# AMATH 482/582: HOMEWORK 1

HAI VAN LE

*Applied Mathematics Department, University of Washington, Seattle, WA*
levh@uw.edu

Abstract. This project is to illustrate the method to locate a submarine in the Puget Sound using noisy acoustic data. It is important to note that the submarine is in motion and emits an unknown acoustic frequency so the focus is to find its location and trajectory, using Fast Fourier Transform.

## 1. Introduction and Overview

This project is to implement Gaussian filter to locate a moving submarine in the Puget Sound area and determine its path using broad spectrum recording of its acoustics data acquired during 24 hours in half-hour increments. This submarine has been equipped with advanced technology that emits an unknown acoustic frequency that needs to be detected. The tasks represent an example of object detection and tracking, a technique widely used in animal detection, surveillance, medical imaging, etc.

## 2. Theoretical Background

This project deploys Fourier Seriers, Fourier Transform and Fast Fourier Transform [2], all of which are explained briefly below.

2.1. **Fourier Series.** The foundation for Fourier Series is that all functions and their derivatives can be represented as sums of two trigonometric series: sines and cosines:

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty}(a_n\ cos(nx) + b_n\ sin(nx))\ , x \in (\pi, \pi]$$

where

$$a_n = \frac{1}{n}\int_{-\pi}^{\pi} f(x)cos(nx)dx, n \geq 0$$

,

$$b_n = \frac{1}{n}\int_{-\pi}^{\pi} f(x)sin(nx)dx, n > 0$$

are the Fourier coefficients. These coefficients can be obtained by using the orthogonality property between sines and cosines in the $[0, 2\pi)$ interval.
From here, we can get the Fourier series on the domain $x \in [-L, L]$:

$$f(x) = \sum_{-\infty}^{\infty} c_n e^{\frac{in\pi x}{L}}, x \in [-L, L]$$

*Date*: January 25, 2024.

with the coefficients

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

.

### 2.2. Fourier Transform.

**Fourier Transform.** The Fourier Transform is the continuous form of the Fourier Series over the entire line $x \in [-\infty, \infty]$:

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

The inverse of the Fourier Transform is defined as:

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

2.3. **Fast Fourier Transform.** The Fast Fourier Transform was developed by Cooley and Tukey [1] to calculate the forward and backward Fourier transforms. It is efficient, only costing $O(NlogN)$ with a higher accuracy. Thus, Fast Fourier Tranform has contributed to the development of digital signal processing and imaging, radar detection and filtering. It is used to analyze and manipulate data in the frequency domain.

The ideas of spectral filtering is often used to improve the signal detection by denoising as it allows us to filter information at specific frequencies. For example, we may want to filter white noise, a noise that affects all frequencies the same. In this study, the fact below is helpful:

*It is known that adding mean zero white noise to a signal (Gaussian noise) is equivalent to adding mean zero white noise (Gaussian noise) to its Fourier series coefficients.*

This fact allows us to come up with a simple yet effective noise filtering technique in cases where multiple measurements are available that are subject to the same noise, such as in imaging or acoustics applications like this study's submarine problem. Given that it is random, the noise should average to zero over many samples. Consequently, it is highly likely that averaging the measurements in the Fourier domain will reduce the noise.

2.4. **Gaussian Filter.** Gaussian filter is defined as:

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

where $\tau$ meadures the bandwidth of the filter, $k$ is the wave number, and $k_0$ is the center frequency. In this project, our Gaussian filter is:

$$G(k_x, k_y, k_z) = e^{-0.1((k_x-k_{0x})^2+(k_y-k_{0y})^2+(k_z-k_{0z})^2)}$$

## 3. Algorithm Implementation and Development

This project deploys different library interfaces such as *NumPy* to load the data, *matplotlib.pyplot* - a plotting library for the Python programming language and its numerical mathematics extension NumPy. Most importantly, it uses *scipy.fft*, including *fftn* and *fftshift* for higher dimensional Fourier transform.

Specifically, the data provided is a huge matrix of size 262144 x 49 (columns contain flattened 3d matrix of size 64x64x64). After loading the data, the following steps were taken:

- Define spatial and frequency domain
- Scale frequency domain by $\frac{\pi}{L}$
- Apply Fast Fourier Transform Shift
- Create 3D arrays with x, y, z coordinates and kx, ky, kz frequency
- Computes the Fourier transform of the data for each time step and sums them up
- Unravel the index for the position with the strongest signal i.e. center frequency
- Build a simple Gaussian filter based on (2.4)

- Applies the 3D Fourier transform to the reshaped data and applies the Gaussian filter
- Locate the submarine by finding the indexes with strongest signal at each time step

Finally, we plotted the 3d and 2d path of the submarine during the 24-hour period.

## 4. Computational Results

The coordinate of the center frequency is [5.340707511102648, 2.199114857512855, -6.911503837897545]. Figure 1 illutrates the 3d path of the submarine during the 24-hour period while figure 2 shows its x,y-coordinate.

