A Python-based Measurement Model Toolbox for Impedance Spectroscopy

William Watson and Mark Orazem
Department of Chemical Engineering
University of Florida
Gainesville, FL, 32611

April 2020

Table of Contents

List of Figures

List of Tables

Preface

The measurement model described in this guide – and this guide itself – are based on one developed by Dr. Mark Orazem (Department of Chemical Engineering, University of Florida). Work on the original measurement model began in 1989 with a program written primarily in MATLAB, with some Fortran code backend. The new code was written from the ground-up in Python, although some portions of the backend libraries, such as `numpy`, use C.

The objective of the work was to find a way to account for measurement errors in the analysis of electrochemical impedance data. Collaborators in this effort have been Professors Luis García-Rubio (Department of Chemical Engineering, University of South Florida) and Oscar Crisalle (Department of Chemical Engineering, University of Florida), Drs. Bernard Tribollet and Claude Deslouis (Laboratoire de Physique des Liquides et Électrochimie, UPR15 du CNRS, Université Pierre et Marie Curie, Paris), and students Pankaj Agarwal, Madhav Durbha, Steven Carson, and Pavan Shukla. This work has also benefited from contributions from Hisasi Takenouti, Jose Carlos Cardoso Filho, Kenneth Jeffers, Paul Wojcik, Michael Membrino, Douglas Riemer, Touriya El Moustafid, Hercilio G. de Melo, and Isabelle Frateur. This work was done in support of research projects supported by the U.S. Office of Naval Research, The National Science Foundation, the Centre National de la Recherche Scientifique, Alza Corporation, and S. C. Johnson and Son, Inc.

The concept which motivated development of the measurement model was that interpretation of impedance data requires both a model which describes the physics of the system under study and a quantitative assessment of the of the measurement. While the for most radiation-based spectroscopic measurements such as absorption spectroscopy and light scattering can be readily identified, the experimental difficulty of quantifying the error structure for electrochemical impedance spectroscopy had prevented application of an error analysis approach for interpretation of spectra. The error analysis approach was successful for some optical spectroscopy techniques because these systems lend themselves to replication and, therefore, to the independent identification of the different errors that contribute to the total variance of the measurements. In contrast, the stochastic contribution to the error structure of electrochemical impedance spectroscopy measurements cannot generally be obtained from the standard deviation of repeated measurements because even a mild non-stationary behavior introduces a significant time-varying bias contribution to the error.

The measurement model approach for identification of the error structure of electrochemical impedance spectroscopy was developed as an extension of that previously used

for optical spectroscopies. The measurement model was used as a filter for lack of replication to allow assessment of the stochastic contribution to the error structure from repeated measurements. Since the model satisfies the Kramers-Kronig relations, it could be used to identify portions of the spectrum that were inconsistent with these relations. This level of statistical information allows enhanced interpretation of impedance spectra. For example, using normal weighting strategies impedance spectra for n-GaAs diodes, information concerning deep-level states could not be obtained; whereas, evaluation of the error structure allowed interpretation in terms of concentrations and energy levels of deep-level electronic states. Similar improvement was demonstrated for interpretation of electrohydrodynamic impedance spectra in terms of transport properties. Such enhanced interpretation was possible because sophisticated process models could be used to interpret the measurements in terms of well-defined physical properties. Recently, an approach was devised to allow a statistically-based simplistic characterization of systems for which such detailed process models are unavailable.

This document is intended to provide an introductory guide to the measurement model toolbox. Some of the concepts which underlie the program are described in the references given at the end of the document.

William Watson and Mark E. Orazem
Gainesville, Florida

Chapter 1

Getting Started

The code comes packaged in Microsoft installer which, when executed, will place all the necessary files into the correct locations, assign default file types, and create shortcuts as desired. The program is written in Python and comes in an executable; the executable contains what is essentially a standalone Python interpreter. Although this causes the program to be relatively large, it also means that a separate Python installation is not necessary to make use of it. An overview of the application of the measurement model toolbox for statistical analysis of impedance data was presented in a series of papers by Agarwal *et al.*[1–3]

The program can be launched by running `Measurement Model.exe`. At first, a popup will appear displaying copyright information (see Figure 1.1). The software is licensed under the GNU General Public License; this can be accessed by clicking the link in the popup or by visiting https://www.gnu.org/licenses/. Press `Start program` or `Enter` to begin the program. Pressing `Cancel` or otherwise closing the alert will terminate the program. There are three main regions to the program layout (see Figure 1.2): the title bar (to the top), the navigation pane (to the left), and the main window (to the right).

The title bar serves to indicate the main purpose of the current tab, while clicking an icon in the navigation pane will switch to that particular tab. The current active tab icon is highlighted blue in the navigation pane. General information about a component in a tab can usually be found by hovering over that component for a few seconds and reading the resulting tooltip.

The program also makes use of six types of files (see Table 1.1). When the installer runs, the file types will be assigned and recieve icons; they can then be opened with the Measurement Model program. The file types will be discussed further as they arise.

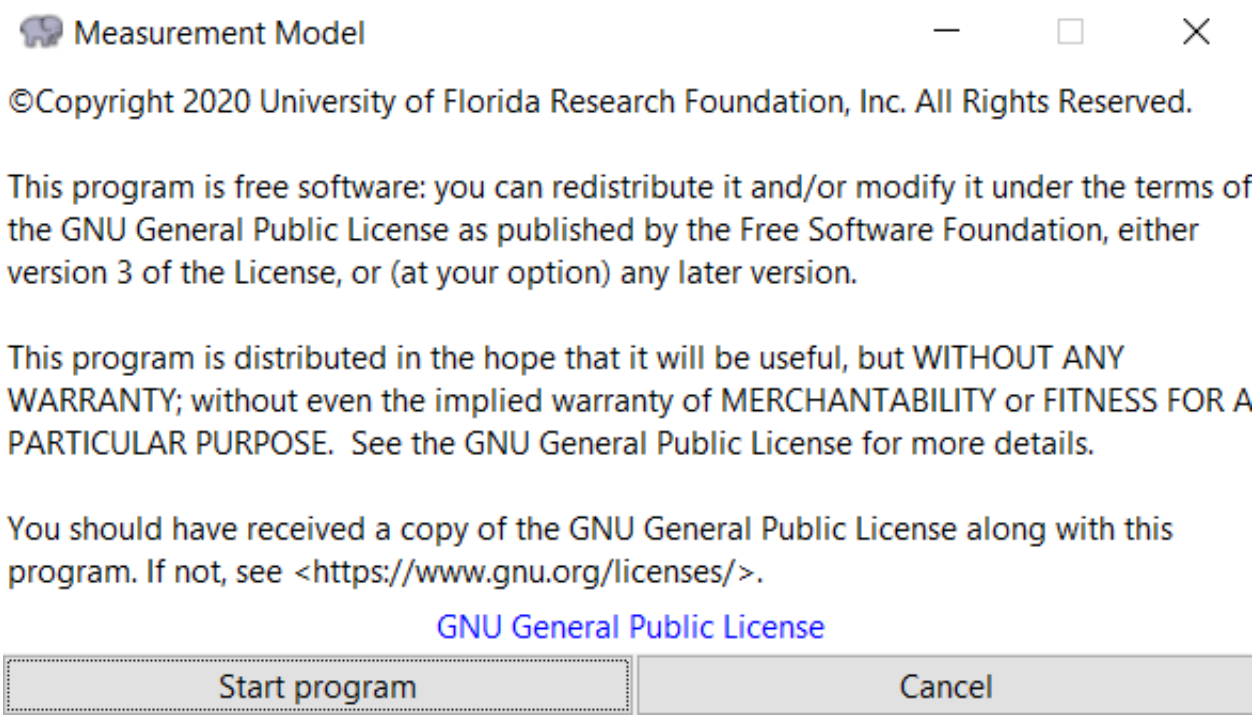


Figure 1.1: Initial copyright popup

Table 1.1: File Extensions used by the Program

Extension	Purpose	Image
mmfile	Holds impedance data (frequency, real, and imaginary parts)	
mmfitting	Holds information from a previous fit	
mmresiduals	Holds residual errors from a fit	
mmerrors	A combined set of *mmresiduals* files	
mmcustom	Holds a custom fitting	
mmformula	Holds a custom formula	

The program is broken up into seven tabs (see Table 1.2). Each tab will be discussed in detail in a following section.

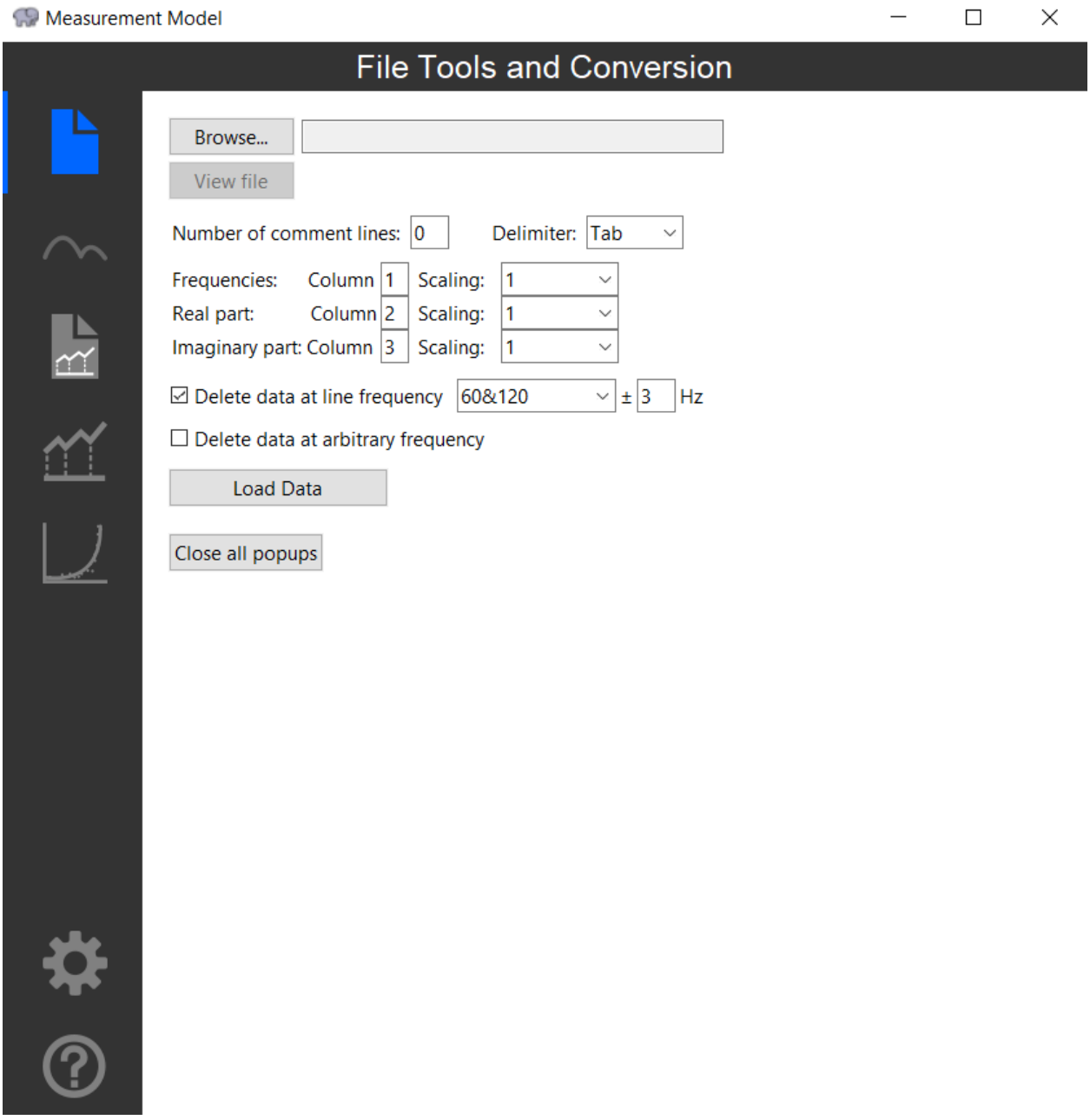


Figure 1.2: Program layout

Table 1.2: Program Tabs

Tab	Image	Purpose
File Tools and Conversion		Convert files to a program-usable format
Measurement Model		Fit data to the measurement model
Error File Preparation		Combine residual error files for analysis
Error Analysis		Fit errors to an error structure
Custom Formula Fitting		Allow for the fitting of custom functions
Settings		Change program preferences
Help and About		Program information and assistance

1.1 Close all popups

The `Close all popups` button appears at the bottom of every tab (except Help and About). When clicked, it will close all open popups created from all tabs. This can be useful to ”clean up” if numerous popups have been opened without closing.

Chapter 2

File Tools and Conversion

The first step towards analyzing any impedance spectra is to convert the file into a format the program will understand, namely a file with a *.mmfile* extension. These files are simple text files with three columns: the first is frequency, the second is real impedance, and the third is imaginary impedance. Creating these files is the job of the first tab: File Input and Conversion. Access this tab by clicking the document icon ( ) at the top of the navigation pane. Note that none of the tools in this tab will edit the original file. The input file tab can be seen in Figure 2.1.

2.1 Input

Files of any kind can be loaded by using the `Browse...` button. This will bring up the standard operating-system file chooser dialog, from whence a file can be chosen. Upon selecting a file, the name of this file will appear in the text bar to the right of the button. The contents of the file can be viewed using the `View file` button. This button is initially disabled and will only be clickable after a file has been chosen with the `Browse...` button. This will bring up a small popup with line numbers and the raw text of the file.

2.2 Number of Comments

The `Number of comment lines` box indicates the number of lines that the program ignores before the actual data begin. This number can be set manually; it can be helpful to use the `View file` button mentioned in Section 2.1 to determine on which line number the data start. The program will also attempt to determine the number of comment lines automatically. This is done by reading through the file and looking for the first line on which consistently delimited text data appears. This can be disabled/enabled in the Settings tab.

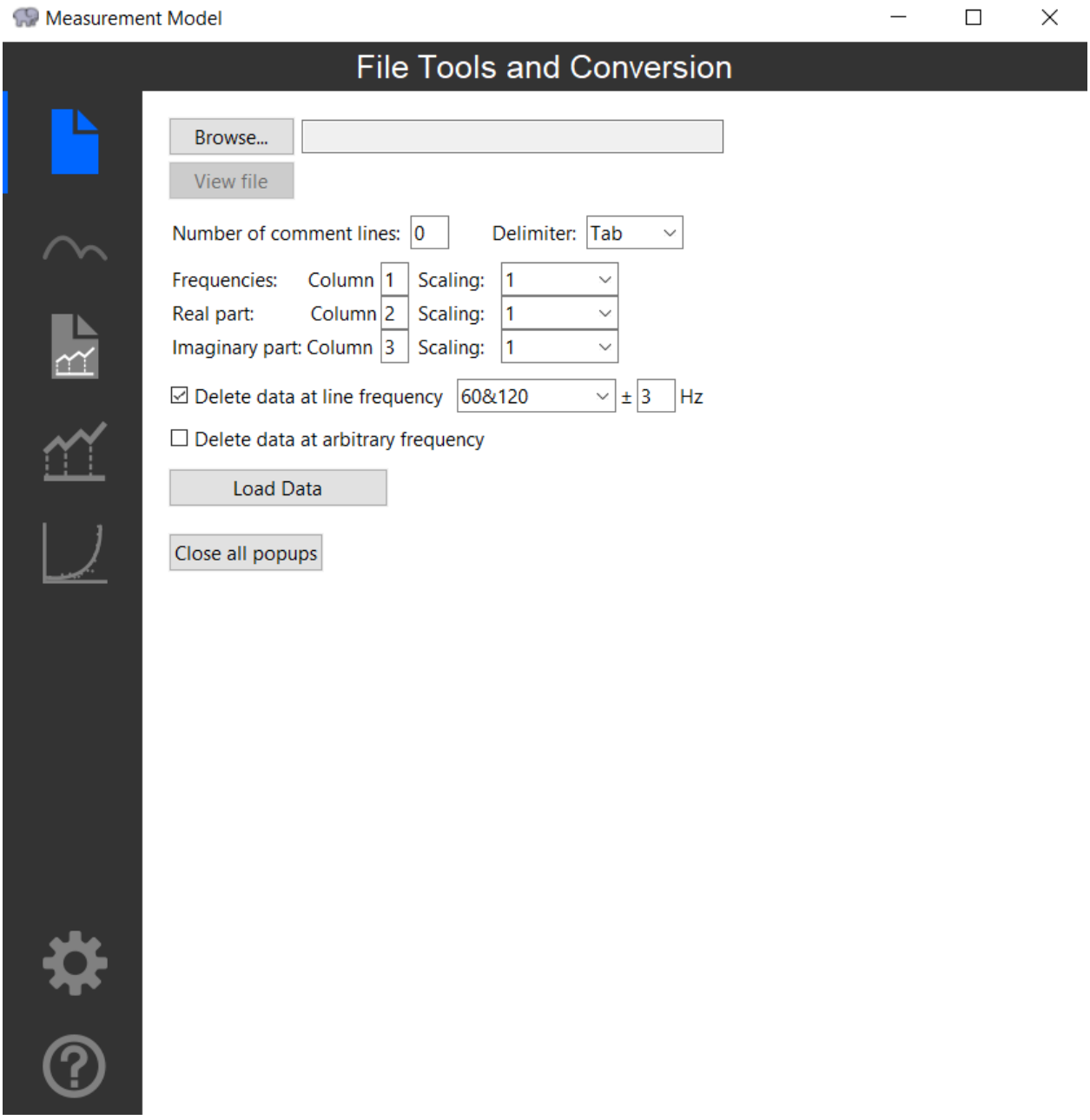


Figure 2.1: The input file tab

2.3 Delimiter

The `Delimiter` is the character used to indicate separation between columns. The delimiter can be chosen manually as one of:

- Tab
- Space
- Semicolon (;)
- Colon (:)
- Comma (,)
- Pipe (|)

As with the number of comment lines, the program will attempt to determine the delimiter automatically. This can be disabled in the Settings tab.

2.4 Recognized File Types

If automatic number of comments detection is enabled, the program will also attempt to automatically parse certain common file types. These are listed in Table 2.1.

Table 2.1: Recognized File Types

Company	Extension	Comments
AutoLab	.txt	
Biologic	.mpt	
Biologic	.mpr	This file is binary, not plain text; therefore, a simplified version is parsed and displayed
CH Instruments	.txt	
Gamry	.DTA	Can detect if experiment was aborted and parse appropriately
Parstat	.txt	
PowerSuite	.txt	
VersaStudio	.par	Will only detect data from the first segment
	.z	Can detect files with and without metadata

2.5 Columns

The program will look for data separated into three columns: one each for frequency, real impedance, and imaginary impedance. These columns should be separated using the delimiter mentioned in Section 2.3. The column numbers should be entered manually. The `Scaling` is the multiplier used on the input data. This defaults to `1` (i.e. the data will be the same as in the input file). Other choices are available from the dropdown menu, and arbitrary values can be entered by typing in the box as well (the value must a real number).

2.6 Deleting Data

Impedance measurements taken near the frequency of the power source can be noisy. Noise from the power source can be superimposed upon the input and output signals of the frequency response analyzer. Even one or two spuriously noisy data points near the line frequency can influence the regression of the impedance spectrum. To avoid this problem, data taken at frequencies close to the first and second harmonic of the power source frequency should be removed from the analysis. The line frequency in the US and Canada is 60 Hz; therefore, impedance measurements taken near 60 Hz and or 120 Hz should be deleted from the analysis of data collected in the US or Canada. The line frequency in Europe is 50 Hz; therefore, impedance measurements taken near 50 Hz and or 100 Hz should be deleted from the analysis. The `Delete data at line frequency` dropdown provides this functionality. The data at the selected frequency(ies) will be removed in the range specified by the $\pm$ box. For instance, if `50&100` $\pm$ `3` Hz is chosen, then all data from 47-53 and 97-103 Hz would be removed. The default is 60-120 $\pm$ 3.

Sometimes, data at different frequencies may need to be discarded. The `Delete data at arbitrary frequency` checkbox, when checked, will provide more options for doing so. To the left will be a listbox (initially empty). This will be propagated with frequencies as they are added. The first entry field to the right indicates the main frequency to be deleted, while the second entry field indicates the range around the first frequency to be deleted. For instance, entering `42` $\pm$ `5` would delete all data from 37-47 Hz. Click the `Add` button to add this deleted frequency to the listbox (see Figure 2.2).

To remove an item from the listbox (and hence have that frequency range *not* be deleted), simply select the item in the listbox and click the `Remove` button, or right-click the item and select `Remove`.

