Amath 482 Homework 1

Oscar Zhang

January 27, 2021

Abstract

In this project, we are asked to track the location of a new submarine when given 3-D noisy acoustic data. The data is consisted of 49 columns for measurements over 24-hour span at half-hour increment in time. In this project, we are going to determine the center frequency generated by the submarine by averaging the spectrum. After this, we will apply the Gaussian filter to denoise the data and determine the path of the submarine. Eventually we will find the location of the submarine of the very last moment.

1. Introduction

1.1 Main methodology

In order to track the location of the submarine, we are going to use some 3-D data. However, the data is full of noise, which will negatively influence our computation and analysis. Since then we first need to adjust our data – we will apply Fournier transform to make our data projected to our desired frequency domain. To find such a center frequency, we are going to take the average of the Fournier transform. After finding out the center frequency, we need to add a filter around this frequency and transfer back to our time domain with the purpose of denoise. On top of this, we will be able to find out the location and the path of our submarine. Since we get the location and the path, we can easily determine the location of the submarine at the very last moment.

1.2 Data and programming language

The data we have is a 64^3 * 49 matrix, which stores the space and time and the programming we used is Matlab.

2. Theoretical Background

2.1 Fournier transform

Fournier series states that any function, either continuous or not, can be written by a combination of sine function and cosine one :

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty}(a_n \cos \mathrm{kx} + b_n \sin \mathrm{kx}),\ \mathrm{x} \in [-\pi, \pi]$$

where $\frac{a_0}{2}$ is for shifting the function up and down.

We also have the function that after being operated by the Fournier series as the following:

$$F(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-ikx} dx$$

and its inverse:

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} F(x)e^{ikx}dk$$

The main purpose that we apply Fournier Transform and its inverse is that we are able to project one function from either space or time domain to frequency domain.

2.2 Gaussian Filer

In addition, as mentioned before, we are going to apply the Gaussian filter around the frequency to remove the signals which are noise or we don't need, the Gaussian function will be as the following:

$$G(x) = e^{-\tau(k - k_0)^2}$$

where τ stands for the width of the filter and k_0 stands for the center frequency of the filter. For our task, we are going to apply a 3-D Gaussian filter, so it will be the following:

$$G(x, y, z) = e^{-\tau[(k - k_x)^2 + (k - k_y)^2 + (k - k_z)^2]}$$

where x, y, and z stand for the center of the filter in a 3-D field.

3 Algorithm Implementation and Development

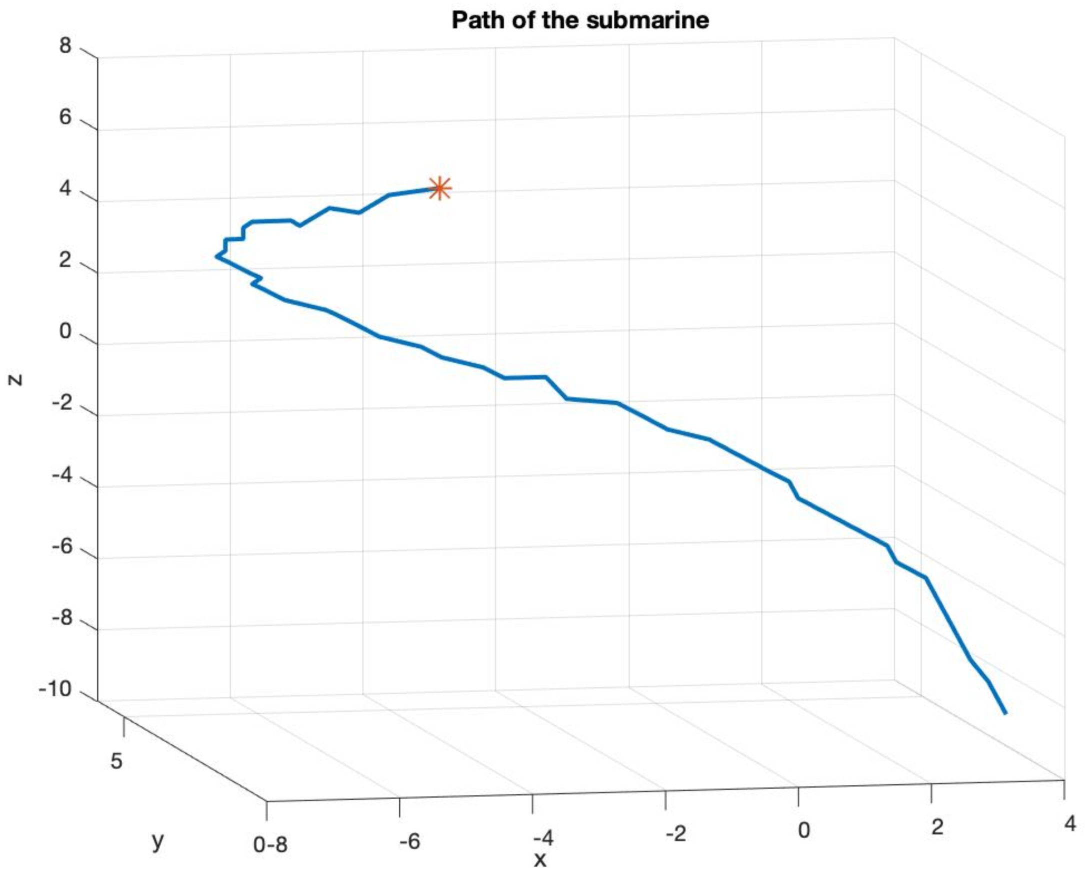
Firstly, we need to set up the spectrum in 3-D field, we have to define some variables: domain, the frequency domain, and Fourier node. One thing to note that, since we are using Fournier transform, we have to define our frequency in the following order – from 0 to (n/2 - 1) and (-n/2) to -1. Also, usually we define FFT with 2π as a period but in our task, we should rescale the period as 2π/L.

