Some background info on the programs `FastEQCM.py` and `FastQCM.py`

The code uses the multifrequency lockin amplifier (MLA) from Intermodulation Products to interrogate the resonance of QCM. Some general remarks are collected here. Further remarks are contained as comments in the code.

- This text assumes that the reader knows the QCM, basically.
- This text assumes that the reader already installed the IMP MLA software supplied by Intermodulation Products
- The key feature of the MLA is the interrogation of resonances in one shot with a “comb”.
- The company marketing the MLA (Intermodulation Products SE, Stockholm) had the AFM in mind as the instrument of application. We are not concerned with the AFM, but the analogies between the resonances of an AFM-cantilever and the resonances of the QCM are far-reaching.
- The inventors of the MLA had a certain type of nonlinearities in mind (the “intermodulation”). We have worked on nonlinearities a bit, in the past. However, nonlinearities are outside the scope, here.

The two codes have a few subroutines in common. Differences are:

- The program `FastEQCM.py` runs an electrochemical modulation QCM. There is a trigger, needed for synchronization with the potentiostat. The code accumulates and averages over the modulation cycles (many of them). The different overtones are interrogated one after the other.
- The code `FastQCM.py` simply runs a fast QCM. It interrogates a few overtones in parallel. This code might be applied to droplet drying. There is no modulation, there is no accumulation.

A few more remarks

- For reasons, that we do not understand, thoroughly, the MLA has lower noise than the other vector network analyzers we know. We calculate a drift-corrected noise from Hadamard variance as $\delta f_{rms} = <(f_{i+1} - 2f_i + f_{i+1})/6>^{1/2}$. Of course, the noise depends on the integration time. If we run an experiment with no modulation (sacrificing time resolution) and average for 1 s, we find an rms-noise of a few mHz. With other network analyzers, the noise is closer to a few tens of mHz. The noise goes down if the output is set to 12 V (rather than 1.95 V, which is the default setting). That is an option for measurements in the liquid phase because the resonators then are damped and the resistance is correspondingly high (→ low currents). When running the resonators in air, there is a danger of overload on the input channel (or even damage to the input channel).
- “Tuning” is needed. Tuning (done by the MLA) slightly shifts all frequencies, such that any frequency is an integer fraction of the data acquisition rate. This avoids what is called “Fourier leakage” in the manual.
- This version of the code does the fitting on the same core as data acquisition. More than half of the total time is spent on fitting (rather than data acquisition). We have a version, which solves that problem with multithreading (send us a mail).
- The best possible time resolution is the inverse distance between the different frequencies of a comb. If the time spent on fitting is a consideration, one may compromise on time and set the parameter `n_averages` to some integer larger than one. The software then widens the comb, but still evaluates the values at the rate prescribed by the parameter `Time_Resolution`. This amounts to an integration on the board of the MLA.
- We do not bother about calibration. We simply fit a phase-shifted Lorentzian to the admittance traces and assume that less-than-perfect calibration leaves the *shifts* resonance frequency and half bandwidth unchanged.