> **Tip!** When using auto-integration, measurements taken near the line frequency and its first harmonic (*e.g.*, , 60$\pm$5Hz and 120$\pm$5Hz) can take a long time to converge. To minimize the time required for the impedance measurement, design the experiment to avoid these frequencies rather than making measurements that will subsequently be discarded.

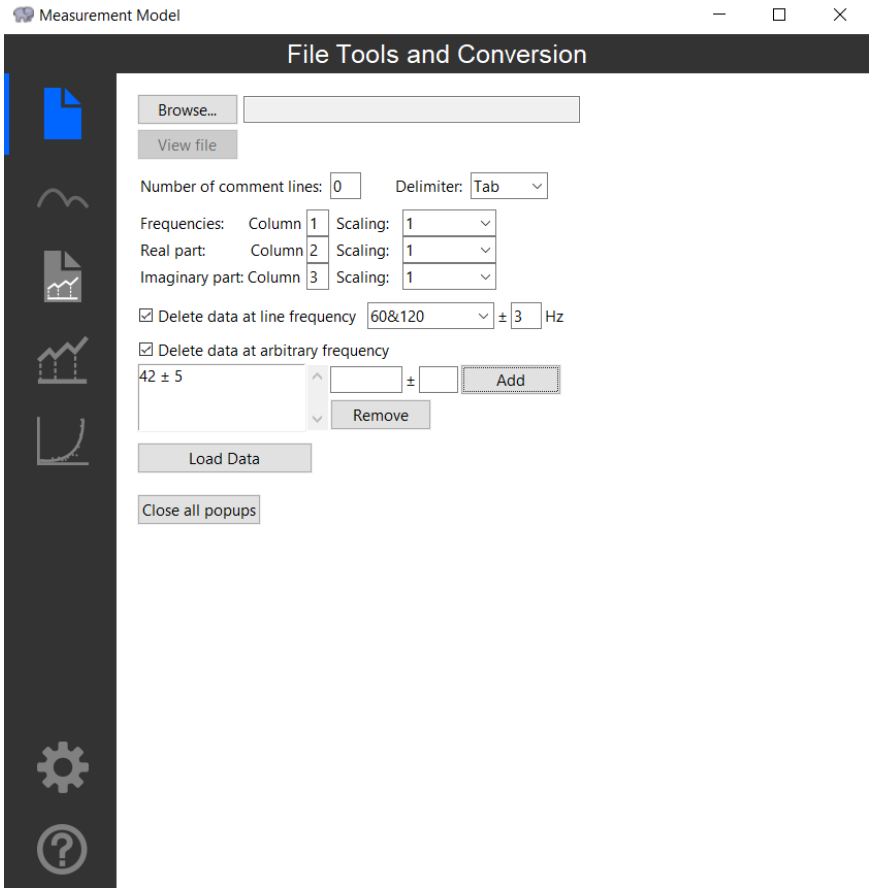


Figure 2.2: Deleting data at arbitrary frequency

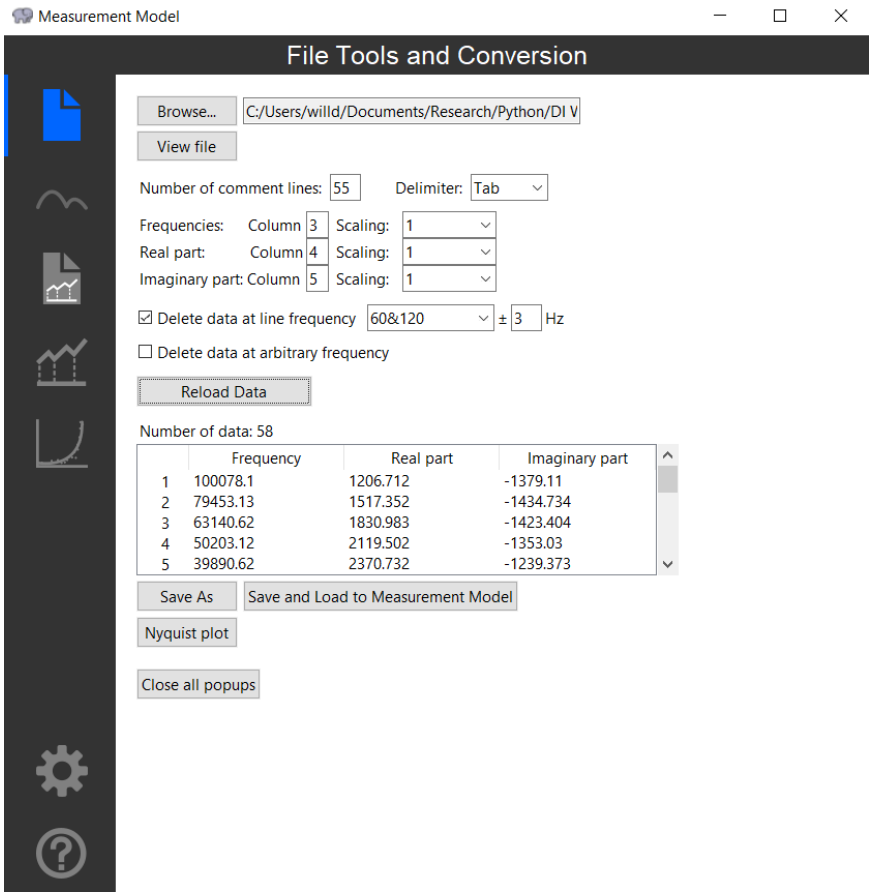


Figure 2.3: DI water data when loaded

2.7 Loading Data

After all the preceding choices have made, the data is ready to be loaded. Click the `Load data` button to do so. Once data is loaded, this button will be renamed to `Reload Data`. In order for changes made to the choices to have an effect, this button must be pressed.

When the data is loaded, a listbox will appear below with four columns (see Figure 2.3). Each row of the listbox holds one data point. The first column is the data point's "number", the second is the frequency, the third is the real part of the data, and the fourth is the imaginary part. Above this listbox will be the `Number of data`, which will indicate how many data points were loaded in.

Below the listbox are three buttons. One is `Save As`, which will save the data shown in the listbox to a *.mmfile*. The second is `Save and Load to Measurement Model`, which will save the data as a *.mmfile* and then open it in the second tab (see Chapter 3). The third is `Nyquist plot`, which will open a popup displaying a Nyquist plot of the data (see Figure 4). Every plot generated by the program will have the same controls, which are discussed in Section 2.8.

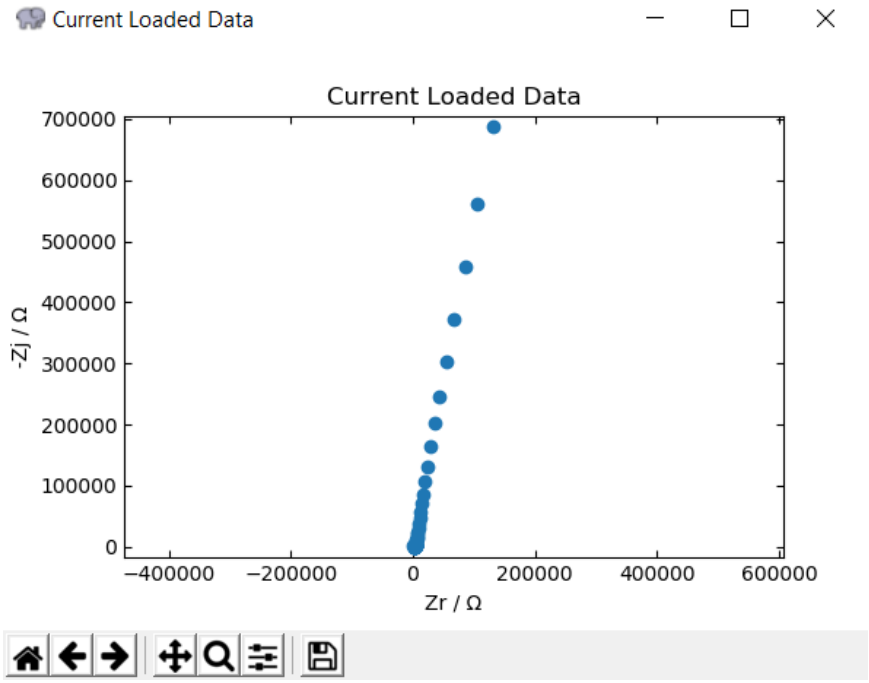


Figure 2.4: A Nyquist plot of DI water; the same layout is used for all plots in the program

2.8 Graphing Controls

Each plot generated by the program has the same controls, which are shown in the toolbar below a given plot (see Figure 2.4). The leftmost three buttons work similarly to Internet browser control buttons. The leftmost button (with a house) returns the view to its original state. The second button (a back arrow) changes the plot back to the state it was immediately before the current one, which the third button (a forwards arrow) changes the plot to a newer one (if the back button has been clicked). The fourth button (with two crossing double-headed arrows) allows the plot view to be moved by dragging with the left mouse button. Dragging with the right mouse button will cause the plot to be zoomed. The fifth button (a magnifying glass) allows a rectangular region of the plot to be selected; this area will then be zoomed to be the new view. Use the back or home buttons to zoom out, or the move button by dragging with the right mouse button. The sixth button is rarely used, if ever; further information can be found by looking up the `matplotlib` GUI control bar. The last button is the save button, which will open a popup to save the current figure as an image file.

Chapter 3

Measurement Model

After a *.mmfile* has been created using the File Tools and Conversion tab, it can be loaded into the Measurement Model tab to be fit to a measurement model consisting of a certain number of Voigt elements. Access this tab by clicking the Nyquist plot icon (◠) at the top of the navigation pane. Note that the tools in this tab will not edit the *.mmfile*. The measurement model tab can be seen in Figure 3.1 below.

3.1 About the Regression

The measurement model works by regressing a model for impedance to the actual impedance data. This model consists of an ohmic resistance in series with a number of Voigt elements – these Voigt elements consist of a resistor in parallel with a capacitor. The general structure for the measurement model to be regressed is seen in Equation 3.1.

$$\widehat{Z} = R_e\,(+C) + \sum_{i=1}^{n} \frac{R_i}{1 + j(2\pi f)R_iC_i} = R_e\,(+C) + \sum_{i=1}^{n} \frac{R_i}{1 + j(2\pi f)\tau_i} \tag{3.1}$$

R_e is the ohmic resistance, C is the (optional) capacitance, R_i is the i-th Voigt element's resistance, and τ_i is the i-th Voigt element's time constant (equivalent to R_iC_i). A visual depiction of this circuit can be seen in Figure 3.2.

The regression uses a Levenberg-Marquardt algorithm to minimize the sum of squared of the objective function seen in Equation 3.2.

$$\chi = \frac{Z_{data} - \widehat{Z}}{\sigma} \tag{3.2}$$

σ is the weighting, described in Section 3.7.

3.2 File Input

Files with two types of extensions can be loaded to the measurement model: *.mmfile* and *.mmfitting*. The first holds a single set of impedance measurements and is created via the

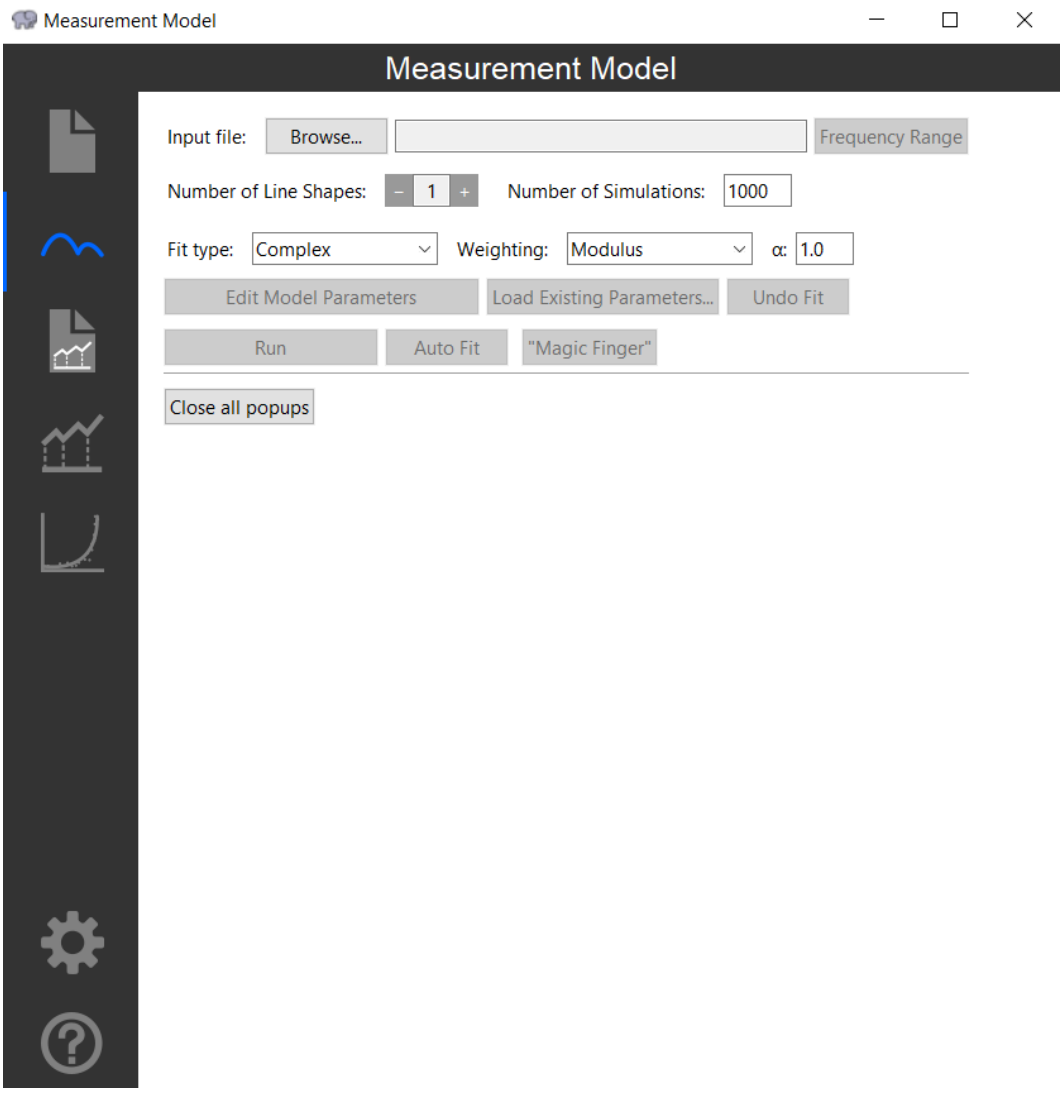


Figure 3.1: The measurement model tab

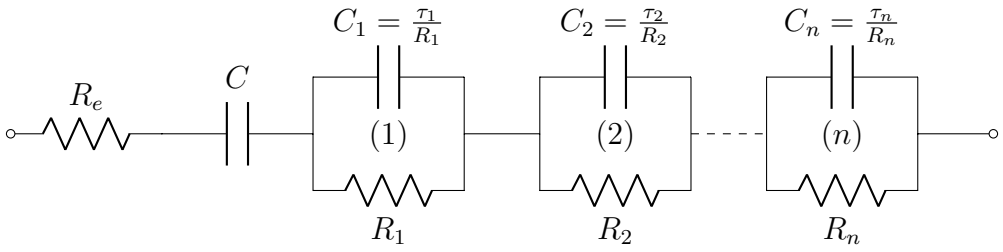


Figure 3.2: Measurement model circuit

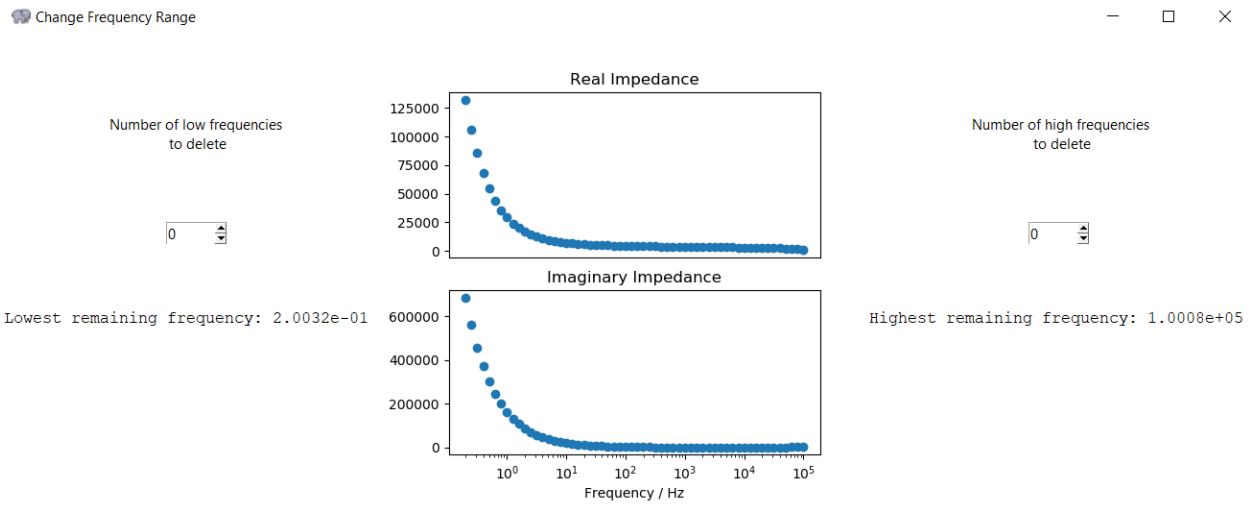


Figure 3.3: Changing the frequencies used for fitting

File Tools and Conversion tab (see Chapter 2), while the second holds a set of fitted parameters and link to an existing *.mmfile*; *.mmfitting* files can be created in the Measurement Model tab after a successful fitting is performed. The `Browse...` button in the top row can be used to open a dialog and select a file for use. After a file is chosen, its path will appear in the textbox to the right. For *.mmfile*s, the buttons below will be activated upon loading. If a *.mmfitting* file was chosen, the path will be the *.mmfile* used to create the fitting; in addition, fitting results will appear below.

3.3 Changing Frequencies

Depending on the specific data used, it may be necessary to remove high or low frequencies from the fit. This can be done by using the `Frequency Range` button. A popup similar to Figure 3.3 will appear to allow frequencies to be deleted.

The graph in the middle is used to visually indicate which frequencies have been removed; the upper graph is the real impedance vs frequency, while the lower is minus imaginary impedance vs frequency. The graph shows all frequencies present in the data set. Frequencies in use are shown as filled circles, while removed frequencies are shown as open circles. The arrows on the two spinboxes to either side can be used to change the number of low or high frequencies being removed. The lowest and highest remaining frequencies (i.e. the ones present after deletion) are shown below the spinboxes; these can be highlighted and copied via keyboard shortcuts. Frequency changes are automatically saved as they are made.

3.4 Number of Voigt Elements

The `Number of Line Shapes` box is used to change the number of Voigt elements being fit. The `−` button on the left is used to decrease the number of shapes (to a minimum

of 1) and the [+] button on the right is used to increase the number of shapes. These buttons will not be active until a file is loaded.

> **Tip!** The addition of Voigt elements is sequential. For a new data set of a given type, it is best to start with one Voigt element and to add more elements sequentially until the confidence interval for one or more parameters includes zero.

3.5 Number of Simulations

The confidence intervals used in the plots are generated through Monte Carlo simulations. The `Number of Simulations` entry field accepts positive integers only and is used to indicate how many Monte Carlo simulations should be performed. A greater number will produce a better/more accurate result, but at the cost of taking more time. The random number generator used in the simulations is seeded with a constant value of `1234`, which ensures that the exact same fitting with the same number of simulations will result in the same confidence intervals. The default number of simulations is 1000.

3.6 Fit type

The portion of the data to which the model is fit can be chosen using the `Fit type` dropdown. The options are:

- `Real` - fit to the real part of the data
- `Imaginary` - fit to the imaginary part of the data (cannnot not fit the ohmic resistance)
- `Complex` - fits to both the real and imaginary parts fo the data

The default choice is complex.

3.7 Weighting

Impedance data may span several orders of magnitude. Thus, it is necessary to weight the regression to obtain a decent fit. This can be done by using the `Weighting` dropdown. There are four options for weighting as seen in Table 3.1. The default weighting is `Modulus`.

