Homework 1: An ultrasound problem

Alyssa Jin

01/24/2020

Abstract

The dog Fluffy swallowed a marble. We need to use ultrasound to track the position of marble in suspected area in intestines of Fluffy. This report shows a method of denoising noise for an ultrasound data set by Fast-Fourier transform (FFT) and a Gaussian filter. The data set consists 20 different measurements that were taken in time. The frequency signature (center frequency) generated by the marble is located in frequency domain through averaging of the spectrum. We can use the data around the center frequency to determine the path of the marble. Then breakup the marble with an intense acoustic wave at the 20th data measurement.

I. Introduction and Overview

In this problem, we have a data set of 20 ultrasound measurements from Fluffy, which is a dog with marble in its intestine. To save Fluffy, we need to locate and compute the find the path of the marble, and report the position of the marble at the last measurement, then we can apply an intense acoustic wave to breakup the marble. To compute the problem, we need to determine the center frequency, which is the wave signature of the marble, then we can focus on this wave to save Fluffy.

II. Theoretical Background

II.i Fourier Transform

The Fourier Transform is an integral transform defined over the entire line $x \in [-\infty, \infty]$. The Fourier transform is defined as

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

The inverse transform is

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

II.ii Fast-Fourier Transform (FFT)

The Fast Fourier transform (FFT) is specifically for forward and backward Fourier transforms. FFT is an algorithm, which has a low operation count: O(N logN), the solutions on finite interval have periodic boundary conditions, and the number of points in a finite domain should be 2^n. The FFT has excellent accuracy properties, which is better than standard discretization schemes.

III. Algorithm Implementation and Development

(1) Load data from Testdata.mat
(2) Define axes of spatial domain. Divide $x, y, z \in [-L, L]$ to n values along each axes.
(3) Define axes of frequency domain. Because I use Fast-Fourier Transform (FFT) in code, the periodic should be 2π. Therefore, I need to rescale frequencies by multiplying with $\frac{2\pi}{L}$ in $[-\frac{n}{2}, \frac{n}{2})$. The axes should be changed from $[0, k_{max}]$ to $[k_{min}, 0]$
(4) Build the coordinate systems by *meshgrid*
(5) Reshape the data to 3-dimensional by *reshape*
(6) Plot the ultrasound data of the first measurement
(7) Using $fftn$ for FFT with the 20 data, and calculate the average of spectrum by dividing the sum of absolute value in **uave** with the largest value **specmax**. Becuase **uave** is a complex double matrix, so i need to use absolute value
(8) Plot isosurface in frequency domain
(9) Find central frequency by $find$
(10) Build a 3-dimensional Gaussian filter around the central frequency
(11) Multiply Gaussian filter on each spectrum **Un**and get **unf** by $ifftn$.
(12) Locate the strongest signal, which is the largest absolute value, by $ind2sub$ with each element in **unf**
(13) Plot the strongest signal by $plot3$
(14) Find the location of the marble at the 20th measurement: **finallocation**

IV. Computational Results

The isosurface of the dataset with isovalue equals to 0.4 for the 1st measurement is plotted in Figure 1, which is scattered to almost every position in Figure 1, so it is very difficult to tell the exact location of the marble.
The isosurface of the dataset with isovalue equals to 0.6 in the frequency domain after averaging is plotted in Figure 2. The central frequency is $[kx; ky; kz] = [1.885, -1.047, 0]$.
The corresponding path of the isosurface of the dataset with isovalue equals to 0.6 of all positions is plotted in Figure 3.
The final position of the marble is computed as $x = -5.625; y = 4.2188; z = 6.0938$.

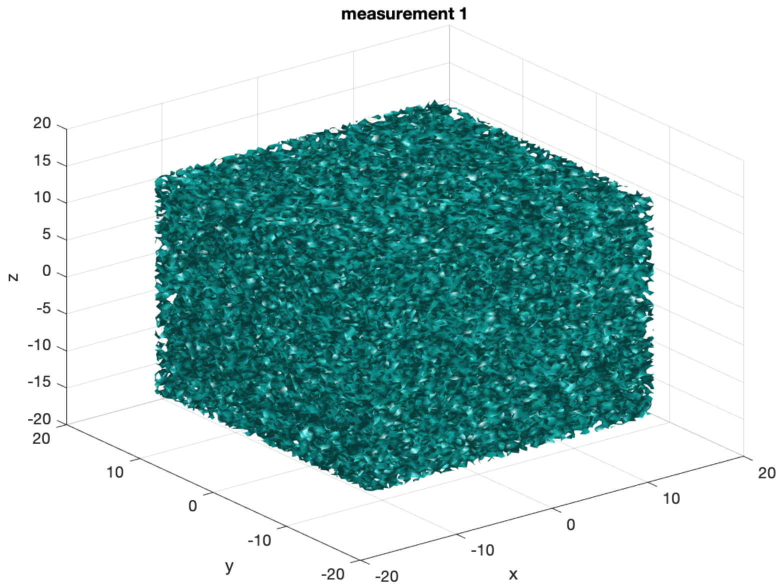

FIGURE 1. The isosurface of the dataset with isovalue equals to 0.4 for the 1st measurement

