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

Last Updated: 1/14/2023

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, ΔD.

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

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

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

The variables are explained in section 5. This one-liner is fully equivalent to the Voigt-Kelvin formalism.[2,3] To the best of our knowledge, it was first written down (in slightly different form) by the Tel Aviv group in the Ref. 4.

Most of you have fitted data with a fit function before. Why is this manual 20-odd 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.

The above remarks struggle with the Einstein quote[5]: "Everything should be made as simple as possible, but not simpler" The bullet points pay recognition to "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.[2,6,7,8] 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 $\Delta\tilde{f}/f_0 = \mathrm{i}/(\pi Z_q)\,\tilde{Z}_L$, 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 $\Delta\tilde{f} = \Delta f + \mathrm{i}\Delta\Gamma$. QTM mostly displays overtone-normalized complex frequency shifts, $\Delta\tilde{f}/n = \Delta f/n + \mathrm{i}\Delta\Gamma/n$

3.2 *Half bandwidth or dissipation factor*

Internally, PyQTM quantifies dissipative processes by the half bandwidth, Γ. Γ 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 $\Delta\tilde{f}/n = \Delta f/n + \mathrm{i}\Delta\Gamma/n = \Delta f/n + \mathrm{i}f_{res}\Delta D/2$. PyQTM offers a choice between the use of $\Delta\Gamma/n$ or ΔD.

3.3 *Roughness*

PyQTM implements shallow, small-scale roughness following Ref. 9 and Eq. 25. More technically, the shear-wave impedance of the liquid is 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 assumptions, which enter this algorithm.

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).

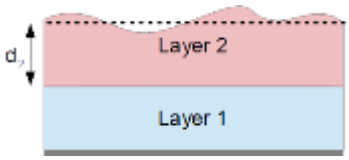

Figure 1: With rough interfaces, the mean height is equal to the thickness of the last layer.

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. The applicability can also be tested by comparing the 3rd-order result to the 5th-order result. If the 3rd-order result and the 5th-order result largely 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[i] must be known in order to derive the film's viscoelastic parameters.

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".

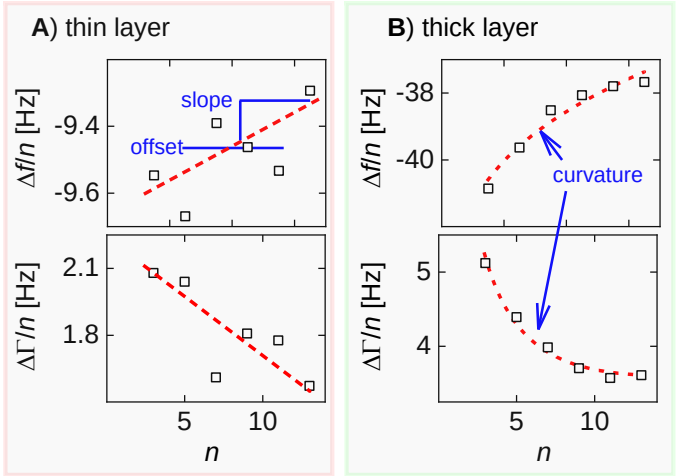

Figure 2: For very thin films (**A**), the fractional noise is too large to let a curvature in plots of $\Delta f/n$ and $\Delta\Gamma/n$ versus n be determined reliably. Interpretation must rely on the offsets and the slopes for $\Delta f/n$ and $\Delta\Gamma/n$ (totaling in four parameters). If the 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.

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).[ii]
b) PyQTM can model thick, soft films, but the derived fit results often are not unique.

---

[i] Gold: $Z_{Au} \approx 23.9 \cdot 10^6$ kg/(m^2 s)

[ii] 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 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, expanded to many layers. However, there is a rather easy way to predict Δf and $\Delta\Gamma$ for such situations, solving the underlying differential equation. Sample code is contained in "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. The interface was meant to be self-explaining.

4.1 *Versions*

PyQTM can be run from Python.[iii] The Clausthal group uses Spyder. The IDLE environment appears to also work. You see the code and can make changes.[iv] QTM.py is the main file. QTM.py imports code from the other files. Start QTM.py to launch the program.

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

4.2 *Input and output*

On start-up, PyQTM reads certain general status information from the file PyQTM.ini. PyQTM saves results into files with the extensions ~tmp.qtm, ~tmp.qtd, and ~tmp.qtf. Fit results – in case they have been produced – are saved into the file ~tmp.qtr. These are ASCII files.

There is a choice in the menu at the top between "Start from Default" or "Start from Previous Config". In case "Start from Default" is selected, PyQTM starts from the file "Default.qtm". Users can edit these settings. For instance, they might set the variable "tt_io_format" to their preferred instrument. Examples for the formats as implemented by the different instruments (QSoft, QSoft_new, AWSensors) are contained in the folder distributed with the code. These examples were kindly provided by Annemarie Maan, Ralf Richter, and Ilya Reviakine.

Time traces acquired are imported with the button "Import". *The data to be imported must have the shape of $(n \times m)$-arrays.* Missing values in a column are not tolerated. In case those exist, the data for importing need to be preprocessed by the user to avoid that. For instance, 0's might be inserted where values are missing. Many QCM-Ds produce an incomplete last line at the end of the output file because they allow the user to stop the measurement in the middle of an overtone sequence and write this last, incomplete sequence to the end of the file. For this reason, PyQTM always ignores the last line of an input file.

[iii] 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).

[iv] In particular, users have adapted the output formats to their needs. This may be achieved by modifying the routine FitPars_to_Clpbd() in QTM.py

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 properties change quickly, an apparent dependence on overtone order may in fact be caused by a dependence on time. PyQTM infers the time derivative of the data and uses with information to correct the data for this artifact. The correction is negligible in most cases.

