AMATH 482: HOMEWORK 1

NICHOLAS NUGRAHA

Applied Math Department, University of Washington, Seattle, WA
nickjn@uw.edu

Abstract. This paper analyzes submarine tracking using acoustic frequency data recorded over 24 hours in the Puget Sound. By employing Fast Fourier Transform (FFT), the study identifies dominant frequencies in the data and applies Gaussian filtering to reduce noise, enabling clearer trajectory visualization. The analysis involves computing N-dimensional Fourier transforms, averaging magnitudes, and applying 3D Gaussian filters to reconstruct the submarine's path in a 3D space. While the method effectively identifies the dominant frequency and visualizes the trajectory, some residual noise persists, suggesting the need for advanced filtering techniques for more precise tracking in practical applications.

1. Introduction and Overview

Underwater submarines are a crucial piece of technology that are used in a multitude of settings, including military operations, scientific research and exploration, and even industrial applications as well. In order to keep track of these submarines, sonar systems located on the submarine give off acoustic frequencies, which scientists can then use to measure their location within the ocean.

In this homework, our goal is to locate a submarine that is moving in the Puget Sound. The submarine we are trying to track is a new technology that emits an unknown acoustic frequency, one that we will try and detect in this assignment.

We are given a data file that contains a broad spectrum recording of acoustics pressure data obtained over a 24 hour period in half-hour increments. The data itself is a matrix with 49 columns of data, one for each increment, that correspond to the measurements of acoustic pressure taken over the 24 hour period. Essentially, the data is four-dimensional because we are provided 3D pressure measurements as a function of time.

To analyze the frequencies, we must first break down the given signal (in our case the submarine data) into its frequency components. However, doing so gives us no information about *when* in time the frequency components occurred. This is where the Fast Fourier Transform (FFT) is most helpful to us. Using FFT, we are able to analyze and manipulate the data in the frequency domain, allowing us to create visuals for this submarine as well.

Due to the random nature of how electromagnetic fields behave when coming into contact with other electromagnetic fields, signals like our submarine data contain lots of noise. Noise makes it more difficult to detect the frequencies we are trying locate, which is why we also implemented a filter to determine a more robust path of the submarine.

2. Theoretical Background

In order to understand the Fourier transform, we must first dive a little into the Fourier series. In the early 19th century Fourier discovered a way to represent a periodic function as a sum of sine and cosine terms:

Date: January 25, 2025.

$$f(x) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos(nx) + b_n \sin(nx)\right), \quad x \in (-\pi, \pi].$$

This equation breaks down the function into a sum of harmonic frequencies where the coefficients a_n and b_n are the Fourier coefficients [3]. The Fourier coefficients describe how much of each sine and cosine wave is present in the function. This is essentially the amplitude and phase of the frequency components. The complex version of this expansion can be written as:

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{in\pi x/L}, \quad x \in [-L, L] \tag{1}$$

with the corresponding Fourier coefficients

$$c_n = \frac{1}{2L} \int_{-L}^{L} f(x) e^{-in\pi x/L}\, dx$$

In equation (1) the frequencies are discrete because $f(x)$ is periodic. In the case of the Fourier transform, $f(x)$ is no longer periodic and is defined over all $x \in \mathbb{R}$. As a result, the Fourier transform and its inverse are defined as:

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

In the transform, the interval grows to infinity ($L \to \infty$) and the discrete frequencies become a continuous variable k that represents all possible frequencies. The Fourier coefficients c_n become the Fourier transform which represent the amplitudes of the frequency k. In order to locate the submarine, we must find its dominant frequency by finding the maximum amplitude of the frequencies.

As is with most frequency data, there comes a lot of noise that makes it more difficult to locate the dominant frequency. For this, we need to use a simple Gaussian filter to denoise the data:

$$F(k) = \exp\left(-\tau(k - k_0)^2\right)$$

where τ is the bandwidth of the filter (what you are trying to filter out) and k is the wavenumber (frequencies of the submarine). Since we know the wavenumber, we can filter around this frequency to remove the undesired frequencies and white noise picked up during detection.

3. Algorithm Implementation and Development

The algorithm we will use to process the signal is the Fast Fourier transform (FFT), provided in the `numpy` Python package [1], which is basically just a fast version of the discrete Fourier transform. There is one caveat that applies to this dataset that adds a level of complexity. Earlier we mentioned that the data was essentially four-dimensional (3D data + time). Because of this, we must use an N-dimensional Fourier transform. Luckily the `numpy` package has a built in function that computes the N-dimensional discrete Fourier transform: `numpy.fft.fftn()`.