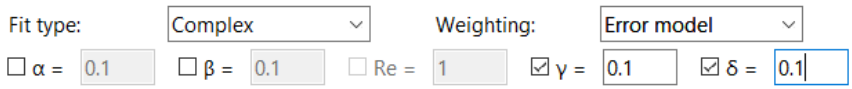


Figure 3.4: Error structure weighting options

Table 3.1: Weighting options for regression

Weighting	Description	Formula
None	No weighting is used	$\sigma_r = \sigma_j = 1$
Modulus	Each frequency is weighted by the modulus of the impedance	$\sigma_r = \sigma_j = \alpha\sqrt{Z_r^2 + Z_j^2}$
Proportional	Real and imaginary values are weighted by real and imaginary impedances, respectively	$\sigma_r = \alpha Z_r$ $\sigma_j = \alpha Z_j$
Error model	Each frequency is weighted by the error structure (see Chapter 5)	$\sigma_r = \sigma_j = \alpha\lvert Z_j\rvert + \beta\lvert Z_r - R_e\rvert + \gamma\lvert Z\rvert^2 + \delta$

Tip! Modulus weighting provides a good estimate for the error structure of impedance data obtained under potentiostatic modulation or under variable-amplitude galvanostatic modulation. The constant weighting strategy (no weighting) is better for impedance data obtained under fixed-amplitude galvanostatic modulation. Once a refined model for the error structure has been determined, it should be used for subsequent regression analysis.

3.7.1 Alpha

The parameter α is the "assumed noise" of the fit and is multiplied by the chosen weighting. It is available only for modulus and proportional weightings.

3.7.2 Error Model

Error model weighting uses the error structure (see Chapter 5) as weighting. Choosing this option from the dropdown will cause another row to appear with further options. (see Figure 3.4).

Each box can be checked to use it in the error structure. Unchecking a box will cause its entry field to be disabled and will remove it from the error structure. The R_e option is only available if β is chosen as well. If R_e is chosen, the value entered will be used as a constant. In other words, the value entered must be an *a priori* estimate for the ohmic resistance as the actual value found through regression may differ from it. All values must be entered as real numbers.

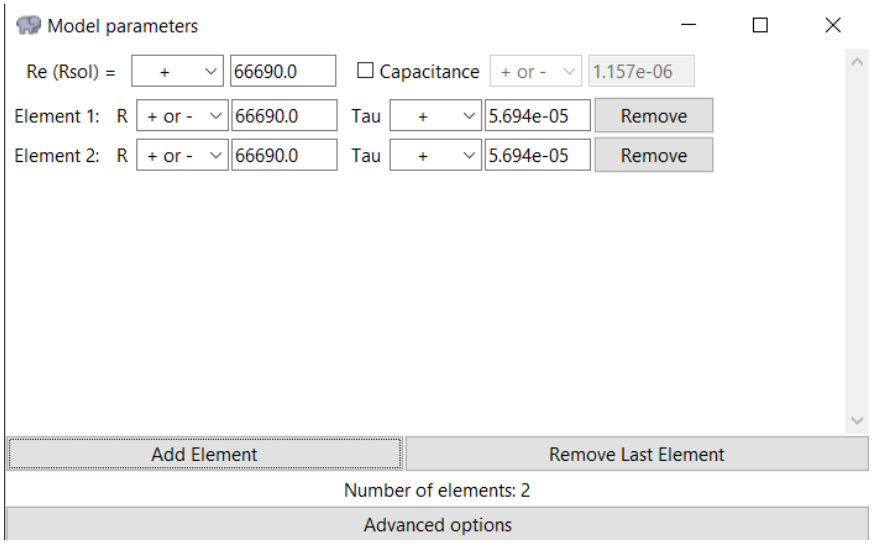


Figure 3.5: Model parameter options

3.8 Edit Model Parameters

The `Edit Model Parameters` button will open a popup (as seen in Figure 3.5) that allows for the current parameters to be adjusted prior to executing a fit. This button is not active until a file has been loaded.

Each parameter is listed with a dropdown and an entry field, which are described in Sections 3.8.1 and 3.8.2. When more than one element is present, a `Remove` button will appear next to each element; clicking this will remove that particular one. Checking the box labelled `Capacitance` will add a capacitance in series with the Voigt elements; this is the optional capacitance seen in Equation 3.1. The `Add Element` and Remove Last Element buttons will add or remove Voigt elements in exactly the same way as the + and − buttons described in Section 3.4. The Number of elements label below these buttons is simply a count of the number of Voigt elements present. The button at the very bottom is `Advanced options`, which will open another popup, which will be described in Section 3.9.

3.8.1 Parameter Bounds

The dropdown by each variable allows limits to be placed on that parameter. There are four options, as seen in Table 3.2.

Table 3.2: Parameter bounds

Option	Meaning
`+`	Force the parameter to be positive
`-`	Force the parameter to be negative
`+ or -`	No bounds on the parameter
`fixed`	Don't vary the parameter (i.e. do not fit)

Negative and no bounds (`-` and `+ or -`) are available for the resistances and capacitance, but not for time constants.

3.8.2 Parameter Initial Guesses

Each parameter also has an entry field beside it. The value in this field is used as the initial guess for the regression. Default initial guesses are calculated before any regression is performed based on the data being used. The formulas for these guesses are shown in Table 3.3.

Table 3.3: Default initial guesses

Parameter	Formula
R	$\left(\max\left(Z_r\right) + \min\left(Z_r\right)\right)/2$
τ	$\left(\log_{10}\left(\frac{\max(f)}{\min(f)}\right)\right) / \left(\max(f) - \min(f)\right)$
C	$1/\left(-2\pi\left(f\right) Z_j \left(\min\left(f\right)\right)\right)$

After fittings have been performed the initial guess formulas are updated, as shown in Table 3.4.

Table 3.4: Updated initial guesses

Parameter	Formula
R	$\left(\sum_{i=1}^{n} R_i\right)/n$
τ	$10^{\left(\sum_{i=1}^{n} \log_{10}(\tau_i)\right)/n}$

where n is the number of Voigt elements

Tip! The value of τ_i is constrained by the frequency range of the measurement. For an upper frequency limit of 10 kHz, the initial guess for τ_k should not be smaller than 10^{-6}s.

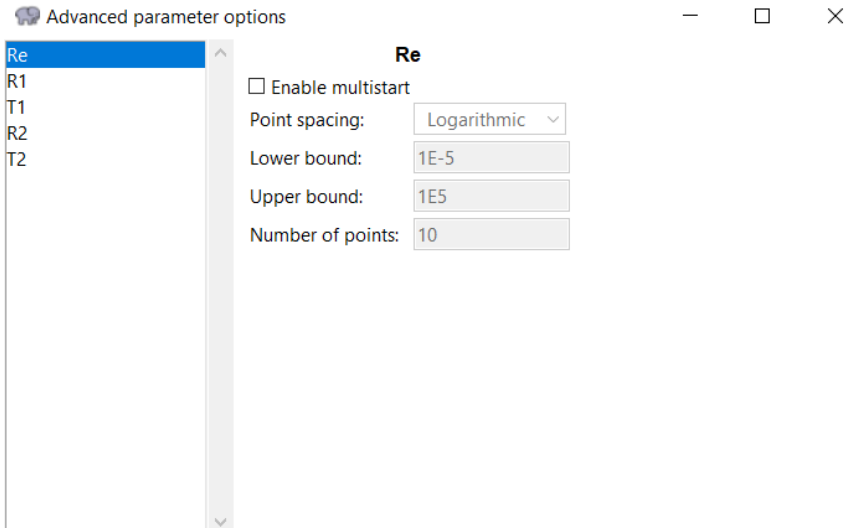


Figure 3.6: Advanced parameter options

Tip! The automated guess of parameters for added line shapes is based on a logarithmic average of the time constants. This works well for capacitive systems, but not so well for systems showing inductive behavior. Manual selection of new guesses is necessary to model capacitive systems. For an inductive loop at low frequencies, select a large time constant and an negative resistor value.

Tip! Remember that $\tau = 1/(2\pi\omega_c)$. To address a poor fit at low frequency, a large time constant is required.

Tip! If the program fails to converge when a lineshape is added, try using a smaller guessed value for R_i.

3.9 Advanced Options

The `Advanced options` button is present in the Edit Model Parameters popup. When clicked, it will open a separate popup window – seen in Figure 3.6 – with further choices for each parameter.

The bar on the left allows a particular parameter to be chosen; when chosen, the parameter will be highlighted on the left, and its name will appear at the top of window to the right.

3.9.1 Multistart

Each parameter has a checkbox labelled `Multistart`. Checking this will activate multistart for that parameter and will enable the other fields present. Multistart works by causing fits to be done at multiple different initial guesses; how these guesses are chosen can be specified here.

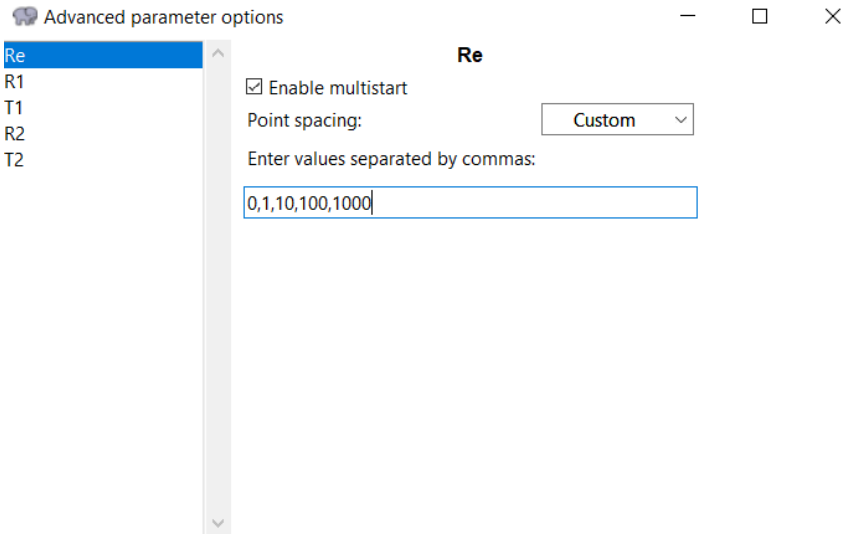


Figure 3.7: Custom multistart guesses

`Point spacing` has four options:

- `Linear` spaces the points linearly
- `Logarithmic` spaces the points logarithmically (therefore the bounds must be on one side of zero only)
- `Random` chooses random points between the bounds
- `Custom` allows for specific guesses to be input

`Lower bound` and `Upper bound` are the limits between which the guesses will be drawn from. `Number of points` is the number of initial guesses to be chosen. The initial guess in the Edit Model Parameters popup will be tried in addition, so the total number of trials for a parameter will be `Number of points`+1. Note that if multiple parameters have multistart activated, each parameter combination will be tried. This means that multistarting multiple parameters will be exponentially slower.

If `Custom` spacing is chosen, the other options will disappear and an entry field will appear as in Figure 3.7.

3.10 Load Existing Parameters

Previously fit parameters can be loaded by clicking the `Load Existing Parameters...` button; these parameters should be located in a *.mmfitting* file. Unlike loading a *.mmfitting* file with the `Browse...` button, this will only load the parameters as initial guesses – it will not change other settings, and it won't load a new *.mmfile*.

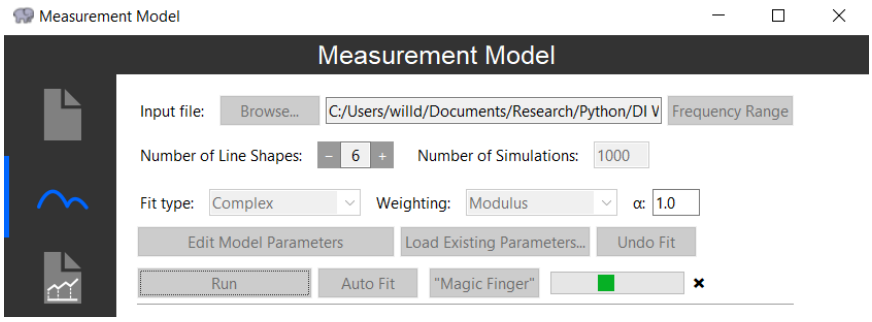


Figure 3.8: Indeterminate progress

3.11 Undo Fit

Sometimes a bad fit is performed and it is necessary to "step back" to the previous fit. This can be done with the `Undo Fit` button. It will only be active once at least two fittings have been performed. This will reset all fit parameters to their previous states, as well as changing back all fitting settings, such as `Weighting` and `Fit Type`.

> **Warning!** Once a fit is undone, it cannot be recovered automatically (i.e. the fitting will have to re-performed), so caution is warranted when using this feature.

3.12 Run

The fitting is actually performed by using the `Run` button. This will perform the regression discussed in Section 3.1 using the settings mentioned throughout this chapter. The button will only become active once a file is loaded. When a fitting begins, all other buttons and entry fields will be disabled, including those in popups. These will be re-enabled once the fitting finishes.

If a fitting is performed with multistart enabled on a computer with multiple cores, new processes may be spawned to parallelize the computations. Specifically, this will occur if the number of parameters multiplied by the number of combinations is greater than 1000. Multiple processes will allow CPU-bound computations to be done in parallel, but there is substantial overhead in creating these processes; therefore, these numbers provide an estimation as to when this tradeoff is profitable. The number of processes spawned will be equal to the number of logical cores available.

3.12.1 Progress Bars

The progress of the fit is indicated by a progress bar. If multistart is not chosen, the completion percentage cannot be estimated and an "indeterminate" progress bar is shown as in Figure 3.8.

If multistart is chosen, the progress bar will show how many multistart combinations have been tried; it will also provide information as to when processes are being created and when Monte Carlo simulations are occurring. This is shown in Figure 3.9.

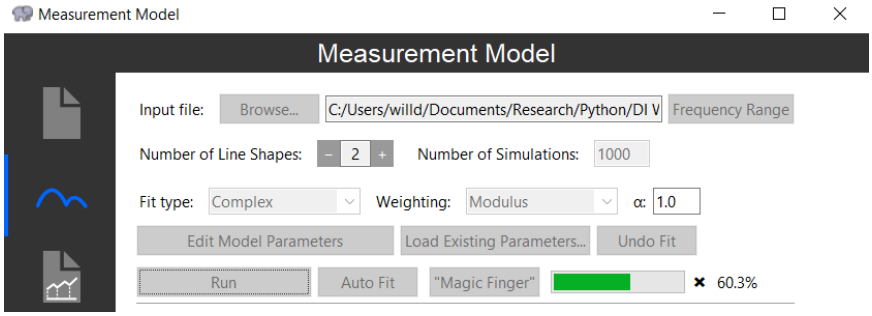


Figure 3.9: Determinate progress

On Windows 7 and above the state of the progress bar will be mirrored in the program's taskbar icon.

3.12.2 Cancelling

Once a fitting is started, it can be cancelled by clicking the `X` next to the progress bar. This will kill the fitting thread, as well as any spawned processes, and will leave the current fit in place. All buttons will be reactivated. A "cancel" also occurs when the overall program is closed.

3.12.3 Finishing

When the fitting is finished, a number of things can occur:

- Nothing, if the fitting was cancelled
- An alert on failure
- The results of the regression appear; this will be discussed starting in Section 3.15

3.13 Auto Fit

Clicking the `Auto Fit` button will bring up a popup (see Figure 3.10) with settings that allows the program to attempt to automatically fit the loaded *.mmfile*. Most of the options are exactly as in the standard Measurement Model tab, including `Number of Simulations` (see Section 3.5), `Weighting` (see Section 3.7), and `Fit type` (see Section 3.6).

3.13.1 Max Voigt Elements

The `Max Voigt Elements` entry takes positive integers only; if the program can do so, it will fit up to this number of line shapes, inclusive.

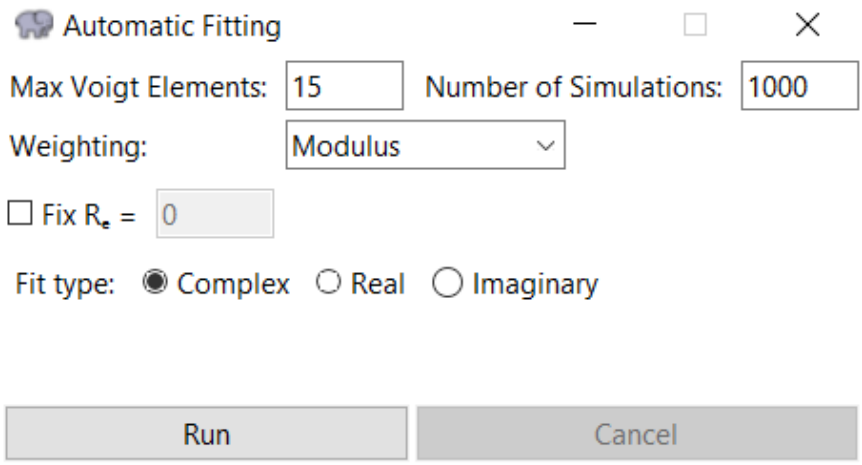


Figure 3.10: The auto fit window

3.13.2 Fix R_e

R_e can be fixed (i.e. not fit) by using the `Fix Re` checkbox; this will enable the textbox to the right, where the value of R_e can be set.

3.13.3 Run

Upon clicking `Run`, the program will attempt to automatically fit the data up to specified number of line shapes using the specified options by sequentially stepping and fitting. The program uses the initial guess heuristics described in Table 3.4 for each new step.While the fitting is running, other options will be disabled (in the same manner as when a normal fit is run). A fitting is considered to "fail" if a fit cannot be found or the confidence intervals on the parameters cannot be estimated or are greater than 100%. Once a fitting fails, it will attempt a 10-point multistart on the last fit's τ value (from $10^{-5} - 10^{5}$). If this fails and fewer than 5 Voigt elements have been fit, it will attempt to increase the number of line shapes; otherwise, the fitting will end and the results will be displayed in the same way as if a normal fitting had been done.

If `Run` is clicked after a fit has already been performed and results are displayed, a confirmation box will appear. Clicking `OK` will cause the autofit to run; clicking `Cancel` will return to the autofit window. If a fitting is already running from the main window, the program will not let you run an autofit as well.

During the fit, a textbox will appear in the autofit window showing which element the program is currently trying. For instance:

```
Trying 8 Voigt elements with multistart fit...
```

3.13.4 Cancel

Once a fitting is running, the `Cancel` button will activate. If clicked, it will stop the fit completely; none of the fitted parameters will be displayed as results.

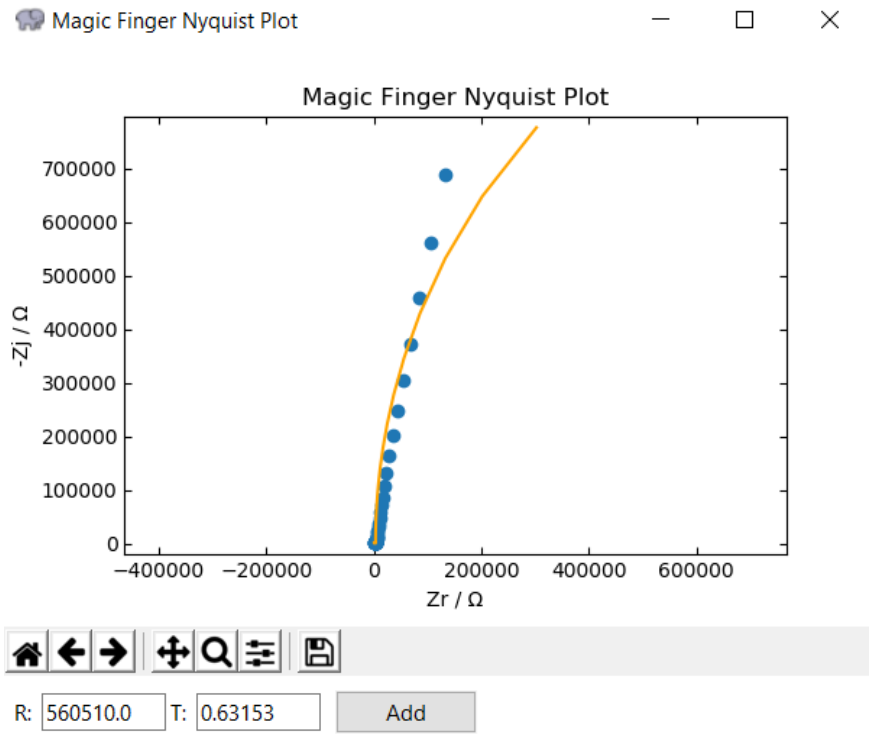


Figure 3.11: The "magic finger" window

Tip! If an autofit looks poor when `Plot` (see Section 3.19) is clicked, try clicking the main window's `Run` button to attempt a fit using autofit's parameters as initial guesses.