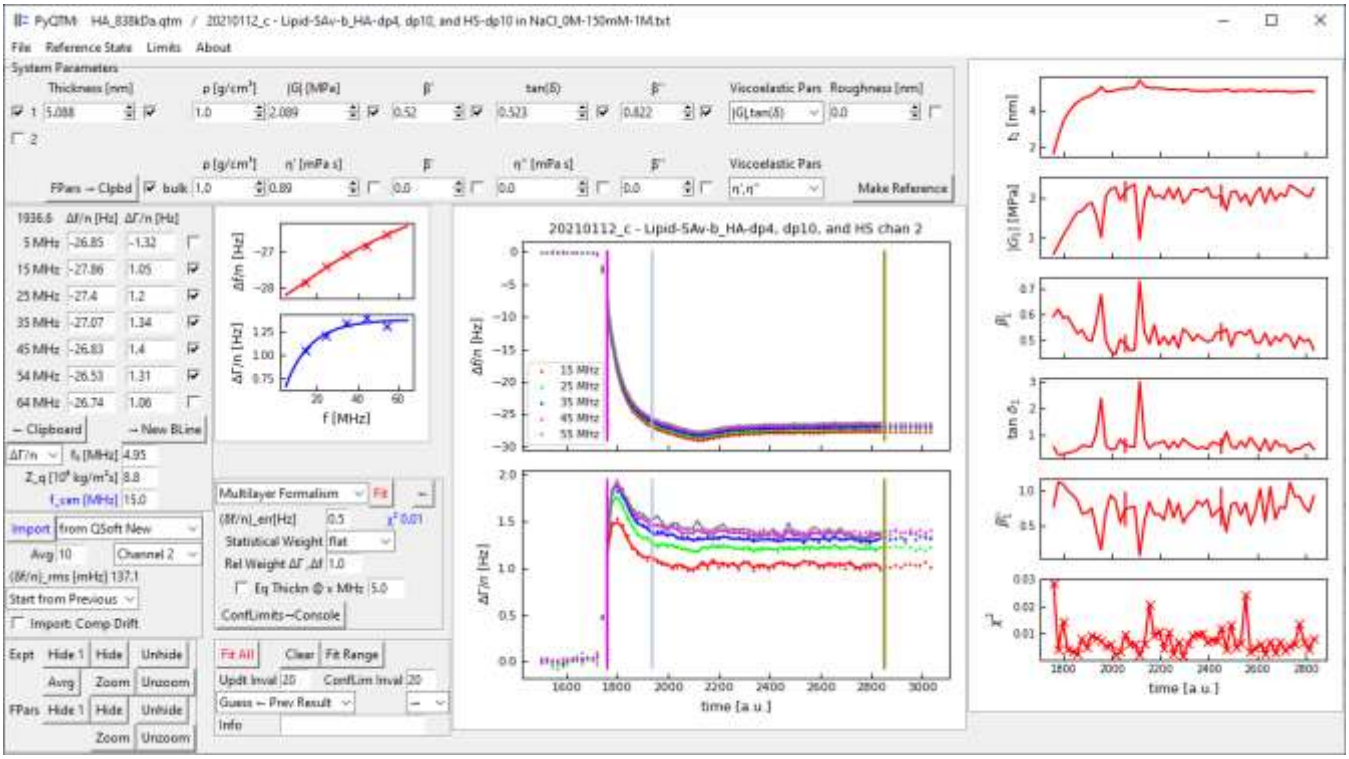

Figure 3: The user interface

The information contained in the temporary files is transferred to the file of your choice when you save. The temporary files contain NaNs (NaN: “not a number”).

~tmp.qtm contains control variables as well as experimental frequency shifts, fit parameters, and the frequency shifts predicted by the fit for *one single data set*. ~tmp.qtm has the structure of a configuration file (also: “ini-file”). One can guess the meanings of the variables to some extent. In case you want to discuss fitting procedures with a collaborator, work on the same qtm-file. PyQTM will collaborator, work on the same qtm-file. PyQTM will then be in the same state on both sides.

The file ~tmp.qtd contains the fit parameters and the standard errors for a time trace. The file ~tmp.qtf contains experimental values and the fitted values of $\Delta f/n$ and $\Delta\Gamma/n$ for the time traces.

PyQTM saves certain values into text files. The file “Experimental_Dfcbyn.txt” contains the current values of $\Delta f/n$ and $\Delta\Gamma/n$. The file “Simulated_Dfcbyn.txt” contains the simulated curve. The file “chi2LS_CrossCors.txt” contains the data plotted in the χ^2 landscape form (“Run 1”). The file “chi2LS_Chi2s.txt” contains the data plotted in the χ^2 landscape form (“Run All”).

Upon “Save as …”, PyQTM saves all these files, but replaces “~tmp” with the filename of your choice.

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 "Zero" and selecting a point in the graph.*

- 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 studied. (It is therefore changed more often than f_0 or Z_q.)
- 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. Your changes are then lost.
  The default limits are such that the density and the viscoelastic parameters can never go to zero. For instance, the minimum density is 10^{-9} g/cm^3. 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 algorithm exits with an error message.

4.4 *Fitting*

The fit is based on a χ^2-minimization. χ^2 is defined as

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

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".

$(\delta f/n)_{err}$ is the statistical uncertainty on $\Delta f/n$ 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 χ^2, not the set of fit parameters, which minimizes χ^2. If $(\delta f/n)_{err}$ was estimated correctly and if χ^2 is of order unity, the quality of the fit is as good as it can be.

The statistical noise can always be lowered with preaveraging during import. However, "noise" in this context includes irregularities between different experiments and crystals, which may be larger than the statistical noise in one single experiment. 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. The magnitude of such irregularities is in the range of 0.1 Hz.

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 ΔΓ , Δf"). This makes sense when $\Delta\Gamma$ is much smaller than Δf. If the weight of $\Delta\Gamma$ is increased, the fit gives these small values more statistical weight than they would usually have.

When fitting entire time traces, there is a choice to either start all fits from the same guess (the values in the parameters panel, "Guess ← Pars Panel") or from the previous fit result ("Guess ← Prev Result"). The latter option should run faster, but carries the danger that one bad fit will let all subsequent fits fail because the guess values from thereon are poor. If the option "Guess ← Prev Result" is chosen, 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 edit fields 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 box next to "Updt 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 ρ 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 μg/cm^2. (With ρ = 1 g/cm^3, a mass per unit area of 1 μg/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 3rd 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 about five overtones) is bound to cause correlated errors (Figure 4).

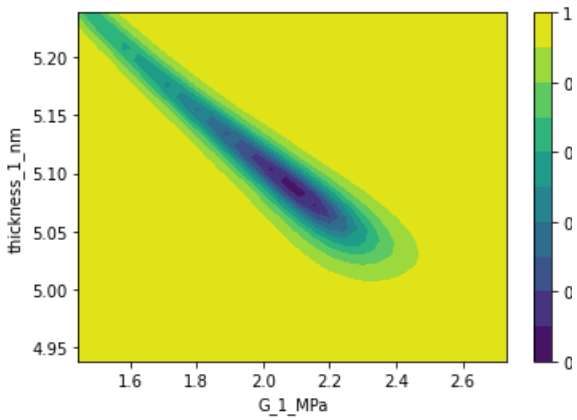

Figure 4: A plot on confidence limits produced by lmfit. Color encodes the quality of the fit. The valley is elongated because the curvature in the QCM data (Figure 2) is determined with less accuracy than the offsets and the slopes. Such elongated valley in the χ^2-are often seen.

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 values. It then performs a fit on those values and obtains a result slightly different from the result obtained on the original data set. The distribution of these results and the standard deviations of those distributions indicate the confidence range. Standard errors from bootstrapping tend to be more pessimistic than the standard errors from the fit routine.

4.5.2 χ^2 Landscape

Inspecting the χ^2 landscape can be instructive. There are three 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 χ^2 versus the varied parameter. One hopes to see a well-defined minimum in χ^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 χ^2. One hopes to see well-defined minima in all plots. The range is specified in the entry field called "Width Factor". The minimum and the maximum are the values from the fit divided by the Width Factor and multiplied by the Width Factor, respectively. An exception are the power law exponents. These run between their limits as set in the Limits form.
- "Run 2D" varies 2 parameters and plots χ^2 as a contour plot. This mode serves to study correlated errors.

4.5.3 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.
- When fitting entire time traces, PyQTM computes confidences limits in intervals specified in the box next to ConfLim Inval. It writes the information into the file "ConfLims.txt". See the source code after "def ConfLims_lmfit_2_Textline():" in QTM_Core.py.
  Switch this off by setting the interval to zero.

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 functions calls might be changed, as well.

