

## PyQTM: A Software Package for the Analysis of QCM-D Data

1	Preface .....	1
2	The Core and Some Extensions of PyQTM.....	2
3	Background.....	2
3.1	<i>Viscoelastic layer systems</i> .....	2
3.2	<i>Half bandwidth or dissipation factor</i> .....	2
3.3	<i>Roughness</i> .....	3
3.4	<i>Perturbation analysis</i> .....	3
3.5	<i>Strengths and limitations</i> .....	3
4	Tour Through PyQTM.....	4
4.1	<i>Versions</i> .....	4
4.2	<i>Input and output</i> .....	5
4.3	<i>Reference state, bounds for fitting</i> .....	5
4.4	<i>Fitting</i> .....	6
4.5	<i>Confidence Limits</i> .....	7
4.5.1	Bootstrapping.....	7
4.5.2	Added random noise .....	7
4.5.3	$\chi^2$ Landscape.....	8
4.5.4	Confidence Limits Calculated by lmfit.....	8
4.6	<i>Fuzzy Interface to the Bulk</i> .....	8
4.7	<i>Choice of parameters quantifying viscoelasticity</i> .....	9
4.8	<i>Thin films with moderate softness</i> .....	11
4.9	<i>Thick, soft adsorbates</i> .....	13
4.10	<i>Uniqueness of fit results if a thin sample looks Sauerbrey-like</i> .....	13
4.11	<i>Fitting in the <math>\{J', J''\}</math> representation, leaving <math>\beta</math> "fixed"</i> .....	14
4.12	<i>Remarks to details</i> .....	15
5	Underlying Equations .....	16
5.1	<i>Multilayer formalism</i> .....	16
5.2	<i>Roughness</i> .....	18
5.3	<i>Perturbation analysis</i> .....	19
6	Glossary .....	19

### 1 Preface

PyQTM implements various sets of equations for the analysis of shifts of frequency and bandwidth ( $\Delta f(n)$  and  $\Delta \Gamma(n)$  with  $n$  the overtone order), acquired with a quartz crystal microbalance with dissipation monitoring (QCM-D). Mostly, PyQTM derives the thickness and the softness of the layer under study. A QCM-D is any instrument, which acquires frequency and bandwidth on a number of different overtones. The shift in half bandwidth,  $\Delta \Gamma$ , carries information largely equivalent to the information contained in the shift of the dissipation factor,  $\Delta D$ .

For an introduction to the science behind the QCM see the Ref. 1. This manual is short on the algebraic details. Users can always look up details in the source code.

As you will notice (or have noticed earlier), uniqueness of the fit results can be problematic. The software cannot change that. We have seen numerous cases, where we ended up not formulating conclusions. (We have seen other cases, as well.)

Numerous people have contributed to PyQTM. Specific mentions go to Ilya Reviakine, Arne Langhoff, Philipp Sievers, Judith Petri, and Christian Leppin.

Diethelm Johannsmann

## 2 The Core and Some Extensions of PyQTM

Most of the time, PyQTM fits the data with the equation

$$\frac{\Delta f + i\Delta\Gamma}{f_0} = \frac{-\tilde{Z}_f \tilde{Z}_f \tan(\tilde{k}_f d_f) - i\tilde{Z}_{liq}}{\pi Z_q \tilde{Z}_f + i\tilde{Z}_{liq} \tan(\tilde{k}_f d_f)} - \frac{i\tilde{Z}_{liq}}{\pi Z_q} \quad \text{Eq. 1}$$

The variables are explained in section 6. This one-liner is fully equivalent to the Voigt-Kelvin formalism.<sup>2,3</sup> To the best of our knowledge, it was first written down (in slightly different form) by the Tel Aviv group in the appendix of Ref. 4.

Most of you have fitted data with a fit function before. Why is this manual ~20 pages long? PyQTM goes beyond Eq. 1 in a few regards:

- $Z$  and  $k$  in Eq. 1 depend on the viscoelastic parameters. Firstly, these depend on frequency and secondly, there are different choices for the viscoelastic parameters. When trying to understand what a fit means, suitable choices for the parameters depend on the question to be addressed.
- You (and PyQTM) need to keep an eye on the baseline and the reference state, to which this baseline corresponds.
- We believe to owe you the mathematics, which covers multilayers, as well (hence the term “multilayer formalism”). Also, we believe to owe you the mathematics covering viscoelastic profiles (section 4.6).
- In some rare cases, the multilayer formalism reports a negative shear modulus, which is unphysical. When that happens, the perturbation formalism must be used (section 5.3, also see the file `Solve_Generalized_Lu_Lewis.py`, the file `Equations_for_Perturbation_Analysis.pdf`, and Box 2 in Ref. 1). This problem mostly concerns polymer layers in the dry. The Shull group has extensively worked on the analysis of such samples.<sup>5</sup>

Einstein is quoted with<sup>6</sup>: “Everything should be made as simple as possible, but not simpler”. The bullet points above address “but not simpler”.

## 3 Background

### 3.1 Viscoelastic layer systems

There is an established formalism to calculate the periodic stress at the resonator surface caused by planar layer systems.<sup>2,7,8,9</sup> PyQTM calls this algorithm the “multilayer formalism” (section 5.1). The multilayer formalism calculates the periodic stress at the resonator surface and derives the complex frequency shift from the relation  $\tilde{\Delta f}/f_0 = i/(\pi Z_q) \tilde{Z}_L$ , where  $\tilde{Z}_L$  (the ratio of stress to velocity) is the load impedance. The latter relation is the small-load approximation. See the glossary in section 6 for the meaning of the other variables. The tilde denotes a complex number. The complex frequency shift is given as  $\tilde{\Delta f} = \Delta f + i\Delta\Gamma$ . QTM mostly displays overtone-normalized complex frequency shifts,  $\tilde{\Delta f}/n = \Delta f/n + i\Delta\Gamma/n = \Delta f/n + if_{res}\Delta D/2$

### 3.2 Half bandwidth or dissipation factor

Internally, PyQTM quantifies dissipative processes by the half bandwidth,  $\Gamma$ .  $\Gamma$  is related to the “dissipation factor” by the relation  $D = Q^{-1} = 2\Gamma/f_{res}$ . The overtone-normalized complex frequency shift may be written as  $\tilde{\Delta f}/n = \Delta f/n + i\Delta\Gamma/n = \Delta f/n + if_{res}\Delta D/2$ . PyQTM offers a choice between the use of  $\Delta\Gamma/n$  or  $\Delta D$ .

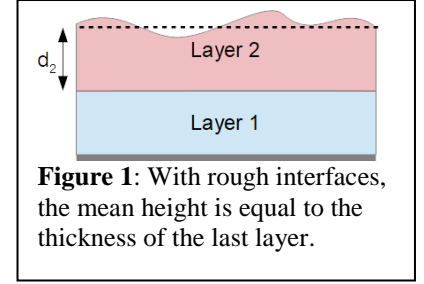
### 3.3 Roughness

PyQTM implements shallow, small-scale roughness following Ref. 10 and Eq. 25. More technically, the shear-wave impedance of the liquid can be replaced by what is called  $Z_{liq,eff}$  in Eq. 25. The results obtained with these equations must be treated with some care because of the inherent assumptions.

Because the model applies to shallow roughness, the parameter “Aspect Ratio” should not be larger than unity. The aspect ratio is the ratio of the vertical scale of roughness to the lateral scale of roughness.

The roughness is always assumed to be the roughness at the interface with the bulk liquid (Figure 1).

This option is activated with a check button at the upper right of the parameters frame.



**Figure 1:** With rough interfaces, the mean height is equal to the thickness of the last layer.

### 3.4 Perturbation analysis

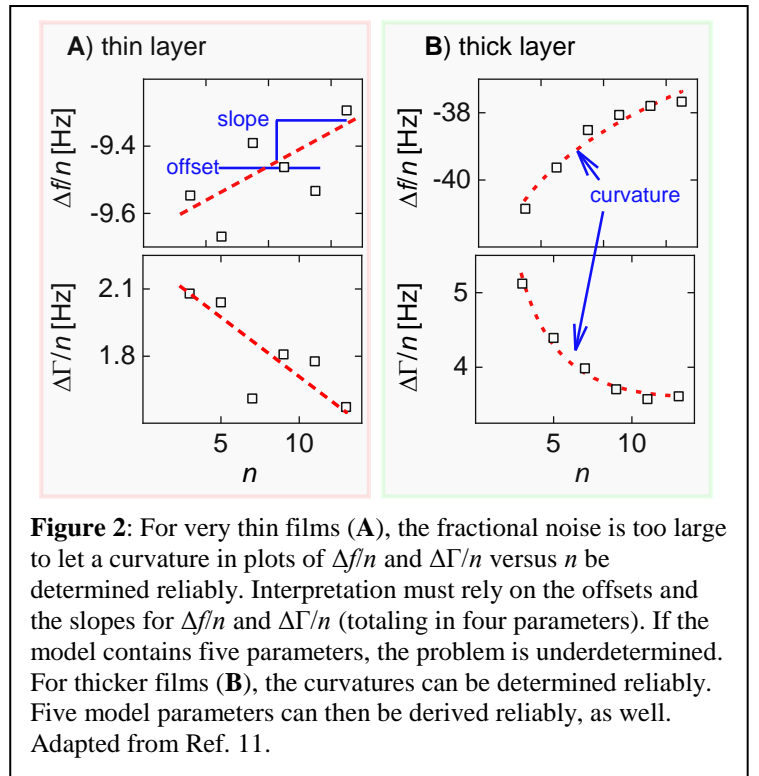
Some short-comings inherent to the small-load approximation are avoided by the “perturbation analysis” (Box 2 in Ref. 1). However, the perturbation analysis only covers thin films. It fails at the film resonance. In cases, where the multilayer formalism predicts a film resonance, the perturbation analysis is not applicable. Whether the perturbation analysis can be trusted can also be inferred from a comparison between the 3<sup>rd</sup>-order result and the 5<sup>th</sup>-order result. If the two agree, one is safe.

The perturbation analysis is needed for stiff films in air. For layer systems in liquids, the differences between the results obtained with the multilayer formalism and with the perturbation analysis are small. For stiff films in air, the shear-wave impedance of the electrode material<sup>1</sup> must be known in order to derive the film’s shear modulus.

