Computational Physics Homework 3

Hao Wang

October 21, 2025

Summary

In this assignment we investigated three distinct problems that highlight the power of Fourier analysis in computational physics. First, we examined the long-term record of monthly sunspot numbers, detecting the well-known solar cycle through both time-domain inspection and frequency-domain analysis. Second, we applied Fourier filtering techniques to financial data, specifically the Dow Jones Industrial Average, to explore how truncating Fourier coefficients can smooth noisy signals and reveal underlying trends. Finally, we tackled an image processing problem: deblurring a grayscale photograph that had been convolved with a Gaussian point spread function. By constructing the point spread function, transforming both the blurred image and the PSF into Fourier space, and performing deconvolution, we attempted to recover a sharper image.

These exercises collectively demonstrate how Fourier methods provide a unifying framework for analyzing periodicity, filtering signals, and restoring degraded data. Although each problem comes from a different domain (astronomy, finance, and imaging), the underlying mathematics is the same, underscoring the versatility of spectral methods.

Methods

Problem 1: Sunspot Periodicity

The dataset `sunspots.txt` contains monthly sunspot counts from January 1749 onward. We wrote a Python program to:

1. Read the data into arrays of time (months) and sunspot numbers.
2. Plot the time series to visually estimate the cycle length.
3. Subtract the mean to remove the DC component.
4. Compute the discrete Fourier transform (DFT) using `numpy.fft.fft`.
5. Plot the power spectrum $|c_k|^2$ as a function of Fourier index k.
6. Identify the dominant peak and convert its index into a physical period in months.

Problem 2: Fourier Filtering of Dow Jones Data

The dataset `dow.txt` contains daily closing values of the Dow Jones Industrial Average from late 2006 to the end of 2007. Our program:

1. Loaded the data as a one-dimensional array.
2. Computed the real Fourier transform using `numpy.fft.rfft`.
3. Zeroed out all but the first fraction of coefficients (10% and later 2%).
4. Performed the inverse transform with `numpy.fft.irfft`.
5. Plotted the original and filtered signals together for comparison.

This procedure effectively acts as a low-pass filter, retaining only the slow variations in the data.

Problem 3: Image Deblurring

The file `blur.txt` contains a 1024×1024 grayscale image blurred with a Gaussian point spread function (PSF) of width $\sigma = 25$. We implemented the following steps:

1. Loaded the image into a 2D NumPy array.
2. Constructed the Gaussian PSF on the same grid, centered at the origin (upper-left corner) and made periodic by wrap-around.
3. Computed the Fourier transforms of both the blurred image and the PSF using `numpy.fft.rfft2`.
4. Divided the Fourier coefficients of the blurred image by those of the PSF, with a cutoff $\varepsilon = 10^{-8}$ to avoid division by near-zero values.
5. Applied the inverse transform with `numpy.fft.irfft2` to obtain the recovered image.

Results

Sunspot Data

The time series plot clearly shows oscillations in sunspot numbers, with peaks roughly every 11 years. The Fourier power spectrum displayed a strong peak at period of about 131 months ($\sim$11 years). This matches the known solar cycle, validating our analysis.

Dow Jones Data

The original Dow Jones data showed significant day-to-day fluctuations. After filtering with 10% of Fourier coefficients, the smoothed curve retained the overall market trend while suppressing short-term noise. With only 2% of coefficients, the curve became even smoother, highlighting long-term behavior but losing finer detail.

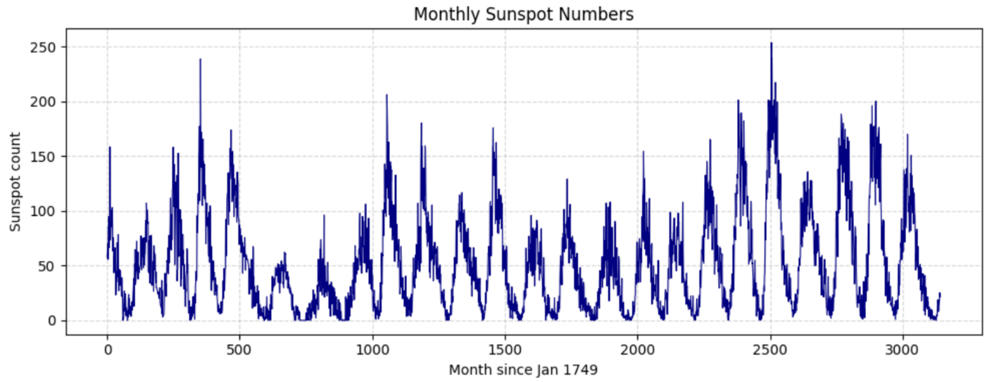

Figure 1: Monthly sunspot numbers from 1749 onward. The oscillatory pattern is obvious.