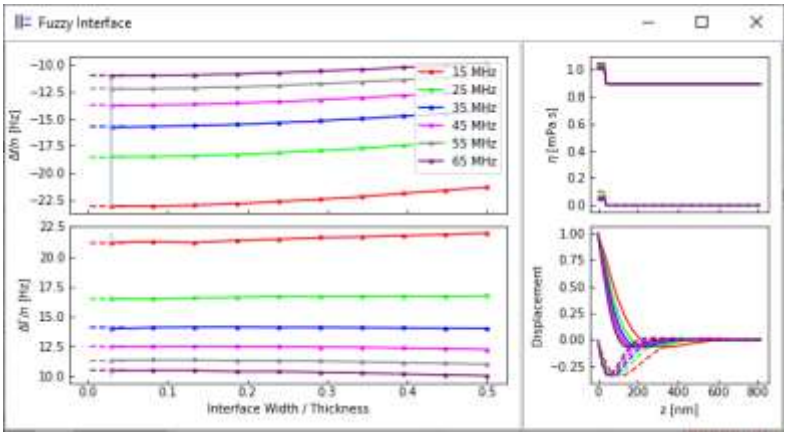

Figure 5: An output from the window "Fuzzy Interface". The panel on the upper right shows the viscoelastic profile, where the width of the interface corresponds to the blue vertical line in the panel to the left. The lower right shows the displacement pattern (dashed for the imaginary part both at the 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.

4.6 *Fuzzy Interface to the Bulk*

Basically, PyQTM requires 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 is sketched in on the upper right in Figure 5.

Predicting values $\Delta f/n$ 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. Arguable, different values of this width should be used for the different parameters and the different overtones.

What can still be done (and what is implemented in PyQTM), is to start from a sharp interface (as assumed everywhere in PyQTM except in the window named "Fuzzy Interface"), to gradually increase the width, and to check if this widening of the interface grossly changes $\Delta f/n$ and $\Delta\Gamma/n$. This is what the window named "Fuzzy Interface" does. Again: This is only meant to be a check as to how wide a smooth interface may be without invalidating the results obtained with a sharp interface.

To the details: If the 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:

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

This is done with "solve_ivp". The complex frequency follows shift from $\hat{u}(z = 0)$ and $\mathrm{d}\hat{u}/\mathrm{d}z(z = 0)$ as

$$\frac{\Delta f + i\Gamma}{f_0} = \frac{i}{\pi Z_q}\tilde{Z}_L = \frac{i}{\pi Z_q}\frac{\hat{\sigma}_S}{\hat{v}_S} = \frac{i}{\pi Z_q}\frac{-\tilde{G}\ z=0\ \left.\frac{\mathrm{d}\hat{u}\ z}{\mathrm{d}z}\right|_{z=0}}{i\omega\hat{u}\ z=0\ |} \qquad \text{Eq. 4}$$

PyQTM assumes hyperpolic tangents with some width for the profiles. And that is all.

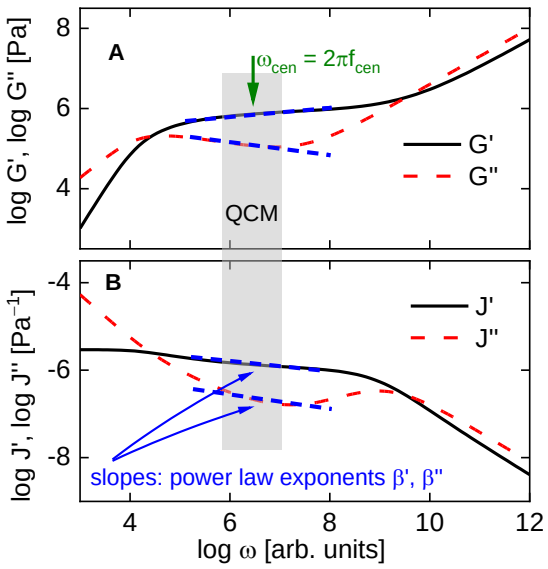

Figure 6
The shear modulus of viscoelastic materials depends on frequency. The plot shows a typical rheological spectrum of a solution of a long-chain linear polymer. The frequency scale extends over many decades, while the QCM only covers about one decade. In this limited frequency range, $G'(\omega)$ and $G''(\omega)$ can be approximated by power laws (dashed blue lines).
For the QCM, the set $\{J'(\omega),J''(\omega)\}$ is more practical than the set $\{G'(\omega), G''(\omega)\}$.

The maximum in G'' at low frequencies is typically interpreted in terms of disentanglement. The figure is not meant to imply that disentanglement would happen for typical soft adsorbates to QCM-D. It is meant to be illustration of viscoelastic dispersion.

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

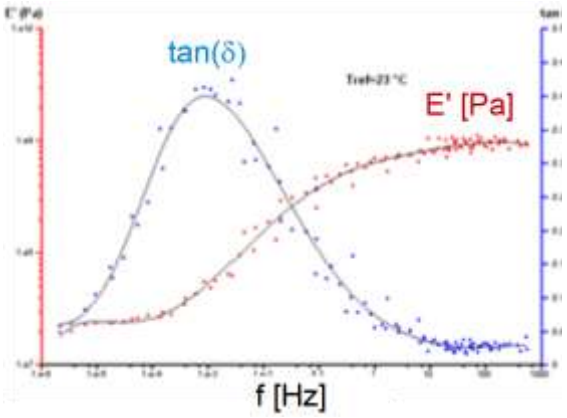

Figure 7: The loss tangent (blue) has a peak at a frequency corresponding to the rate of relaxation, which governs the dynamics. *https://commons.wikimedia.org/wiki/File:Master_curves_on_polymer.png#/media/File:Master_curves_of_a_polymer.png*, (slightly modified for

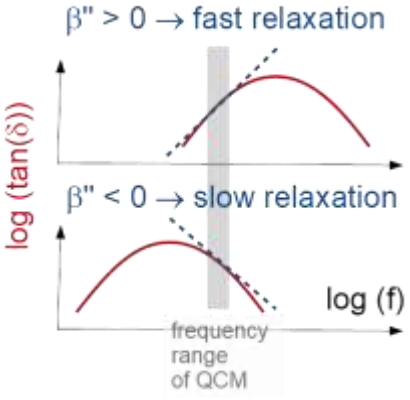

Figure 8: The sign of the power-law exponent pertaining to tan(δ) indicates, whether the main relaxation is faster or slower than the frequency of the QCM.

Importantly, the viscoelastic parameters depend on frequency. This may create the impression that the problem was underdetermined because there are separates values of J' and J'' for every single overtone. However, the frequency dependence of J' and J'' usually is smooth (Figure 6). PyQTM assumes power laws of the form

Eq. 5

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

or (for instance)

Eq. 6

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

f_{cen} is a frequency in the center of the accessible range. The parameter tan(δ) 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 η. 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(δ) (Figure 7,Figure 8).

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

4.8 *Thin films with moderate softness*

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

$$\frac{\Delta f}{n} + \mathrm{i}\frac{\Delta\Gamma}{n} \approx \frac{-2nf_0^2}{Z_q} m_f \left[1 - \mathrm{i}n\left(2\pi f_0 \frac{\rho_{bulk}}{\rho_f}\eta_{bulk}\right)(J_f' - \mathrm{i}J_f'')\right] \quad \text{a)} \qquad \text{Eq. 7}$$

$$\frac{\Delta f}{n} = \frac{-2nf_0^2}{Z_q} m_f \left[1 - n\left(2\pi f_0 \frac{\rho_{bulk}}{\rho_f}\eta_{bulk}\right)J_f''\right] \quad \text{b)}$$

$$\frac{\Delta\Gamma}{n} = \frac{2nf_0^2}{Z_q} m_f n\left(2\pi f_0 \frac{\rho_{bulk}}{\rho_f}\eta_{bulk}\right)J_f' \quad \text{c)}$$

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

$$\frac{\Delta\Gamma}{-\Delta f} \approx \frac{n\left(2\pi f_0 \frac{\rho_{bulk}}{\rho_f}\eta_{bulk}\right)J_f'}{1 - \left\{n\left(2\pi f_0 \frac{\rho_{bulk}}{\rho_f}\eta_{bulk}\right)J_f''\right\}} \approx 2\pi n f_0 \eta_{bulk} J'_f = \omega\eta_{bulk} J'_f \qquad \text{Eq. 8}$$

The second step assumes that J''_f is small and that $\rho_{bulk} \approx \rho_f$. The term in curly brackets can also be written as J''_f/J''_{bulk}. This parameter is displayed on the right-hand side in the parameters frame in the PyQTM user interface. **If J''_f/J''_{bulk} is much smaller than unity, Eq. 8 applies (only then).**

Following Eq. 8, an approximate value of β' can be obtained by plotting $\log(\Delta\Gamma/(-\Delta f))$ versus $\log(n)$ and reading the slope of this line. However, the approximations leading to Eq. 8 will also affect the slope. PyQTM does not make these approximations when inferring β' 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

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

and

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

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 (section 5.3.2).