In order to capture the film resonance and still avoid the limitations inherent to the small-load approximation, one may numerically solve the generalized Lu-Lewis equation (Eq. 40 in Ref. 1). Python code doing this is provided in the file “Solve\_Generalized\_Lu\_Lewis.py”.

### 3.5 Strengths and limitations

- a) Robust results are obtained for thin films, if the curvature in plots of  $\Delta f/n$  and  $\Delta\Gamma/n$  versus  $n$  can be determined with confidence (Figure 2).<sup>2</sup>
- b) PyQTM can model thick, soft films, but the derived fit results often are not unique.



**Figure 2:** For very thin films (A), the fractional noise is too large to let a curvature in plots of  $\Delta f/n$  and  $\Delta\Gamma/n$  versus  $n$  be determined reliably. Interpretation must rely on the offsets and the slopes for  $\Delta f/n$  and  $\Delta\Gamma/n$  (totaling in four parameters). If the model contains five parameters, the problem is underdetermined. For thicker films (B), the curvatures can be determined reliably. Five model parameters can then be derived reliably, as well. Adapted from Ref. 11.

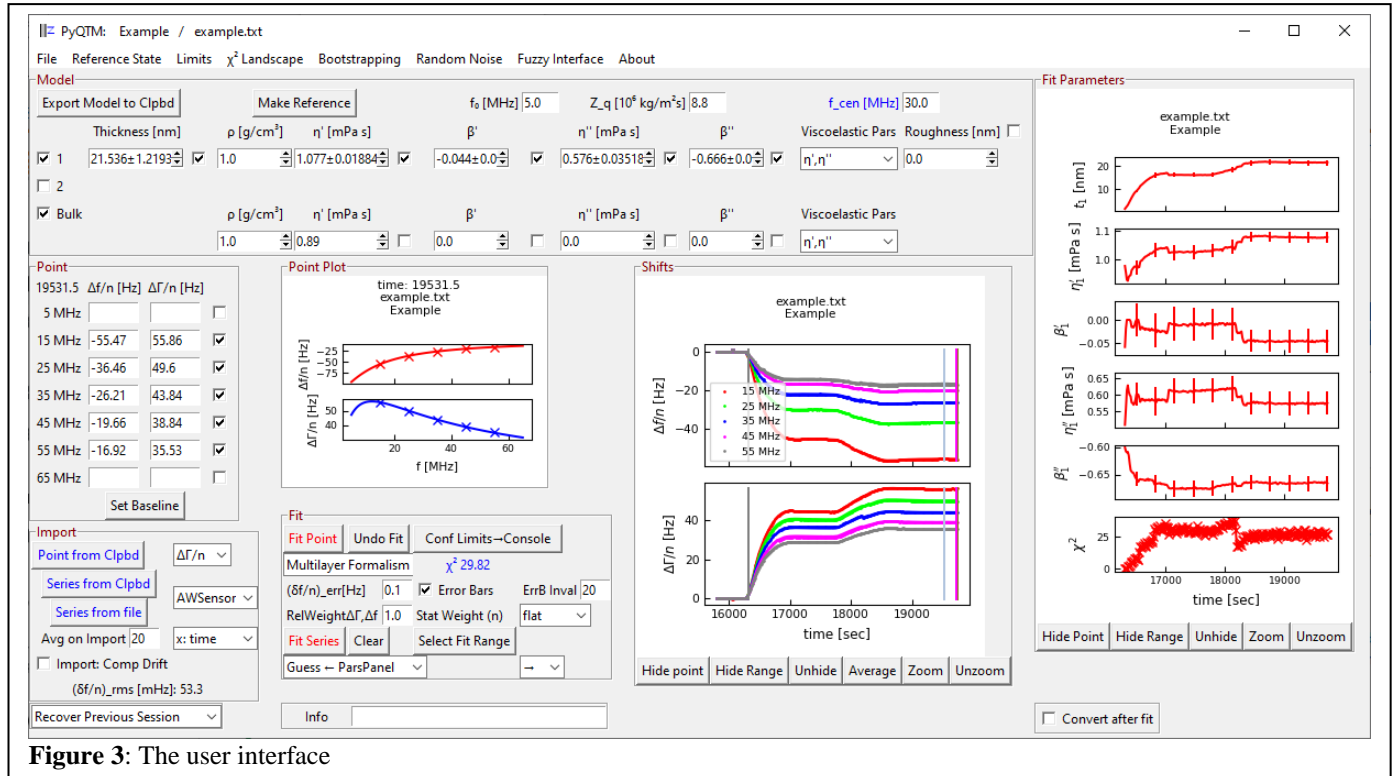
<sup>1</sup> Gold:  $Z_{Au} \approx 23.9 \cdot 10^6 \text{ kg}/(\text{m}^2 \text{ s})$

<sup>2</sup> If viscoelasticity is described as proposed in section 4.7, the model has five free parameters (thickness plus four parameters for viscoelasticity). If the experimental data can be aggregated into only four robust parameters (two offsets, two slopes, see Figure 2A), the problem is underdetermined. A model of viscoelasticity with three free parameters would be needed. The Clausthal group has tried a number of such models (variants of the Maxwell model), but has concluded that these produce more confusion than insight.

Such a model would amount to some other way of calculating  $\tilde{J}$  in the routine Calc\_J\_SI in QTM\_Core.py

- c) Flexural motion of the plate is not covered. PyQTM is based on the parallel-plate model.
- d) Piezoelectric stiffening is not covered.
- e) Structured samples are not covered, basically. Small-scale, shallow, random roughness is an exception.
- f) PyQTM allows for two discrete layers, at most. Viscoelastic profiles are not covered.

In principle, viscoelastic profiles (for instance produced by a polymer brush) might be covered by the multilayer formalism, extended to many layers with small thickness, each. However, there is a rather easy way to predict  $\Delta f$  and  $\Delta \Gamma$  for such situations, solving the underlying differential equation. Sample code is contained in



**Figure 3:** The user interface

“VE\_Profile\_Solve\_ODE.py”. This Python program solves the wave equation for continuous profiles  $\{G'(z), G''(z), \rho(z)\}$  and derives the shifts of frequency and bandwidth from the solution (section 4.6.3 in Ref. 1). The profiles leading to agreement with experiment may or may not be unique (see b)).

## 4 Tour Through PyQTM

Figure 3 shows the user interface. It was meant to be self-explaining.

### 4.1 Versions

PyQTM can be run from Python.<sup>3</sup> The Clausthal group uses Spyder. The IDLE environment appears to work as well. You see the code and can make changes.<sup>4</sup> PyQTM.py is the main file. PyQTM.py imports code from the other files.

An older version (written in Delphi) is called QTM.

<sup>3</sup> The numba package is required. Alternatively, all lines containing `@jit(nopython = True)` can be commented away, but that will let PyQTM run much slower.

The lmfit package is required. Installation of lmfit can be difficult for older versions of Python. At Clausthal, we needed to install the newest version of Anaconda (and then everything went smoothly).

<sup>4</sup> In particular, users have adapted the output formats to their needs. This may be achieved by modifying the routine `FitPars_to_ClPbd()` in PyQTM.py

## 4.2 Input and output

On start-up, PyQTM reads certain general status information from the file `PyQTM.ini`. While you work, all information is saved into temporary files in a subfolder to the working directory named `"/tmp"`. The file `"Master.qtm"` is organized like a config-file and contains the most relevant information. Various other files (with extension `.qt2`) contain supplementary information. These are ASCII files. They contain NaNs (NaN: "not a number"). "Save as ..." asks the user to select a folder and suggest a prefix to the existing filenames, which was meant to be descriptive (can be changed). It then copies all files from the `tmp`-folder into this folder (with the names changed). When the user opens a file, PyQTM reads in the information contained in these files. *The master file to be opened must contain the string `"_Master.qtm"` in the filename.*

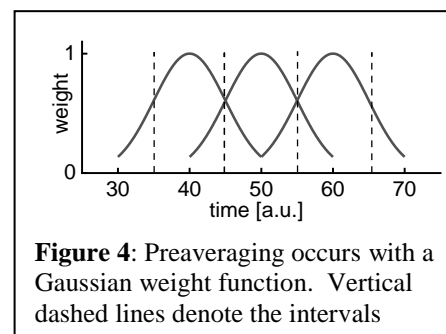
The submenu item under File named "Export Fit Params and + Simulation Data" exports results in a format, which is more suitable for input to Excel than the files with extension `.qt2`. PyQTM never imports those data.

There is a choice between "Start: Default" or "Start: Previous". In case "Start: Default" is selected, PyQTM starts from the file `"Default.qtm"`. Users can edit these settings. For instance, they might set the default value of `"tt_io_format"` to their preferred instrument.

Experimental data are imported from "Import" frame. The files must be text files. Examples for the formats as implemented by the different instruments (QSoft, QSoft\_new, AWSensors, openQCM, QCM-I, 3TAnalytik) are contained in the folder distributed with the code. These examples were kindly provided by Annemarie Maan, Ralf Richter, Ilya Reviakine, Martin Dienwiebel, Osheen Joseph, and Adam Olsson. Once in a while, the companies change their output formats (a repeating source of delight). You then may either figure it out yourself (and adapt the source code in `QTM_TT_IO.py`) or send us a mail.

If the option "Import: Comp Drift" is checked, PyQTM applies a small correction to the data, which is meant to account for the fact that the overtones are interrogated sequentially. If the sample's properties change quickly, an apparent dependence on overtone order may in fact be caused by a dependence on time. PyQTM calculates the time derivative of the data on each overtone and uses this information to correct the data from the different overtones for this artifact. The correction is negligible in most cases.

The statistical noise can always be lowered with preaveraging during import (Figure 4). Preaveraging also lowers the number of data points, which speeds up the fits on the entire time series. Statistical noise does not usually dominate the standard error of the fit parameters. The different overtones often slightly deviate from the expectations, even for Sauerbrey films or Newtonian liquids. Presumably, these irregularities go back to poorly controlled effects of compressional waves, the latter being caused by small admixtures of flexural motion to the thickness-shear deformation. These vary between crystals, but not over time during one experiment. Usually, these deviations amount to more than the statistical noise.



Data for one single point can be edited in the frame "Point". They can also be imported from the Clipboard with the button "Point From Clpbd". The folder, which contains the Python files, should also contain an Excel file named `Example_Input_from_Clipboard.xlsx`, which shows the required format.

Consider updating  $f_{\text{cen}}$  (see Eq. 5) after import. (It may be preferable to always fit with the same  $f_{\text{cen}}$  for better comparison between experiments.)