3.14 "Magic Finger"

The "Magic Finger" is a option to graphically choose a new parameter's initial guesses. Clicking the button will open a popup (see Figure 3.11) showing a Nyquist plot with the data points and the current fit, if any.

The plot operates in the same manner as other plots (see Section 2.8). Once a point is clicked, the initial guess corresponding to that point will be shown in the textboxes at the bottom of the window; they can be modified by typing in the boxes. Click `Add` to add the new parameter. If the guess is not a real number, an error will appear.

3.15 Regression results

After a fit is successfully performed, the results will appear in the bottom half of the tab (see Figure 3.12 for a 10-Voigt-element fit for DI water).

First will be listed the ohmic resistance, polarization impedance, and overall capacitance. An example for a 10-Voigt-element fit for DI water is shown below:

```
Ohmic Resistance = 137.7 ± 33
```

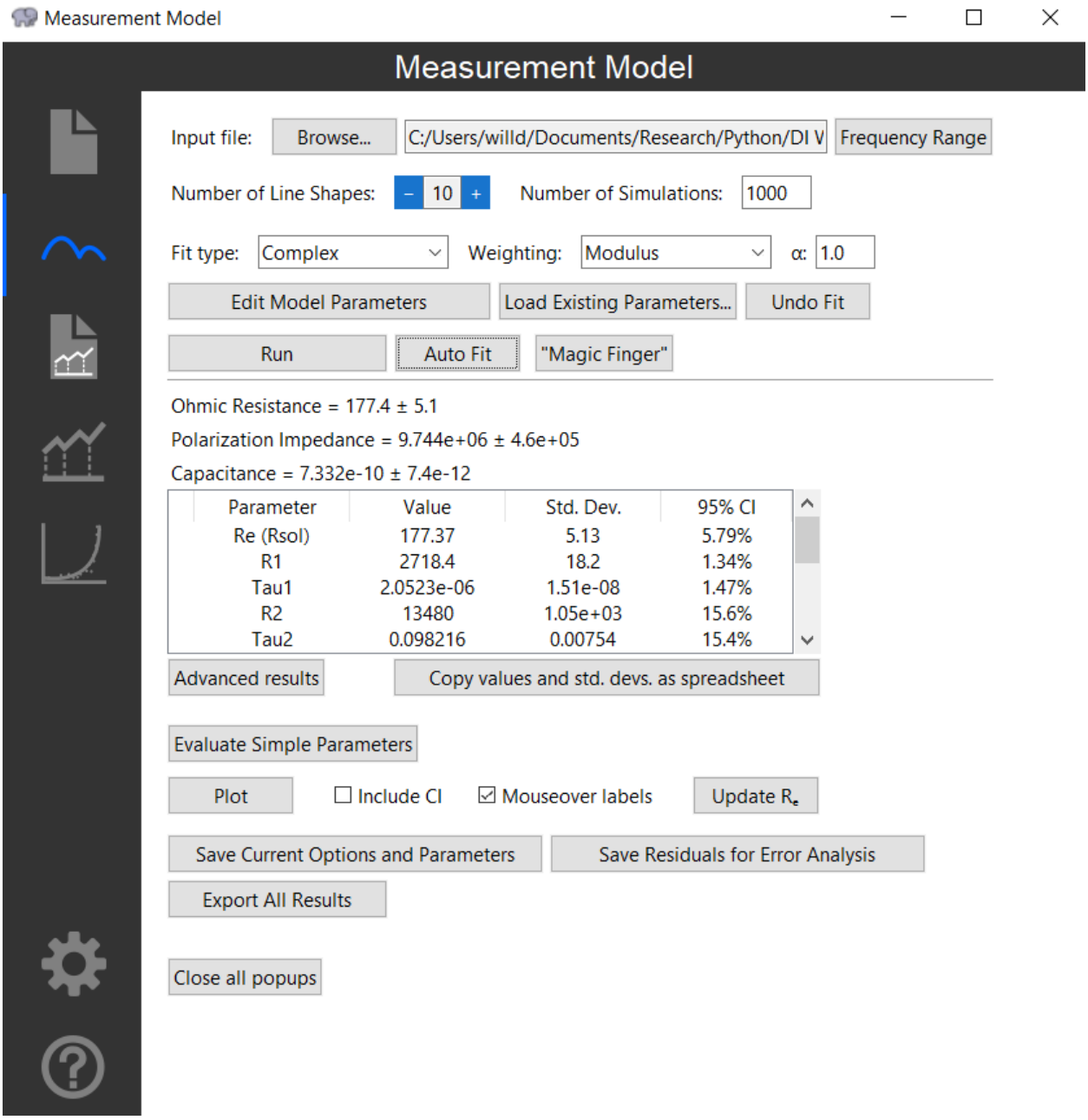


Figure 3.12: Fitting results for 10 Voigt elements on DI water

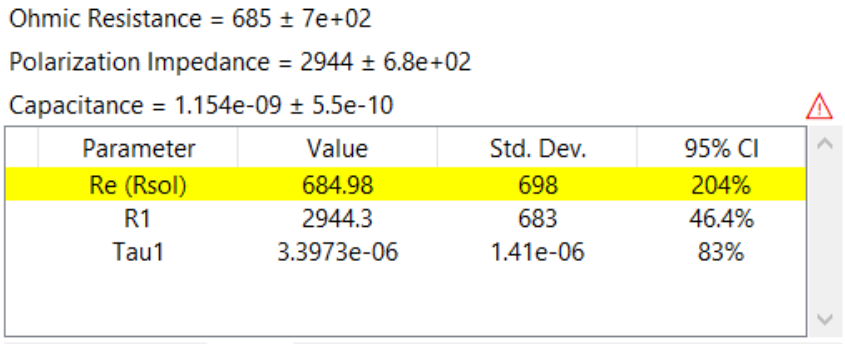


Figure 3.13: A bad fit with a single Voigt element for DI water

```
Polarization Impedance = 2.711e+05 ± 7.2e+03
Capacitance = 7.068e-10 ± 6e-11
```

Below that will appear a listbox with each fitted parameter, their regression results, their standard deviations, and their confidence intervals. When the confidence interval given in the fourth column is greater than 100 percent, the parameter can have no statistical significance, and the number of parameters should be reduced. Typically, the number of parameters is increased incrementally until the confidence interval for one or more parameters includes zero.

If a parameter has a confidence interval greater than 100% (or a confidence interval of `nan`), it will be highlighted yellow in the listbox and a red alert triangle will appear to the top right of the listbox (see Figure 3.13).

> **Tip!** The success of a non-linear regression is sensitive to the initial guess for parameters. If the quality of the fit is poor, but the confidence intervals indicate that a maximum number of parameters has been achieved, consider a) trying a different initial guess, b) trying a multistart fit, or b) checking the data for consistency with the Kramers-Kronig relations.

3.16 Advanced results

Further information about the fitting can be found under the `Advanced results` button. The information included in the popup is:

- The file name that was fit
- The amount of data that were fit
- The number of parameters fit ($Number of parameters = 2*Number of lineshapes+1$)
- The parameters and standard deviations, with more digits
- The zero freqency impedance
- The polarization impedance, with more digits

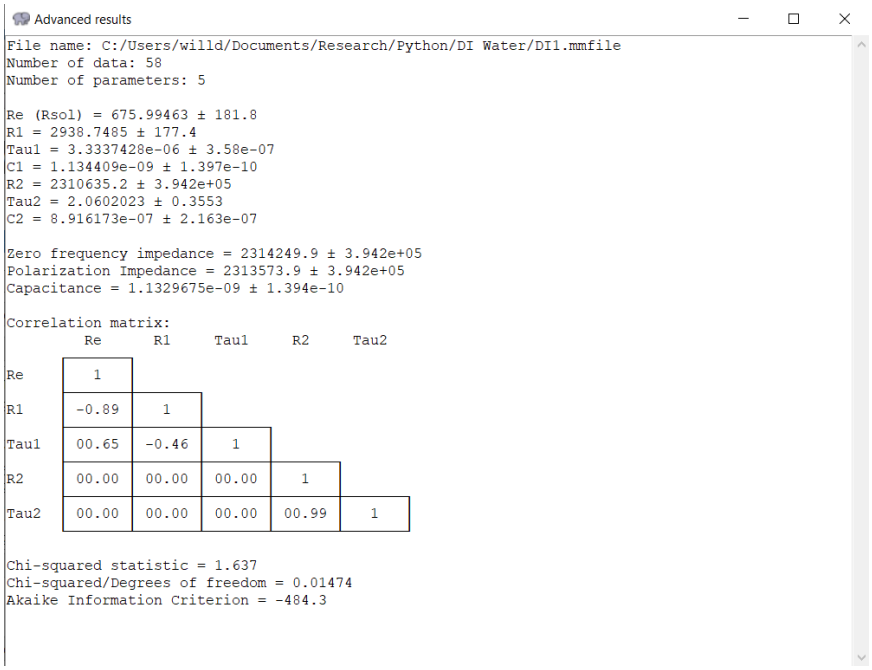


Figure 3.14: Advanced results for a 2-line-shape fit on DI water

- The capacitance, with more digits
- A correlation matrix of every parameter
- The χ^2 statistic
- The χ^2 statistic divided by the degrees of freedom
- The Akaike Information Criterion (AIC)

For example, see the results of a 2-Voigt-element fit on DI water (Figure 3.14).

3.17 Copy values and std. devs. as spreadsheet

To export the results of a fit in a manner that can be pasted easily into a spreadsheet program, use the `Copy values and std. devs. as spreadsheet` button. This will sort all parameter pairs by time ascending time constant, and will also include the file name, capacitances for each parameter pair, the zero frequency impedance, the polarization impedance, and the overall capacitance.

3.18 Evaluate Simple Parameters

The `Evaluate Simple Parameters` button uses a zero finding routine to identify the maxima and minima of the imaginary part of the impedance model in order to identify

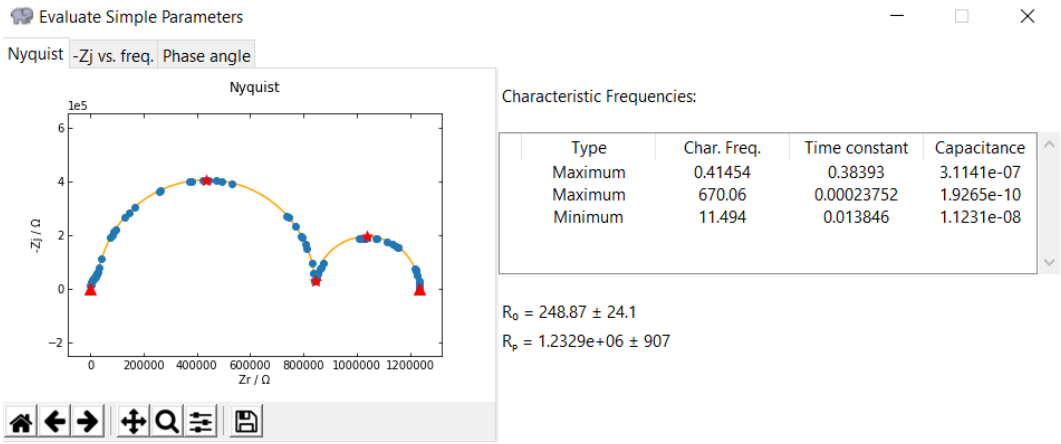


Figure 3.15: Evaluating characteristic frequencies of synthetic data

characteristic frequencies, time constants, and capacitances. This can cause a delay of a second or two between when the button is clicked and when the window is ready.

To the right will be a listbox containing all the identified maxima and minima, along with their corresponding characteristic frequencies, time constants, and capacitances (calculated per Equation 3.3).

$$C = \frac{1}{(2\pi f_c R_p)} \tag{3.3}$$

R_0 and R_p are calculated and display below the listbox (see Figure 3.15)..

To the left of the window is a graph with three tabs above. Clicking these tabs will change the currently displayed graph. `Nyquist` will show a Nyquist plot (the default), `-Zj vs. freq.` will show a graph of the imaginary impedance against frequency, and `Phase angle` will show a graph of the phase angle against frequency.

The red stars in each graph correspond to a maxima or minima shown in the listbox (for the phase angle graph, they will be inflection points, not maxima or minima). Hovering over a point will show its frequency, Z_j (for Nyquist and imaginary), Z_r (for Nyquist), and phase angle (for phase angle). The red triangles in the Nyquist plot represent the asymptotic behavior of the impedance.

Tip! Spurious characteristic frequencies will be identified unless the model provides a good fit to the data.

Warning! The extrapolation shown in Figures 3.15 is based on synthetic data that are known to satisfy the Kramers-Kronig relations. Nevertheless, extrapolation is extremely dangerous. Whenever possible, it is better to have experimental data at the characteristic frequencies.

Warning! Note that the parameters obtained by the simple analysis do not substitute for development and regression of a process model based on physical and chemical arguments.

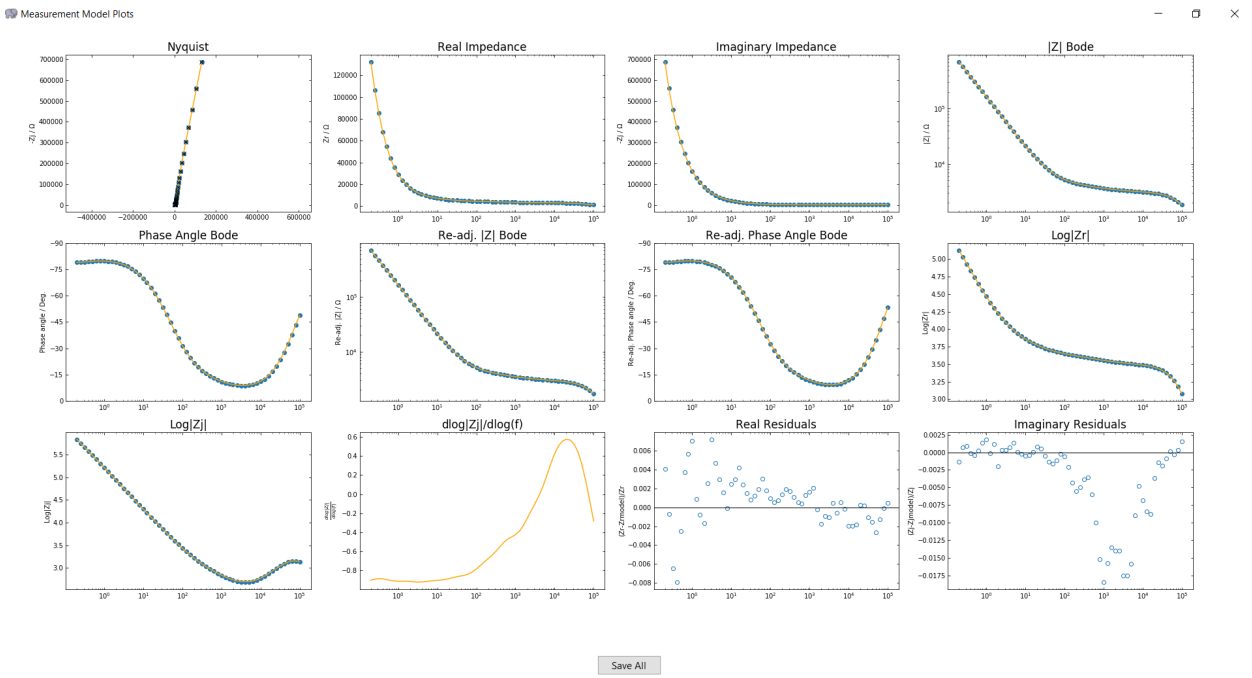


Figure 3.16: Plots of a 10-line-shape fit of DI water

3.19 Plot

The `Plot` button will open a fullscreen window containing twelve plots (see Figure 3.16):

- Nyquist plot
- Real impedance vs. frequency
- Imaginary impedance vs. frequency
- Bode plot of the modulus of the impedance $\left(\sqrt{Z_r^2 + Z_j^2}\right)$
- Bode plot of the phase angle $(\arctan\left(Z_j/Z_r\right))$
- Bode plot of the ohmic resistance corrected modulus of the impedance $\left(\sqrt{\left(Z_r - R_e\right)^2 + Z_j^2}\right)$
- Bode plot of the ohmic resistance corrected phase angle $(\arctan\left(Z_j/\left(Z_r - R_e\right)\right))$
- Base-10 logarithm of real impedance vs. frequency $(\log_{10}|Z_r|)$
- Base-10 logarithm of imaginary impedance vs. frequency $(\log_{10}|Z_j|)$
- Derivative of base-10 logarithm of imaginary impedance with respect to base-10 logarithm of freq $\left(\frac{\partial \log_{10}|Z_j|}{\partial \log_{10} f}\right)$

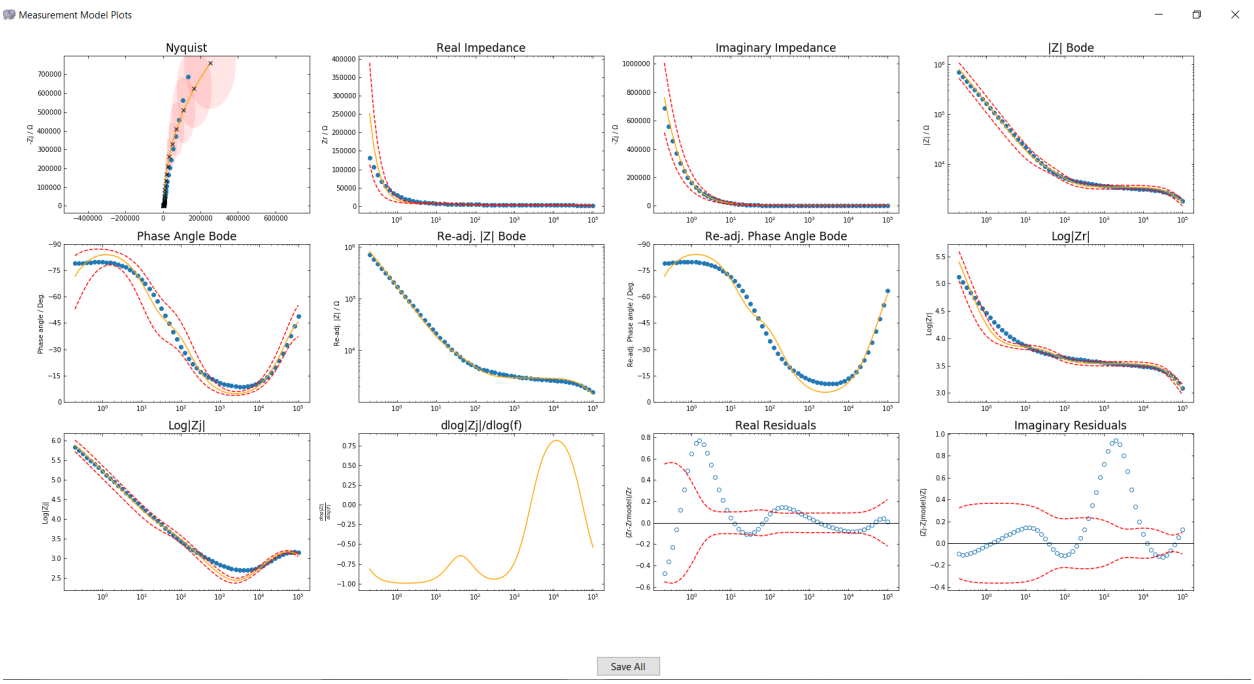


Figure 3.17: Plots of a 3-line-shape fit of DI water with confidence intervals

- Real residuals $\left(\left(Z_r - \widehat{Z}_r\right)/Z_r\right)$
- Imaginary residuals $\left(\left(Z_j - \widehat{Z}_j\right)/Z_j\right)$

Clicking on a plot will open a larger popup version of that plot, which can be operated on using the controls described in Section 2.8. The x-axis labels do not appear for the plots in order to save space; however, they are present in the larger popup version. Note that the plots do not auto-update when a new fit is done, so the `Plot` button must be clicked again.

3.19.1 Include Confidence Interval

If `Include Confidence Interval` checkbox is checked and then `Plot` is clicked, the window will include confidence intervals around every plot except the ohmic-resistance-corrected Bode plots and the derivative plot (see Figure 3.17).

The Nyquist plot will include confidence ellipses whose width indicate the deviation in the Z_r direction and whose height indicate the deviation in the Z_j direction.

3.19.2 Mouseover Labels

The `Mouseover labels` checkbox - when checked (the default) - will cause small labels to appear next to data points when the mouse is over them. This occurs on the larger popup plots, not the main plot window. These labels include the frequencies and "y-axis" data; for the Nyquist plot and real and imaginary impedance plots both Z_r and Z_j are shown.