Assume a single film in a bulk liquid. This problem has a total of 5 unknowns, which are d, $J'(f_{cen})$, $J''(f_{cen})$, β', and β". If viscoelasticity is represented with $\{J', J''\}$ 4 of these 5 parameters can be inferred with good accuracy from plots as shown in Figure 9. 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.
- β' is about equal to the slope in a log-log plot shown in Figure 9b.

The parameter β" 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, β" 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 10).

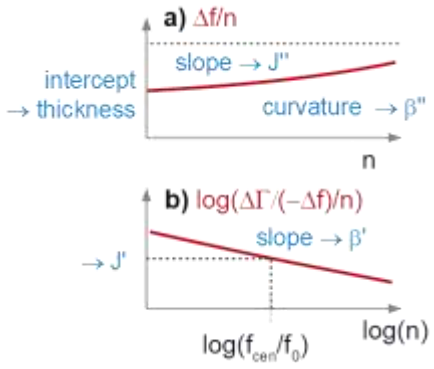

Figure 9: Thin film in a liquid

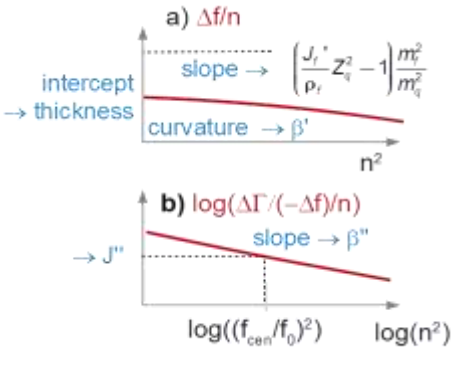

Figure 10: Thin film in air

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

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

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 structure of such films from QCM data. Plotting χ^2 versus assumed thickness (section **4.5.2**), one finds a broad minimum. Further complications are :

- The equations implicitly assume that the power law can be extrapolated to $n = 0$ (section 4.10).
- 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. PyQTM outputs the softness parameter (J''_f/J''_{bulk}) on the lower left of the parameters frame. 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 *Equivalent thickness at some finite frequency*

Interestingly, the one parameter, which typically is the output of a gravimetric QCM experiment – the layer thickness – is determined with considerable uncertainty if the sample is a soft layer in a liquid and if the sample shows viscoelastic dispersion. The problem becomes virulent, if the plot of $\Delta f/n$ versus n shows large negative curvature, as in red in Figure 11.[v] If one extrapolates the red line at the top of Figure 11 to $n = 0$, a large negative intercept of the red line with the y-axis results and the apparent layer thickness becomes correspondingly large. However, this extrapolation to $n = 0$ implicitly invokes the assumption that the power law used to describe viscoelastic dispersion extends to low frequencies. This is assumption may or may not be justified. It is justified for Newtonian liquids, but not usually justified for polymers. The problem is illustrated in Figure 12.

In order to deal with this problem, QTM offers to infer an equivalent thickness from the value of $-\Delta f/n$ (with the Sauerbrey equation) at some finite frequency.[vi] The option is called "Eq. Thickn @ x MHz". "x" stands for the frequency, which can be chosen in the edit field next to the checkbox. The chosen frequency is shown as a vertical grey line in Figure 11. Of course there is some ambiguity in this procedure because the frequency (10 MHz in Figure 11) can be chosen in a certain range. This frequency should be less than the lowest frequency of the experiment but not much less. The absolute values of the thickness derived this way cannot be trusted, but they can be compared between experiments.

4.11 *Uniqueness of fit results if a thin sample looks Sauerbrey-like*

When $-\Delta f/n$ is the same on all overtones and when $-\Delta f >> \Delta\Gamma$, one typically interprets this finding as being caused by a rigid film. There is, however, another possibility. If the layer is viscous ($\tan(\delta) >> 1$, $J'' > J'$), and if, further, J'' scales as $1/n$, one again finds $-\Delta f/n \approx$ const and $-\Delta f >> \Delta\Gamma$. 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.

---

[v] This happens, when β'' in a representation with $\{J',J''\}$ approaches -1.

[vi] This option is only active if the reference state is the bare resonator (in air or in liquid) and if the sample contains one layer, only.

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 χ^2 landscape plotted as a function of tan(δ) is rather flat in these cases.

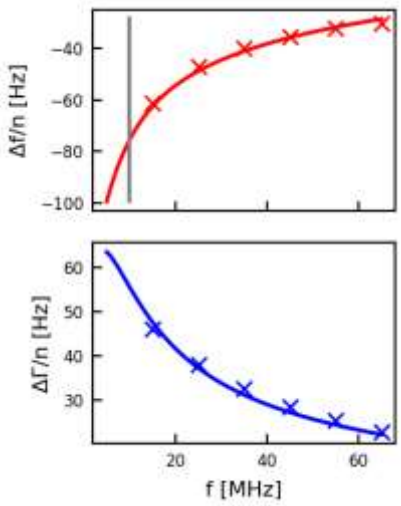

Figure 11
In the presence of viscoelastic dispersion, the fitted data sometimes agree well with experimental values, but when extrapolating the fitted curve to $n = 0$, a large negative value of $\Delta f/n$ is obtained. This can lead to an unreasonably large thickness. In such cases, it is advised to derive the thickness from $\Delta f/n$ at some finite frequency (grey vertical line).

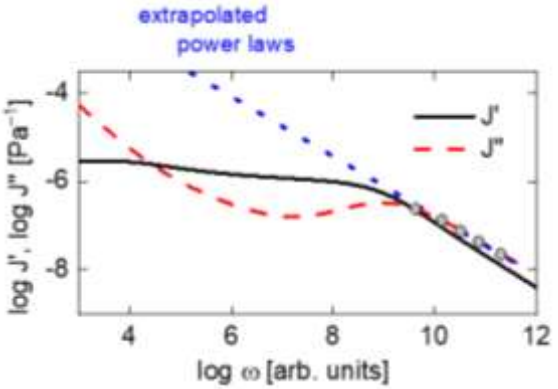

Figure 12
The power laws should not be extrapolated to small frequencies. J'' is not as large at low frequencies, as the dotted blue line suggests.