## 4.3 Reference state, bounds for fitting

- Shifts of frequency and bandwidth are always understood as shifts with respect to some reference state. The reference state is edited in the Reference Form. The reference state corresponds to the Zero on the scales of  $\Delta f/n$  and  $\Delta \Gamma/n$  (that is: corresponds to the baseline). Change this offset by clicking "→ Set Baseline". (I can make

sense to average data over a certain range before setting the baseline.) PyQTM then uses the current values of  $\Delta f/n$  and  $\Delta\Gamma/n$  as the new baseline.

**Do not confuse the baseline with the reference state.** The baseline is a set of values for  $\Delta f/n$  and  $\Delta\Gamma/n$ . The reference state is a set of system parameters. The two need to correspond to each other. The reference state can be edited in the window “Reference State”. It can also be changed with the button “Make Reference State”.

- The resonance frequencies of a crystal usually change by a few Hz, when the user dismounts them from the holder and mounts them again. (In-between the user might apply a film by spin-casting). These changes occur because of static stress. They can be lowered to some extent by mounting crystals with as little stress as possible.
- There are parameters unrelated to the sample. These are the frequency of the fundamental,  $f_0$ , the shear-wave impedance of AT-cut quartz,  $Z_q$ , and the parameter  $f_{cen}$  from Eq. 5. The parameter  $f_{cen}$  ideally is in the center of the frequency range analyzed. If data from the overtones at 15, 25, 35, and 45 MHz are analyzed,  $f_{cen}$  should be 30 MHz. One might always adapt  $f_{cen}$  to the active harmonics or keep it fixed (for better comparison between experiments).
- The fits are constrained by bounds, which can be edited in the Limits Form. When the viscoelastic parameters are changed (for instance from  $J$  to  $G$ ), PyQTM automatically resets the limits to the default values. In case you have manually changed the limits, your changes will get lost when the viscoelastic parameters are changed.

The default limits are such that  $G$ ,  $J$ , and  $\eta$  can never be zero. For instance, the minimum density is  $10^{-9}$  g/cm<sup>3</sup>. With these bounds applied to the fits, the shear-wave impedance,  $Z$ , and the wave number,  $k$ , always remain finite (which avoids divisions by zero). It is advised to not allow zeros for the density and the viscoelastic parameters. When the fit routine attempts to set the density to zero, the program exits with an error message.

#### 4.4 Fitting

The fit is based on a  $\chi^2$ -minimization.  $\chi^2$  is defined as

$$\chi^2 = \frac{1}{2n_{ovt} - n_{fitpar}} \sum_n w_n \left[ \left( \frac{\Delta f_n / n - \Delta f_{n,fit} / n}{(\delta f / n)_{err}} \right)^2 + \left( \frac{\Delta \Gamma_n / n - \Delta \Gamma_{n,fit} / n}{(\delta \Gamma / n)_{err}} \right)^2 \right] \quad \text{Eq. 2}$$

$n_{ovt}$  is the number of overtones included in the analysis.  $n_{fitpar}$  is the number of fit parameters. The factor of 2 before  $n_{ovt}$  occurs because every overtone contributes two data points ( $\Delta f/n$  and  $\Delta\Gamma/n$ ).

The standard errors reported by PyQTM are those returned by the fit routine of the package “lmfit”. (Remember the possibility of correlated errors.)

$(\delta f/n)_{err}$  is the uncertainty in the measurements of  $\Delta f/n$  and  $\Delta\Gamma/n$ . The uncertainty is about the same on  $\Delta f/n$  and  $\Delta\Gamma/n$ , hence only one parameter. PyQTM’s default value is  $(\delta f/n)_{err} = 0.1$  Hz.  $(\delta f/n)_{err}$  only affects the value of  $\chi^2$ , not the set of fit parameters, which minimizes  $\chi^2$ . If  $(\delta f/n)_{err}$  was estimated correctly and if  $\chi^2$  is of order unity, the fit is as good as it can be.

PyQTM also allows to assign certain statistical weights,  $w_n$ , to the different overtones. The default is “flat”. One may also assign different weights to  $\Delta f/n$  and  $\Delta\Gamma/n$  (“Rel Weight  $\Delta\Gamma$ ,  $\Delta f$ ”). This makes sense when  $\Delta\Gamma$  is much smaller than  $\Delta f$ . If the weight of  $\Delta\Gamma$  is increased, the fit will give these small values more statistical weight than they would usually have.

When fitting entire time series, there is a choice to either start all fits from the same guess (the values in the parameters panel (“Guess ← ParsPanel”) or from the previous fit result (“Guess ← Prev Result”). The latter option makes the fits run faster, but carries the danger that one bad fit will let all subsequent fits fail because the guess values from thereon are poor. The fits can run from start to stop (“→”) or from stop to start (“←”).

A few more comments:

- A parameter is turned into an active fit parameter with the check box next to it.
- Entries into an edit field become active, after the user hits RETURN (only then, do not forget).
- The panel to the right-hand side in Figure 3 shows the fitted parameters with error bars. Only a few error bars are shown in order to avoid overcrowding of the figure. Increase or decrease the number of bars with the parameter in the field next to “ErrB Inval”. This parameter also controls how often the user interface is updated while PyQTM runs fits on all data in a time trace.
- The density  $\rho$  cannot be a fit parameter.  
The QCM cannot independently determine the thickness and the density of a layer. Only the product of the two (the mass per unit area) enters the equations. PyQTM nevertheless reports the thickness, because the density often is known with good accuracy and because many users are more familiar with the unit “nm” than with the unit “ $\mu\text{g}/\text{cm}^2$ ”. (With  $\rho = 1 \text{ g}/\text{cm}^3$ , a mass per unit area of  $1 \mu\text{g}/\text{cm}^2$  corresponds to a thickness of 10 nm.) Again, the derived thickness depends on the chosen density.
- The density of the bulk cannot be a fit parameter because the QCM only determines the viscosity-density product of the bulk.
- It is recommended to discard data from the fundamental. For poorly understood reasons, the frequency shifts measured on the fundamental often do not match the expectations. The data from the higher harmonics in these cases do form a consistent picture. Depending on the experimental conditions, data from the 3<sup>rd</sup> overtone can be problematic as well.

#### 4.5 Confidence Limits

Confidence limits are a bit of a thorny issue. It is difficult to conceive a model with less than five free parameters. Extracting five parameters independently from a set of about ten data points ( $\Delta f/n$  and  $\Delta \Gamma/n$  on five overtones) is bound to cause correlated errors (Figure 5).

The solutions provided by PyQTM are often not unique. The thickness, in particular, can be uncertain. There is a similar problem in optics, where the thickness of a layer sometimes cannot be inferred from reflectometry without independent knowledge of the refractive index.

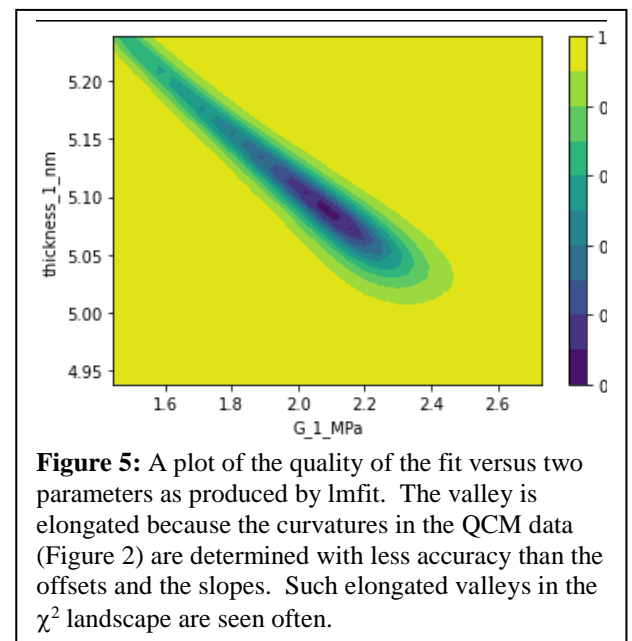
PyQTM offers three tools to study confidence limits:

##### 4.5.1 Bootstrapping

Bootstrapping provides for a quick estimate of the confidence range, not based on the covariance matrix. (The standard errors from the fit routine are based on the covariance matrix.) The bootstrapping algorithm resamples the experimental values, allowing for multiple draws of the same value. It then performs a fit on those sets and obtains a result slightly different from the result obtained from the original data set. The distribution of these results and the standard deviations of those distributions indicate the confidence range.

##### 4.5.2 Added random noise

Another quick estimate of the robustness of a fit is obtained by adding random noise to the data. PyQTM adds Gaussian noise with a width of the distribution as given in the field  $(\delta f/n)_{\text{err}}$ . Otherwise, the window is the same as the Bootstrapping window.



#### 4.5.3 $\chi^2$ Landscape

Inspecting the  $\chi^2$  landscape can be instructive. There are two modes:

- “Run One” varies the parameter chosen in the box next to the button in the range indicated in the entries “Min” and “Max”. PyQTM fits the remaining free parameters and plots these fitted parameters as well as  $\chi^2$  versus the varied parameter. One hopes to see a well-defined minimum in  $\chi^2$ . The other graphs show the cross correlations. For instance, the stiffness may be anticorrelated with the thickness.
- “Run All” essentially does the same as “Run One”, but does so for all free parameters and only plots  $\chi^2$ . One hopes to see well-defined minima in all plots. The range is specified in the field called “Width Factor”. The minimum and the maximum are the values from the best fit, divided by the Width Factor and multiplied by the Width Factor. An exception are the power law exponents. These run between their limits as set in the Limits Form.