By performing the discrete Fourier transform on each increment of time, we can average the magnitudes of the frequencies across the 24-hour period to determine the dominant frequency. However, `numpy`'s FFT algorithm does some unique things to the data: it shifts the data so that the zero-frequency component is at the start of the array. This is inconvenient for analysis reasons since we expect the zero-frequency to be in the middle. To fix this, we use the `numpy.fft.fftshift()` function to shift the zero-frequency component to the center.

In order to deal with the noise, we also implement a simple Gaussian filter. A 3D Gaussian filter is the product of three Gaussian filters:

$$G(\kappa_x, \kappa_y, \kappa_z) = e^{-\tau\left((\kappa_x-\kappa_x^*)^2+(\kappa_y-\kappa_y^*)^2+(\kappa_z-\kappa_z^*)^2\right)}$$

This can be applied to N dimensions as a product of N Gaussian filters, which we compute in the code [2].

4. Computational Results

After performing the N-dimensional FFT, shifting the frequencies, and averaging over all the magnitudes, we can find the index of the dominant frequency by taking the maximum frequency. This tells us where our dominant frequency is located in our signal. In our case, the dominant frequency was located at the indices `(39, 49, 10)`. By indexing the magnitudes with our dominant frequency index, we found the dominant frequency to be at 2.53. In order to visualize this better, we plotted a 1D scatter plot of the flattened average magnitudes in Figure 1

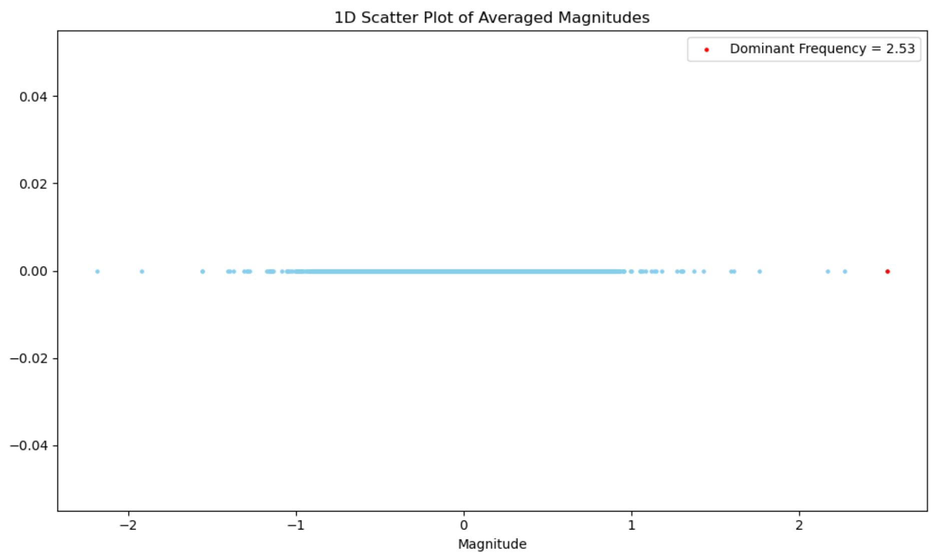


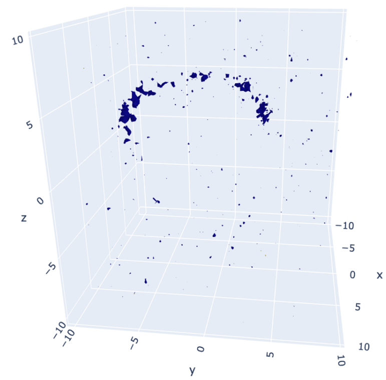
Figure 1. Scatter plot of averaged magnitudes, showing the dominant frequency at the right end of the plot

Figure 1 plots all the magnitudes in one dimension so the dominant frequency will appear at the right end of the plot. The dominant frequency has a magnitude at 2.53, which aligns with what our plot shows as well.