After the basic setting up, we are going to find the center frequency. First, we set up an empty matrix to store our average value. Then we do a for loop to update the average values. Within the for loop, we first use "reshape" to convert our 2-D(64^3 x 49) matrix to a 4-D(64 x 64 x64 x 49) matrix. Since we transferred our data from space domain to frequency domain, our data turned into 3-D field. Due to this, we apply fftn to our data. After summing up our average values, we do fftshift and calculate the mean. After those procedures, our average matrix is full of average values. We apply [maximum, index] = max() command to sort the maximum value and its index. Then we apply ind2sub command to find out the center frequency of x, y, and z.

After calculating the frequency of x, y, and z, we are able to apply our Gaussian filter formula. We set up the filter width and a matrix to store the location of our submarine. Then we runa for loop again, we firstly reshape our data as above and transfer it into frequency domain by command "fftn". After this, we apply our filter to our new data and invert it back to space domain by command "ifftn". As above, we do [maximum, index] = max() to sort the maximum value and its index. Apply command "ind2sub" again to find out the center. After the for loop, we then are able to find out the path of our submarine.

4 Computational Results

The center frequency of x, y, and z are 5.3407, -6.9115, and 2.1991. The path is the following:



The final location of the submarine is [-5, 0.9375, 6.5625]

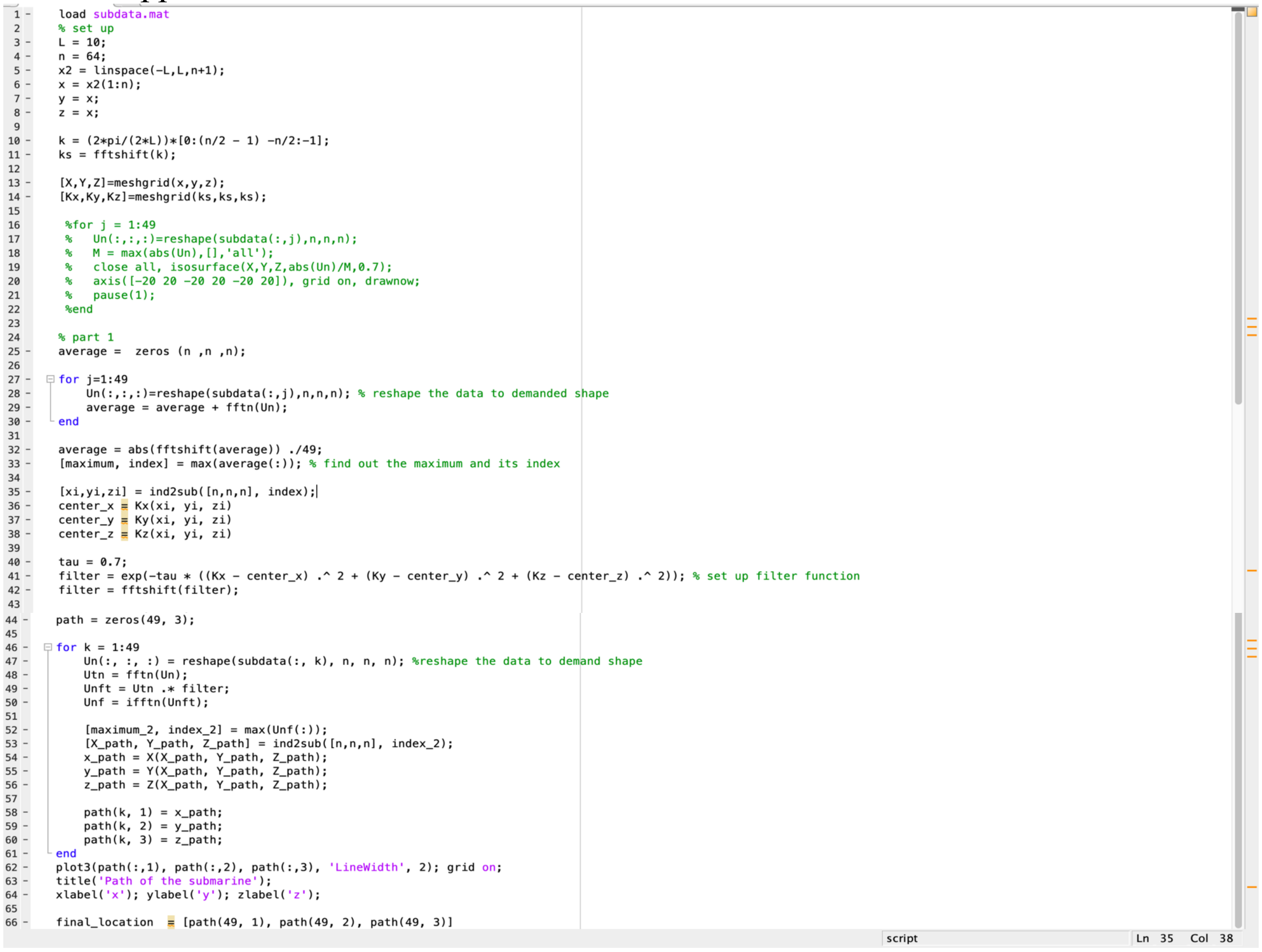
5 Conclusion

For this project, our purpose is to track the location of a submarine when given some data. Sometimes the data will be noisy, and, in that case, we need to apply spectrum averaging to find our center frequency. After this, we are going to apply filter around the center frequency to denoise our data. As we can see, in the whole procedure, FFT and IFFT are used a lot since we need to conversion between space domain and frequency domain frequently. The purpose of spectrum averaging and applying Gaussian filter around the center frequency are always utilized together for the purpose of eliminating the noisy data and reshape the data so that it could just focus on what we are interested in.