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 executed the fit.
- If the choice of the viscoelastic parameters is $\{J',J''\}$, the accuracy on the power law exponent β' is better than the accuracy on β''.
- The QCM can be used to determine the complex viscosity at high frequencies. The technique also carries the name "high frequency rheology". For reasons, which are poorly understood, η'' often is reported as being negative by the QCM. One reason is uncontrolled adsorption. η'' is proportional to $\Delta\Gamma^{-2} - \Delta f^{-2}$. Adsorption lowers Δf and may thereby drive $\Delta\Gamma^{-2} - \Delta f^{-2}$ 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).
- 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 Hadamar variance, $\delta f_{Had}^2 = 1/6 < (f_{i+1} - 2f_i + f_{i-1})^2 >$. In the absence of drift, the square root of the Hadamard variance is the same as the root-mean-square noise. This calculation assumes, that the differences between neighboring points are indeed governed by noise. This is not the case, if the data have been extensively averaged before import.
  The rms-noise is computed from the time traces inside the fitting range. If no fitting range is selected, the rms-noise is not displayed.

– Some settings can only be changed from the file ~tmp.qtm, meaning that the user interface does not contain the respective dialogs. Most of these a linked to what might be called a "development version".

5 Underlying Equations

5.1 *Multilayer formalism*

The multilayer formalism is the canonical model of the QCM in contact with viscoelastic layers.[6,2,7,8,9] 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 (no roughness) still fits into one line:

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

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 $\tilde{Z}_{\mathrm{liq,eff}}$ from Eq. 25.

PyQTM does not directly apply this equation. It rather uses an algorithm, which can be extended to arbitrary numbers 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 $-z$, respectively. The waves are of the form $\hat{u}^+\exp(\mathrm{i}\omega t - \mathrm{i}kz)$ and $\hat{u}^-\exp(\mathrm{i}\omega t + \mathrm{i}kz)$. In the bulk, there is no wave traveling inwards ($\hat{u}^-_{\mathrm{bulk}} = 0$). The amplitude of the other wave may be set to 1, meaning that all other amplitudes may be normalized to $\hat{u}^+_{\mathrm{bulk}}$. Continuity of stress at an interface between layers 1 and 2 can be expressed as

Eq. 13

$$\tilde{\eta}_1 \frac{\mathrm{d}}{\mathrm{d}z}\left(\hat{u}_1^- + \hat{u}_1^+\right) = \tilde{\eta}_2 \frac{\mathrm{d}}{\mathrm{d}z}\left(\hat{u}_2^- + \hat{u}_2^+\right)$$

$$\tilde{\eta}_1\left(\mathrm{i}\tilde{k}_1\hat{u}_1^- - \mathrm{i}\tilde{k}_1\hat{u}_1^+\right) = \tilde{\eta}_2\left(\mathrm{i}\tilde{k}_2\hat{u}_2^- - \mathrm{i}\tilde{k}_2\hat{u}_2^+\right)$$

$$\tilde{\eta}_1\left(\mathrm{i}\frac{\omega}{\tilde{c}_1}\hat{u}_1^- - \mathrm{i}\frac{\omega}{\tilde{c}_1}\hat{u}_1^+\right) = \tilde{\eta}_2\left(\mathrm{i}\frac{\omega}{\tilde{c}_2}\hat{u}_2^- - \mathrm{i}\frac{\omega}{\tilde{c}_2}\hat{u}_2^+\right)$$

$$\mathrm{i}\omega\tilde{\eta}_1\left(\sqrt{\frac{\rho_1}{\tilde{G}_1}}\hat{u}_1^- - \sqrt{\frac{\rho_1}{\tilde{G}_1}}\hat{u}_1^+\right) = \mathrm{i}\omega\tilde{\eta}_2\left(\sqrt{\frac{\rho_2}{\tilde{G}_2}}\hat{u}_2^- - \sqrt{\frac{\rho_2}{\tilde{G}_2}}\hat{u}_2^+\right)$$

$$\sqrt{\tilde{G}_1\rho_1}\left(\hat{u}_1^- - \hat{u}_1^+\right) = \sqrt{\tilde{G}_2\rho_2}\left(\hat{u}_2^- - \hat{u}_2^+\right)$$

$$\tilde{Z}_1\left(\hat{u}_1^- - \hat{u}_1^+\right) = \tilde{Z}_2\left(\hat{u}_2^- - \hat{u}_2^+\right)$$

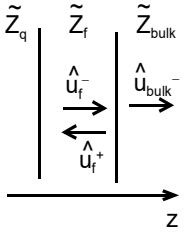

Figure 13
Geometry and variables underlying Eq. 13 – Eq. 19

The relations $\tilde{k} = \omega/\tilde{c}$, $\tilde{c} = (\tilde{G}/\rho)^{1/2}$, $\tilde{G} = \mathrm{i}\omega\tilde{\eta}$, 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

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

Combination of Eq. 13 and Eq. 21 leads to

$$\mathrm{I}: \left(\hat{u}_1^- - \hat{u}_1^+\right) = \frac{\tilde{Z}_2}{\tilde{Z}_1}\left(\hat{u}_2^- - \hat{u}_2^+\right) \qquad \text{Eq. 15}$$

$$\mathrm{II}: \hat{u}_1^- + \hat{u}_1^+ = \hat{u}_2^- + \hat{u}_2^+$$

$$\mathrm{I{+}II}: 2\hat{u}_1^- = \left(\frac{\tilde{Z}_2}{\tilde{Z}_1} + 1\right)\hat{u}_2^- + \left(-\frac{\tilde{Z}_2}{\tilde{Z}_1} + 1\right)\hat{u}_2^+$$

$$\mathrm{I-II}: -2\hat{u}_1^+ = \left(\frac{\tilde{Z}_2}{\tilde{Z}_1} - 1\right)\hat{u}_2^- + \left(-\frac{\tilde{Z}_2}{\tilde{Z}_1} - 1\right)\hat{u}_2^+$$

$$\begin{pmatrix}\hat{u}_1^+ \\ \hat{u}_1^-\end{pmatrix} = \frac{1}{2}\begin{bmatrix} 1+\frac{\tilde{Z}_2}{\tilde{Z}_1} & 1-\frac{\tilde{Z}_2}{\tilde{Z}_1} \\ 1-\frac{\tilde{Z}_2}{\tilde{Z}_1} & 1+\frac{\tilde{Z}_2}{\tilde{Z}_1}\end{bmatrix}\begin{pmatrix}\hat{u}_2^+ \\ \hat{u}_2^-\end{pmatrix} = P_{\text{interf}}\begin{pmatrix}\hat{u}_2^+ \\ \hat{u}_2^-\end{pmatrix}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

The perturbation analysis modifies this result to:

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

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

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

Taylor expansion to 1st order in the mass per unit area yields

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

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".[10] 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:[11]

$$\frac{\Delta f}{f_0} = \frac{-Z''_{liq,eff}}{\pi Z_q} \approx \frac{-1}{\pi Z_q}\sqrt{\frac{\rho_{liq}\omega\eta_{liq}}{2}}\left(1+3\sqrt{\pi}\frac{h_r^2}{l_r\delta}-2\frac{h_r^2}{\delta^2}\right) \qquad \text{Eq. 25}$$

$$\frac{\Delta\Gamma}{f_0} = \frac{Z'_{liq,eff}}{\pi Z_q} \approx \frac{1}{\pi Z_q}\sqrt{\frac{\rho_{liq}\omega\eta_{liq}}{2}}\left(1+2\frac{h_r^2}{\delta^2}\right)$$

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, η_{liq} is replaced by its complex analog, $\tilde{\eta}_{liq}$.