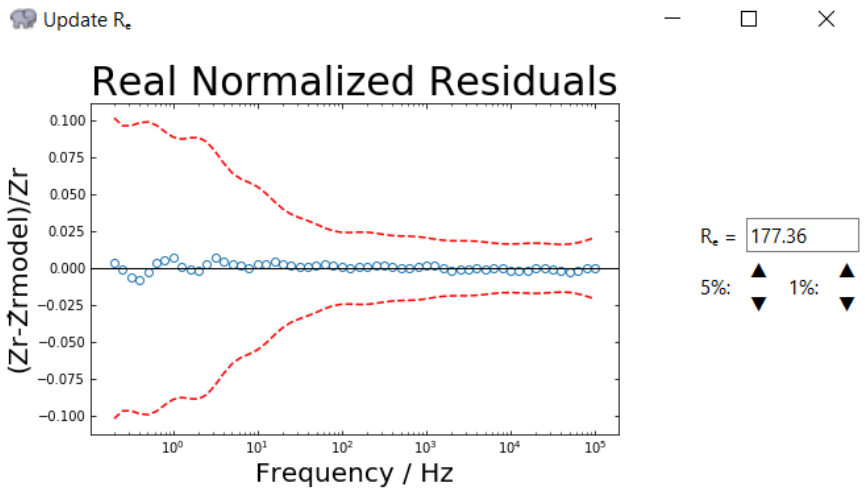


Figure 3.18: The ohmic resistance updating window

3.19.3 Save All

The `Save All` button located at the bottom of the plots window can be used to save all of the plots simultaneously. A prompt will appear asking for the directory in which they will be saved. They will be saved as PNG files; the name format is *file name* + - + *type of fit and number of Voigt elements* + *_which plot*. The x-axis labels will appear on the saved plots.

3.20 Update R_e

Under some circumstances, most commonly when doing an imaginary fit, it may be necessary to manually adjust the ohmic resistance. This can be done with the `Update Re` button. This will open a popup (see Figure 3.18); to the left is a graph of the real residuals - this will be automatically updated as the R_e is updated.

The ohmic resistance can be entered manually in the box to the right. Otherwise, the update and down arrows below can be used: the left arrows will change the ohmic resistance by 5%, while the right arrows will change the ohmic resistance by 1%. There will be a confirmation alert prior to updating the ohmic resistance, as updating it will cause the current fit and confidence interval for R_e to be lost.

3.21 Save Current Options and Parameters

The `Save Current Options and Parameters` button will create a new *.mmfitting* file. This file includes all currently fitted parameters, the current fitting options (number of Voigt elements, number of simulations, fit type, weighting, parameter constraints, and frequency changes (if any), and a link to the current *.mmfile*. The entire current fit can then be loaded in the future (see Section 3.2) or just the parameters can be loaded (see Section 3.10).

3.22 Save Residuals for Error Analysis

The `Save Residuals for Error Analysis` button will create a new *.mmresiduals* file consisting of the current frequencies, data, fitted ohmic resistance, and real and imaginary residuals. This file can be used to produce a *.mmerrors* file in the Error File Preparation tab (see Chapter 4), from whence it can be used to fit an error stucture in the Error Analysis tab (see Chapter 5).

3.23 Export All Results

The `Export All Results` button will create a *.txt* file containing all information about the current fitting results, including the frequencies, data, weighting, and model values; the fitted parameters (including capacitances) and their confidence intervals; and the chi-squared value.

Chapter 4

Error File Preparation

Prior to fitting an error structure under the Error Analysis tab (see Chapter 5), the *.mmresiduals* files saved from fittings must be combined into a *.mmerrors* file. This can be accomplished in the Error File Preparation tab. To access this tab, click the file icon with an error graph in the middle ( ). Figure 4.1 shows the tab before any files have been loaded.

It is important that all files combined into a *.mmerrors* file have been fit with the same number of statistically significant parameters. Likewise, all files must have the same frequencies.

> **Tip!** If the same number of parameters could not be obtained for each data file, either reduce the number of parameters to the lowest common value or eliminate the file yielding a smaller number of parameters.

4.1 File Input

To begin, use the `Add File(s)` button to choose one or more *.mmresiduals* files. Once the file dialog is confirmed, the file name(s) will appear in the listbox to the upper right of the tab. If more than one file is loaded, the listbox below will display the average values and standard deviations of the residuals at each frequency, and the `Plot` and `Save errors` buttons will become active (see Figure 4.2).

The `Real Mean` is the average of the real residual values at a given frequency, and the `Real Std. Dev.` is the standard deviation of the real residuals between the loaded files at that frequency.

Once one file is loaded, a line will appear under the listbox displaying the average ohmic resistance and average overall capacitance of the loaded files.

If a file has a different number of frequencies, or the frequency values are different, than files that are already present, an error will occur. If a file is loaded that has already been loaded, a warning will appear but the file will still be loaded.

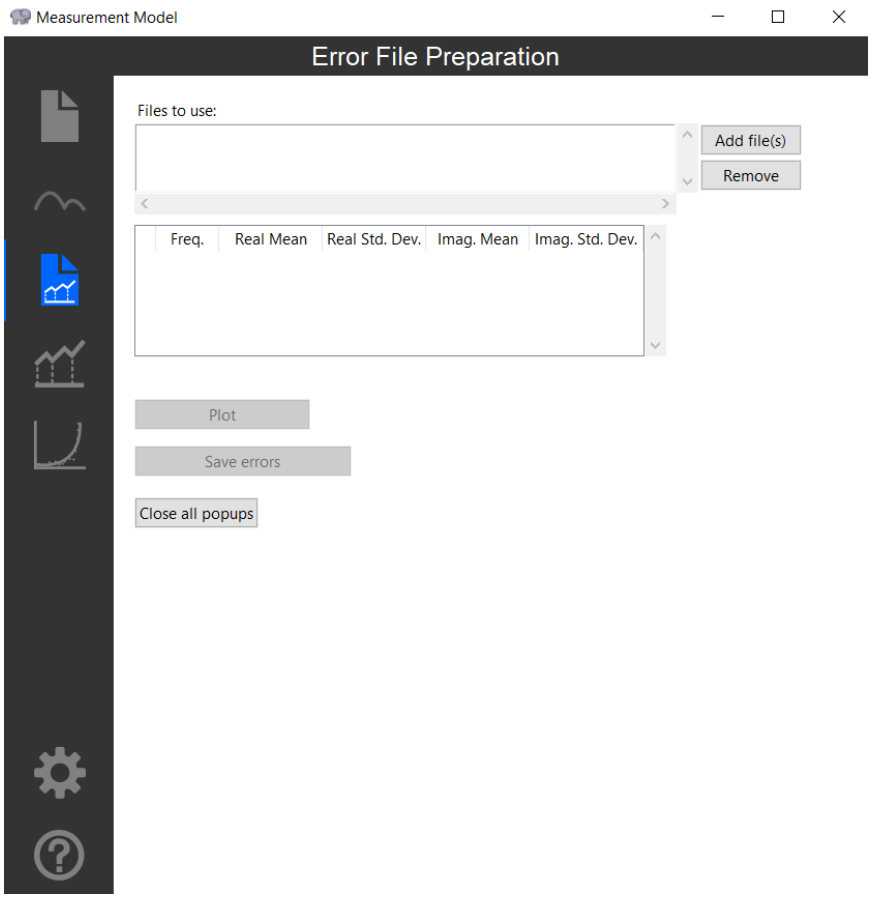


Figure 4.1: The Error File Preparation tab

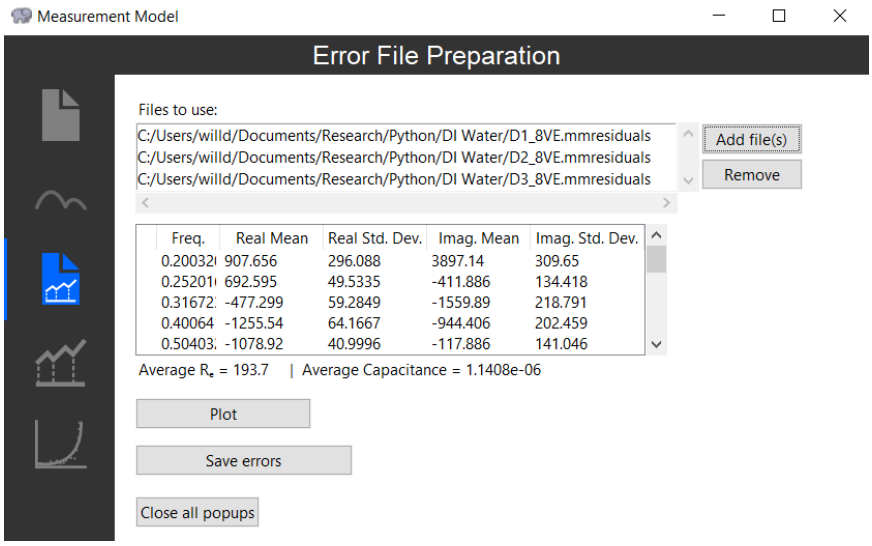


Figure 4.2: The Error File Preparation tab after loading three residuals files for DI water, each fit with eight Voigt elements

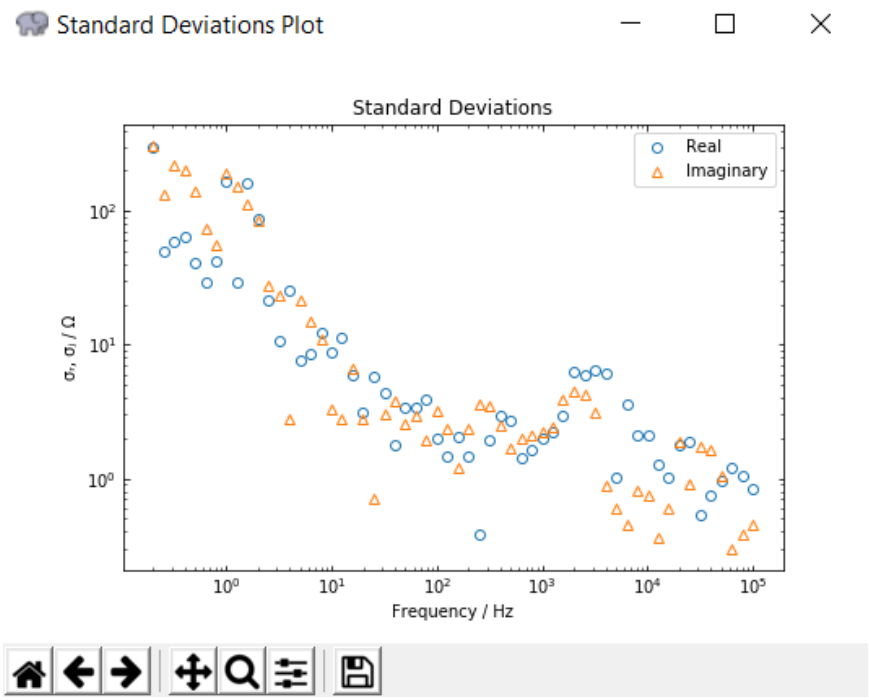


Figure 4.3: Standard deviations of residuals for three measurements of DI water fit with eight Voigt elements

4.1.1 Remove

In order to remove a file from the list, click the desired file so it is highlighted in the listbox, then click the `Remove` button or right-click and click `Remove file`.

4.2 Plot

Once at least two files are added, the `Plot` button will become active. This button will bring up a plot of the overlayed real and imaginary standard deviations of the residuals (i.e. columns 3 and 5 of the listbox seen in Figure 4.2) against frequency (see Figure 4.3). The real standard deviations are represented as blue circles, while the imaginary are represented as orange triangles.

4.3 Save errors

Once the desired *.mmresiduals* files have been loaded, a new *.mmerrors* file can be saved by using the `Save errors` button. This file will include the frequencies, real and imaginary standard deviations of the residuals, the real and imaginary impedance data, and the standard deviations between the real and imaginary standard deviations at a given frequency (i.e. σ_σ).

Chapter 5

Error Analysis

The principle behind using the measurement model to estimate the stochastic contribution to the error structure was presented by Agarwal *et al*.[2] The approach has been applied to a number of electrochemical[4–8] and non-electrochemical systems.[9,10]

The error analysis tab can be accessed with the error graph icon ( ) in the navigation pane (see Figure 5.1).

5.1 File Input

The file input is very similar to the Error File Preparation tab (see Section 4.1). Use the `Add File(s)` button to choose one or more *.mmerrors* files. Once the file dialog is confirmed, the file name(s) will appear in the listbox to the upper right of the tab.

5.1.1 Remove

In order to remove a file from the list, click the desired file so it is highlighted in the listbox, then click the `Remove` button or right-click and click `Remove file`.

5.2 Error structure

The complete error structure is of the form seen in Equation 5.1.

$$\sigma = \alpha |Z_j| + \beta |Z_r - R_e| + \gamma |Z|^2 + \delta \tag{5.1}$$

The `Error structure model` line beneath the file listbox shows the current error structure to be fit. This can be modified with the checkboxes below - checking a box will cause that parameter to be added to the fitting and will update the `Error structure model` line. If β is checked, the R_e checkbox will appear below it; if unchecked, R_e will disappear. The default error structure choice is $\gamma |Z|^2 + \delta$.

> **Tip!** The model which includes the role of R_e in equation (5.1) appears to give a better representation of the standard deviation for systems which have a large solution resistance, but is more awkward to use because it requires

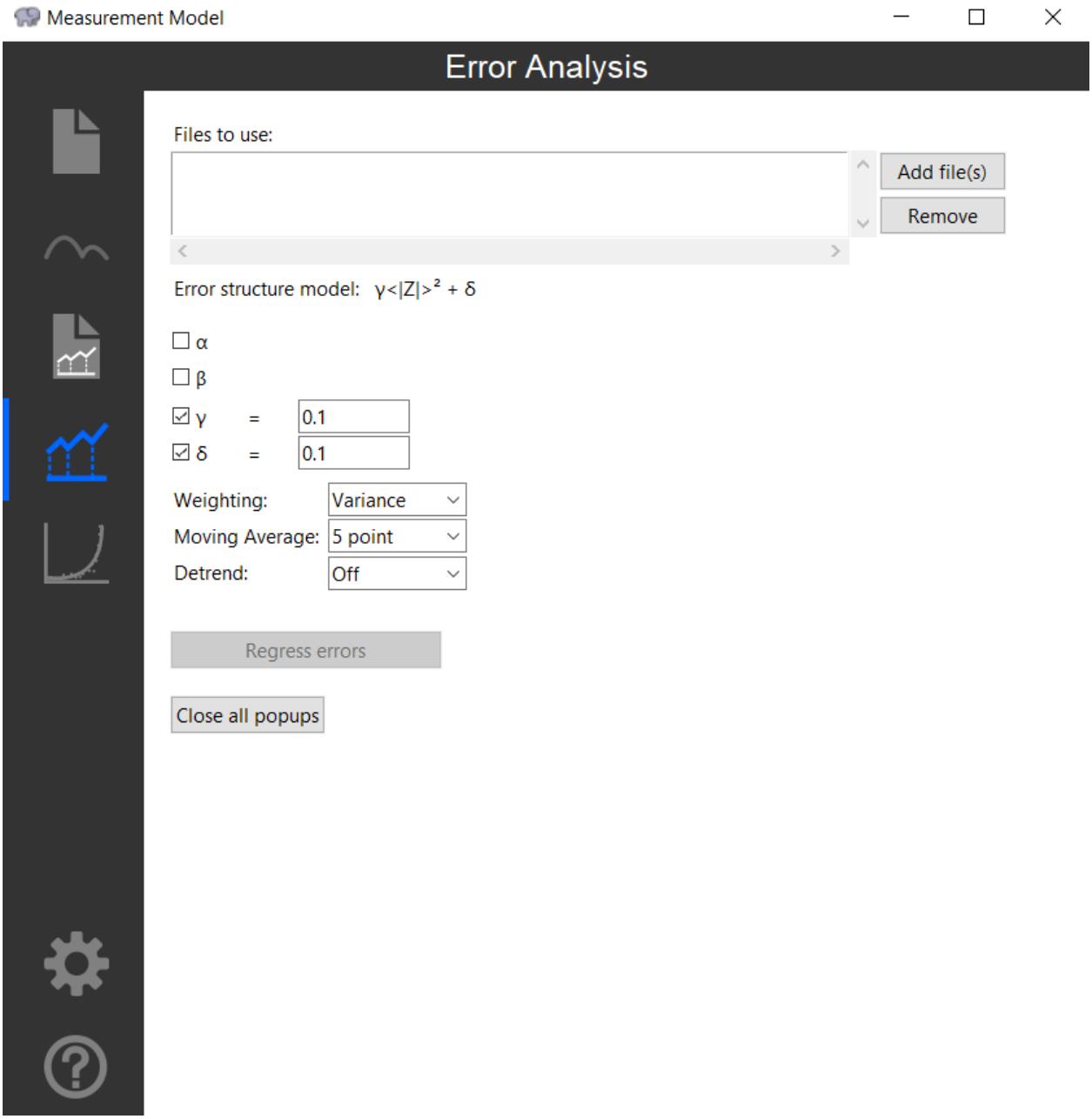


Figure 5.1: The Error Analysis tab

an *a priori* estimate of R_e. For systems that have a relatively small solution resistance, it is sufficient to ignore this contribution.

5.3 Fitting options

Due to the heteroskedastic nature of the residuals, it is important to choose the proper options when fitting in order to obtain a good regression.

5.3.1 Weighting

The weighting used for the regression can have a significant influence because the standard deviation is a very strong function of frequency. The `Weighting` dropdown provides options for this. `None` means that no weighting will be used (all weights will be 1), while `Variance` (the default) means that each frequency will be weighted by the standard deviation between the real and imaginary parts of the standard deviations of the residuals. If `None` is chosen, the `Moving Average` dropdown will disappear.

5.3.2 Moving Average

The variances used in weighting can fluctuate between nearby frequencies. In order to smooth the weighting out, a moving average can be taken of nearby points. This can be done using the `Moving Average` dropdown. `None` means that no moving average will be taken; `3 point` means that the variance will be averaged with the ones immediately before and after it; `5 point` (the default) means that the variance will be averaged with the two before and two after it.

The `Moving Average` dropdown is only available if `Variance` weighting is chosen.

5.3.3 Detrend

Fitting can potentially be improved by centering all data around 0. Detrending can accomplish this by subtracting the average value of a series from every point in that series. This can be done with the `Detrend` dropdown. `Off` (the default) means that no detrending will be used; `On` means that all standard deviations and impedance values will be detrended.

> **Tip!** Try fitting with and without detrending to see which yields a better fit and better confidence intervals on the parameters.

5.4 Regress Errors

The `Regress errors` button will use a Levenberg-Marquardt regression to fit the chosen parameters using the current settings. If a fitting fails, an alert will appear. Otherwise, the results will appear below.

Both the real and imaginary standard deviations are fit at the same time. The equality of the real and imaginary parts of the impedance is now supported by a large amount

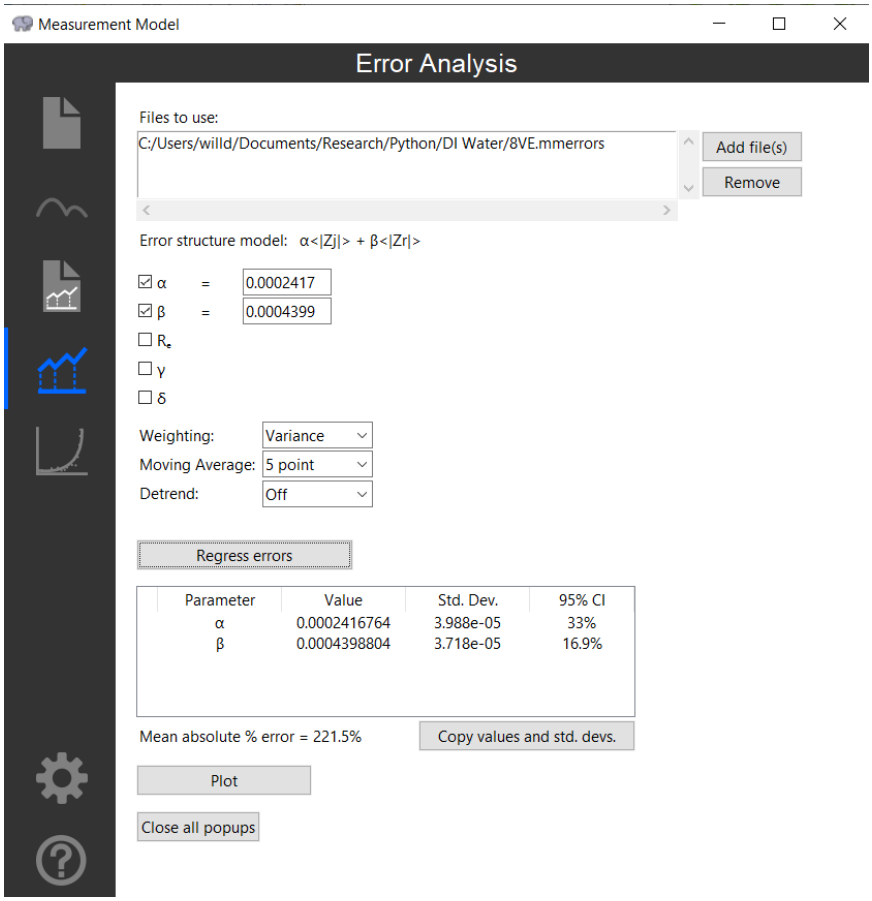


Figure 5.2: A successful fit of the error structure $\alpha |Z_j| + \beta |Z_r|$ for DI water originally fit with eight Voigt elements

of experimental evidence,[9,6,11] by theoretical arguments based on the Kramers-Kronig relations,[10,12,13] and by arguments based on propagation of errors from time to frequency domain.[14,15]

5.5 Results

After a regression has completed successfully, the resulting parameter values will be shown in a listbox beneath the `Regress errors` button. Figure 5.2 shows a successful error structure fit.