#### 4.5.4 Confidence limits calculated by lmfit

PyQTM also calls routines from the library lmfit. It does so in two ways:

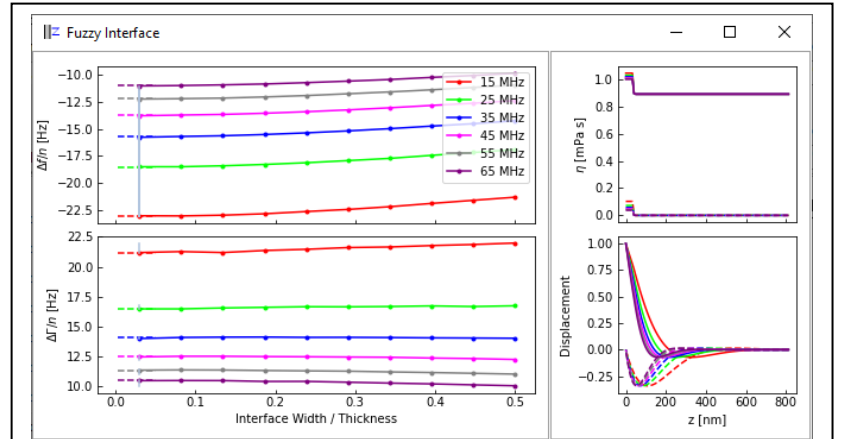
- The button “ConfLimits → Console” lets PyQTM throw all kinds of tables and diagrams to the Python Console. Please refer to the source code after “def ConfLims\_lmfit\_2\_Console():” in QTM\_Core.py. Documentation is contained in <https://lmfit.github.io/lmfit-py/confidence.html>.
- lmfit is rather powerful. It should be not so difficult to call other lmfit-routines from inside the routine ConfLims\_lmfit\_2\_Console() (adapting the source code). The keyword arguments of the existing function calls might be changed, as well.

#### 4.6 Fuzzy Interface to the Bulk

PyQTM assumes sharp interfaces. That was a choice made at the beginning. However, many samples of interest would be better characterized by profiles of the density and viscoelasticity, such as  $\tilde{G}(z)$  and  $\rho(z)$ . One such profile ( $\eta'$  and  $\eta''$  versus  $z$ ) is sketched in on the upper right in Figure 6.

Predicting values of  $\Delta f/n$  and  $\Delta \Gamma/n$  produced by profiles is not particularly difficult (section 4.6.3 in Ref. 1). The problem is the number of free parameters of such models. There would have to be at least one more parameter, which is the width of the interface.

What can still be done (and what is implemented in PyQTM), is to start from a sharp interface, gradually increase the width of the interface, and check whether increased interface width grossly changes  $\Delta f/n$  and  $\Delta \Gamma/n$ . PyQTM assumes hyperbolic tangents with some width for the profiles. This is what the window named “Fuzzy Interface” does. Again: This is only meant to be a check on how wide a smooth interface may be without invalidating the results obtained with a sharp interface.



**Figure 6:** An output from the window “Fuzzy Interface”. The panel on the upper right shows the viscoelastic profile, where the width of the interface corresponds to the blue vertical line in the panel to the left. The lower right shows the displacement pattern (dashed for the imaginary part both at the top and the bottom). The graph to the left shows to what degree  $\Delta f/n$  and  $\Delta \Gamma/n$  are changed when the interface is made wider. The perfectly sharp interface corresponds to an interface width of zero. The results calculated assuming a sharp interface are shown as dashed horizontal lines on the left. In the case shown here, a fractional interface width of 10% yields values similar to what is obtained with a sharp interface.

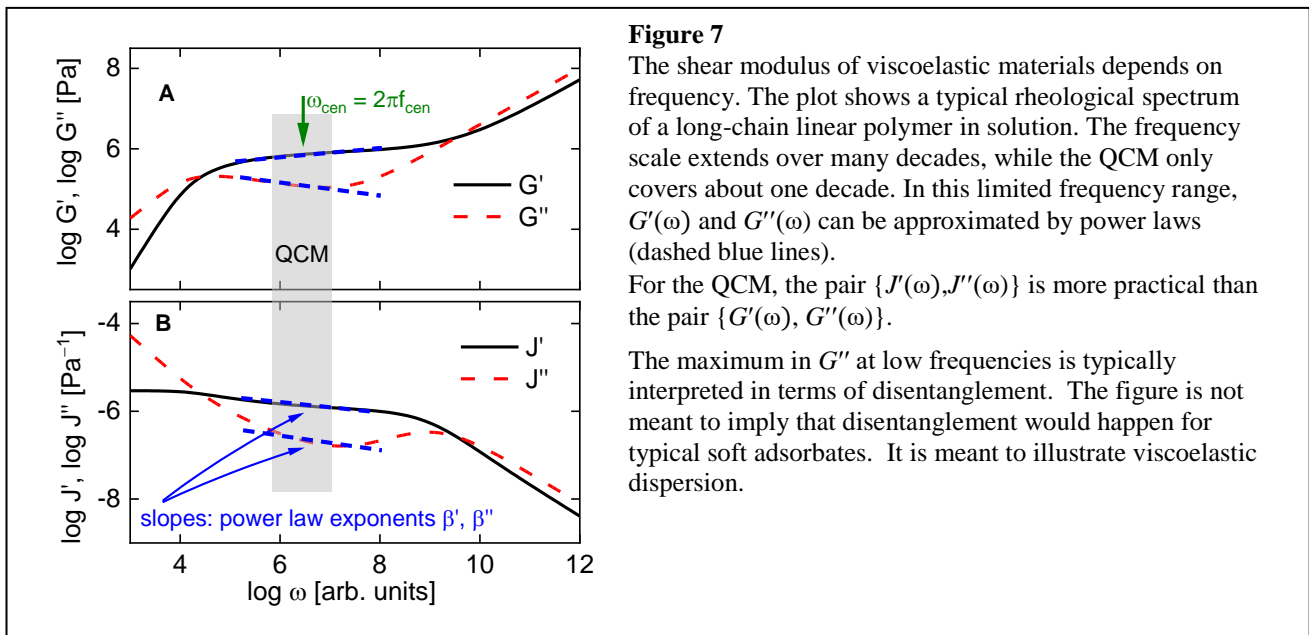


To the details: If profiles of the complex shear modulus,  $\tilde{G}(z) = G'(z) + iG''(z)$ , and of the density,  $\rho(z)$ , are given, one may calculate the displacement  $\hat{u}(z)$  by solving the differential equation:

$$-\rho(z)\omega^2\hat{u}(z) = \frac{d\hat{\sigma}}{dz} = \frac{d}{dz}\left(\tilde{G}(z)\frac{d\hat{u}(z)}{dz}\right) \quad \text{Eq. 3}$$

This is done with routine “solve\_ivp”. The complex frequency shift follows from  $\hat{u}(z=0)$  and  $d\hat{u}/dz(z=0)$  as

$$\frac{\Delta f + i\Gamma}{f_0} = \frac{i}{\pi Z_q} \tilde{Z}_L = \frac{i}{\pi Z_q} \frac{\hat{\sigma}_s}{\hat{v}_s} = \frac{i}{\pi Z_q} \frac{-\tilde{G}(z=0) \frac{d\hat{u}(z)}{dz} \Big|_{z=0}}{i\omega \hat{u}(z=0)} \quad \text{Eq. 4}$$



#### 4.7 Choice of parameters quantifying viscoelasticity

In rheology, a material's stiffness is usually described in terms of the shear modulus,  $\tilde{G}$ , where the tilde denotes a complex number. Sometimes, the viscosity,  $\tilde{\eta} = \tilde{G}/(i\omega)$ , is used. For the QCM, it is often convenient to use the compliance,  $\tilde{J} = 1/\tilde{G}$ . The reasons are discussed below Eq. 7. For ease of interpretation, PyQTM offers different sets of parameters, which are

- $J'$  and  $J''$
- $G'$  and  $G''$
- $\eta'$  and  $\eta''$
- $G'$  and  $\eta'$
- $|\tilde{J}|$  and  $\tan(\delta)$
- $|\tilde{G}|$  and  $\tan(\delta)$
- $|\tilde{\eta}|$  and  $1/(\tan(\delta))$

Importantly, the viscoelastic parameters depend on frequency. This may create the impression that the problem was underdetermined because there are separate values of  $J'$  and  $J''$  (or of  $G'$  and  $G''$  or of ...) for every

single overtone. However, the frequency dependence of  $J'$  and  $J''$  usually is smooth (Figure 7). PyQTM assumes power laws of the form <sup>5</sup>

$$J'(f) = J'_{cen} \left( \frac{f}{f_{cen}} \right)^{\beta'}$$

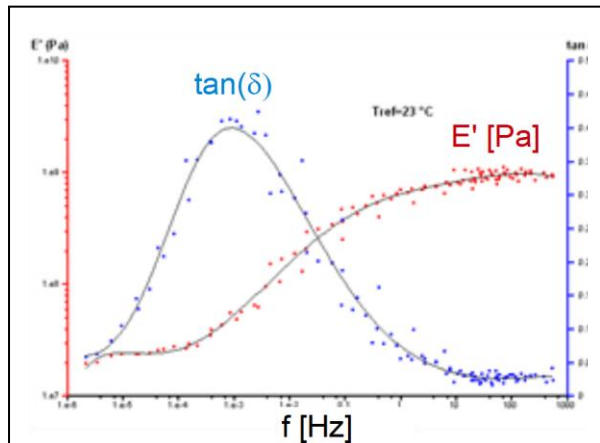
$$J''(f) = J''_{cen} \left( \frac{f}{f_{cen}} \right)^{\beta''}$$
Eq. 5

or (for instance)

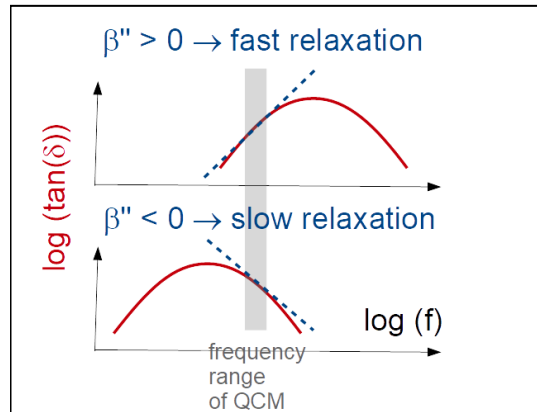
$$|G|(f) = |G|_{cen} \left( \frac{f}{f_{cen}} \right)^{\beta'}$$

$$\tan(\delta)(f) = \tan(\delta)_{cen} \left( \frac{f}{f_{cen}} \right)^{\beta''}$$
Eq. 6

$f_{cen}$  is a frequency in the center of the accessible range. The parameter  $\tan(\delta)$  is defined as  $\tan(\delta) = G''/G' = J''/J' = \eta'/\eta''$ . The loss tangent is independent of whether viscoelasticity is quantified with  $G$ ,  $J$ , or  $\eta$ . If the loss tangent has a peak at some frequency, the medium under study undergoes relaxations with rates in the range of the peak in  $\tan(\delta)$  (Figure 8, Figure 9).



