AMATH 482 Winter 2020

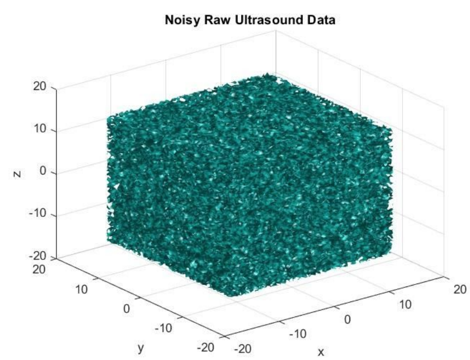
Homework 1: An Ultrasound Problem
Tanner Graves

Abstract:

This project demonstrates the use of spectral methods, specifically the Fast Fourier Transform (FFT), for signal filtering in 3 spatial dimensions. Additionally, by introducing a time dimension, it is demonstrated that the Fourier Transform is not sensitive to position data. Here this is utilized to discern an object across several data sets by use of its 'frequency signature'. Ultimately, the FFT proves to be an effective tool for filtering.

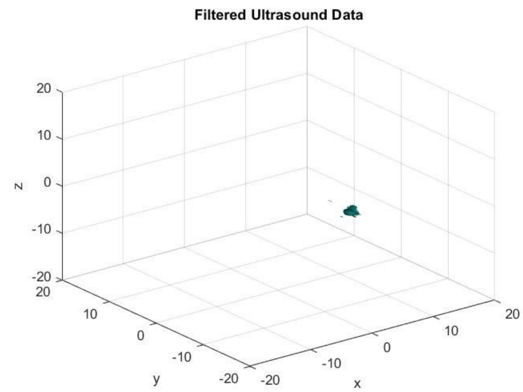
I. Introduction and Overview

The problem is as follows: my beloved dog Fluffy has ingested a marble some time ago and as it works its way into the dogs intestines, the vet fears complications could arise. To investigate further the vet uses an ultrasound. 19 consecutive, 3-dimensional measurements were taken of Fluffys interior. However, the fluids contained in the dogs organs combined with constant movement has resulted in noisy data.



To make sense of this data, and to be able to find out where the marble is, the ultrasound data will need to be filtered.

Filtering will be done using the FFT, decomposing the ultrasound data into its x, y , and z frequency components for each measured point in time. The resulting data in frequency space will consist primarily of undesirable information. The majority of frequencies will correspond to the noise visible in the figure above. The problem is now to decide what information we wish to keep and which we want to discard. Since the marble is a prominent feature present at each time, the goal is to find which frequencies correspond to the marble. This is done by averaging all the frequencies over time, and finding the largest. Once this signature frequency is obtained, it may be used as the center for the filter by which all the other frequencies are scaled. This allows to discriminately lessen the frequencies that correspond to noise and keep those that correspond to the marble. From here, we may use the Inverse Fourier Transform return back to space.



Now that the position of the marble can be obtained, we may plot the course that it is taking though time, and extrapolate where it will be on the 20th time so we may destroy it with a concentrated sonic vibration.

II. Theoretical Background

The Fast Fourier Transform is a variation of the fourier Transform that has been discretized and optimized for numerical calculation. In function, however, they remain the same in that they take a signal and deconstructs into its constituent frequencies.

The fourier transform is defined as follows:

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

And its inverse:

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

Immediately, we are presented a problem for its application. We cannot compute an improper integral, so any practical implementation the Fourier transform will be computed on a finite interval, and this is the case for the FFT. The FFT enjoys has several notable features that affect its application, both specifically and in scope. The FFT is incredibly efficient, taking only N log N time; it must work on a periodic function on a bounded interval; the sampled points must be a power of two, as a result of its optimization; it behaves very nicely under differentiation: $F' = ikF$,where F is the Fourier Transform of a function; and as is typical amount spectral methods, it enjoys remarkable accuracy when compared to alternative methods.

The Fourier Transform was useful in this application, as it allowed for filtering. By taking noisy spatial data and converting it into constituent frequencies, we may then selectively remove frequency information that corresponds to noise while maintaining the frequency data that corresponds to our desired signal. A common choice for such a filter is a gaussian. By multiplying each data point to a corresponding point on the filter, we can limit the frequency information.

III. Algorithm Implementation and Development

The first task was to calculate frequency signature of the marble. To do this I utilized the starter code in the assignment to analyze the fourier transform of the data at each of the given 19 times. While doing this I would summate resulting 64 x 64 x 64 frequency space matrices to compute the average. Of all the component frequencies over all 19 times. The frequency with the maximum absolute value, will correspond the the signature frequency of the marble. Because of the implementation of the max function in MATLAB where its default behaviour is only to max over one dimension, I first converted the data into a 1 x 64^3 matrix and maxed over that. I was then able to translate the resulting index back to the standard row, column, place notation of the 64 x 64 x 64 matrix. K values may then be retrieved at these indices to get the signature frequency.