5.3 *Perturbation analysis*

The equations can be abbreviated by the use of the following variables:

Reduced mass: $\mu_e = m_e/m_q$, $\mu_f = m_f/m_q$
Reduced shear-wave impedance

$$\zeta_e(\omega) = \frac{Z_q^2}{\tilde{Z}_e^2(\omega)} - 1, \quad \zeta_f(\omega) = \frac{Z_q^2}{\tilde{Z}_f^2(\omega)} - 1, \quad \xi_{liq}(\omega) = \frac{\tilde{Z}_{liq}(\omega)}{Z_q} \qquad \text{Eq. 26}$$

Index *e*: first layer (“electrode”)
Index *f*: second layer (“film”)
Index *liq*: bulk medium (“liquid”)

For brevity, the reduced shear-wave impedance was written without the tilde.

5.3.1 Semi-infinite liquid

Eq. 27

SLA result: $\frac{\Delta\tilde{f}}{f_0} = \frac{\mathrm{i}}{\pi Z_q}\tilde{Z}_{liq}$

3rd order: $\frac{\Delta\tilde{f}}{f_{ref}} = \frac{\mathrm{i}}{n\pi}\left(\xi_{liq} + \frac{1}{3}\xi_{liq}^3\right)$

5th order: $\frac{\Delta\tilde{f}}{f_{ref}} = \frac{\mathrm{i}}{n\pi}\left(\xi_{liq} + \frac{1}{3}\xi_{liq}^3 + \frac{1}{5}\xi_{liq}^5\right)$

5.3.2 Viscoelastic film in air

Eq. 28

SLA result: $\frac{\Delta\tilde{f}}{f_0} = \frac{-\tilde{Z}_f}{\pi Z_q}\tan\left(\tilde{k}_f d_f\right)$

3rd order: $\frac{\Delta\tilde{f}}{f_{ref}} = -\mu_f + \mu_f^2 - \left(1 + \frac{1}{3}(n\pi)^2\zeta_f\right)\mu_f^3$

5th order:

$$\frac{\Delta\tilde{f}}{f_{ref}} = -\mu_\mathrm{f} + \mu_\mathrm{f}^2 - \left(1 + \frac{(n\pi)^2}{3}\zeta_\mathrm{f}\right)\mu_\mathrm{f}^3 + \left(1 + \frac{4(n\pi)^2}{3}\zeta_\mathrm{f}\right)\mu_\mathrm{f}^4 - \left(1 + \frac{10(n\pi)^2}{3}\zeta_\mathrm{f} + \frac{(n\pi)^4}{15}(1 - 2\zeta_\mathrm{f})\zeta_\mathrm{f}\right)\mu_\mathrm{f}^5$$

5.3.3 Viscoelastic film in liquid

SLA result: $$\frac{\Delta\tilde{f}}{f_0} = \frac{-\tilde{Z}_f}{\pi Z_q}\frac{\tilde{Z}_f \tan\left(\tilde{k}_f d_f\right) - \mathrm{i}\tilde{Z}_{liq}}{\tilde{Z}_f + \mathrm{i}\tilde{Z}_{liq}\tan\left(\tilde{k}_f d_f\right)} - \frac{\mathrm{i}\tilde{Z}_{liq}}{\pi Z_q}$$ Eq. 29

3rd order:

$$\frac{\Delta\tilde{f}}{f_{ref}} = \frac{\mathrm{i}}{n\pi}\left(\xi_{liq} + \frac{1}{3}\xi_{liq}^3\right) - \left(1 + \frac{\mathrm{i}\xi_{liq}}{n\pi} - \zeta_f\xi_{liq}^2\right)\mu_f + \left(1 + \left(\frac{\mathrm{i}}{n\pi} + \mathrm{i}n\pi\zeta_f\right)\xi_{liq}\right)\mu_f^2 - \left(1 + \frac{(n\pi)^2}{3}\zeta_f\right)\mu_f^3$$

5th order

$$\begin{aligned}\frac{\Delta\tilde{f}}{f_{ref}} = {} & \frac{\mathrm{i}\xi_{\mathrm{liq}}}{n\pi} + \frac{\mathrm{i}\xi_{\mathrm{liq}}^3}{3n\pi} + \frac{\mathrm{i}\xi_{\mathrm{liq}}^5}{5n\pi} + \left(-1 - \frac{\mathrm{i}\xi_{\mathrm{liq}}}{n\pi} + \zeta_{\mathrm{f}}\xi_{\mathrm{liq}}^2 + \frac{\left(-5\mathrm{i} + 15\mathrm{i}\zeta_{\mathrm{f}}\right)\xi_{\mathrm{liq}}^3}{15n\pi} + \zeta_{\mathrm{f}}\xi_{\mathrm{liq}}^4\right)\mu_{\mathrm{f}} + \\ & \left(1 + \left(\frac{\mathrm{i}}{n\pi} + \mathrm{i}n\pi\zeta_{\mathrm{f}}\right)\xi_{\mathrm{liq}} - 4\zeta_{\mathrm{f}}\xi_{\mathrm{liq}}^2 + \frac{\left(5\mathrm{i} - 45\mathrm{i}\zeta_{\mathrm{f}}\right)}{15n\pi} + \frac{\left(15\mathrm{i}\pi^2\zeta_{\mathrm{f}} - 15\mathrm{i}\pi^2\zeta_{\mathrm{f}}^2\right)}{15n\pi}\xi_{\mathrm{liq}}^3\right)\mu_{\mathrm{f}}^2 + \\ & \left(-1 - \frac{1}{3}n^2\pi^2\zeta_{\mathrm{f}} + \left(-\frac{\mathrm{i}}{n\pi} - 4\mathrm{i}n\pi\zeta_{\mathrm{f}}\right)\xi_{\mathrm{liq}} + \left(10\zeta_{\mathrm{f}} + \frac{n^2\left(-10\pi^3\zeta_{\mathrm{f}} + 20\zeta_{\mathrm{f}}^2\right)}{15\pi}\right)\xi_{\mathrm{L}}^2\right)\mu_{\mathrm{f}}^3 + \\ & \left(1 - \frac{4}{3}n^2\pi^2\zeta_{\mathrm{f}} + \left(\frac{\mathrm{i}}{n\pi} + 10\mathrm{i}n\pi\zeta_{\mathrm{f}} + \frac{n^3\left(-5\pi^4\zeta_{\mathrm{f}} + 10\pi^4\zeta_{\mathrm{f}}^2\right)}{15\pi}\right)\xi_{\mathrm{liq}}\right)\mu_{\mathrm{f}}^3 + \\ & \left(-1 - \frac{10}{3}n^2\pi^2\zeta_{\mathrm{f}} - \frac{1}{15}n^4\pi^4\zeta_{\mathrm{f}}\left(-1 + 2\zeta_{\mathrm{f}}\right)\right)\mu_{\mathrm{f}}^5\end{aligned}$$

5.3.4 Two viscoelastic films in air

SLA-Result:

$$\frac{\Delta\tilde{f}}{f_0} = \frac{-1}{\pi Z_q}\frac{\tilde{Z}_f \tan\left(\tilde{k}_f d_f\right) + \tilde{Z}_e \tan\left(\tilde{k}_e d_e\right)}{1 - \tilde{Z}_f / \tilde{Z}_e \tan\left(\tilde{k}_f d_f\right)\tan\left(\tilde{k}_e d_e\right)}$$ Eq. 30