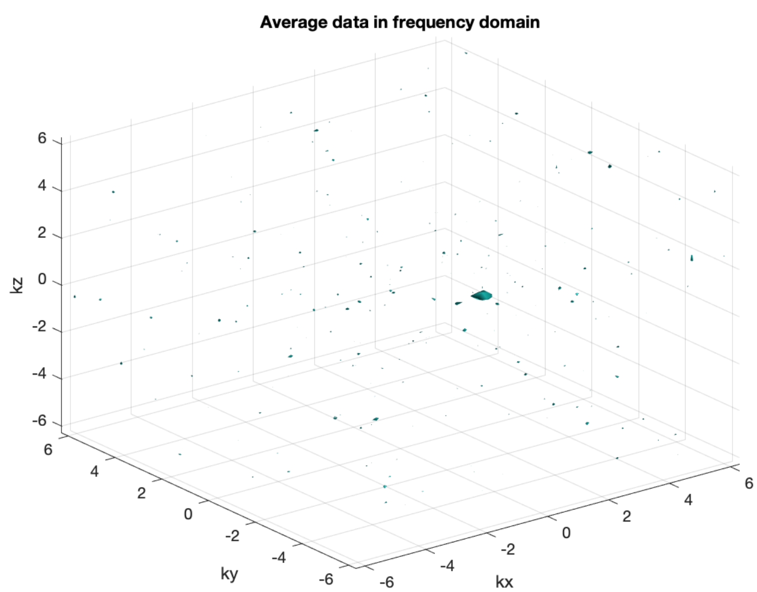

FIGURE 2. The isosurface of the dataset with isovalue equals to 0.6 in the frequency domain after averaging

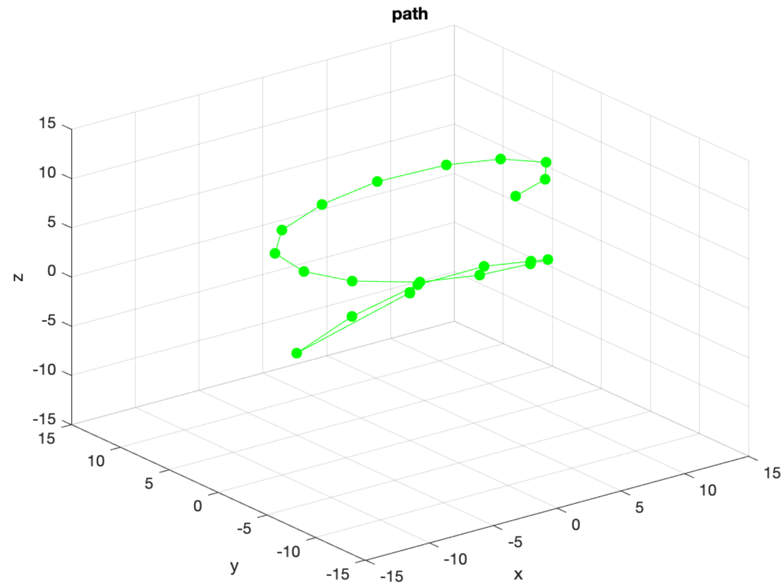

FIGURE 3. The corresponding path of the isosurface of the dataset with isovalue equals to 0.6 of all positions

V. Summary and Conclusions

We saved Fluffy. At first, the figure looks scattered, so I cannot find the exact place of marble in the intestine of Fluffy. I use Fast-Fourier transform to average the spectrum, and then find the center frequency, which help me locate the position of marble. I use Gaussian filtering around that center frequency to determine the path of the marble by using *plot*3. I use the inverse Fourier transform to find the new measurement of the marble location, then I saved Fluffy. The final position of the marble is at (-5.625, 4.219, -6.094), where an intense acoustic wave should be focused.