Appendix A. Matlab Functions.

1. Uf = fftn(Un) : when a function is multidimensional, we apply Fournier transform for it to convert between time/space domain and frequency domain.
2. Un = ifftn(Un) : this is the inverse Fournier transform.
3. [maximum, index] = max() : sort the maximum value within a matrix or vector and find its corresponding index.
4. Un(:, :, :) = reshape(subdata(:, i), :, :, :) : reshape the input data size to the size we want.
5. fftshift(x) = rearrange data x by putting all zero frequency data to the center of the data.

Appendix B. Matlab codes



```
load subdata.mat
% set up
L = 10;
n = 64;
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);

[X,Y,Z]=meshgrid(x,y,z);
[Kx,Ky,Kz]=meshgrid(ks,ks,ks);

 %for j = 1:49
 %   Un(:,:,:)=reshape(subdata(:,j),n,n,n);
 %   M = max(abs(Un),[],'all');
 %   close all, isosurface(X,Y,Z,abs(Un)/M,0.7);
 %   axis([-20 20 -20 20 -20 20]), grid on, drawnow;
 %   pause(1);
 %end

% part 1
average =  zeros (n ,n ,n);

for j=1:49
    Un(:,:,:)=reshape(subdata(:,j),n,n,n); % reshape the data to demanded shape
    average = average + fftn(Un);
end

average = abs(fftshift(average)) ./49;
[maximum, index] = max(average(:)); % find out the maximum and its index

[xi,yi,zi] = ind2sub([n,n,n], index);
center_x = Kx(xi, yi, zi)
center_y = Ky(xi, yi, zi)
center_z = Kz(xi, yi, zi)

tau = 0.7;
filter = exp(-tau * ((Kx - center_x) .^ 2 + (Ky - center_y) .^ 2 + (Kz - center_z) .^ 2)); % set up filter function
filter = fftshift(filter);

path = zeros(49, 3);

for k = 1:49
    Un(:, :, :) = reshape(subdata(:, k), n, n, n); %reshape the data to demand shape
    Utn = fftn(Un);
    Unft = Utn .* filter;
    Unf = ifftn(Unft);

    [maximum_2, index_2] = max(Unf(:));
    [X_path, Y_path, Z_path] = ind2sub([n,n,n], index_2);
    x_path = X(X_path, Y_path, Z_path);
    y_path = Y(X_path, Y_path, Z_path);
    z_path = Z(X_path, Y_path, Z_path);

    path(k, 1) = x_path;
    path(k, 2) = y_path;
    path(k, 3) = z_path;
end
plot3(path(:,1), path(:,2), path(:,3), 'LineWidth', 2); grid on;
title('Path of the submarine');
xlabel('x'); ylabel('y'); zlabel('z');

final_location  = [path(49, 1), path(49, 2), path(49, 3)]
```