Parameters with confidence intervals greater than 100% are highlighted yellow, as they are not statistically different than 0 (see Figure 5.3).

These parameters should be removed from the fitting. Although the fitting permits parameters to have negative values and does not highlight these results, care should be taken with negative results based on the argument that there should be no negative contributions to a standard deviation.

Tip! Continue to delete or add model parameters α, β, γ, and δ until the

Parameter	Value	Std. Dev.	95% CI
α	0.0002244399	4.221e-05	37.6%
β	0.000553278	0.0001023	37%
δ	-0.3796469	0.3194	168%

Figure 5.3: An unsuccessful fit of the error structure $\alpha |Z_j| + \beta |Z_r| + \delta$ for DI water originally fit with eight Voigt elements

regression yields positive parameters with confidence intervals smaller than 100% of the parameter value.

5.5.1 Mean absolute percent error

As the weighting and detrending choices can have a significant impact on the χ^2 value, the `Mean absolute % error` is reported instead. This value is found by taking the average of the absolute values of the percent differences between the model and the real and imaginary standard deviations at each frequency. This value should be comparable between different fittings; a lower value indicates a "better" fit.

5.5.2 Copy values and std. devs.

The `Copy values and std. devs.` button will copy the files names, fitting settings, parameter values and standard deviations, and correlation matrix in a format that can be easily pasted into a spreadsheet program.

5.5.3 Plot

The `Plot` button will bring up a plot for each file fit of standard deviations against frequency (see Figure 5.4). The plot will include the real and imaginary standard devations as blue circles and orange triangles, respectively. The error structure model will be plotted as a green line.

Tip! In many, but not all cases, a common error structure can be identified which seems to be a function only of the instruments used. It is worthwhile to attempt to find such a common set of parameters because the resulting model can give a good estimate for the error structure of other data collected under the same conditions.

Tip! A large error structure may signify a need to improve experimental parameters or conditions.

Tip! For measurements that do not introduce a bias error, the standard deviation of the real part of the measurement should be equal to the standard

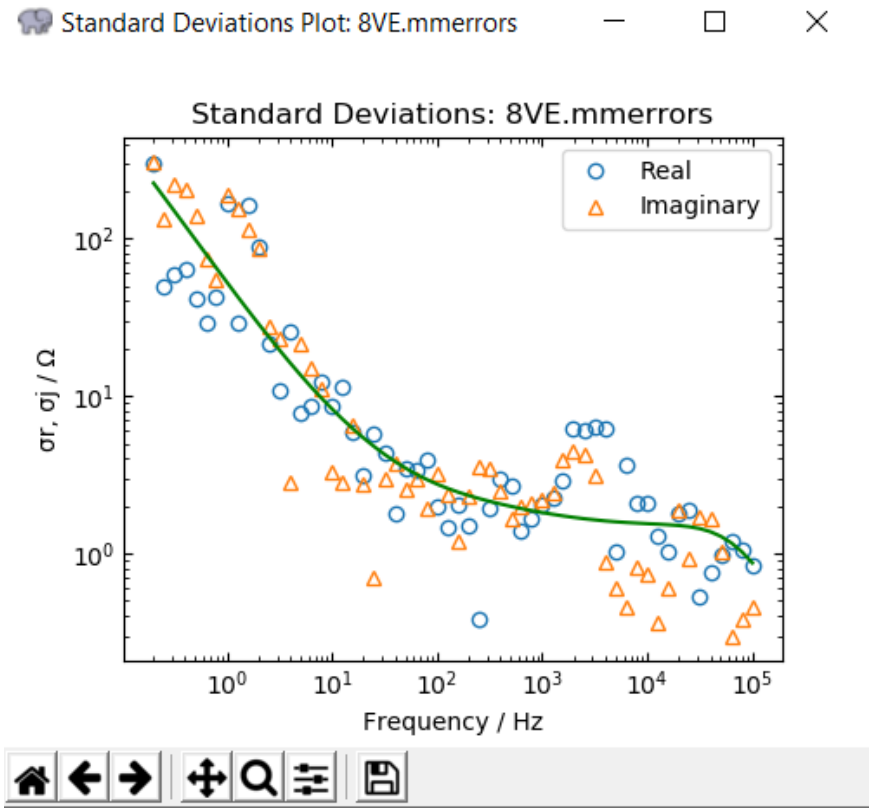


Figure 5.4: The error structure plot for a fitting of $\alpha |Z_j| + \beta |Z_r|$ to DI water originally fit with eight Voigt elements

deviation of the imaginary part. Deviation from this result may arise from problems with the experiment.

Chapter 6

Custom Formula Fitting

A user-defined formula can be fit in the Custom Formula Fitting tab (see Figure 6.1), accessible with the graph icon ( ).

The fitting requires the formula to be input as Python code; there are many free, high-quality resources online to learn the basics of Python syntax.

> **Warning!** Almost all Python code can be typed in and exected; this code could potentially damage the main program or even the computer!

6.1 File Input

A file can be loaded using the `Browse...` button at the top left of the tab. Available file types include *.mmfile* and *.mmcustom*. After loading a file succesfully, its name will appear in the textbox to the right.

If a *.mmcustom* file is chosen, it will load its linked *.mmfile* as well as its code, fitting choices, and fitting parameters.

6.2 Fitting Options

The `Frequency Range` button, `Fit type` dropdown, `Number of Simulations` textbox, and α textbox act exactly as they do in the Measurement Model tab; see Sections 3.3, 3.6, 3.5, and 3.7.1, respectively.

The `Weighting` dropdown acts almost exactly as it does in Section 3.7. However, it includes an additional option of `Custom`. If this is chosen, the weighting used will be taken from a variable called `weighting` in the code. This weighting will be the same for the real and imaginary parts at a given frequency.

6.3 Defining a Custom Formula

Custom Python code can be defined in the `Code` box. The program makes available three pre-defined lists: `Zr`, `Zj`, and `freq`, which are the real impedance values, the imaginary impedance values, and the frequencies, respectively.

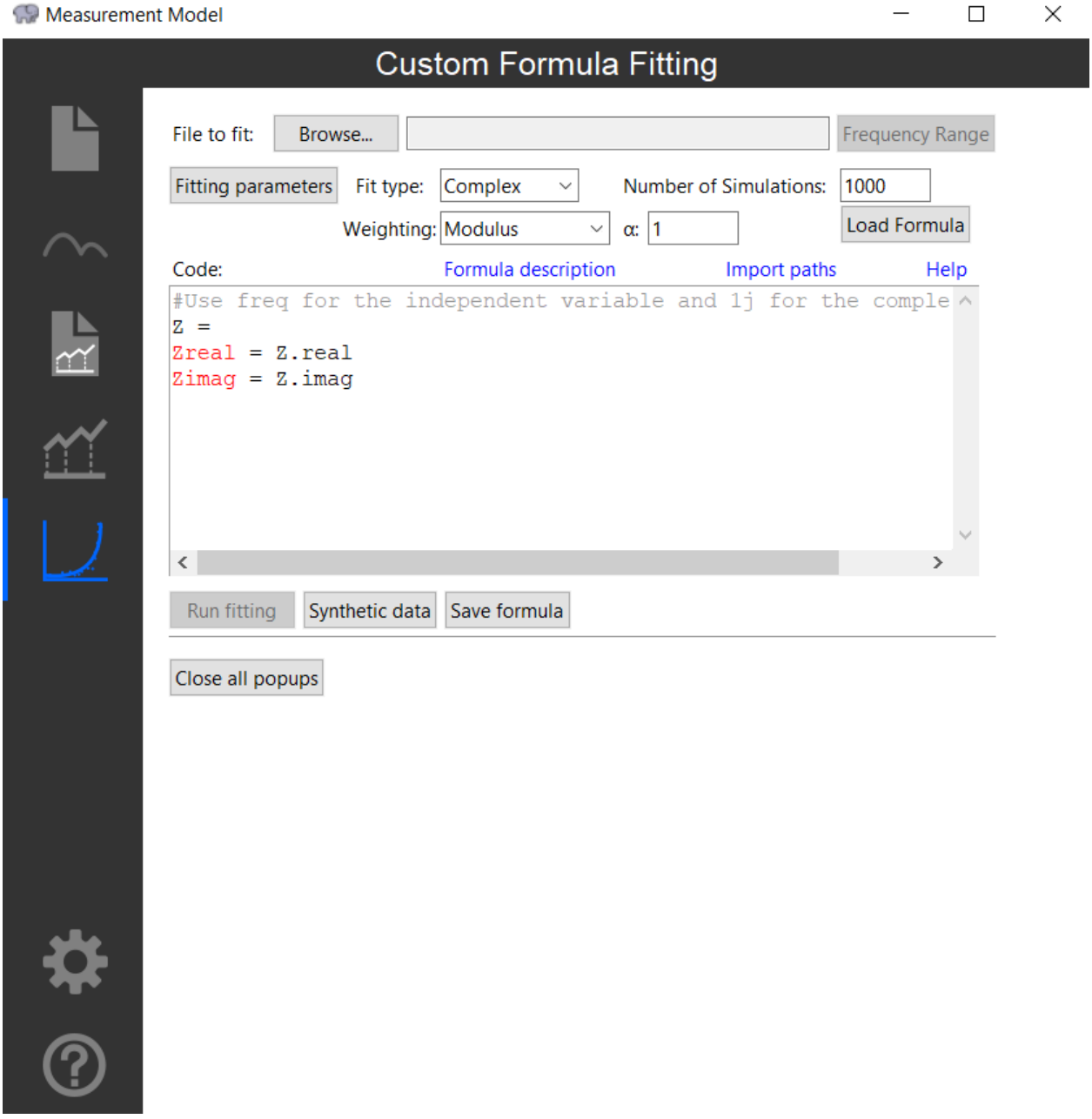


Figure 6.1: The Custom Formula Fitting tab

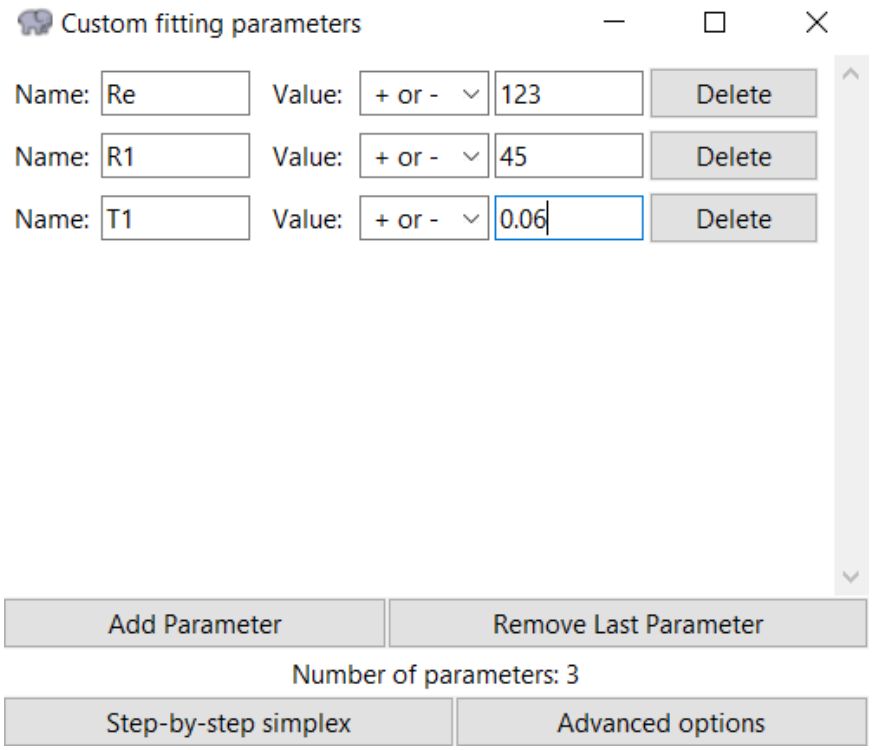


Figure 6.2: Three named custom parameters with initial guesses

The program expects to find two lists: `Zreal` and `Zimag`, in which are held the real and imaginary model impedances, respectively. If `Custom` weighting has been chosen, then the program will expect a third variable: `weighting`.

Python can operate on whole lists/arrays at once, meaning loops are often unnecessary and code can be short. Note that \can be used to break long lines of code and continue on the next line.

6.3.1 Fitting Parameters

All parameters which the program will fit must be added using the `Fitting parameters` button. This will open a popup which will display all current parameters (see Figure 6.2).

At first, the popup will be empty. Add a parameter with the `Add Parameter` button. This will add a new line to the popup with a new parameter. The parameter's name can be defined in the leftmost textbox; this name should be used in the code. The dropdown to the right can be used to restrict the parameter's value, and the textbox to the right holds the initial guess for the parameter. The `Delete` button can be used to remove a given parameter, and the `Remove Last Parameter` button will remove the parameter farthest down the popup. These parameters cannot be recovered once removed. The `Step-by-step simplex` button will become active once a file is loaded (see Section 6.3.2), and the `Advanced options` button will open a popup containing further choices for each parameter, as described in Section 6.3.3.

Parameter names must be alphanumeric, cannot be duplicates, cannot be any of the following predefined names: `freq`, `Zr`, `Zj`, `Zreal`, `Zimag`, or `weighting`, and cannot be a Python reserved word:

False	break	else	if	not	while
None	class	except	import	or	with
True	continue	finally	in	pass	yield
and	def	for	is	raise	
as	del	from	lambda	return	
assert	elif	global	nonlocal	try	

6.3.2 Step-by-step simplex

The `Step-by-step simplex` button will perform the first "step" in the Nelder-Mead simplex algorithm using the current code and loaded file. The initial simplex is constructed using the current parameter values, and the fitting will respect fixed parameters and parameter constraints. This feature can be used as a more robust method for determining appropriate initial guesses for use with the main Levenberg-Marquardt algorithm.

6.3.3 Advanced options

The `Advanced options` button will open a popup containing further choices for each parameter, as seen in Figure 6.3. At the top are lower and upper limits; the fitting will not allow the parameter's value to go below the lower limit, or above the upper limit. The values must be real numbers (with the upper limit being greater than the lower); however, `-inf` and `inf` are accepted as well. These mean no lower or upper limit, respectively. The dropdown by the parameter values in the main parameter popup can also control the upper and lower imits. `Fixed` will disable both the upper and lower limit box, `-` will make the upper limit `0` and the lower limit `-inf`, `+ or -` will make the upper limit `inf` and the lower limit `-inf`, `+` will make the upper limit `inf` and the lower limit `0`, and `Custom` will open the advanced options popup. Changing the values in the advanced options popup itself will set the dropdown to `Custom`, unless the values are a combination listed above (for instance, setting the upper limit to `inf` and the lower limit to `-inf` will set the dropdown's value to `+ or -`).

The multistart options work exactly as they do in the Measurement Model tab (see Section 3.9.1).

6.3.4 Built-ins

Python packages can be imported for more complex operations; `numpy` is already imported as `np`. As in standard Python, the imaginary number can be accessed as `1j`.

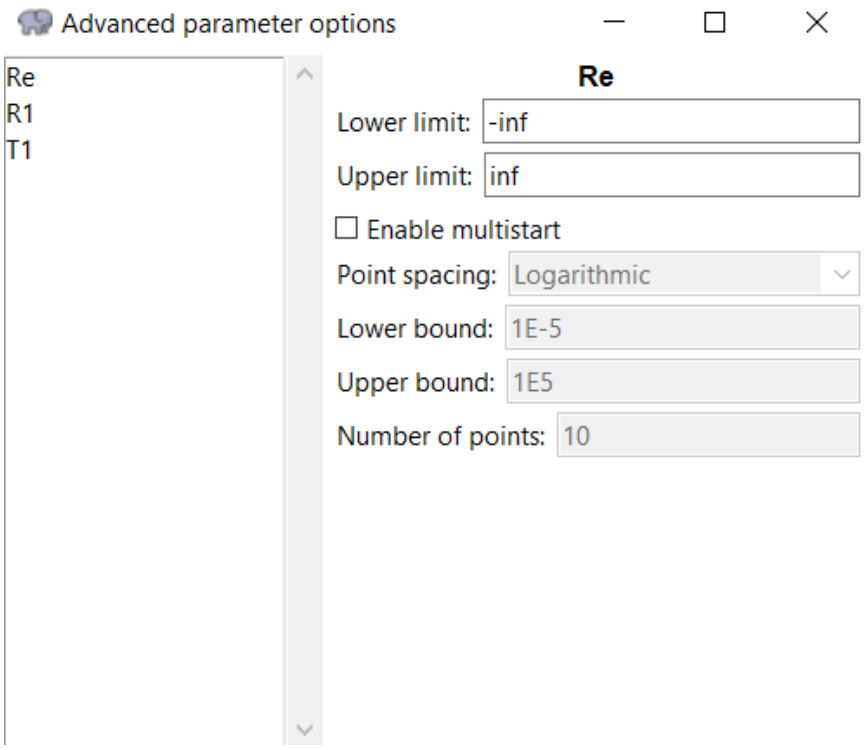


Figure 6.3: Advanced options popup for custom fitting

A number of functions are already included. These can be seen in Table 6.1.

Table 6.1: Included functions

Function (as in code)	Meaning
PI	$\pi = 3.141592653589793$
SQRT	$\sqrt{x}$
ABS	$\lvert x \rvert$
EXP	e^x
SIN	$\sin(x)$
COS	$\cos(x)$
TAN	$\tan(x)$
ARCSIN	$\sin^{-1}(x)$
ARCCOS	$\cos^{-1}(x)$
ARCTAN	$\tan^{-1}(x)$
SINH	$\sinh(x)$
COSH	$\cosh(x)$
TANH	$\tanh(x)$
ARCSINH	$\sinh^{-1}(x)$
ARCCOSH	$\cosh^{-1}(x)$
ARCTANH	$\tanh^{-1}(x)$
LN	$\ln(x)$
LOG	$\log_{10}(x)$
RAD2DEG	Converts radians to degrees: $x * 180/\pi$
DEG2RAD	Converts degrees to radians: $x * \pi/180$

Note: (Arc)trig functions take/return arguments in radians

Practically any other mathematical operation can be using Python libraries, references for which can be found online; see, for example, integration and differential equations.

6.3.5 Import paths

As mentioned above, common python modules can be imported by simply using the `import` keyword. However, the Python interpreter will need to know where to locate custom-written or otherwise less-common modules. This can be done either by placing the Python file in the same directory as the actual executable (not its shortcut or its installer) or by using the `Import paths` link above the code textbox. Clicking this will open a popup (see Figure 6.4).

