Computational Physics Homework 3

Michael Albergo

25 October 2019

1 Problem 1

a) **Plot the sunspot count as a function of months. Mark off the length of one cycle.**

The number of sunspots per month is plotted below, along with an estimate of one period of the main oscillatory behavior of the dataset.

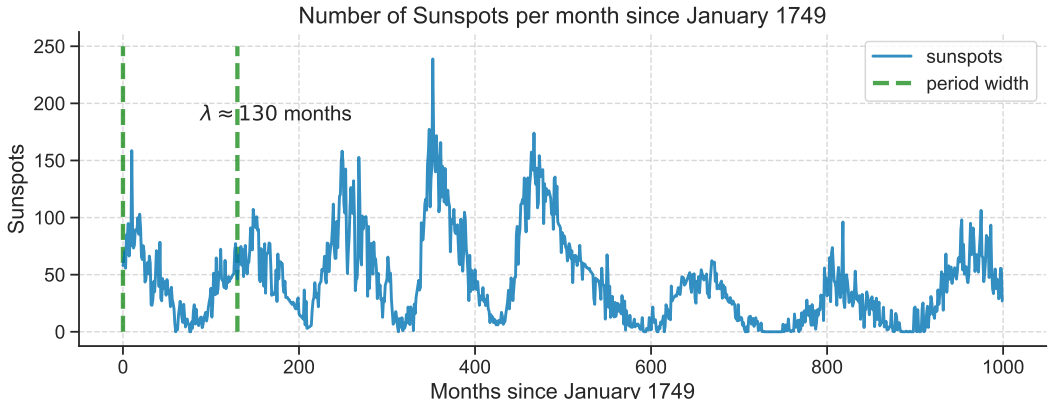

Figure 1: Number of sunspots per month from January 1749. A region of one empirical period of oscillation for the fluctuating time series is marked off by vertical green lines, estimated to be 130 months.

One empirical period of oscillation of the primary sinusoidal structure in the time series is marked off in the figure, estimated at 130 months.

b) **Calculate the Fourier coefficients, and plot $|c_k|^2$ as a function of k.**

The implementation of the DFT can be found in the Jupyter notebook in the repository. The plot of the power spectrum is provided in Figure 2.

c) **Find the approximate argmax of the the power spectrum and calculate the period of the sine wave with that value of k.**

The argmax (k corresponding to the peak) of the power spectrum is $k = 24$. The period of the sine wave of associated with the coefficient is $b = \frac{2*\pi*k}{N}$ for $\sin(bx)$ is given by

$$T = \frac{2\pi}{b} = \frac{2\pi}{2\pi\frac{k_{max}}{N}} = \frac{N}{k_{max}} = \frac{3143}{24} \approx 130.958 \text{ months} \tag{1}$$

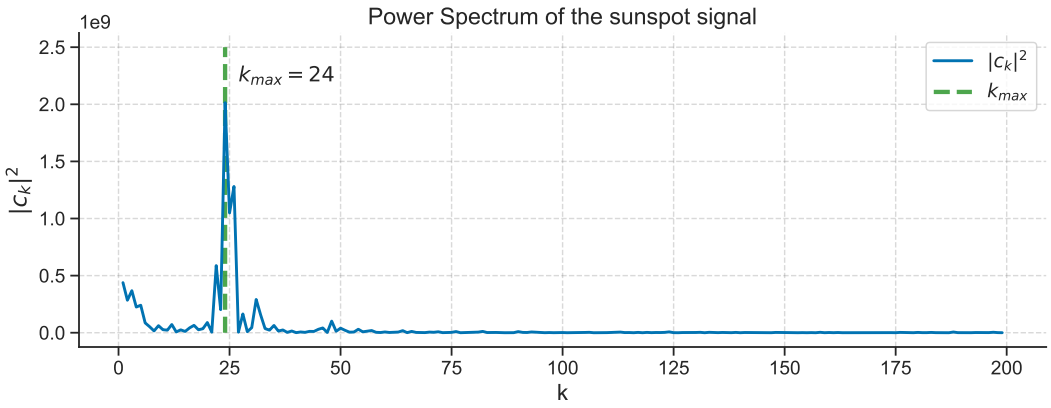

Figure 2: Power spectrum of the function of sun spots over time. One can locate the wavenumber of the predominant oscillatory mode of the signal at $k_{max} = 24$.