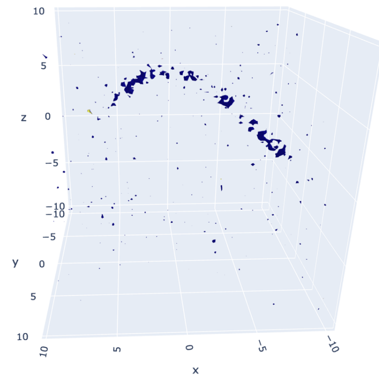
Now that we have found the dominant frequency, we can use the Gaussian filter to denoise the data in order to plot its movement on a 3D grid (Figure 2).

Figures 2a, 2b, and 2c show a clear trajectory of the submarine's movement in space (x, y, z). The congregation of points that form the path are the same dominant spatial frequencies we found previously. We can also see the depth the submarine went by analyzing the z-axis. Because of the Gaussian filter, we can visualize the path of the submarine much clearer. However, there is still some noise that you can see in the plot. In this case, it's enough to make out the trajectory, but in real-world applications this may not be sufficient.

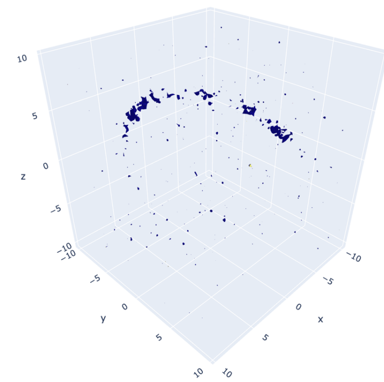
We also plotted just the x and y coordinates of the submarine's path in Figure 3, in order to get a top-down view of the submarine if you were to ever track it from above.



(A) Left view of the 3D path.



(B) Right view of the 3D path.



(C) Center view of the 3D path.

FIGURE 2. Different views of the 3D path.

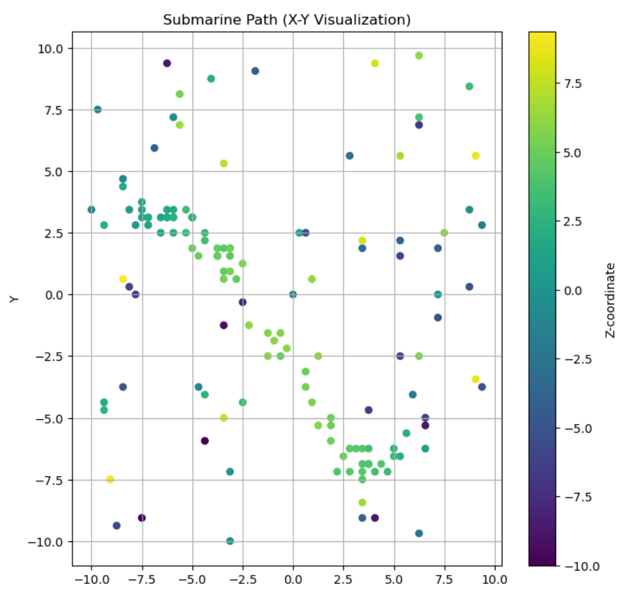


FIGURE 3. Path of the submarine in the X-Y plane. Also contains a depth scale for the z-coordinate

In Figure 3, the X-Y path is easy to see, making it easier to track from above, as would be the case in most situations. Again, the noise is still visible, but the path remains clear.

5. Summary and Conclusions

In this analysis, the Fourier transform and Gaussian filtering techniques were used to analyze frequency measurements coming from a submarine. The main goal was to reconstruct the path of the submarine by locating the dominant frequencies in the data, filtering to get rid of the noise, and creating a 3D spatial representation of its path.

While the trajectory of the submarine was clear enough to visualize, there still remained some noise within the frequencies. The Gaussian filter is a very simple filter, so using a more advanced filter could potentially help with that.

Acknowledgements

The author is thankful to Professor Natalie Frank for useful lectures about Fourier Transforms. We are also thankful to TA Rohin Gilman for assisting in office hours. We also acknowledge that artificial intelligence was used to make edits to this document, fix minor bugs, and gain a better understanding of the concepts.

References

[1] T. N. Community. *NumPy Documentation*, 2025. Accessed: 2025-01-25.
[2] N. Frank. Amath 482: Computational methods for data analysis. Lecture notes, University of Washington, 2025.
[3] J. N. Kutz. *Data-driven modeling & scientific computation: methods for complex systems & big data*. OUP Oxford, 2013.
[4] OpenAI. Chatgpt. `https://chat.openai.com/`, 2025.