3rd order:

$$\frac{\Delta\tilde{f}}{f_{ref}} = -\mu_{\mathrm{e}} + \mu_{\mathrm{e}}^2 - \left(1 + \frac{(n\pi)^2}{3}\zeta_{\mathrm{e}}\right)\mu_{\mathrm{e}}^3 - \left(1 - 2\mu_{\mathrm{e}} + 3\left(1 + \frac{(n\pi)^2}{3}\zeta_{\mathrm{e}}\right)\mu_{\mathrm{e}}^2\right)\mu_{\mathrm{f}} + \left(1 - 3\left(1 + \frac{(n\pi)^2}{3}\zeta_{\mathrm{e}}\right)\mu_{\mathrm{e}}\right)\mu_{\mathrm{f}}^2 - \left(1 + \frac{(n\pi)^2}{3}\zeta_{\mathrm{f}}\right)\mu_{\mathrm{f}}^3$$ Eq. 31

5th order:

$$\frac{\Delta\tilde{f}}{f_{ref}} = -\mu_e + \mu_e^2 + \left(-1-\frac{1}{3}n^2\pi^2\zeta_e\right)\mu_e^3 + \left(1+\frac{4}{3}n^2\pi^2\zeta_e\right)\mu_e^4 +$$
$$\left(-1-\frac{10}{3}n^2\pi^2\zeta_e - \frac{1}{15}n^4\pi^4\zeta_e\left(-1+2\zeta_e\right)\right)\mu_e^5 +$$
$$\left(-1+2\zeta_e + \left(-3-n^2\pi^2\zeta_e\right)\mu_e^2 + \left(4+\frac{16}{3}n^2\pi^2\zeta_e\right)\mu_e^3 + \left(-5-\frac{50}{3}n^2\pi^2\zeta_e - \frac{1}{3}n^4\pi^4\zeta_e\left(-1+2\zeta_e\right)\right)\mu_e^4\right)\mu_f +$$
$$\left(1+\left(-3-n^2\pi^2\zeta_e\right)\mu_e + \left(6+8n^2\pi^2\zeta_e\right)\mu_e^2 + \left(-10-\frac{100}{3}n^2\pi^2\zeta_e - \frac{2}{3}n^4\pi^4\zeta_e\left(-1+2\zeta_e\right)\right)\mu_e^3\right)\mu_f^2 +$$
$$\left(-1-\frac{1}{3}n^2\pi^2\zeta_f + \left(4+\frac{4}{3}n^2\pi^2\left(3\zeta_e+\zeta_f\right)\right)\mu_e + \left(-10-\frac{1}{3}n^4\pi^4\zeta_e\left(-2+3\zeta_e+\zeta_f\right)\right) - \frac{10}{3}n^2\pi^2\left(9\zeta_e+\zeta_f\right)\mu_e^2\right)\mu_f^3 +$$
$$\left(1+\frac{4}{3}n^2\pi^2\zeta_f + \left(-5-\frac{1}{3}n^4\pi^4\zeta_e\left(-1+2\zeta_f\right) - \frac{10}{3}n^2\pi^2\left(3\zeta_e+2\zeta_f\right)\right)\mu_e\right)\mu_f^4 +$$
$$\left(-1-\frac{10}{3}n^2\pi^2\zeta_f - \frac{1}{15}n^4\pi^4\zeta_f\left(-1+2\zeta_f\right)\right)\mu_f^5$$

Eq. 32

5.3.5 Two viscoelastic films in a liquid

SLA-Result:

$$\frac{\Delta\tilde{f}}{f_0} = \frac{-Z_e}{\pi Z_q}\frac{\tilde{Z}_f\left(\tilde{Z}_e\tan\left(\tilde{k}_e d_e\right) + \tilde{Z}_f\tan\left(\tilde{k}_f d_f\right)\right) + \mathrm{i}\,\tilde{Z}_{liq}\left(\tilde{Z}_e\tan\left(\tilde{k}_f d_f\right)\tan\left(\tilde{k}_e d_e\right) - \tilde{Z}_f\right)}{\tilde{Z}_f\left(\tilde{Z}_e - \tilde{Z}_f\tan\left(\tilde{k}_f d_f\right)\tan\left(\tilde{k}_e d_e\right)\right) + \mathrm{i}\,\tilde{Z}_{liq}\left(\tilde{Z}_e\tan\left(\tilde{k}_f d_f\right) + \tilde{Z}_f\tan\left(\tilde{k}_e d_e\right)\right)} - \frac{\mathrm{i}\tilde{Z}_{liq}}{\pi Z_q}$$

Eq. 33

3rd order:

$$\frac{\Delta\tilde{f}}{f_{ref}} = \frac{\mathrm{i}}{n\pi}\left(\xi_{liq} + \frac{1}{3}\xi_{liq}^3\right) - \left(1+\frac{\mathrm{i}}{n\pi}\xi_{liq} - \zeta_e\xi_{liq}^2\right)\mu_e + \left(1+\left(\frac{\mathrm{i}}{n\pi}+\mathrm{i}\,n\pi\zeta_e\right)\xi_{liq}\right)\mu_e^2$$
$$-\left(1+\frac{1}{3}(n\pi)^2\zeta_e\right)\mu_e^3$$
$$-\left(1+\frac{\mathrm{i}}{n\pi}\xi_{liq} - \zeta_f\xi_{liq}^2 - 2\left(1+\left(\frac{\mathrm{i}}{n\pi}+\mathrm{i}\,n\pi\zeta_e\right)\xi_{liq}\right)\mu_e + 3\left(1+\frac{1}{3}(n\pi)^2\zeta_e\right)\mu_e^2\right)\mu_f$$
$$+\left(1+\left(\frac{\mathrm{i}}{n\pi}+in\pi\zeta_f\right)\xi_e - 3\left(1+\frac{1}{3}(n\pi)^2\zeta_e\right)\mu_e\right)\mu_f^2$$
$$-\left(1+\frac{1}{3}(n\pi)^2\zeta_f\right)\mu_f^3$$

Eq. 34

5th order:

$$\frac{\Delta\tilde{f}}{f_{ref}} = \frac{\mathrm{i}\,\xi_{liq}}{n\pi} + \frac{\mathrm{i}\,\xi_{liq}^3}{3n\pi} + \frac{\mathrm{i}\,\xi_{liq}^5}{5n\pi} + \left(-1 - \frac{\mathrm{i}\,\xi_{liq}}{n\pi} + \zeta_e\xi_{liq}^2 + \frac{\left(-5\mathrm{i}+15\mathrm{i}\,\zeta_e\right)\xi_{liq}^3}{15n\pi} + \zeta_e\xi_{liq}^4\right)\mu_f +$$

$$\left(1 + \left(\frac{\mathrm{i}}{n\pi} + \mathrm{i}\,n\pi\zeta_e\right)\xi_{liq} - 4\zeta_e\xi_{liq}^2 + \left(\frac{5\mathrm{i}-45\mathrm{i}\,\zeta_e}{15n\pi} + \frac{n\left(15\mathrm{i}\,\pi^2\zeta_e - 15\mathrm{i}\,\pi^2\zeta_e^2\right)}{15n\pi}\right)\xi_{liq}^3\right)\mu_e^2 +$$

$$\left(-1 - \frac{1}{3}n^2\pi^2\zeta_e + \left(-\frac{\mathrm{i}}{n\pi} - 4\mathrm{i}\,n\pi\zeta_e\right)\xi_{liq} + \left(10\zeta_e + \frac{n^2\left(-10\pi^3\zeta_e + 20\pi^3\zeta_e^2\right)}{15\pi}\right)\xi_{liq}^2\right)\mu_e^3 +$$

$$\left(1 - \frac{4}{3}n^2\pi^2\zeta_e + \left(\frac{\mathrm{i}}{n\pi} + 10\mathrm{i}\,n\pi\zeta_e + \frac{1}{3}\mathrm{i}\,n^3\pi^3\zeta_e\left(-1+2\zeta_e\right)\right)\xi_{liq}\right)\mu_e^4 +$$