2 Problem 2

a) **Load and plot the image.**

This is done using matplotlib.pyplot.imshow and shown on the top left in Figure 3:

b) **Implement the point spread function.**

The point spread function can be found in the jupyter notebook. The plot of it can be found on the top right in Figure 3.

c) **Write a program that performs the image deconvolution using 2D FFTs and inverse FFTs.**

The unblurred image can be seen on the bottom in Figure 3.

d) **Why can't we perfectly unblur images?**

Dividing by zero will, of course, make the division of the FFT of the blurred image by the FFT of the point spread be undefined (and anything very very close to zero numerically unstable). In practice we do not have the a priori knowledge of the exact point spread function. Here we were given one to use that works decently well. It's true that there is hidden information in the structure of the blurring, but we can't access all that information unless we know precisely how the image is blurred. There's usually noise in the recording process of the image that can't be exactly emulated.

3 Extra Problem 3 (7.7) !

I accidentally did problem 7.7 as well, before realizing I was supposed to do 7.9. It's in the Jupyter notebook if you care to see it.

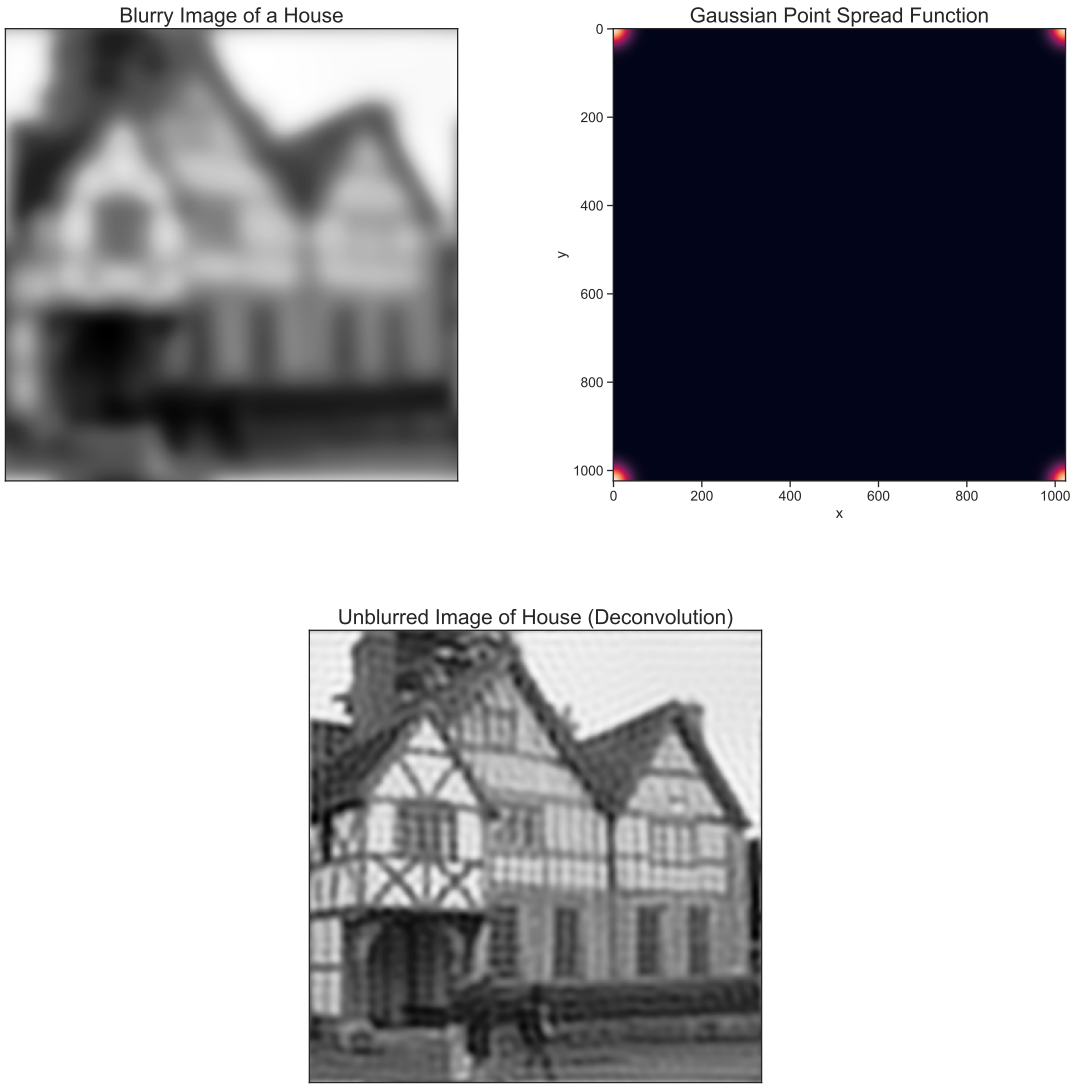

Figure 3: Parts and results of the deconvolution. Top Left: The original blurry image under a grayscale color map. Top Right: The Gaussian point spread function used in the deconvolution. Bottom: The unblurred image, derived via the inverse FFT of the division of the FFT of the blurry image by the FFT of the point spread function.