered data
    isosurface(X,Y,Z,abs(UnFilter)/max(abs(UnFilter), [], 'all'),0.8)
    axis([-L L -L L -L L]), grid on, drawnow

    % find the coordinate of the center of the marble
    [ind1 ind2 ind3] = ind2sub([n,n,n], find(abs(UnFilter) == max(abs(UnFilter), [],
'all')));
    marble(j,:) = [X(ind1, ind2, ind3), Y(ind1, ind2, ind3), Z(ind1, ind2, ind3)];
    pause(0.2);
end
% plot the trajectory of the marble
plot3(marble(:,1), marble(:,2), marble(:,3), 'm.-', 'MarkerSize', 20);
xlim([-L L]); ylim([-L L]); zlim([-L L]);
title("Trajectory of the Marble");
xlabel("X"); ylabel("Y"); zlabel("Z");
grid on

%% 3) Where should an intense acoustic wave be focused to breakup the
%     marble at the 20th data measurement.
acoustic_wave = marble(end,:) % Fluffy lives!
hold on
plot3(acoustic_wave(1), acoustic_wave(2), acoustic_wave(3), 'r.', 'MarkerSize', 50)

```