# Lab assignment #5: Solution

Nico Grisouard (Instructor), Ahmed Rayyan (TA & Marker)
& Mohamed Shaaban (TA)

Due Friday, 16 October 2020, 5 pm

---

1. **Basic applications of the Fourier transform.**

   (a) Newman 7.2: Detecting periodicity.

   The code (not required) is in `Lab05-407-2018-Q1a.py`.

Part (a) See fig. 1: I count 24 cycles (visual inspection) in 3142 months (last value of first column). That's one cycle every 130.9 months, or 11.4 years.

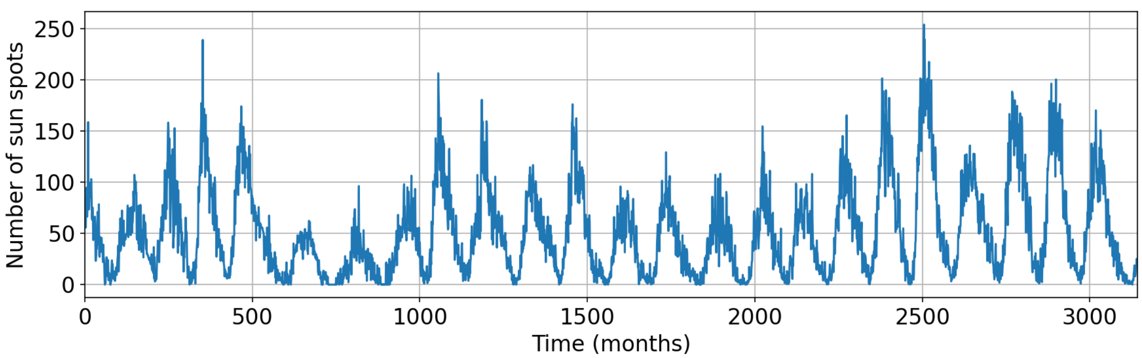


Figure 1: Number sun spots over the months.

- See fig. 1.

Part (b) We can clearly see a peak at $k = 24$. Coincidence? I don't think so. If the formula for a DFT coefficient is

$$c_k = \sum_{n=0}^{N-1} y_n \exp\left(-i\frac{2\pi nk}{N}\right), \quad (1)$$

then $n/N$ is a discretized form of $t/T$, with $t$ the time coordinate and $T$ the total duration of the record. Therefore, the exponential function is

$$\exp\left(-i\frac{2\pi t}{T_k}\right), \quad (2)$$

with $T_k = T/k$, which is also the period for this specific $k$ coefficient.

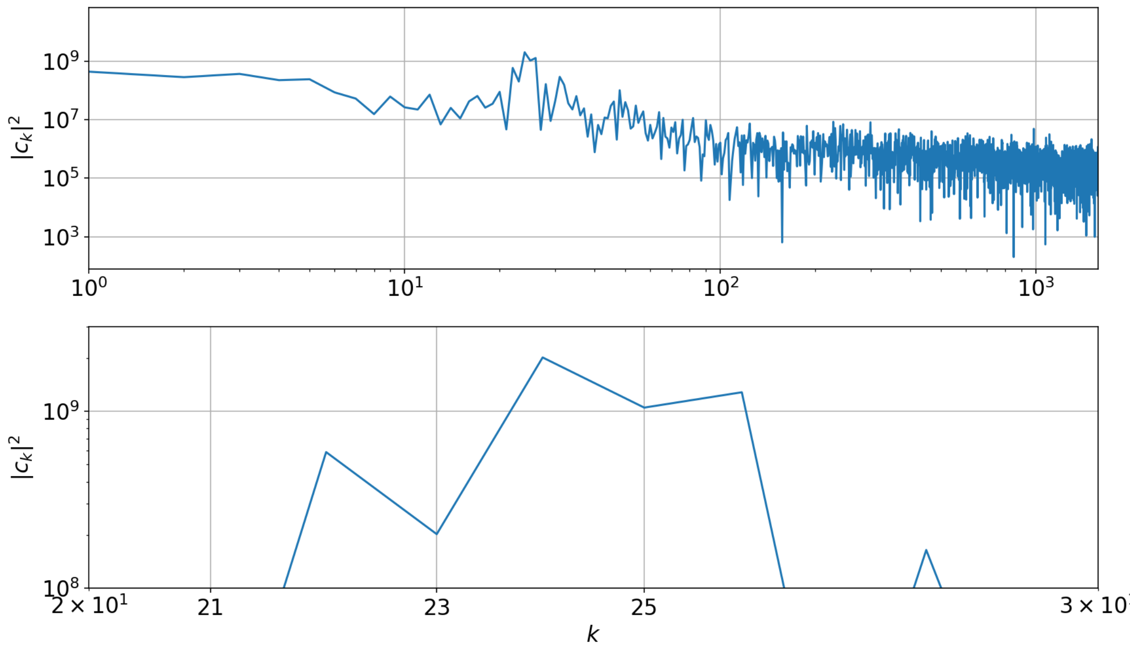


Figure 2: Sun spots cycle: the Fourier transform. Bottom panel is just a close-up around the peak of the top panel.