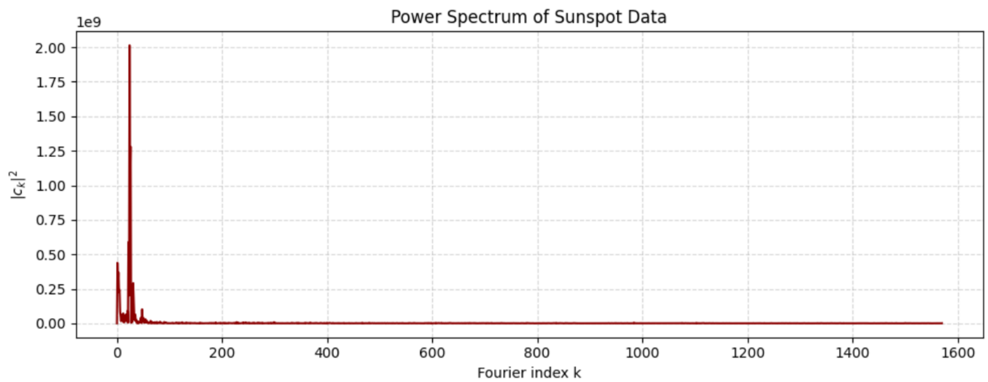

Figure 2: Power spectrum of sunspot data. The dominant peak corresponds to the ∼11-year solar cycle.

Image Deblurring

The blurred image appeared washed out, with edges and fine details obscured. The Gaussian PSF showed bright symmetric spots in all four corners, consistent with periodic wrap-around. After Fourier deconvolution, the recovered image was noticeably sharper, though not perfectly restored. Fine details remained lost due to suppression of high-frequency components in the blur.

Code

All Python scripts used for this assignment are available in the GitHub repository:
https://github.com/hw3926/Computational-Physics/tree/main/HW3

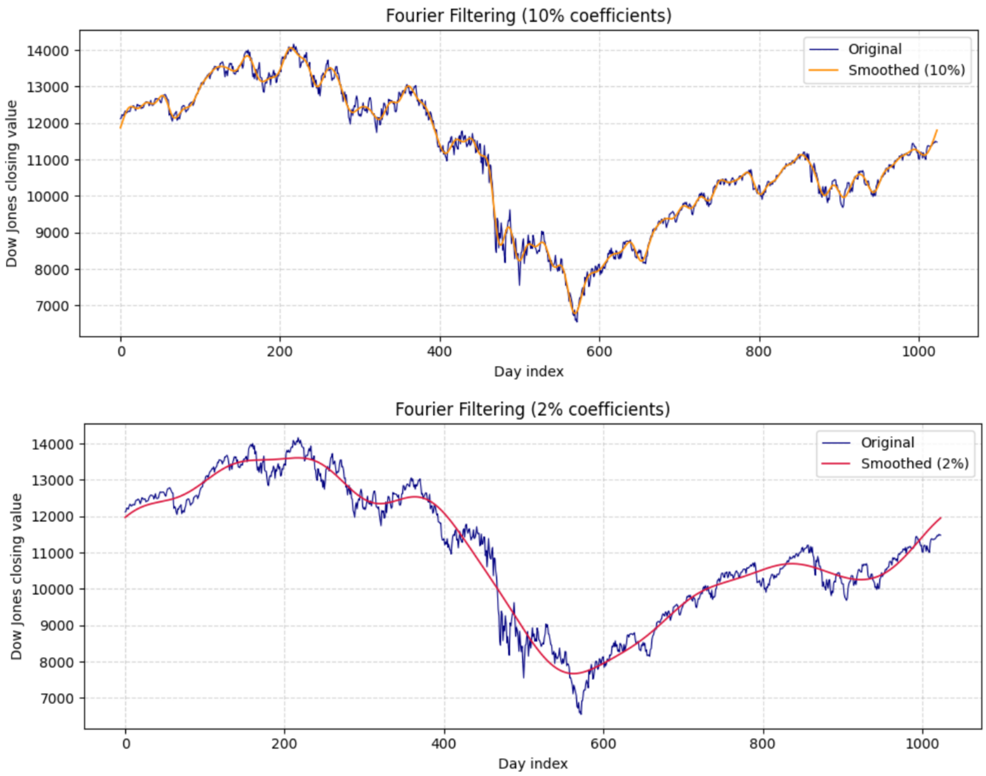

Figure 3: Dow Jones data (blue) with Fourier-smoothed versions using 10% (orange) and 2% (red) of coefficients.

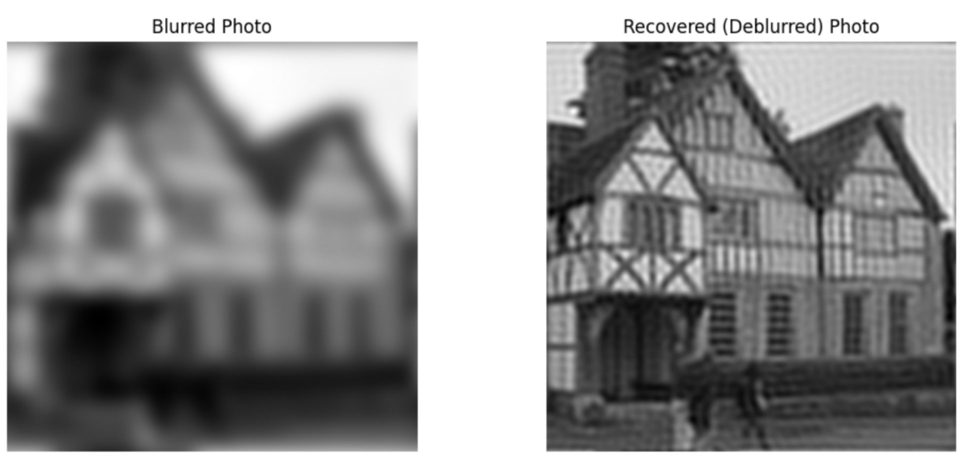

Figure 4: Recovered image after Fourier deconvolution. Sharper than the blurred input, but not perfectly restored.