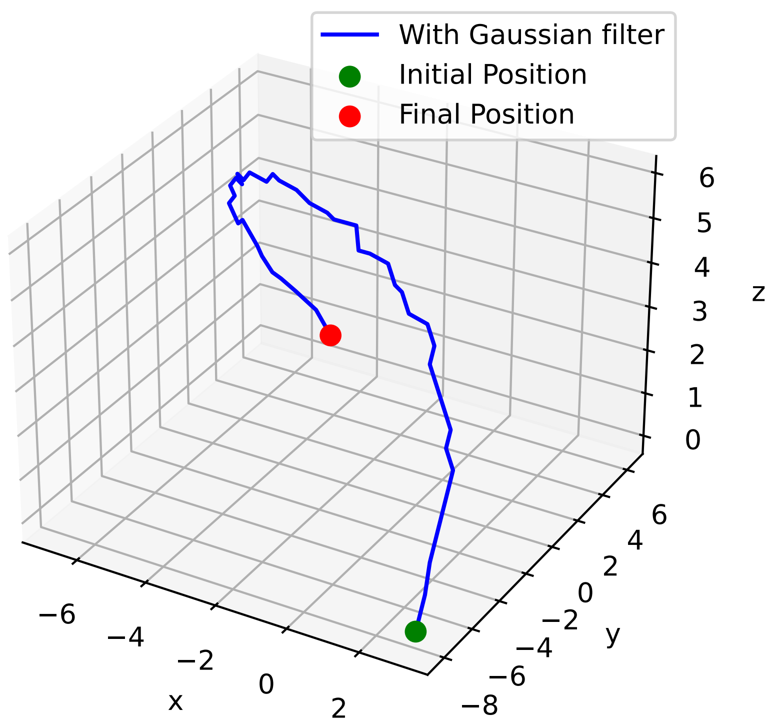


Figure 1. Path of the submarine during the 24-hour period

## 5. Summary and Conclusions

This project attempted to illustrate a simple application of the Gaussian filtering to denoise acoustic data and determine the path of a submarine. The techniques of averaging and filtering in the Fourier space helped us determine the path of the submarine despite the fact that it is in motion and the data is noisy. Given her limited capacity, the author has not explored other technique for noise filtering. Thus, possible future work is to explore the area of signal processing and object detection.

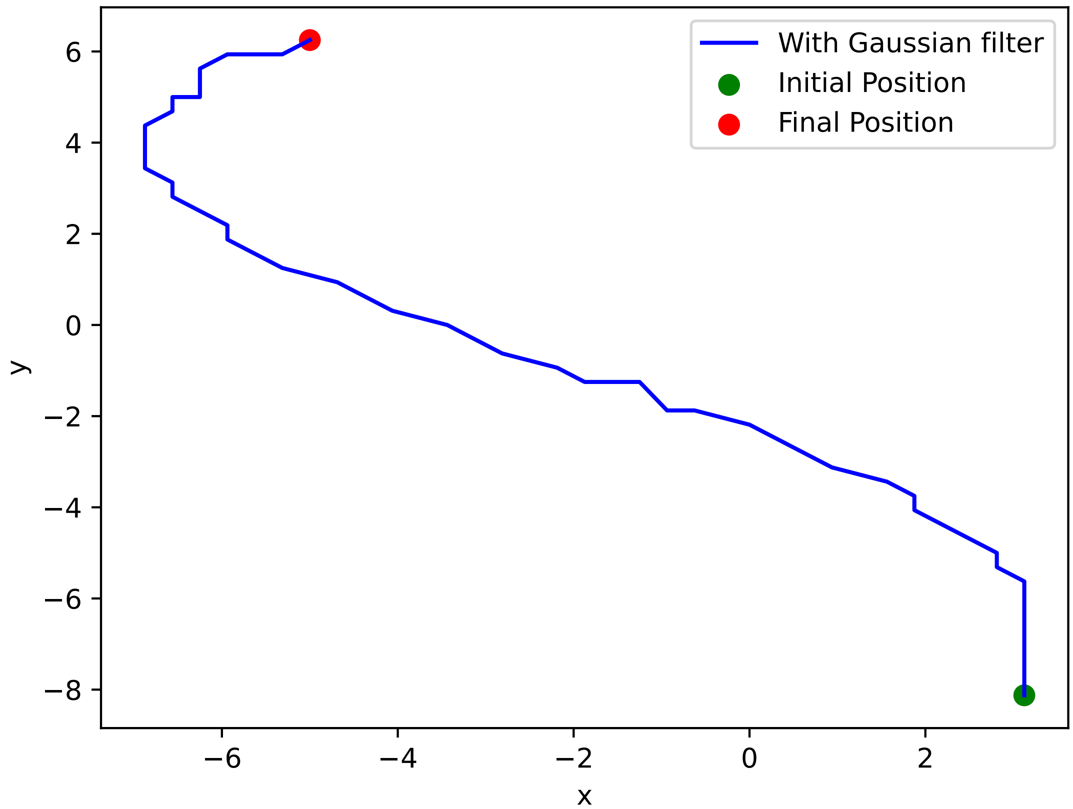


FIGURE 2. x,y-coordinates of the submarine during the 24-hour period

## Acknowledgements

The author is thankful to Prof. Eli for useful discussions about the FFT algorithm and Python libraries. She is also thankful to the TAs who answered all my questions regarding class materials and resources.

## References

[1] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

[2] J. Kutz. *Methods for Integrating Dynamics of Complex Systems and Big Data*. Oxford University Press, Oxford, 2013.