Therefore, the period is again $(T = 3142 \text{ months})/(k = 24) = 130.9$ months.

(b) Newman 7.4: Fourier filtering and smoothing.

The code is in `Lab05-407-2018-Q1b.py`.

Part (a) See fig. 3, blue curve.

Part (d) See fig. 3, orange curve, and in particular the close-up on the bottom panel. The effect is that of a low-pass filtering: by removing the high-frequency modes, we retrieve a smoother, less "jittery" (though noise) time series. Note that this way of filtering is a bit too abrupt, and may lead to Gibbs oscillations (see Q1c). Other, more clever methods, exist (see e.g. https://en.wikipedia.org/wiki/Filter_(signal_processing)).

Part (e) See fig. 3, green curve. The result is an even stricter low-pass-filtered version of the raw data.

(c) Newman 7.6: Comparison of the DFT and DCT.

The code (not required) is in `Lab05-407-2018-Q1c.py`.

Part (a) See fig. 4. The additional artifact is that at the beginning and end of the time series, the filtered data sharply tends towards a value that is the same as the value on the opposite end of the time series. This is because the DFT/FFT enforces periodicity of the data. The filtered curve actually oscillates a little at each end, although it is

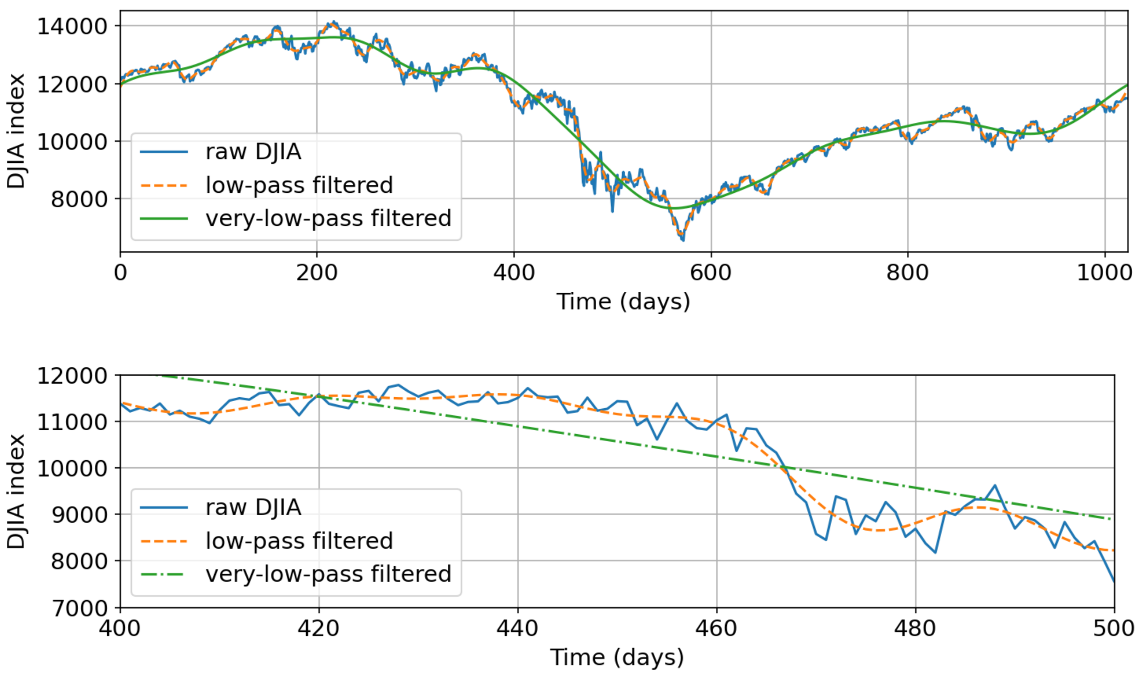


Figure 3: Top: Dow Jones Industrial Average (DJIA) from 2006 to 2010, before and after applying various filters. Bottom: Close-up of the index between days 400 and 500, to see the effect of low-pass filtering.

mostly visible that the day=0 end of the time series with a visible overshoot.