Appendix A. MATLAB functions used and brief implementation explanation

(1) $linspace(x1, x2, n)$: Return n points. The spacing between the points is $(x2-x1)/(n-1)$.
(2) $fftshift(X)$: Rearranges a Fourier transform X by shifting the zero-frequency component to the center of the array.
(3) $meshgrid(x, y, z)$: Returns 3-D grid coordinates defi
ned by the vectors x, y, and z. The grid represented by X, Y, and Z has size length(y)-by-length(x)-by-length(z).
(4) $reshape(A, sz1, ..., szN)$: Reshapes A into a sz1-by-...-by-szN array where sz1,...,szN indicates the size of each dimension.
(5) $isosurface(X, Y, Z, V, isovalue)$: Computes isosurface data from the volume data V at the isosurface value specified in isovalue.
(6) $fftn(X)$: Returns the multidimensional Fourier transform of an N-D array using a fast Fourier transform algorithm.
(7) $[I1, I2, ..., In] = ind2sub(sz, ind)$: Returns n arrays I1,I2,...,In containing the equivalent multidimensional subscripts corresponding to the linear indices ind for a multidimensional array of size sz.
(8) $[row, col] = find(X)$:Returns the row and column subscripts of each nonzero element in array X.
(9) $zeros(sz1, ..., szN)$: Returns an sz1-by-...-by-szN array of zeros where sz1,...,szN indicate the size of each dimension.
(10) $ifftn(Y)$: Returns the multidimensional discrete inverse Fourier transform of an N-D array using a fast Fourier transform algorithm.
(11) $plot3(X1, Y1, Z1, ...)$: Plots one or more lines in three-dimensional space through the points whose coordinates are the elements of X1, Y1, and Z1.

Appendix B. MATLAB codes

```
%% load data
clear all; close all; clc;
load Testdata

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);

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

% plot the ultrasound data of the first measurement
figure(1);
Un1(:,:,:)=reshape(Undata(1,:),n,n,n);
isosurface(X,Y,Z,abs(Un1),0.4);
axis([-20 20 -20 20 -20 20]), grid on, drawnow
xlabel('x'); ylabel('y');zlabel('z');
title("measurement 1");
pause(1);
print(gcf,'-dpng','Figure 1.png');

%% part 1
% Through averaging of the spectrum, determine the frequency signature
% (center frequency) generated by the marble.

% Define uave as the average of all points in dataset
uave = zeros(n,n,n);

% Use for loop to calculate all measurement
for j=1:20
    Un(:,:,:)=reshape(Undata(j,:),n,n,n); % construct 3d matrix
    u = fftn(Un); % frequency domain
    uave = uave + u;
end

% average the spectrum
specmax = max(max(max(abs(uave))));
specuave = abs(uave)/specmax;

% plot isosurface in frequency domain
close all;
figure(2);

isosurface(Kx,Ky,Kz,fftshift(specuave),0.6);
```

```
axis([-2*pi 2*pi -2*pi 2*pi -2*pi 2*pi]), grid on, drawnow;
xlabel('kx'); ylabel('ky');zlabel('kz');
title("Average data in frequency domain");
pause(2);
print(gcf,'-dpng','Figure 2.png');

%% part 2
% Filter the data around the center frequency determined above in order to
% denoise the data and determine the path of the marble.
close all;

% find max in frequency domain, which is the central frequency
[i, j, m] = ind2sub([n,n,n], find(fftshift(abs(uave)) == ...
    max(max(max(fftshift(abs(uave)))))));

% make a filter
filter = exp(-1.0*(kx-Kx(i,j,m)).^2) .* exp(-1.0*(ky-Ky(i,j,m)).^2) .* ...
    exp(-1.0*(kz-Kz(i,j,m)).^2);

% compute and plot the data
location = zeros(n,3);

figure(3); % for plotting the isosurface in spatial domain

 for l = 1:20
 % apply filter on each spectrum and ifft back
 Un(:,:,:)=reshape(Undata(l,:),n,n,n);
 unf = ifftn(filter.* (fftn(Un)));

 % locate the strongest signal
   [i1,j1,m1] = ind2sub([n,n,n], find(abs(unf) == max(max(max(abs(unf))))));
   location(l,:) = [X(i1,j1,m1), Y(i1,j1,m1), Z(i1,j1,m1)];
 end

 % plot the trajectory
plot3(location(:,1), location(:,2), location(:,3), 'g.-', 'MarkerSize', 20);
axis([-L L -L L -L L]);
title("path");
xlabel("x"); ylabel("y"); zlabel("z");
grid on, drawnow
print(gcf,'-dpng','Figure 3.png');

%% part 3
% The location the intense acoustic wave should be focused to breakup the
% marble at the 20th data measurement.
```

```
close all;

finallocation = location(20,:)
```