$$\left(-1 - \frac{10}{3}n^2\pi^2\zeta_e - \frac{1}{15}n^4\pi^4\zeta_e\left(-1+2\zeta_e\right)\right)\mu_f^5 +$$

$$\left(-1 - \frac{\mathrm{i}\,\xi_{liq}}{n\pi} + \zeta_f\xi_{liq}^2 + \frac{\left(-5\mathrm{i}+15\mathrm{i}\,\zeta_f\right)\xi_{liq}^3}{15n\pi} + \zeta_f\xi_{liq}^4 +\right.$$

$$\left(2 + \left(\frac{2\mathrm{i}}{n\pi} + 2\mathrm{i}\,n\pi\zeta_e\right)\xi_{liq} + \frac{\left(-90\pi\zeta_e - 30\pi\zeta_f\right)\xi_{liq}^2}{15\pi} + \left(\frac{10\mathrm{i} - 60\mathrm{i}\,\zeta_e - 30\mathrm{i}\,\zeta_f}{15n\pi} - 2\mathrm{i}\,n\pi\zeta_e\left(-1+\zeta_f\right)\right)\xi_{liq}^3\right)\mu_e +$$

$$\left(-3n^2\pi^2\zeta_e + \left(-\frac{3i}{n\pi} - 12\mathrm{i}\,n\pi\zeta_e\right)\xi_{liq} + \left(\frac{n^2\left(45\pi^3\zeta_e^2 + 15\pi^2\zeta_e\left(-2+\zeta_f\right)\right)}{15\pi} + \frac{405\pi\zeta_e + 45\zeta_f}{15\pi}\right)\xi_{liq}^2\right)\mu_e^2 +$$

$$\left(4 + \frac{16}{3}n^2\pi^2\zeta_e + \left(\frac{4\mathrm{i}}{n\pi} + 40\mathrm{i}\,n\pi\zeta_e + \frac{4}{3}\mathrm{i}\,n^3\pi^3\zeta_e\left(-1+2\zeta_e\right)\right)\xi_{liq}\right)\mu_e^3 + \left(-5 - \frac{50}{3}n^2\pi^2\zeta_e - \frac{1}{3}n^4\pi^4\zeta_e\left(-1+2\zeta_e\right)\right)\mu_e^4\Big)\mu_f +$$

$$\left(1 + \left(-\frac{\mathrm{i}}{n\pi} + \mathrm{i}\,n\pi\zeta_f\right)\xi_{liq} - 4\zeta_f\xi_{liq}^2 + \left(\frac{5\mathrm{i}-45\mathrm{i}\,\zeta_f}{15n\pi} + \frac{n\left(15\mathrm{i}\,\pi^2\zeta_f - 15\mathrm{i}\,\pi^2\zeta_f^2\right)}{15\pi}\right)\xi_{liq}^3 +\right.$$

$$\left(-3 - n^2\pi^2\zeta_e + \left(-\frac{3\mathrm{i}}{n\pi} - 3\mathrm{i}\,n\pi\zeta_e\left(3\zeta_e+\zeta_f\right)\xi_{liq} + 2n^2\pi^2\zeta_e\left(-1+2\zeta_f\right) + \frac{270\pi\zeta_e + 180\zeta_f}{15\pi}\right)\xi_{liq}^2\right)\mu_e +$$

$$\left(6 + 8n^2\pi^2\zeta_e + \left(\frac{6\mathrm{i}}{n\pi} + \mathrm{i}\,n^3\pi^3\zeta_e\left(-2+3\zeta_e+\zeta_f\right) + \frac{n\left(810\mathrm{i}\,\pi^2\zeta_e + 90\mathrm{i}\,\pi^2\zeta_f^2\right)}{15\pi}\right)\xi_{liq}\right)\mu_e^2 +$$

$$\left(-10 + \frac{100}{3}n^2\pi^2\zeta_e + \frac{2}{3}n^4\pi^4\zeta_e\left(-1+2\zeta_e\right)\right)\mu_f^2 +$$

$$\left(-1 - \frac{1}{3}n^2\pi^2\zeta_f + \left(-\frac{\mathrm{i}}{n\pi} - 4\mathrm{i}\,n\pi\zeta_f\right)\xi_{liq} + \left(10\zeta_f + \frac{n^2\left(-10\pi^3\zeta_f + 20\pi^3\zeta_f^2\right)}{15\pi}\right)\xi_{liq}^2 \quad +\right.$$

$$\left(1 - \frac{4}{3}n^2\pi^2\left(3\zeta_e+\zeta_f\right) + \left(\frac{4\mathrm{i}}{n\pi} + \frac{4}{3}\mathrm{i}\,n^3\pi^3\zeta_e\left(-1+2\zeta_f\right) + 8\mathrm{i}\,n\pi\left(3\zeta_e+2\zeta_f\right)\right)\xi_{liq}\right)\mu_e +$$

$$\left(-10 + \frac{1}{3}n^4\pi^4\zeta_e\left(-2+3\zeta_e+2\zeta_f\right) - \frac{10}{3}n^2\pi^2\left(9\zeta_e+\zeta_f\right)\right)\mu_e^2\Big)\mu_f^3 +$$

$$\left(1 + \frac{4}{3}n^2\pi^2\zeta_f + \left(\frac{\mathrm{i}}{n\pi} + 10\mathrm{i}\,n\pi\zeta_f + \frac{n^2\left(-5\mathrm{i}\,n\pi^4\zeta_f + 10\pi^4\zeta_f^2\right)}{15\pi}\right)\xi_{liq} + \left(5 - \frac{1}{3}n^4\pi^4\zeta_e\left(-1+2\zeta_f\right) - \frac{10}{3}n^2\pi^2\left(3\zeta_e+2\zeta_f\right)\right)\mu_e\right)\mu_f^4 +$$

$$\left(1 - \frac{10}{3}n^2\pi^2\zeta_f - \frac{1}{15}n^4\pi^4\left(-1+2\zeta_f\right)\right)\mu_f^5$$

Eq. 35

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	$Z_q = 8.8 \cdot 10^6$ kg m^{-2} s^{-1}
β', β''	Power law exponents	Eq. 5
Γ	Imaginary part of a resonance frequency, Half-band half width	
δ	Penetration depth of a shear wave	Newtonian liquids: $\delta = (2\eta_{liq}/(\rho_{liq}\omega))^{1/2}$
δ_L	Loss angle	$\tan\delta_L = G''/G' = J''/J'$ often called δ in rheology
Δ	As a prefix: A shift induced by the sample	
$\tilde{\eta}$	Viscosity	$\tilde{\eta} = \tilde{G}/(i\omega)$
μ	Non-dimensional mass	Eq. 26
ξ_{liq}	Non-dimensional shear-wave impedance of the bulk liquid	Eq. 26
ζ	Non-dimensional inverse square shear-wave impedance	Eq. 26

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] Attributed to Einstein by Roger Sessions in a New York Times article from Jan 8, 1950.

[6] 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.

[7] Johannsmann, D., Viscoelastic analysis of organic thin films on quartz resonators. *Macromolecular Chemistry and Physics* **1999,** 200, (3), 501-516.

[8] 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.

[9] 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.
[10] Voinova, M. V.; Jonson, M.; Kasemo, B., 'Missing mass' effect in biosensor's QCM applications. *Biosensors & Bioelectronics* **2002,** 17, (10), 835-841.
[11] 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. *Analytical Chemistry* **2002,** 74, (3), 554-561