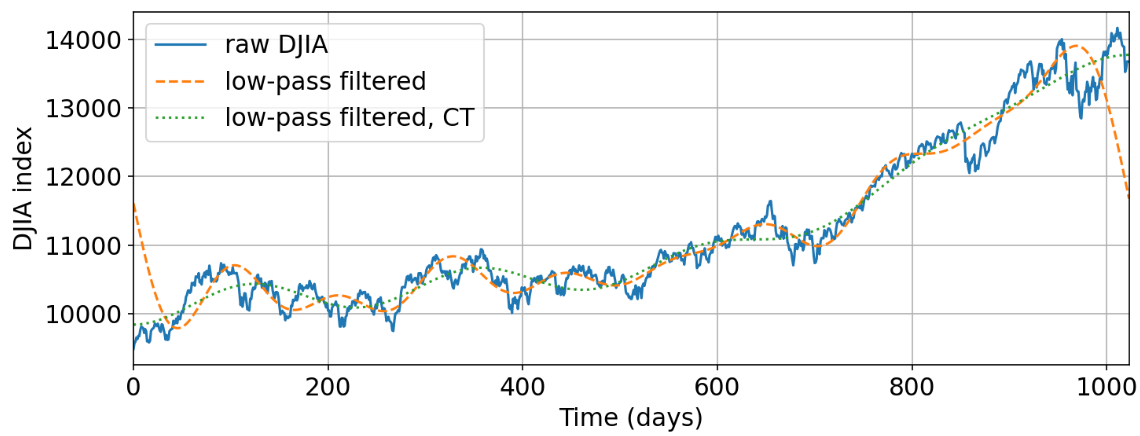


Figure 4: Dow Jones Industrial Average (DJIA) from 2004 to 2008, before and after applying various filters. The blue curve is the daily data, the orange, dashed curve ("low-pass filtered") is the data, after applying a DFT/FFT, followed by a low-pass filter, and an inverse DFT. The green, dotted curve ("low-pass filtered, CT") is the data, after applying a DCT, followed by a low-pass filter, and an inverse DCT.

Part (b) See fig. 4 again. What I meant when I said that the DCT filter was not as good a fit is very visible in the intervals 250–500, 800–900, 950–end. In these intervals, the index goes through a series of relatively quick and ample fluctuations that are more or less captured by the cropped DFT, but not at all by the cropped DCT.

(d) Newman 7.3: musical instruments.

The code (not required) is in `Lab05-407-2018-Q1d.py`.

Part (a) See fig. 5. For the same note, the piano is a lot more monochromatic than the trumpet. This is because fundamentally, the piano is a simple instrument: just a bunch of strings that get hit. Here, a string got hit and resonated at one frequency.

In comparison, the trumpet has a lot of overtones, because in essence, it relies on a lot of tricks (mouthpiece effect, bell efffect, pedal tone. . . ) to create one note.

Part (b) The frequency is about 523 Hz on the piano, twice that of $C_4$ (middle C). The note played is therefore $C_5$ (tenor C). This frequency is not the dominant one in the spectrum of the trumpet, which is normal.

2. **Newman 7.9: Image deconvolution**

The code (not required) is in `Lab05-407-2018-Q2.py`.

Part (a) See fig. 6 (left) for the blurred picture.

Part (b) See fig. 7 for a plot of the point spread function.

Part (c) See fig. 6 (right) for the unblurred picture.