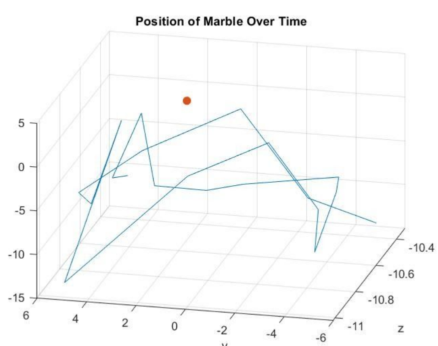
Next was the issue of filtering the ultrasound data to ascertain the position of the ball. Again for each of the given times I would reformat the data into a 64 x 64 x 64 matrix, transform using the 3D FFT to frequency space, filter using a 3D gaussian about the previously found signature frequency, and transform back after removing the noise. Isosurface was useful not only for plotting, but it has additional functionality in that you can request the vertices that comprises the Surface of the isoplot. By taking the mean of these points at each time will give you the balls position at that time.

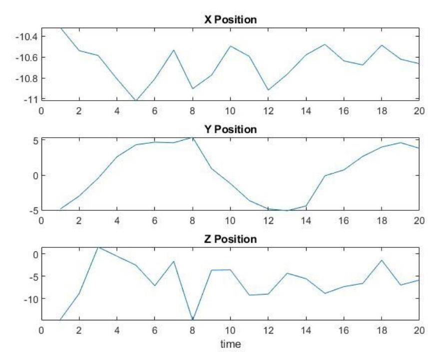
The final aspect of the assignment was to anticipate where the marble will be at the 20th time. To do this I used cubic extrapolation to get an additional value.

IV. Computational Results

Problem 1: The center frequency of the marble was found to be (1.8850, -1.0472, 0).

Problem 2: The X and Z components seem rather noisy, which may indicate a mistake. However, the Y position looks quite sensible. Below is a 3D plot of the marbles position in space as well as its component coordinates over time.





Problem 3: Using cubic extrapolation, we may predict the position of the marble at time 20 to be (-10.5776, 1.7682, 1.8286).

V. Summary and Conclusions

Given a set of 19 sets of 3-dimensional ultrasound data, we were able to filter that data using a FFT and 3D gaussian distribution to discern the location of a feature despite a very noisy signal. Using the successive times we were then able to extrapolate to predict the location at an additional time. Though these specific results may be untrustworthy, the FFT has proven to be an effective tool for spectrum analysis and filtering.

Appendix A: MATLAB Functions

`Meshgrid` – Useful for calculations that requires a set of matrices that increase linearly in one dimension each

`reshape` – Given a matrix and desired compatible size, will canonically reshape into a matrix of desired dimension
`Fftn` – adaptation of the FFT to work in arbitrary dimension
`Ind2sub` – Converts linear indices to 2D matrix indices
`Isosurface` – Plots a single equipotential surface of a 4 dimensional function in 3D, may also be used to get vertex data
`interp1` – Capable of poth interpolation and extrapolation in one dimension using a variety of methods
`plot3` – used for plotting a curve in 3-space

Appendix B: MATLAB Code

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

L = 15;
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(k,k,k);
Ksum = zeros(64,64,64);
for t = 1:20
    Un(:,:,:)=reshape(Undata(t,:),n,n,n);
    K = fftn(Un);
    Ksum = Ksum+K;
end
Kavg = Ksum/20;
[val, ind] = max(Kavg(:));[r,c,p] = ind2sub(size(Kavg),ind);%Kavg(r,c,p)
k0x = k(c);%took me way too long to figue out rows and colums are flipped
k0y = k(r);
k0z = k(p);
k0 = [k0x;k0y;k0z]; %answer to problem one: find frequency center
%%
%%Problem 2 - filter the spacial data
tau = 1;
filter = exp(-tau*((Kx-k0x).^2 + (Ky-k0y).^2 + (Kz-k0z).^2));
pos = zeros(3,20);
for t = 1:20
    Un(:,:,:)=reshape(Undata(t,:),n,n,n);
    K = fftn(Un);
    Kft = K.*filter;
    Unft = ifft(Kft);
    figure(1);
    close all, isosurface(X,Y,Z,abs(fftshift(Unft)),28);
    axis([-20 20 -20 20 -20 20]), grid on, drawnow;
    figure(2);
```

```
    [f,v] = isosurface(X,Y,Z,abs(Unft),28);
    pos(:,t) = mean(v,1);
    pause(1)
end
%%
Xp = pos(1,:);
Yp = pos(2,:);
Zp = pos(3,:);
plot3(Xp,Yp,Zp);
grid on;
title('Position of Marble Over Time');
xlabel('x');ylabel('y');xlabel('z');
figure(3);
subplot(3,1,1);
plot(Xp);
title('X Position')
subplot(3,1,2);
plot(Yp);
title('Y Position')
subplot(3,1,3);
plot(Zp);
title('Z Position')
xlabel('time')

%%
%problem 3
X21 = interp1(1:20, Xp, 21, 'cubic', 'extrap');
Y21 = interp1(1:20, Yp, 21, 'cubic', 'extrap');
Z21 = interp1(1:20, Zp, 21, 'cubic', 'extrap');
pos21 = [X21;Y21;Z21] %answer to problem 3
figure(2);
hold on;
plot3(X21,Y21,Z21,'.','MarkerSize',25)
```