Foreword: Scripts are supposed to be written in python and previously written py-scripts can be used as starting point. You can use as well jupyter notebook.

Is moving average low-pass filter?

1. Create the time axis from 0.0 s to 5.0 s, such as

   ```
   t0=0.0 ; t1=5.0 ; nt=2001
   t = np.linspace(t0, t1, nt)
   ```

2. Create an array called `s` which corresponds to the signal

   $$s(t) = \sin(4\pi t) + 1.5\sin(10\pi t) + \sin(23\pi t + 0.4) + \sin(45\pi t + 0.1).$$

3. Implement a Fourier transform of this signal (see filter practical). Arrange two subplots in order to have the time serie on the top row and the spectrum on the bottom row (see Fig. 1)

4. Declare a function to realize a moving average (this one involves the convolution product),

   ```
   def moving_average(x, w):
       return np.convolve(x, np.ones(w), 'valid') / w
   ```

   and test it with $n = 20$,

   ```
   nm=20
   smavg=moving_average(s, nm)
   ```

5. In order to allow a comparison of Fourier transforms between the raw and the 'smoothed' signals, the array `smavg` has to be extended to match the size of `s`. One possibility is to fill the beginning and the end of `smavg` with constant values.

   ```
   aa=np.zeros(int(nm/2))+1.0
   smavg0=smavg[0] ; smavgn=smavg[len(smavg)-1]
   aa=aa*smavg0 ; aaa=np.concatenate((aa, smavg), axis=None)
   nn=len(s)-len(aaa) ; aa=np.zeros(nn)+1.0
   aa=aa*smavgn ; smavg=np.concatenate((aaa, aa), axis=None)
   ```

6. Add the computation of the spectrum for `smavg`.

7. Test different values for `nm` (*e. g.* 8, 16, 24, 50, 80). What can you conclude about the 'smoothed' signal? Is the raw signal correctly filtered by this moving average operation?

8. Implement a low-pass filter in order to compare signals.

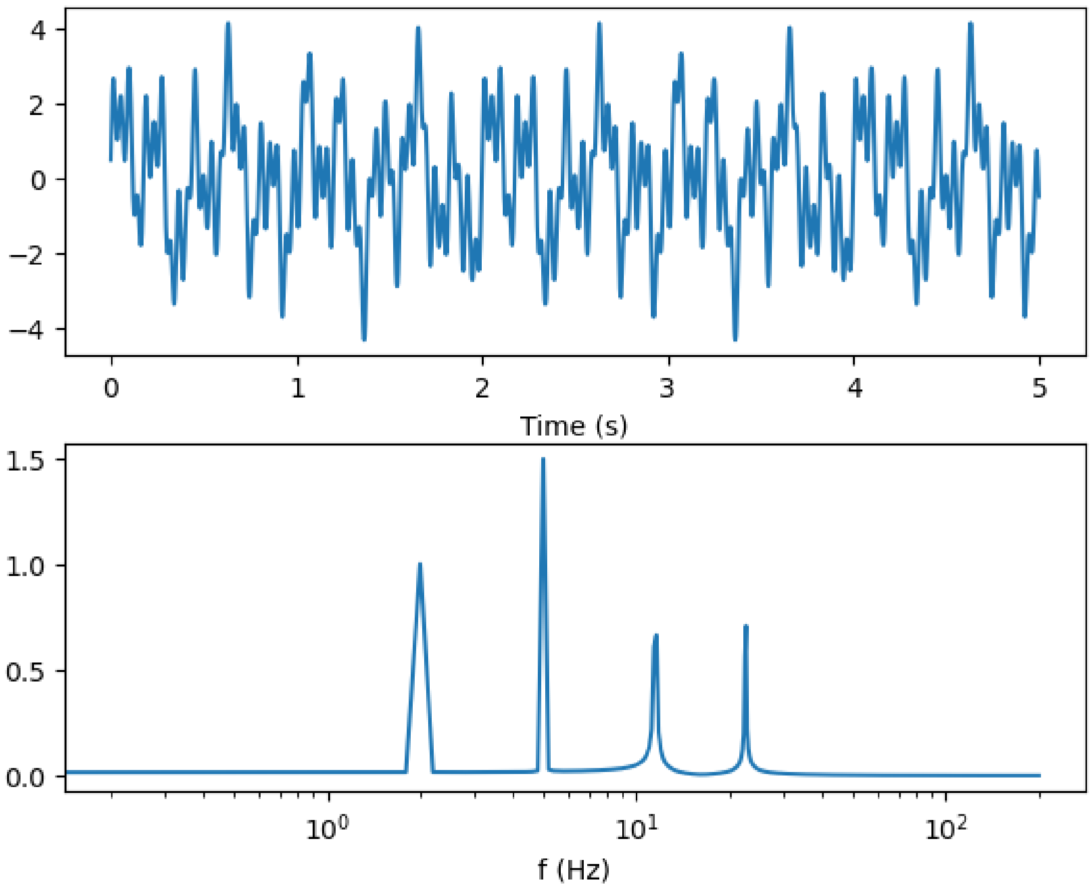

Figure 1: Signal composed by four pure sine components (top). The corresponding spectrum is represented at the bottom.