To the left is a listbox where the current extra import paths will be listed. To the right are two buttons. The `Browse...` button will open a dialog where the directory to be searched can be chosen; the once this directory is selected, it will appear in the listbox to the right, and the interpreter will look for modules in it. After a directory is listed, it can

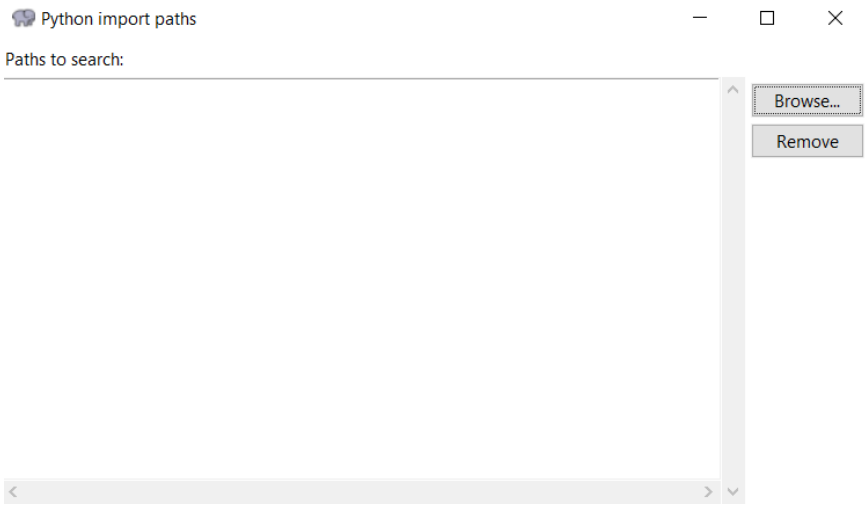


Figure 6.4: The extra import paths popup

be removed by either selecting it and clicking the `Remove` button, or by right clicking and pressing `Remove directory`.

For instance, say one wished to create a fitting utilizing a look-up table program written as a separate Python module, and that the look-up table program file was called `LUT.py`. The fitting could be created as normal, and then the directory where `LUT.py` is located could be loaded into the `Import paths`. Once present, the module could be used by simply calling `import LUT` in the main code.

6.3.6 Syntax Highlighting

Code is automatically syntax highlighted. Supplied variables (`Zr`, `Zj`, and `freq`), fitting variables, and result variables (`Zreal`, `Zimag`, and `weighting`) are highlighted red. Included functions, the imaginary number, and Python built-in functions are highlighted purple. Strings are highlighted green, comments are highlighted gray, and Python reserved words are highlighted blue.

An example of a custom fitting formula using the parameters seen in Figure 6.2 is shown below:

```
Z = Re + R1/(1+1j*2*PI*freq*T1)
Zreal = Z.real
Zimag = Z.imag
```

6.3.7 Save Fitting

The `Save fitting` button will save the current fitting options, fitting parameters, code, description, equation, and a link to the loaded *.mmfile* as a new *.mmcustom* file.

6.4 Formula Files

The `Save formula` button can be used to save the current formula; the information saved includes the parameter names, values, choices (such as limits and multistart options), as well as the current code, extra import paths, description, and equation. This information is saved in a file with the extension *.mmformula*. The default save name is the current loaded formula, if any, and the default save directory is the formula directory. Note that *.mmformula* files are identical to *.mmcustom* files in every respect; the different extensions are merely used to indicate their different uses.

6.4.1 Formula description

The `Formula description` link above the code box can be clicked to open a popup. The textbox at the top of this popup can be used to hold a description; it can be helpful to talk about the code, parameters, or theory. Below the description textbox is an equation textbox. This accepts LaTex-style math formatting to produce an equation, which is displayed at the bottom of the window. The textbox ignores line breaks, but `//` can be used to create a line break in the displayed equation. Consult online guides for help with LaTex math formatting. An example equation is shown in Figure 6.5.

6.4.2 Load Formula

The `Load Formula` button can be used to import an existing *.mmformula* file through a popup window (see Figure 6.6).

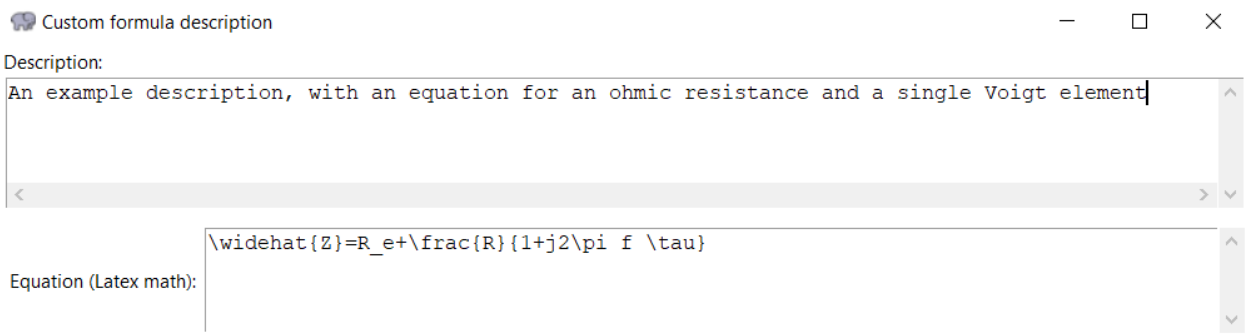
The default top-level path is as specified in Settings (see Chapter 7). This path can be changed by clicking the `Browse...` button. The top-level path will appear at the top of the listbox to the window's left. The directory can be expanded by double-clicking it or by clicking the `+` to the left. This will display all subdirectories (which can be further expanded), alongside all *.mmformula* files. The names are arranged in alphabetical order. Clicking a *.mmformula* file will load its description, code, and equation in the boxes to the right.

Once a formula has been selected, it can be loaded by clicking the `Load Formula` button at the bottom of the popup. The checkboxes to the right control what is loaded: `Parameters` will load the parameter names, values, and choices; `Code` will load the code; and `Other fitting settings` will load the number of Monte Carlo simulations, the weighting, and the fit type. `Parameters` and `Code` are selected by default.

> **Warning!** Loading a formula will overwrite any existing code and parameters without prompting, and these will not be retrievable unless they were saved previously.

6.5 Regression Results

A fitting can be performed using the `Run fitting` button, which will attempt to run the Python code in the `Code` textbox while fitting the parameters under the `Fitting`



$$\widehat{Z} = R_e + \frac{R}{1 + j2\pi f\tau}$$

Figure 6.5: An example description with equation

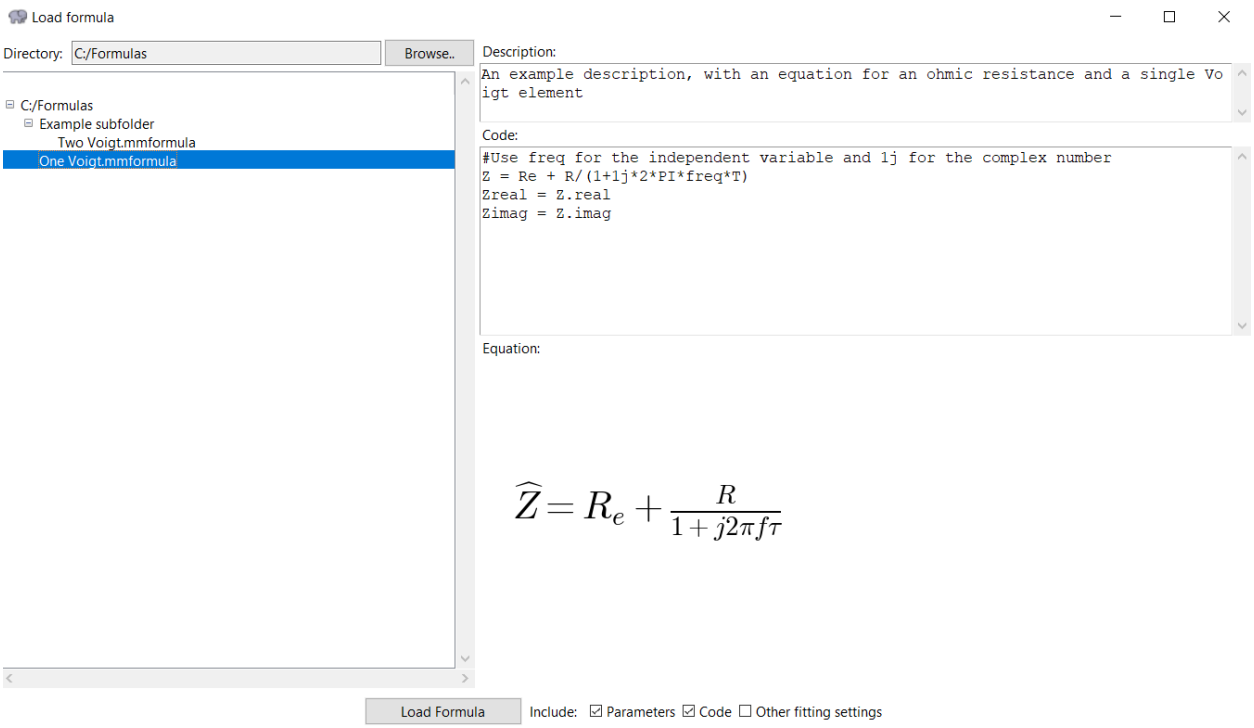


Figure 6.6: An example formula selected in the Load Formula popup

`Parameters` popup using a Levenberg-Marquardt algorithm. If successful, the results will appear in the bottom half of the tab (see Figure 6.7).

The `Run fitting` button will only be active if a file has been loaded. If there is a syntax error in the code the fitting will not run at all. If there is another error in the code or the result variables `Zreal` and `Zimag` are not defined the fitting will result in an error.

Upon a successful fitting, a listbox will appear with all fitted parameters, their values, and their standard deviations and confidence intervals. If a parameter has a confidence interval greater than 100% it cannot be considered statistically different than 0 and it will be highlighted yellow.

The `Advanced results` button will bring up a popup similar to the one discussed in Section 3.16, except that it does not include a correlation matrix.

The `Copy values and std. devs. as spreadsheet` button will include parameter names, values, and standard deviations.

The `Plot` button and `Include Confidence Interval` checkbox act in the same manner as described in Section 3.19. The same plots are present, except for the R_e-adjusted plots and the derivative plot.

The residuals can be saved to a *.mmresiduals* file using the `Save Residuals` button. These can be used in the Error File Preparation tab; the only difference from those created in the Measurement Model Tab is that R_e is set as 0 in the file.

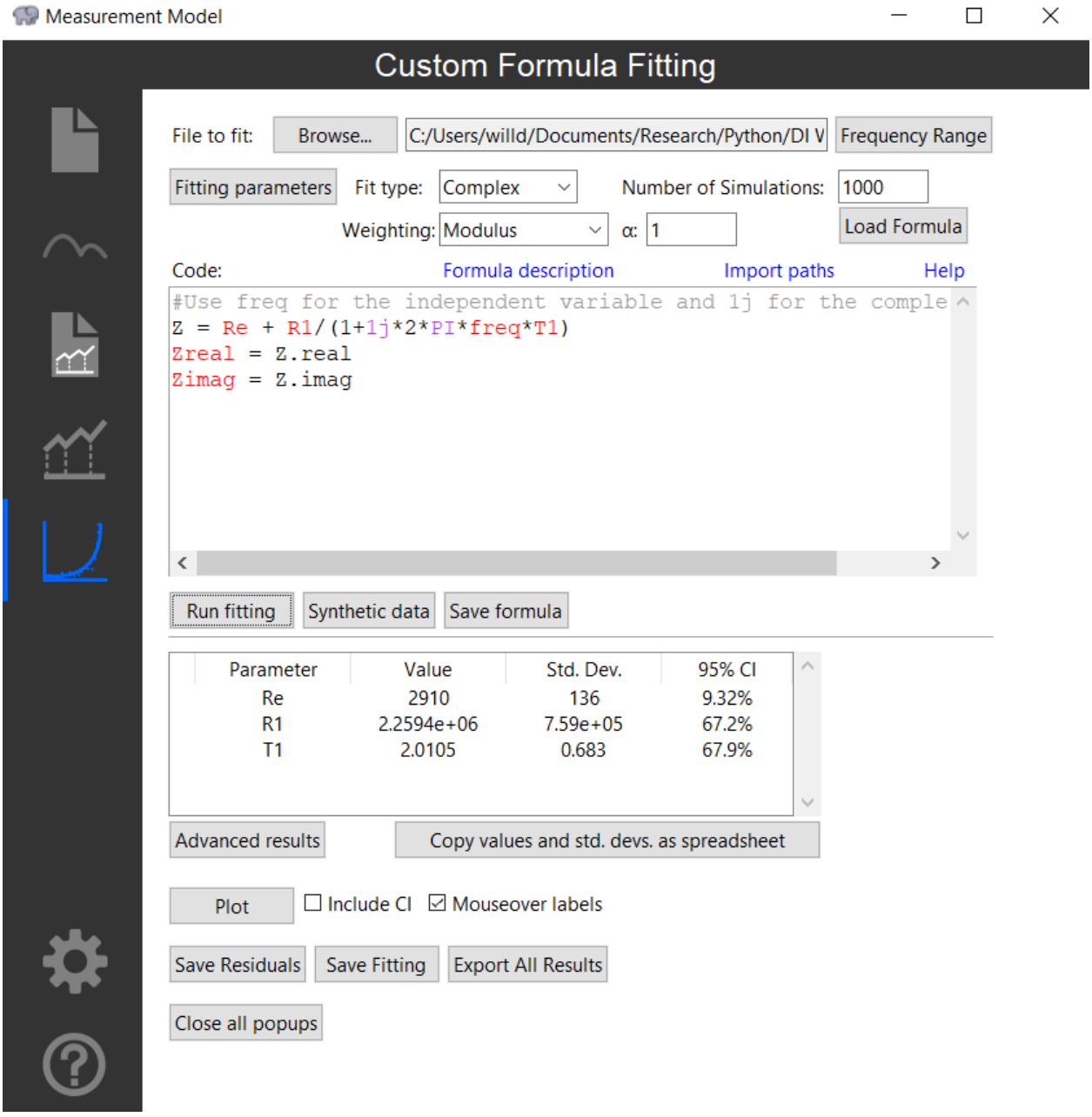


Figure 6.7: A successful fitting of the parameters in Figure 6.2 and the code in Section 6.3.6

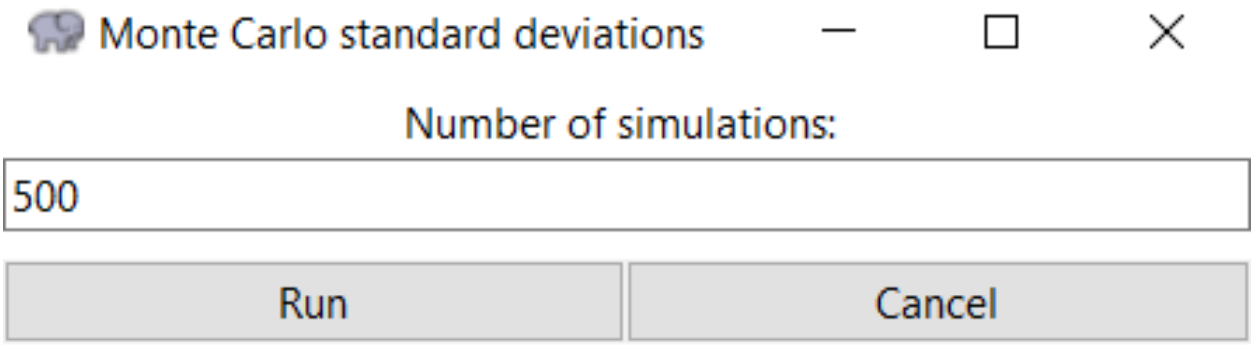


Figure 6.8: Choosing the number of Monte Carlo simulations when estimating parameter standard deviations

6.5.1 Monte Carlo standard deviations

Under certain circumstances, the fitting may succeed but be unable to find the standard deviations of the results. In this case, a prompt will appear after the fitting terminates asking whether or not to perform Monte Carlo simulations in an attemp to estimate the parameter standard deviations. If `No` is chosen, the fitting will finish as normal but with `nan` as each parameter's standard deviation. If `Yes` is chosen, another popup will appear asking for the number of Monte Carlo simulations to perform (see Figure 6.8).

Clicking `Cancel` will prevent any simulations from being performed (in the same manner as if `No` were chosen originally). Otherwise, `Run` can be clicked to perform the fits.

The program will then re-perform the fitting using the current code and original initial guesses. However, it will add normaly distributed "noise" to the data centered at 0 with a variance equal to the standard deviation of the residuals of the initial fitting. After the fits are performed, the standard deviation between the different fitted values of each parameter is used as that parameter's reported standard devation.

During this process, a progress bar will appear in the main window showing what percentage of the fits have been performed. If more than 1000 simulations are chosen, multiple processes will be spawned to perform the fits. The fits can be cancelled as usual with the `Cancel fitting` button; in this case, no Monte Carlo results will be reported.

6.6 Example

We fill fit a custom formula which will be for a single Voigt element in series with a resistance (see Equation 6.1).

$$Z = R_e + \frac{R_1}{1 + j\,(2\pi f)\,\tau_1} \tag{6.1}$$

1. First load a *.mmfile* under `Browse...`

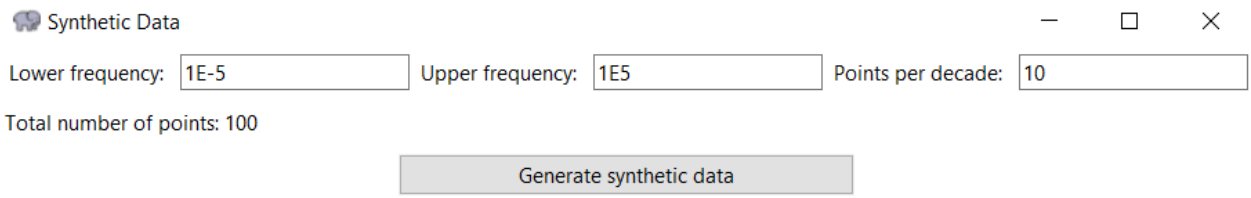


Figure 6.9: The synthetic data popup

2. Next, click `Fitting Parameters`, then click `Add Parameter` three times (one for each variable we will be fitting)
3. Rename `var0` to `Re`, `var1` to `R1`, and `var2` to `T1`, and type their initial guesses into the boxes to the right
4. In the dropdown to the right of `T1`, select `+` to constrain the time constant to be positive
5. Set the desired fit type, number of Monte Carlo simulations, and weighting strategy
6. Enter the following into the `Code` window (the code should syntax-highlight as seen below)

```
Z = Re + R1/(1+1j*2*PI*freq*T1)
Zreal = Z.real
Zimag = Z.imag
```

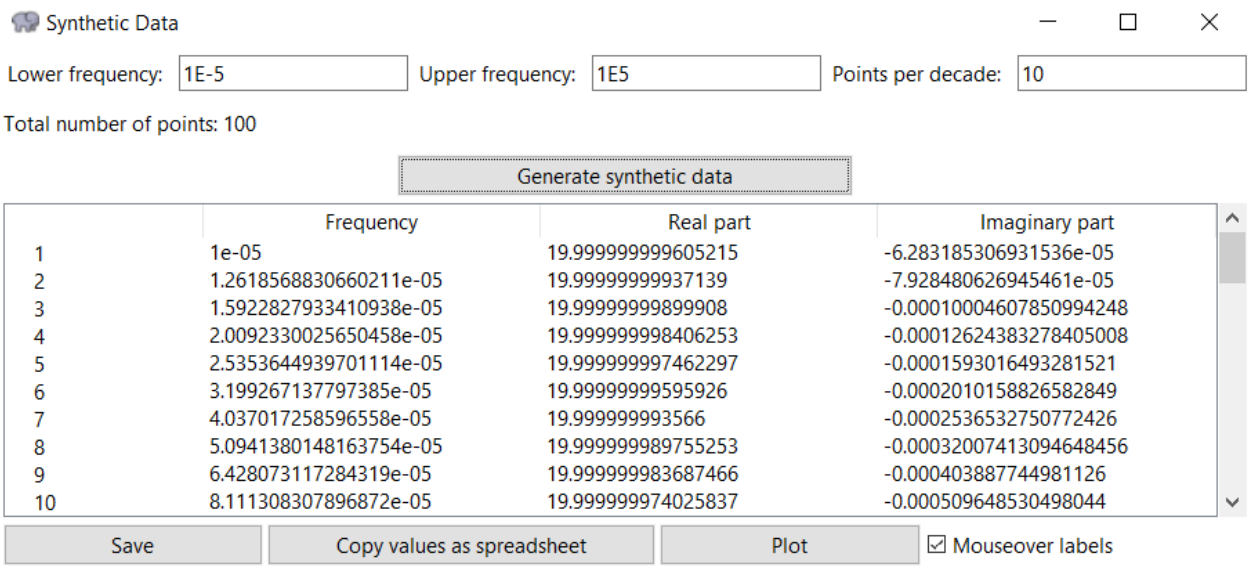
7. Click `Run fitting` and wait for results to appear

6.7 Synthetic data

Synthetic impedance data can be generated by using the current code and parameter values with the `Synthetic data` button. The button will open a popup (see Figure 6.9) where the frequencies can be specified at which data will be generated.