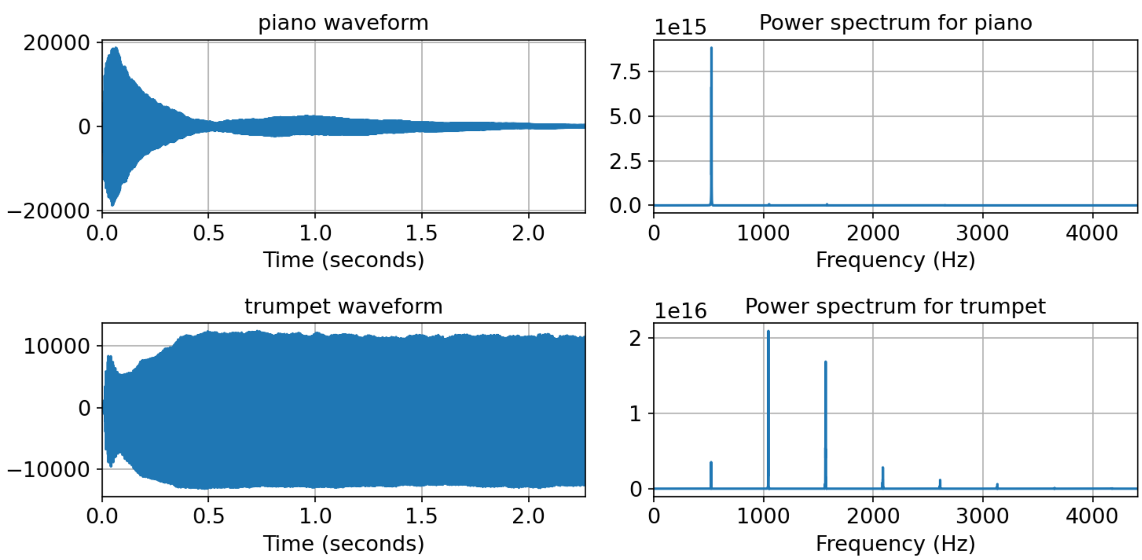


Figure 5: Waveforms (left) and Fourier transforms (right) for a piano (top) and a trumpet (bottom).

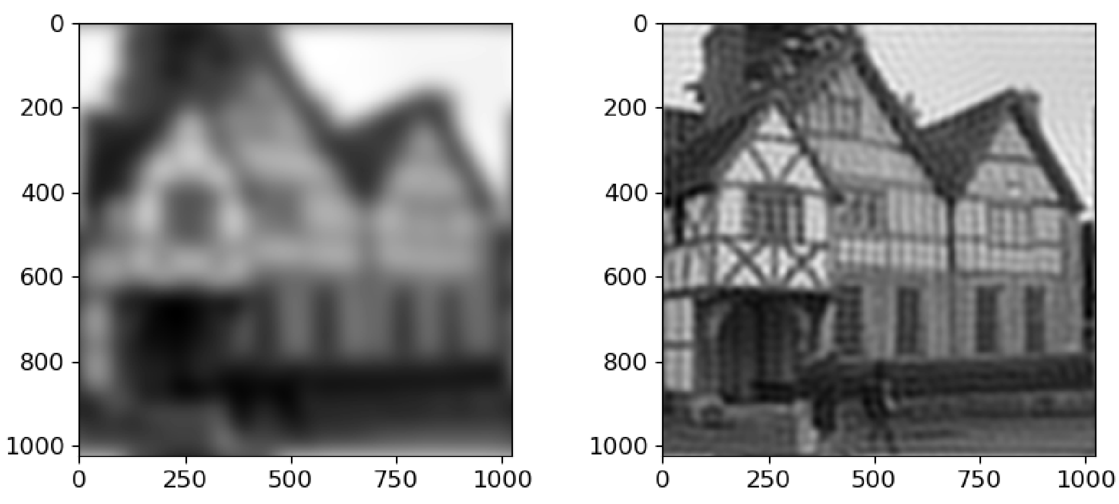


Figure 6: Blurred (left) and unblurred (right) picture.

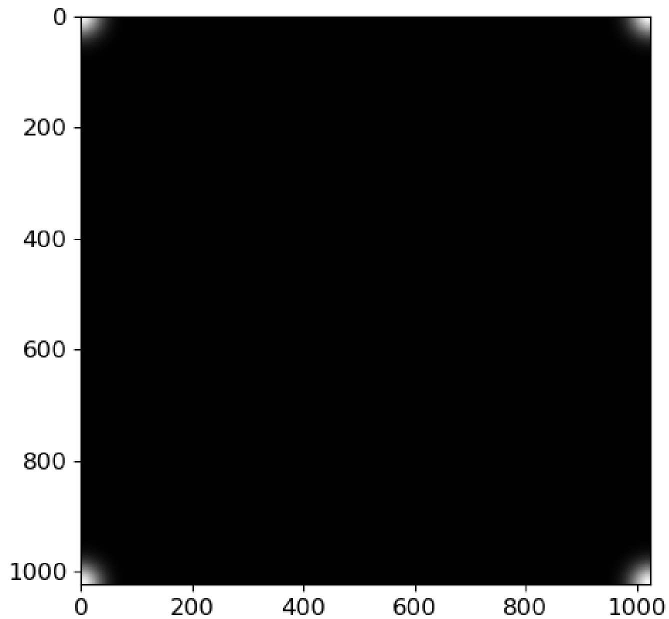


Figure 7: Point spread function. It looks different than Newman's because he saturated his colour scale.