**Figure 8:** The loss tangent (blue) has a peak at a frequency corresponding to the rate of relaxation, which governs the dynamics.  
[https://commons.wikimedia.org/wiki/File:Master\\_curves\\_on\\_polymer.png#/media/File:Master\\_curves\\_of\\_a\\_polymer.png](https://commons.wikimedia.org/wiki/File:Master_curves_on_polymer.png#/media/File:Master_curves_of_a_polymer.png), (slightly modified)



**Figure 9:** The sign of the power-law exponent pertaining to  $\tan(\delta)$  indicates, whether the main relaxation is faster or slower than the frequency of the QCM.

<sup>5</sup> Power laws here are to be understood as approximations. It is not assumed that the sample was a „power-law fluid“ in the narrow sense. Some models of polymer viscoelasticity predict power-law behavior in certain frequency ranges. When they do, the power law exponents often are the same for  $G'(\omega)$  and  $G''(\omega)$ . The spectrum shown in Figure 7 predicts power law behavior in the high-frequency range. It was calculated from the BZW model (Baumgaertel et al., *Rheologica Acta* 29, 1990, 400).

If the viscoelastic parameters are  $\{J', J''\}$ ,  $\{G', G''\}$ ,  $\{\eta', \eta''\}$ , or  $\{G', \eta'\}$ , the Kramers-Kronig relations impose limits on the power-law exponents. For instance, one has  $-2 < \beta' < 0$  and  $-1 < \beta'' < 1$  for the case of  $\{J', J''\}$ . These values are implemented as default limits. For the other choices, the limits are less stringent.

There is a slight problem with power laws: A power law behavior in  $J'(f)$  and  $J''(f)$  does not translate to a power law after converting to  $G'(f)$  and  $G''(f)$  with the relations  $G' = J'/(J'^2 + J''^2)$  and  $G'' = J''/(J'^2 + J''^2)$ . When the representation is changed from  $\tilde{J}$  to  $\tilde{G}$ , the simulated curves slightly change.

A representation with  $\tan(\delta)$  is interesting insofar, as the peak in  $\tan(\delta)$  does not depend on whether  $\tan(\delta)$  is defined as  $J''/J'$ , as  $G''/G'$ , or as  $\eta'/\eta''$ . The position of the peak in  $\tan(\delta)$  is often identified with the rate of the main relaxation (Figure 8). Applied to the QCM, this argument implies that a positive power law exponent of  $\tan(\delta)$  implies a “fast” relaxation, while a negative power law exponent implies a “slow” relaxation (Figure 9).

Occasionally, fitting with sets of parameters involving  $\tan(\delta)$  is less efficient than fitting with the real and imaginary part. In these cases, it is advised to first fit with the real and imaginary part (for example with  $G''$  and  $G'$ ) and to then redo the fit with  $\{|G|, \tan(\delta)\}$ , starting with the values obtained after the first fit.

PyQTM uses the unit MPa and  $\text{MPa}^{-1}$  for the shear modulus and the shear compliance. This unit is convenient because a viscosity of 1 mPa s (about the viscosity of water) corresponds to a shear modulus of 0.1 MPa if converted to  $G$  with the relation  $G'' = i\omega\eta$  and  $\omega = 2\pi \cdot 15 \text{ MHz} \approx 10^8 \text{ s}^{-1}$ . Materials with a shear stiffness  $> 100 \text{ MPa}$  appear as rigid to the QCM. Interesting non-gravimetric behavior is seen when the shear stiffness is a few MPa (that is, if the viscosity is a few times the viscosity of water).

#### 4.8 Thin films with moderate softness

This section is background information. The equations are not actually implemented in PyQTM.

For thin films,  $J'$  and  $J''$  are the most suitable parameters for the analysis of QCM experiments because there are simple and intuitive relations between  $J'$  and  $J''$ , on the one hand, and the data sets  $\{\Delta f/n, \Delta\Gamma/n\}$ , on the other. There are two sets of such relations, one for the film in air and one for the film in a liquid.

For the film *in a liquid*, the relation is

$$\begin{aligned} \frac{\Delta f}{n} + i \frac{\Delta\Gamma}{n} &\approx \frac{-2nf_0^2}{Z_q} m_f \left[ 1 - in \left( 2\pi f_0 \frac{\rho_{bulk}}{\rho_f} \eta_{bulk} \right) (J_f' - iJ_f'') \right] \quad \text{a)} \\ \frac{\Delta f}{n} &\approx \frac{-2nf_0^2}{Z_q} m_f \left[ 1 - n \left( 2\pi f_0 \frac{\rho_{bulk}}{\rho_f} \eta_{bulk} \right) J_f'' \right] \quad \text{b)} \\ \frac{\Delta\Gamma}{n} &\approx \frac{2nf_0^2}{Z_q} m_f n \left( 2\pi f_0 \frac{\rho_{bulk}}{\rho_f} \eta_{bulk} \right) J_f' \quad \text{c)} \end{aligned} \tag{Eq. 7}$$

It is instructive to analyze the ratio  $\Delta\Gamma/(-\Delta f)$ . For the thin film in a liquid, the ratio is linked to  $J_f'$  as

$$\begin{aligned} \frac{\Delta\Gamma}{-\Delta f} &\approx \frac{n \left( 2\pi f_0 \frac{\rho_{bulk}}{\rho_f} \eta_{bulk} \right) J_f'}{1 - \left[ n \left( 2\pi f_0 \frac{\rho_{bulk}}{\rho_f} \eta_{bulk} \right) J_f'' \right]} \\ &\approx 2\pi n f_0 \eta_{bulk} J_f' = \omega \eta_{bulk} J_f' \end{aligned} \tag{Eq. 8}$$

The second step assumes  $J''_f$  to be small and  $\rho_{\text{bulk}} \approx \rho_f$ . The term in curly brackets can also be written as  $J''_f/J''_{\text{bulk}}$ . **If  $J''_f/J''_{\text{bulk}}$  is much smaller than unity, Eq. 8 applies (only then).**

Following Eq. 8, an approximate value of  $\beta'$  can be obtained by plotting  $\log(\Delta\Gamma/(-\Delta f)/n)$  versus  $\log(n)$  and reading the slope of this line (Figure 10b). However, the approximations leading to Eq. 8 will also affect the slope. PyQTM does not make these approximations when inferring  $\beta'$  from  $\Delta f/n$  and  $\Delta\Gamma/n$ . PyQTM fits the full equation for arbitrary thickness to the data.

For the thin film *in air*, one has

$$\frac{\Delta f}{n} + i \frac{\Delta\Gamma}{n} \approx \frac{-2nf_0^2}{Z_q} m_f \left[ 1 + \frac{n\pi^2}{3} \left( \frac{J_f' - iJ_f''}{\rho_f} Z_q^2 - 1 \right) \left( \frac{m_f}{m_q} \right)^2 \right] \quad \text{Eq. 9}$$

and

$$\frac{\Delta\Gamma}{-\Delta f} \approx \frac{n\pi^2}{3} \left( \frac{J_f''}{\rho_f} Z_q^2 \right) \left( \frac{m_f}{m_q} \right)^2 \quad \text{Eq. 10}$$

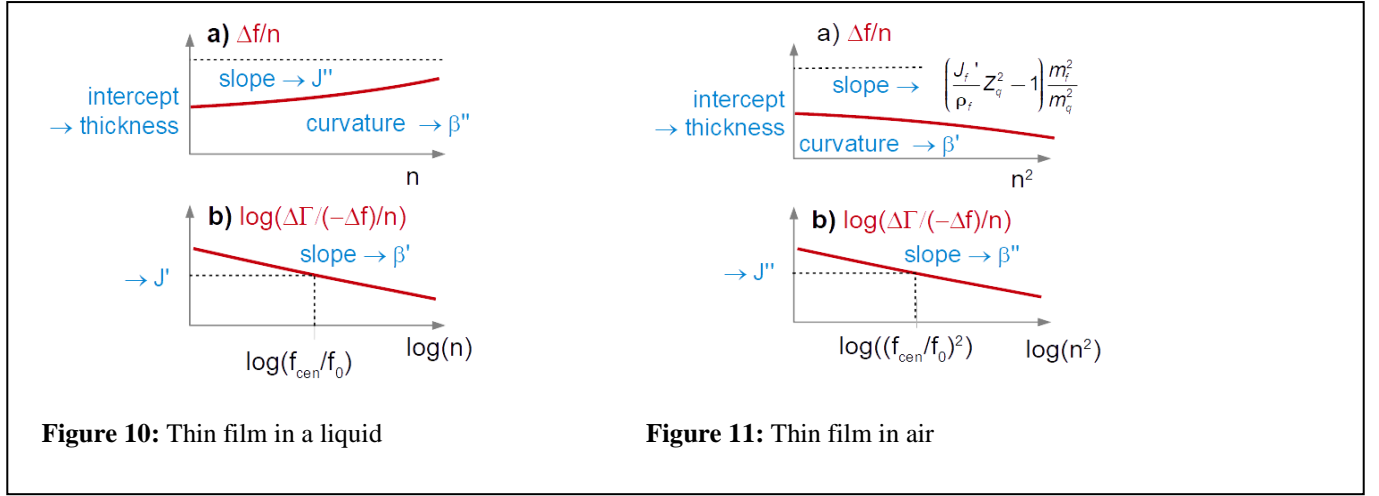
Both in air and in liquid,  $J'$  and  $J''$  appear in the numerator of the viscoelastic correction. The trivial case (Sauerbrey behavior) corresponds to  $J' = J'' = 0$ . The term “−1” in Eq. 9 is non-trivial. It is part of the results from perturbation analysis (see section 1.1.2 in the document [Equations\\_for\\_Perturbation\\_Analysis.pdf](#)).

Assume a single film in a bulk liquid. This problem has a total of 5 unknowns, which are  $d$ ,  $J'(f_{\text{cen}})$ ,  $J''(f_{\text{cen}})$ ,  $\beta'$ , and  $\beta''$ . If viscoelasticity is represented with  $\{J', J''\}$ , 4 of these 5 parameters can be inferred with good accuracy from plots as shown in Figure 10. The correspondences are:

- the thickness is proportional to the intercept with the y-axis in a plot of  $\Delta f/n$  versus  $n$  (note the problem discussed in section 4.9, though).
- $J''$  is proportional to the slope in a plot of  $\Delta f/n$  versus  $n$ .
- $J'$  is roughly proportional to the ratio  $\Delta\Gamma/(-\Delta f)$ , see Eq. 8. An approximation is involved.
- $\beta'$  is about equal to the slope in a log-log plot as shown in Figure 10b.

The parameter  $\beta''$  is linked to the curvature in the plot of  $\Delta f/n$  versus  $n$ . Unfortunately, this curvature cannot usually be derived with confidence. For that reason,  $\beta''$  often remains uncertain. This argument applies if  $J'$  and  $J''$  are chosen as the viscoelastic parameters. Otherwise, the uncertainty spreads to both exponents.

Similar arguments apply to the thin film in air, where the roles of  $J'$  and  $J''$  are interchanged (Figure 11).



#### 4.9 Thick, soft adsorbates

The arguments from section 4.8 only apply, if the denominator in Eq. 8 is about unity. This condition can be rewritten as

$$n \left( 2\pi f_0 \frac{\rho_{bulk}}{\rho_f} \eta_{bulk} \right) J_f'' = \frac{\rho_{bulk}}{\rho_f} \frac{J_f''}{J_{bulk}''} = \frac{\rho_{bulk}}{\rho_f} \frac{\eta_{bulk}'}{\eta_f'} \ll 1 \quad \text{Eq. 11}$$

Dilute adsorbates (such as polymer brushes, which are highly swollen in the solvent) do not always fulfill this criterion. The sample might be a layer with slightly increased viscosity compared to the bulk. It is difficult to infer statements on the geometric parameters of such layers from QCM data. Plotting  $\chi^2$  versus the assumed thickness (section 4.5.3), one finds a broad minimum. Further complications are :

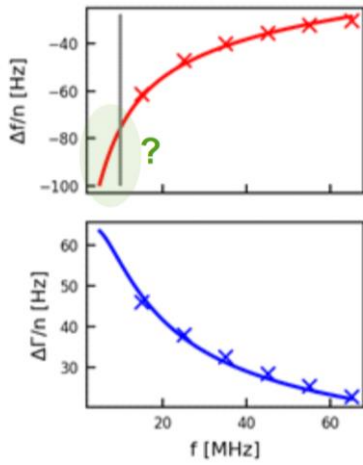
- The equations implicitly assume that the power law can be extrapolated to  $n = 0$  (Figure 13).
- These layers may not be acoustically thin.
- There may be a viscoelastic profile,  $\{G'(z), G''(z), \rho(z)\}$ .

The best one can do is to check, whether the sample is thin and moderately soft in the sense of Eq. 11. A second check can be a fit with zero film thickness and altered viscosity of the bulk. If such a fit leads to reasonable values, the sample probably is thick and soft.

#### 4.10 Uniqueness of fit results if a thin sample looks Sauerbrey-like

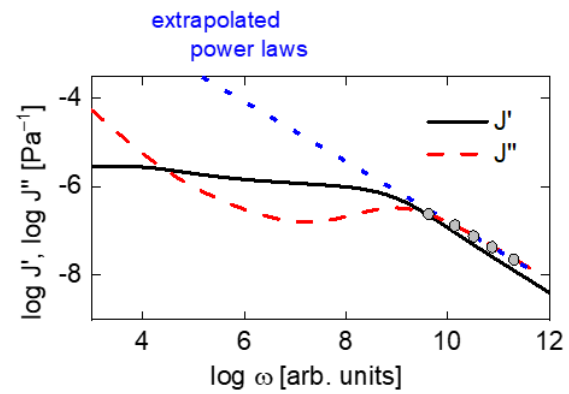
When  $-\Delta f/n$  is the same on all overtones and when  $-\Delta f \gg \Delta\Gamma$ , one typically interprets this finding as being caused by a rigid film. There is, however, another possibility. If the layer is viscous ( $\tan(\delta) \gg 1$ ,  $J'' > J'$ ), and if, further,  $J''$  scales as  $1/n$ , one again finds  $-\Delta f/n \approx \text{const}$  and  $-\Delta f \gg \Delta\Gamma$ . These conditions are realized in Newtonian liquids. A near-surface layer with slightly increased viscosity (common in electrochemistry) looks like a rigid film to the QCM. For more details, see Ref. 11.

PyQTM cannot solve this problem. Depending on the starting values for the fit, it will either predict the layer to be rather rigid or to look like a Newtonian liquid with a viscosity slightly higher than the viscosity of the bulk. The  $\chi^2$  landscape plotted as a function of  $\tan(\delta)$  is rather flat in these cases.



**Figure 12**

In the presence of viscoelastic dispersion, the fitted data sometimes agree well with experimental values, but when extrapolating the fitted curve to  $n = 0$ , a large negative value of  $\Delta f/n$  is obtained. This can lead to an unreasonably large thickness.



**Figure 13**

There may be a danger if power laws are extrapolated to zero frequency. In the case shown here,  $J''$  is not as large at low frequencies, as the dotted blue line suggests. The dotted blue line lets  $\Delta f/n$  diverge to large negative values in Figure 12.

#### 4.11 Fitting in the $\{J', J''\}$ representation, leaving $\beta''$ fixed

Because the power laws applied to the different representations of viscoelasticity are not fully equivalent, the fit results obtained with the different representations are not strictly the same. For the sake of reproducibility and comparison between samples, one might decide to always fit with the same representation. Of course reproducibility in this context must not be mistaken for accuracy. If different representations yield different results (that is, if the different sets of fitted parameters do not agree after conversion between the representations), this points to uncertainties of a fundamental kind. In such cases, the chosen representation amounts to a bias.

The  $\{J', J''\}$  representation is the preferred representation for fitting. First, the relations between the fitted parameters and the plots  $\Delta f/n$ ,  $\Delta \Gamma/n$ , and  $\Delta \Gamma/(-\Delta f)$  versus  $n$  are most transparent (see Figure 10 and Eq. 7). The user can intuitively understand the outcome of the fit. Second, if five free parameters turn out to be too many for practical fitting and if the user still wishes to proceed with some kind of quantitative analysis, keeping  $\beta''$  fixed in the  $\{J', J''\}$  representation is the least problematic option.  $\beta''$  then is the one most uncertain parameter and errors in  $\beta''$  are largely uncorrelated with errors in  $J'$ ,  $J''$ , and  $\beta'$ .  $\beta''$  quantifies the curvature in the plot of  $\Delta f/n$  versus  $n$  (Figure 10B). It does not affect the slope in this plot, which governs  $J''$ . It neither affects the plot of  $\Delta \Gamma/n$  versus  $n$ , nor the plot of  $\Delta \Gamma/(-\Delta f)$  versus  $n$ .  $J'$  is about equal to  $\Delta \Gamma/(-\Delta f) \cdot (\omega \eta_{\text{bulk}})^{-1}$  at  $f = f_{\text{cen}}$ .  $\beta'$  is obtained from the slope in Figure 10B. Fixing  $\beta''$  will, however, affect the derived thickness. If  $\beta''$  is close to  $-1$ , the curvature is negative and the apparent thickness becomes large (red line at the top in Figure 12). If  $\beta''$  close to  $-1$  and if  $\beta''$  fluctuates, the derived thickness will also fluctuate (strongly). Fixing  $\beta''$  avoids those fluctuations. A smooth dependence of thickness on time will be obtained. Again: Different smooth curves will be obtained, depending on the values chosen for  $\beta''$ . These arguments apply in the thin-film limit, only. The conversion is only applied to the display of the fit parameters for the time series on the right-hand side, not to the parameters in the parameters frame.

For the sake of simplicity, one might assume  $\beta'' = 0$ , but this assumption is just arbitrary as, say,  $\beta'' = +0.5$  or  $\beta'' = -0.5$ . (Following from the Kramers-Kronig relations,  $\beta''$  in the  $\{J', J''\}$  representation is between  $-1$  and  $+1$ , see Figure 7B). Negative values of  $\beta''$  imply that the rates of the main relaxation are less than  $5 \cdot 10^6 \text{ s}^{-1}$  (that is:

less than 5 MHz). Positive values imply fast relaxations in the same sense. Again, the value of  $\beta''$  is not easily determined from experiment if the sample is thin.

If  $\beta''$  is strongly negative and if the user fixes  $\beta''$ , the user should simply not report the derived values of the thickness. This is the case, where gravimetry fails, also discussed in section 4.10. If gravimetry fails, that does not imply that the viscoelastic analysis was entirely worthless. Only the absolute values of the thickness will be uncertain. (Trends in the evolution of the thickness versus time will usually be robust).

$J'$  and  $J''$  are not usually the preferred parameters for the discussion with non-specialists. Researchers from polymer rheology typically are familiar with  $|G|$  and  $\tan(\delta)$ . A second case are liquid-like layer, in which case  $\eta'$  and  $\eta''$  allow for a direct comparison with the bulk liquid (and hence are more suitable parameters than  $|G|$  and  $\tan(\delta)$ ). Importantly, the conversion to the preferred representation may occur *after* the fit. Conversion of the laws requires a  $\chi^2$ -minimization because the power laws are not strictly equivalent in all cases.<sup>6</sup> This conversion is routinely done by PyQTM when the user switches representation. PyQTM always updates the plots of the simulated values of  $\Delta f/n$  and  $\Delta\Gamma/n$  (with hardly any noticeable differences). Conversion between representations after the fit is not at all a problem.

#### 4.12 Remarks to details

- The user may change the limits for fitting, in principle. However, PyQTM sets the limits back to the defaults, whenever the user switches the representation of viscoelasticity. Make sure that your choice was still valid when you actually ran the fit.
- The QCM can be used to determine the complex viscosity of bulk liquids at high frequencies. The technique also carries the name “high frequency rheology”. For reasons, which are poorly understood,  $\eta''$  on the higher overtones is sometimes reported as being negative by the QCM. One reason may be uncontrolled adsorption.  $\eta''$  is proportional to  $\Delta\Gamma^2 - \Delta f^2$ . Adsorption makes  $\Delta f$  more negative and thereby drives  $\Delta\Gamma^2 - \Delta f^2$  into the negative range. Another reason may be coupling to the anharmonic side bands, which is more likely on the high overtones and is more likely with broad resonances (produced by a liquid with high viscosity, Fig. 41 in Ref. 1).
- PyQTM reports an rms-noise on the lower left of user interface (“(δf/n)\_rms [mHz]”). This value is computed as the square root of the Hadamard variance,  $\delta f_{Had}^2 = 1/6 \langle (f_{i+1} - 2f_i + f_{i-1})^2 \rangle$ . In the absence of drift, the square root of the Hadamard variance is the same as the root-mean-square noise. The rms-noise is averaged over all active overtones.

The rms-noise is computed from the time series inside the fitting range. If no fitting range is selected, the rms-noise is not displayed.

The calculation of the rms-noise does not take the time base into account. Preaveraging improves the rms-noise.

- Some settings can only be changed from the file Master.qtm, meaning that the user interface does not contain the respective dialogs. Most of these are linked to what might be called a “development version”.
- There is an option to display data either versus time (“x: time” close to the Import button) or versus row number (“x: i” close to the Import button). If the second option is chosen, files can be merged and PyQTM ignores gaps in time between measurements. It just plots the data sets one after the other.
- PyQTM crashes rarely. When it crashes, you see an error message on the console. Please send a mail with the error message. (You also see warnings on the console. Those can be ignored.)

PyQTM does not usually crash, when a fit fails. Still.... it sometimes does. Avoid fitting around on the baseline (will all  $\Delta f/n$ 's  $\approx 0$  and all  $\Delta\Gamma/n$ 's  $\approx 0$ ).

---

<sup>6</sup> PyQTM first converts values at every single overtone,  $n$ , and then fits a power law to these converted values.

## 5 Underlying Equations

### 5.1 Multilayer formalism

The multilayer formalism is the canonical model of the QCM in contact with viscoelastic layers.<sup>7,2,8,9,10</sup> The community agrees on these equations. (Others use other variables, but the equations are the same.)

The multilayer result for a system of two layers in a liquid still fits into one line:

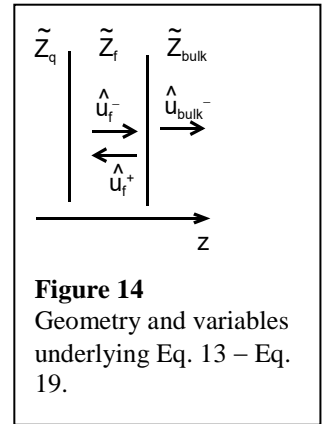
$$\frac{\Delta \tilde{f}}{f_0} = \frac{-\tilde{Z}_e \tilde{Z}_f \left( \tilde{Z}_e \tan(\tilde{k}_e d_e) + \tilde{Z}_f \tan(\tilde{k}_f d_f) \right) + i \tilde{Z}_{liq} \left( \tilde{Z}_e \tan(\tilde{k}_f d_f) \tan(\tilde{k}_e d_e) - \tilde{Z}_f \right)}{\pi Z_q \tilde{Z}_f \left( \tilde{Z}_e - \tilde{Z}_f \tan(\tilde{k}_f d_f) \tan(\tilde{k}_e d_e) \right) + i \tilde{Z}_{liq} \left( \tilde{Z}_e \tan(\tilde{k}_f d_f) + \tilde{Z}_f \tan(\tilde{k}_e d_e) \right)} \quad \text{Eq. 12}$$

Eq. 12 can be a fit function in Excel. The Excel solver should produce the same fits as PyQTM. If the interface to the bulk liquid displays small-scale roughness, PyQTM replaces  $\tilde{Z}_{liq}$  in Eq. 12 by  $\tilde{Z}_{liq,eff}$  from Eq. 25.

PyQTM does not directly apply this equation. It rather uses an algorithm (in the subroutine MultilayerFormalism() in QTM\_Core), which can be extended to an arbitrary number of layers. There is a set of two complex velocity amplitudes,  $\hat{u}^+$  and  $\hat{u}^-$ , pertaining to two waves propagating in the direction of  $+z$  and  $-z$ . The waves are of the form  $\hat{u}^+ \exp(i\omega t - ikz)$  and  $\hat{u}^- \exp(i\omega t + ikz)$ . In the bulk, there is no wave traveling inwards ( $\hat{u}^-_{bulk} = 0$ ). The amplitude of the other wave may be set to 1, meaning that all other amplitudes may be normalized to  $\hat{u}^+_{bulk}$ . Continuity of stress at an interface between layers 1 and 2 can be expressed as

Eq. 13

$$\begin{aligned} \tilde{\eta}_1 \frac{d}{dz} (\hat{u}_1^- + \hat{u}_1^+) &= \tilde{\eta}_2 \frac{d}{dz} (\hat{u}_2^- + \hat{u}_2^+) \\ \tilde{\eta}_1 (i \tilde{k}_1 \hat{u}_1^- - i \tilde{k}_1 \hat{u}_1^+) &= \tilde{\eta}_2 (i \tilde{k}_2 \hat{u}_2^- - i \tilde{k}_2 \hat{u}_2^+) \\ \tilde{\eta}_1 \left( i \frac{\omega}{\tilde{c}_1} \hat{u}_1^- - i \frac{\omega}{\tilde{c}_1} \hat{u}_1^+ \right) &= \tilde{\eta}_2 \left( i \frac{\omega}{\tilde{c}_2} \hat{u}_2^- - i \frac{\omega}{\tilde{c}_2} \hat{u}_2^+ \right) \\ i\omega \tilde{\eta}_1 \left( \sqrt{\frac{\rho_1}{\tilde{G}_1}} \hat{u}_1^- - \sqrt{\frac{\rho_1}{\tilde{G}_1}} \hat{u}_1^+ \right) &= i\omega \tilde{\eta}_2 \left( \sqrt{\frac{\rho_2}{\tilde{G}_2}} \hat{u}_2^- - \sqrt{\frac{\rho_2}{\tilde{G}_2}} \hat{u}_2^+ \right) \\ \sqrt{\tilde{G}_1 \rho_1} (\hat{u}_1^- - \hat{u}_1^+) &= \sqrt{\tilde{G}_2 \rho_2} (\hat{u}_2^- - \hat{u}_2^+) \\ \tilde{Z}_1 (\hat{u}_1^- - \hat{u}_1^+) &= \tilde{Z}_2 (\hat{u}_2^- - \hat{u}_2^+) \end{aligned}$$



The relations  $\tilde{k} = \omega/\tilde{c}$ ,  $\tilde{c} = (\tilde{G}/\rho)^{1/2}$ ,  $\tilde{G} = i\omega\tilde{\eta}$ , and  $\tilde{Z} = (\tilde{G}\rho)^{1/2}$  were used.  $\tilde{Z}$  is the material's shear wave impedance.

The continuity of velocity is expressed as

$$\hat{u}_1^- + \hat{u}_1^+ = \hat{u}_2^- + \hat{u}_2^+ \quad \text{Eq. 14}$$



Combination of Eq. 13 and Eq. 14 leads to

$$\begin{aligned}
\text{I: } (\hat{u}_1^- - \hat{u}_1^+) &= \frac{\tilde{Z}_2}{\tilde{Z}_1} (\hat{u}_2^- - \hat{u}_2^+) \\
\text{II: } \hat{u}_1^- + \hat{u}_1^+ &= \hat{u}_2^- + \hat{u}_2^+ \\
\text{I+II: } 2\hat{u}_1^- &= \left( \frac{\tilde{Z}_2}{\tilde{Z}_1} + 1 \right) \hat{u}_2^- + \left( -\frac{\tilde{Z}_2}{\tilde{Z}_1} + 1 \right) \hat{u}_2^+ \\
\text{I-II: } -2\hat{u}_1^+ &= \left( \frac{\tilde{Z}_2}{\tilde{Z}_1} - 1 \right) \hat{u}_2^- + \left( -\frac{\tilde{Z}_2}{\tilde{Z}_1} - 1 \right) \hat{u}_2^+ \\
\begin{pmatrix} \hat{u}_1^+ \\ \hat{u}_1^- \end{pmatrix} &= \frac{1}{2} \begin{bmatrix} 1 + \frac{\tilde{Z}_2}{\tilde{Z}_1} & 1 - \frac{\tilde{Z}_2}{\tilde{Z}_1} \\ 1 - \frac{\tilde{Z}_2}{\tilde{Z}_1} & 1 + \frac{\tilde{Z}_2}{\tilde{Z}_1} \end{bmatrix} \begin{pmatrix} \hat{u}_2^+ \\ \hat{u}_2^- \end{pmatrix} = P_{\text{interf}} \begin{pmatrix} \hat{u}_2^+ \\ \hat{u}_2^- \end{pmatrix}
\end{aligned} \tag{Eq. 15}$$

In the last step, the two amplitudes were combined into a vector. The propagator matrix links the amplitudes on one side of the interface to the amplitudes on the other side.

The amplitudes at the bottom of a layer with thickness  $d$  are related to the amplitudes at the top by

$$\begin{pmatrix} \hat{u}_{\text{bottom}}^+ \\ \hat{u}_{\text{bottom}}^- \end{pmatrix} = \begin{bmatrix} \exp(ikd) & 0 \\ 0 & \exp(-ikd) \end{bmatrix} \begin{pmatrix} \hat{u}_{\text{top}}^+ \\ \hat{u}_{\text{top}}^- \end{pmatrix} = P_{\text{layer}} \begin{pmatrix} \hat{u}_{\text{top}}^+ \\ \hat{u}_{\text{top}}^- \end{pmatrix} \tag{Eq. 16}$$

With two layers, the amplitudes at the bottom of layer 1 (at the resonator surface) are

$$\begin{pmatrix} \hat{u}_{1,\text{bottom}}^+ \\ \hat{u}_{1,\text{bottom}}^- \end{pmatrix} = P_{\text{layer},1} P_{\text{interface},1,2} P_{\text{layer},2} P_{\text{interface},2,3} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{Eq. 17}$$

The stress-velocity ratio at the resonator surface (the load impedance,  $\tilde{Z}_L$ ) is

$$\tilde{Z}_L = - \frac{\tilde{Z}_1 (\hat{u}_{1,\text{bottom}}^- - \hat{u}_{1,\text{bottom}}^+)}{\hat{u}_{1,\text{bottom}}^- + \hat{u}_{1,\text{bottom}}^+} \tag{Eq. 18}$$

A minus sign enters at the front, because the stress is exerted by the sample onto the resonator plate in the downward direction. The complex frequency shift follows as

$$\frac{\Delta f + i\Delta\Gamma}{f_0} = \frac{\Delta f + i(f_{\text{res}} \Delta D / 2)}{f_0} = \frac{i}{\pi Z_q} \tilde{Z}_L \tag{Eq. 19}$$

For a *single layer in air*, Eq. 12 simplifies to

$$\frac{\Delta f}{f_0} = \frac{-1}{\pi Z_q} \tilde{Z}_f \tan(\tilde{k}_f d_f) \tag{Eq. 20}$$

Taylor expansion of Eq. 20 to 1<sup>st</sup> order in film thickness,  $d_f$ , yields the Sauerbrey result. There is a complication with regard to the Taylor expansion to 3<sup>rd</sup> order in  $d_f$ . This expansion reveals finite-compliance effects in the thin-film limit. Taylor expansion of Eq. 20 to 3<sup>rd</sup> order in film thickness yields

$$\frac{\Delta \tilde{f}}{f_0} = \frac{-\omega m_f}{\pi Z_q} \left( 1 + \frac{(n\pi)^3}{3} \frac{\tilde{J}_f}{\rho_f} Z_q^2 \frac{m_f^2}{m_q^2} \right) = -\frac{n}{f_0} C \rho_f d_f \left( 1 + \frac{(n\pi)^3}{3} \frac{\tilde{J}_f}{\rho_f} Z_q^2 \frac{m_f^2}{m_q^2} \right) \quad \text{Eq. 21}$$

The constant  $C$  was introduced in step 2. One may remember that for 5 MHz crystals and a density of the film of  $\rho_f = 1 \text{ g/cm}^3$ , a film thickness of 1 nm corresponds to a frequency shift of 5.7 Hz (meaning  $C\rho_f = 5.7 \text{ Hz/nm}$ ). The second term in brackets in Eq. 21 is the viscoelastic correction.

The perturbation analysis modifies this result to:

$$\frac{\Delta \tilde{f}}{f_0} = \frac{-\omega m_f}{\pi Z_q} \left( 1 + \frac{(n\pi)^3}{3} \left( \frac{\tilde{J}_f}{\rho_f} Z_q^2 - 1 \right) \frac{m_f^2}{m_q^2} \right) \quad \text{Eq. 22}$$

Importantly, the term  $\tilde{J}_f/\rho_f Z_q^2$  from Eq. 21 is replaced by  $\tilde{J}_f/\rho_f Z_q^2 - 1$ . If the film's stiffness is comparable to the stiffness of the crystal (if  $\tilde{Z}_f \approx Z_q$ ), the difference is substantial. If  $\Delta f(n)$  is naively analyzed with Eq. 21, one may find negative values for  $G'$  or  $J'$ .

For a *single layer in a liquid*, Eq. 12 simplifies to

$$\frac{\Delta \tilde{f}}{f_0} = \frac{-\tilde{Z}_f}{\pi Z_q} \frac{\tilde{Z}_f \tan(\tilde{k}_f d_f) - i \tilde{Z}_{liq}}{\tilde{Z}_f + i \tilde{Z}_{liq} \tan(\tilde{k}_f d_f)} - \frac{-\tilde{Z}_{liq}}{\pi Z_q} \quad \text{Eq. 23}$$

Taylor expansion to 1<sup>st</sup> order in the mass per unit area yields

$$\Delta \tilde{f} = -\frac{2f_0 f}{Z_q} m_f \left( 1 - \frac{\tilde{Z}_{liq}^2}{\tilde{Z}_f^2} \right) = -\frac{2f_0 f}{Z_q} m_f \left( 1 - \frac{\tilde{J}_f}{\rho_f} \tilde{Z}_{liq}^2 \right) \quad \text{Eq. 24}$$

Even for very thin films, this equation is different from the Sauerbrey equation. The term in brackets is sometimes associated with the “missing mass effect”.<sup>12</sup> For films in liquids, finite compliance lowers the apparent mass, if determined with the Sauerbrey equation.

## 5.2 Roughness

Shallow roughness on small scales is modeled as:<sup>10</sup>

$$\begin{aligned} \frac{\Delta f}{f_0} &= \frac{-Z_{liq,eff}''}{\pi Z_q} \approx \frac{-1}{\pi Z_q} \sqrt{\frac{\rho_{liq} \omega \eta_{liq}}{2}} \left( 1 + 3\sqrt{\pi} \frac{h_r^2}{l_r \delta} - 2 \frac{h_r^2}{\delta^2} \right) \\ \frac{\Delta \Gamma}{f_0} &= \frac{Z_{liq,eff}'}{\pi Z_q} \approx \frac{1}{\pi Z_q} \sqrt{\frac{\rho_{liq} \omega \eta_{liq}}{2}} \left( 1 + 2 \frac{h_r^2}{\delta^2} \right) \end{aligned} \quad \text{Eq. 25}$$

In the presence of roughness, PyQTM assigns the effective shear-wave impedance from Eq. 25 to the bulk medium's shear wave impedance. In this way, roughness can be part of the multilayer formalism.

Eq. 25 is formulated for Newtonian bulk media. For these, the wavenumber in the bulk is given as  $\tilde{k}_{liq} = (1 - i)/\delta$ . If the bulk is viscoelastic, all occurrences of  $1/\delta$  in Eq. 25 are replaced by  $\tilde{k}_{liq}/(1 - i)$ . Also,  $\eta_{liq}$  is replaced by its complex analog,  $\tilde{\eta}_{liq}$ .

### 5.3 Perturbation analysis

The equations underlying the perturbation analysis are a bit long and have been moved to a separate document. The name of the file is “Equations\_for\_Perturbation\_Analysis.pdf”.

## 6 Glossary

Variable	Definition	Comment
$C$	Mass-sensitivity constant	Eq. 21
$d$	Thickness of a layer	
$D$	Dissipation factor	$D = 1/Q = 2\Gamma/f_{res}$
$f$	Frequency	
$\tilde{f}$	Complex resonance frequency	$\tilde{f} = f_{res} + i\Gamma$
$f_r$	Resonance frequency (also: $f_{res}$ )	also: “series resonance frequency”
$f_0$	Resonance frequency at the fundamental	
$\tilde{G}$	Shear modulus	
$h_r$	Vertical scale of roughness	Eq. 25, “ $h$ ” for “height”
$h_r/l_r$	Aspect ratio	Eq. 25, “ $l$ ” for “lateral”
$\tilde{J}$	Shear compliance	
$\tilde{k}$	Wave number	
$m$	Mass per unit area	
$n$	Overtone order	
$q$	As an index: quartz resonator	
$ref$	As an index: reference state of a crystal in the absence of a load	
$\tilde{Z}$	Shear-wave impedance	$\tilde{Z} = \rho\tilde{c} = (\rho\tilde{G})^{1/2}$
$\tilde{Z}_L$	Load impedance	
$Z_q$	Acoustic shear-wave impedance of AT-cut quartz	$Z_q = 8.8 \cdot 10^6 \text{ kg m}^{-2} \text{ s}^{-1}$
$\beta', \beta''$	Power law exponents	Eq. 5
$\Gamma$	Imaginary part of a resonance frequency, Half-band half width (“half bandwidth”, for short)	
$\delta$	Penetration depth of a shear wave	Newtonian liquids: $\delta = (2\eta_{liq}/(\rho_{liq}\omega))^{1/2}$
$\delta_L$	Loss angle	$\tan \delta_L = G''/G' = J''/J'$ often called $\delta$ in rheology
$\Delta$	As a prefix: A shift induced by the sample	
$\tilde{\eta}$	Viscosity	$\tilde{\eta} = \tilde{G}/(i\omega)$

## References

- <sup>1</sup> Johannsmann, D.; Leppin, C.; Langhoff, A., Studying Soft Interfaces with Shear Waves: Principles and Applications of the Quartz Crystal Microbalance (QCM). *Sensors* **2021**, 21, 3490.
- <sup>2</sup> Voinova, M. V.; Rodahl, M.; Jonson, M.; Kasemo, B., Viscoelastic acoustic response of layered polymer films at fluid-solid interfaces: Continuum mechanics approach. *Physica Scripta* **1999**, 59, (5), 391-396.
- <sup>3</sup> Eisele, N. B.; Andersson, F. I.; Frey, S.; Richter, R. P., Viscoelasticity of Thin Biomolecular Films: A Case Study on Nucleoporin Phenylalanine-Glycine Repeats Grafted to a Histidine-Tag Capturing QCM-D Sensor. *Biomacromolecules* **2012**, 13, (8), 2322-2332.
- <sup>4</sup> Tsionsky, V.; Daikhin, L.; Gileadi, E., Response of the electrochemical quartz crystal microbalance for gold electrodes in the double-layer region. *Journal of the Electrochemical Society* **1996**, 143, (7), 2240-2245.
- <sup>5</sup> Sadman, K.; Wiener, C. G.; Weiss, R. A.; White, C. C.; Shull, K. R.; Vogt, B. D., Quantitative Rheometry of Thin Soft Materials Using the Quartz Crystal Microbalance with Dissipation. *Analytical Chemistry* **2018**, 90, (6), 4079-4088.

- <sup>6</sup> Attributed to Einstein by Roger Sessions in a New York Times article from Jan 8, 1950.
- <sup>7</sup> Johannsmann, D.; Mathauer, K.; Wegner, G.; Knoll, W., Viscoelastic Properties of Thin-Films Probed with a Quartz-Crystal Resonator. *Physical Review B* **1992**, 46, (12), 7808-7815.
- <sup>8</sup> Johannsmann, D., Viscoelastic analysis of organic thin films on quartz resonators. *Macromolecular Chemistry and Physics* **1999**, 200, (3), 501-516.
- <sup>9</sup> Bandey, H. L.; Martin, S. J.; Cernosek, R. W.; Hillman, A. R., Modeling the responses of thickness-shear mode resonators under various loading conditions. *Analytical Chemistry* **1999**, 71, (11), 2205-2214.
- <sup>10</sup> Daikhin, L.; Gileadi, E.; Katz, G.; Tsionsky, V.; Urbakh, M.; Zagidulin, D. Influence of roughness on the admittance of the quartz crystal microbalance immersed in liquids. *Anal. Chem.* **2002**, 74, 554–561.
- <sup>11</sup> Johannsmann, D.; Langhoff, A.; Leppin, C.; Reviakine, I.; Maan, A. M. C., Effect of Noise on Determining Ultrathin-Film Parameters from QCM-D Data with the Viscoelastic Model. *Sensors* **2023**, 23(3), 1348.
- <sup>12</sup> Voinova, M. V.; Jonson, M.; Kasemo, B., 'Missing mass' effect in biosensor's QCM applications. *Biosensors & Bioelectronics* **2002**, 17, (10), 835-841.