The leftmost textbox holds the lowest frequency, the middle box the uppermost frequency, and the rightmost box holds the number of data points per decade. The total of data points (calculated as the number of decades between the lower and upper frequency multiplied by the number of points per decade) is shown below.

After the choices are made, pressing the `Generate synthetic data` button will perform the calculations and display the results (see Figure 6.10). A listbox will appear with the frequencies to the left, the real impedance in the middle, and the imaginary impedance to the right. The data can be saved with te `Save` button as a measurement mode file (*.mmfile*), a text file (*.txt*), or a comma-separated values file (*.csv*). The data can also be copied in a format consistent with pasting to a spreadsheet program by using the `Copy values as spreadsheet` button. The `Plot` button will open a new window displaying plots of the synthetic data. These plots and the `Mousover labels` checkbox



	Frequency	Real part	Imaginary part
1	1e-05	19.999999999605215	-6.283185306931536e-05
2	1.2618568830660211e-05	19.99999999937139	-7.928480626945461e-05
3	1.5922827933410938e-05	19.99999999899908	-0.00010004607850994248
4	2.0092330025650458e-05	19.999999998406253	-0.00012624383278405008
5	2.5353644939701114e-05	19.999999997462297	-0.00015930164932815 21
6	3.199267137797385e-05	19.99999999595926	-0.0002010158826582849
7	4.037017258596558e-05	19.999999993566	-0.0002536532750772426
8	5.0941380148163754e-05	19.999999989755253	-0.00032007413094648456
9	6.428073117284319e-05	19.999999983687466	-0.000403887744981126
10	8.111308307896872e-05	19.999999974025837	-0.000509648530498044

Figure 6.10: Synthetic data generated for an example formula consisting of one Voigt element with R_e=10, R=10, and $\tau = 10$

work in the same manner as in the Measurement Model tab (see Section 3.19), but without the residuals plots, the R_e-adjusted plots, or the derivative plot.

Chapter 7

Settings

Many of the defaults in the program can be changed in the settings tab, accessible through the gear icon(⚙). The settings are organized into five sub-tabs, each corresponding to a different part of the program (see Figure 7.1).

7.1 Overall

The `Overall` tab includes settings that apply to the program as a whole. The `Light/Dark` toggle button will change the overall program theme (default: light). Light mode includes a white background and black text; dark mode includes a dark gray background and white text. The plot coloration will be changed as well. The `Side bar color` opens a color picker window where the color of the navigation and title panes can be set (default: dark gray). The tab title will change from white to black depending on its contrast with the color. The `Tab` dropdown is which tab the program opens to automatically (default: Input file). `Change tab on scroll`, if checked, will mean that the scrollwheel can be used to change tab so long as the mouse is over the navigation pane (default: checked).

7.2 Paths

`File directory` is the default directory for opening or saving files (default: C:\). `Formula directory` is the default directory for *.mmformula* files (default: C\). `Import paths` are the default extra paths searched by the Python interpreter for imports when performing a custom fit. The `Browse...` button here will add another directory to the listbox (instead of overwriting the current selection as the others do). To remove a directory from the listbox, select it and click `Remove` or right-click and select `Remove directory`.

7.3 Input

The `Detect number of comment lines` and `Detect delimiter` checkboxes make the program attempt to automatically detect the number of lines to ignore and the delimiter for data in the File Input and Conversion tab (default: checked). `Detect delimiter`

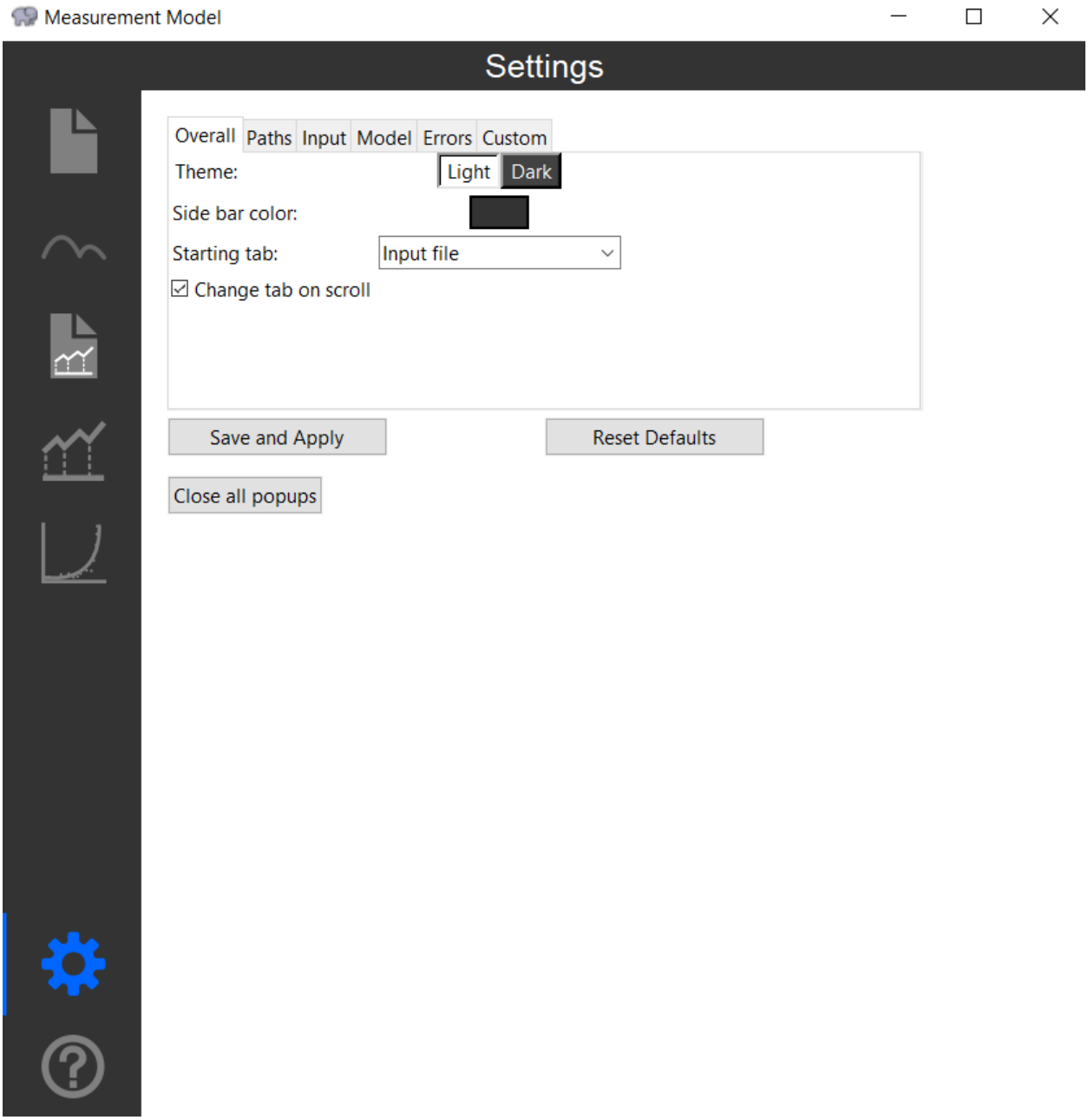


Figure 7.1: The Settings tab

can only be checked if Detect number of comment lines is checked. The textbox and dropdown beneath these two checkboxes is the default choice if automatic detection fails or is turned off (defaults: 0 and Tab, respectively). Alert on close if unsaved will - if checked - provide a confirmation alert if the program is closed while unsaved data is present in the File Input and Conversion tab (default: unchecked).

7.4 Model

Number of simulations, Fit type, Weighting type, and Assumed noise level (α) hold the default values for the fitting options described in Chapter 3 (defaults: 1000, Complex, Modulus, and 1, respectively). The Error lines/ellipse color button will open a color picker which will control the color of the confidence interval ellipses and lines when plotting in the Measurement Model or Custom Formula Fitting tabs (default: red). Keep frequency range on loading new file will, if checked, keep the number of points deleted when a new file is loaded (default: unchecked). Undo frequency range with undo button will, if checked, cause the Undo Fit button to remember the frequency range choices (default: unchecked).

7.5 Errors

The Error Model Parameters checkboxes control which parameters are selected by default (default: γ and δ). If Re is checked, it will check β as well; if β is unchecked, it will uncheck Re. The following three dropdowns control the default variance, moving average, and detrending options, respectively (defaults: Variance, 5 point, and Off, respectively).

7.6 Custom

The Alert on close if unsaved checkbox will - if checked - cause a confirmation prompt if a new custom formula has been entered, and the program is exited without saving it (default: unchecked). Keep frequency range on loading new file will - if checked - cause the frequency range choices (i.e. the number of points deleted) to remain the same when a new file is loaded (default: unchecked).

7.7 Saving and Resetting

The Save and Apply button will save the settings in a manner that will persist when the program is reloaded. It will also apply most appearance changes immediately (certain windows may need to be closed and re-opened for the changes to take effect). The Reset Defaults button will cause all settings to revert to their default state, and will apply those default settings.

Chapter 8

Help and About

The final tab is Help and About, which simply includes links to libraries used in the program, as well as contact information. Also included is a brief copyright notice that the program is distributed under the GNU General Public License. The license can be found online at https://www.gnu.org/licenses. A copy of the license is distributed with the program as an HTML file; it can be opened by clicking the `GNU General Public License` link at the bottom of the tab. The tab can be accessed either by clicking the question mark icon in the navigation pane () or by pressing `F1` from anywhere in the program.

8.1 Modules

Many modules/libraries were used in creating this program:

- The standard library
- Delimiter detection: detect_delimiter
- COM: comtypes
- Copying to/from clipboard: pyperclip
- File detection: impedance.py
- Fitting: LMFIT
- Graphing: matplotlib
- Images: PIL (Pillow)
- MPR decoding: galvani
- Numerical manipulations: numpy
- Other fitting: scipy

8.2 Contact

Dr. Mark Orazem (352) 392-6207
morazem@che.ufl.edu
William Watson williamdwatson1@gmail.com

Chapter 9

Procedure For Analyzing Impedance Data Files

The procedure for analyzing data is presented here in brief outline form.

9.1 Preliminary Analysis for Error Structure

1. Convert all data files to *.mmfile*s using the File Tools and Conversion tab (Chapter 2).
2. Perform complex fits for all replicate data using the Measurement Model tab (Chapter 3).
3. Obtain the maximum number of line shapes for each replicate.
4. Save the residual errors by using the `Save Residuals for Error Analysis` button. Note that, for a given set of replicates, the number of line shapes for each file needs to be the same in order to obtain the error structure. In addition, the number of line shapes for all files must be equal to the number of line shapes of the file with the minimum number of line shapes. For example if there are 5 files for a given set of replicates and 6 line shapes were obtained for 4 files and 5 line shapes for the remaining file, the files with six line shapes need to be saved with five line shapes. This can be done by repeating the regressions until five line shapes have been obtained.
5. Compile the residual errors using the Error File Preparation tab (Chapter 4).
6. Use the Error Analysis tab (Chapter 5) to calculate parameters for the error structure model replicates.

9.2 Assess Consistency with the Kramers-Kronig Relations

There are several ways to assess consistency with the Kramers-Kronig relations. In principle, since the Voigt model is itself consistent with the Kramers-Kronig relations, the

ability to fit this model to data within the noise level of the measurement should indicate that the data are consistent. An ambiguity exists when the data are not fully consistent because the lack of fit of the model could be due to causes other than inconsistency with the Kramers-Kronig relations. Some other possible causes could be that the number of frequencies measured was insufficient to allow regression with a large enough number of Voigt parameters or the initial guesses for the non-linear regression could be poorly chosen.

Regression to one component with subsequent prediction of the other component provides a more sensitive method to assess consistency. A procedure for this analysis is described below:

1. Perform a fit to the imaginary part of the spectrum using error structure weighting. Increase the number of lineshapes used until the maximum number of statistically significant parameters is obtained.

2. Examine the imaginary residual errors to determine if they fall within the error structure. Should a few points lie outside the error structure at intermittent frequency values do not be concerned. Assess prediction of real part of the impedance by examining real residual plots with confidence intervals displayed. Real residual data points that are outside the confidence interval are considered to be inconsistent with the Kramers-Kronig relations and should be removed from the data set.

3. Typically, the number of lineshapes that can be determined in a complex fit will increase when Kramers-Kronig -inconsistent data are removed. Deletion of data that are strongly influenced by bias errors increases the amount of information that can be extracted from the data. In other words, the bias in the complete data set induces correlation in the model parameters which reduces the number of parameters that can be identified. Removal of the biased data results in a better conditioned data set that enables reliable identification of a larger set of parameters.

Tip! R_e cannot be obtained by fitting the Voigt model to the imaginary part of the spectrum. In this sense, R_e should be treated as an adjustable parameter. The `Update Re` button in the Measurement Model tab - described in Section 3.20 - provides a convenient way to manually adjust the value of the solution resistance to determine whether a value can be found that would bring all the real part of the impedance into compliance with the Kramers-Kronig relations.

The use of measurement models is superior to the use of polynomial fitting because fewer parameters are needed to model complex behavior, and because the measurement model satisfies the Kramers-Kronig relations implicitly. Experimental data can, therefore, be checked for consistency with the Kramers-Kronig relations without actually integrating the equations over frequency, avoiding the concomitant quadrature errors. The use of measurement models does require an implicit extrapolation of the experimental data set, but the implications of the extrapolation procedure are quite different from extrapolations reported in the literature. The extrapolations done with measurement models are based on a common set of parameters for the real and imaginary parts and on a model

structure that has been shown to represent adequately the observations. The confidence in the extrapolation using measurement models is, therefore, higher. For the application to a preliminary screening of the data, the use of measurement models is superior to the use of more specific electrical circuit analogues because one can determine whether the residual errors are due to an inadequate model, to failure of data to conform to the Kramers-Kronig assumptions, or to experimental noise. The algorithm proposed in this work, in conjunction with weighting provides a robust way to check for consistency of impedance data.

It should be emphasized that the work presented here is part of an overall assessment of measurement errors. The measurement model is used as a filter for lack of replicacy to obtain a quantitative value for the standard deviation of the measurement as a function of frequency. The mean error identified in this way is equal to zero, thus the standard deviation of the measurement does not incorporate the bias errors. In contrast, the standard deviation of repeated impedance measurements typically includes a significant contribution from bias errors because perfectly replicate measurements can rarely be made for electrochemical systems. Since the line-shapes of the measurement model satisfy the Kramers Kronig relations, the Kramers Kronig relations can then be used as a statistical observer to assess the bias error in the measurement.

9.3 Modeling of Impedance Data

As used here, the measurement model provides much more than a preliminary analysis of impedance data in terms of the number of resolvable time constants and asymptotic values. The measurement model can be used as a filter for lack of replicacy that allows accurate assessment of the standard deviation of impedance measurements. This information is critical for selection of weighting strategies for regression, provides a quantitative basis for assessment of the quality of fits, and can guide experimental design. The measurement model is also used to assess the bias component of the . The next step in the interpretation of these data is the development of deterministic models which can account for the physical phenomena associated with this system. The analysis presented here can be used to ensure that the data used for comparison to the model are not corrupted by bias errors, thus facilitating interpretation in terms of physical parameters.

A preliminary interpretation can be obtained by using the `Evaluate Simple Parameters` button. The procedure would be to:

1. Perform a complex fit to the spectrum using error structure weighting and deleting data points found to be inconsistent with the Kramers-Kronig relations. Increase the number of lineshapes used until the maximum number of statistically significant parameters is obtained.

2. Press the `Evaluate Simple Parameters` button.

Chapter 10

Troubleshooting

From time to time, errors may arise. Errors generated by the program will most often include a number which can be referenced to Table 10.1.

Table 10.1: Error Codes

Error code	How to fix
2	When opening multiple files with the program, all extensions must be *.mmresiduals* or *.mmerrors* (exclusive or, no mixing)
3	There was an error opening the files (try opening them from the program when the program itself is running)
5	There was an error opening files (the file may be corrupted)
6	Number of values for frequency, real impedance, and imaginary impedance don't match; try manually editing file to remove bad points
7	Multiplier value is invalid; ensure that the multiplier is a real number
8	Column number is missing; ensure that the column numbers are all present, and are all positive integers
9	Problem reading the input file; don't manually edit *.mmfitting* files
10	File has an unknown extension; only *.mmfile* or *.mmfitting* files will work
11	The fitting failed (no minimization of the objective function could be found); try multistart/different initial guesses or a different number of parameters
12	A minimization was found, but parameter standard errors couldn't be estimated; try fitting again, or try multistart/different initial guesses/a different number of parameters
18	R_e's initial guess is negative but it is constrained positive; remove the initial guess or change the constraint

Error code	How to fix
19	One or more initial guesses are missing; ensure that all parameters have real-number initial guesses under `Edit Model Parameters`
20	One or more parameters has an initial guess whose sign does not match its constraint; change the initial guess or change the constraint
21	One or more of the initial guesses has an invalid value; initial guesses must be a real number
22	No parameters were chosen for error structure; choose at least one parameter
23	Error stucture has an invalid value; all values must be real numbers, even if not checked
24	All error structure parameters are 0 (similar to 22); make at least one parameter non-zero
25	The assumed noise (α) value is invalid; it must be a real number
26	Input file has an unknown extension; all files must have a *.mmresiduals* extension
27	Number of data is different between files; all input files must have the same number of data, so try manually editing the files or refitting under Measurement Model with the same number of data chosen using `Frequency Range`
28	Frequencies do not match; all frequencies must be the same between files - correct as in 27 or re-do measurements
29	Error reading file; the file may be corrupted
30	Input file has an unknown extension; all files must have a *.mmerrors* extension
31	Number of data do not match within a *.mmerrors* file; do not manually edit *.mmerrors* files
33	Error reading file; the file may be corrupted
34	Fiitting failed as a minimization of the objective function could not be found; try changing the error structure or using different initial guesses
35	Too few values for a 3-point moving average; at least 3 points are needed
36	Too few values for a 5-point moving average; at least 5 points are needed
37	Initial guess for α is invalid; the guess must be a real number
38	Initial guess for β is invalid; the guess must be a real number
39	Initial guess for γ is invalid; the guess must be a real number
40	Initial guess for δ is invalid; the guess must be a real number

Error code	How to fix
41	No parameters are chosen for fitting; choose at least one parameter
42	Error reading file; the file may be corrupted
43	Input file has an unknown extensions; it must have a *.mmfile* or *.mmcustom* extension
44	No formula is present; input a valid formula
45	A variable name is a Python reserved word; see online for a list of Python reserved words, and change the offending variable's name
47	A variable name is a word used in the fitting code (i.e. `Zr`, `Zj`, `freq`, `Zreal`, `Zimag`, or `weighting`); change the offending variable's name
48	A variable name is used more than once; ensure all variables have a unique name
49	Number of Monte Carlo simulations is invalid; ensure the `Number of Simulations` textbox holds a positive integer
50	Assumed noise (α) is invalid; ensure it is a real number
51	Number of comment lines is invalid; ensure it is a positive integer
52	Error applying or saving settings; ensure all values are valid, try saving again, or try restarting the program; do not move or manually edit the *settings.ini* file
53	The *.mmfile* linked in the loaded *.mmcustom* file could not be located; check that the file has not been moved, renamed, or deleted
54	One or more initial guesses is invalid; change the offending variable(s) value to a real number
55	The lower bound for multistart is invalid; change the lower bound to a real number
56	The upper bound for multistart is invalid; change the upper bound to a real number
57	The number of multistarts is invalid; change the number of multistarts to a positive integer
58	The signs of the upper and lower bounds when using logarithmic spacing are different; change the signs so that they match, or use a different spacing
59	Either the upper or lower bound is zero weh nusing logarithmic spacing; change the bound to be non-zero
60	The custom multistart choices couldn't be parsed; ensure that all custom mulltistart values are real numbers only, and that they are all separated by a comma
61	Either the upper or lower limit of a parameter is not a real number; change the offending limit to a real number

Error code	How to fix
62	The formula that was clicked could not be loaded; do not manually edit a .mmformula file
63	Specifiec import path does not exist; change the offending import path so it points to an exisiting directory
64	There is an invalid value for either the upper or lower frequency, or the number of simulated points per decade; ensure that the upper and lower frequencies are real numbers, and that the number of points per decade is a positive integer
65	The total number of simulated points is negative; ensure that the upper frequency is greater than the lower frequency and that the number of points per decade is a positive integer

References

1. P. Agarwal, M. E. Orazem, and L. H. García-Rubio, "Measurement Models for Electrochemical Impedance Spectroscopy: 1. Demonstration of Applicability," *Journal of the Electrochemical Society*, **139** (1992) 1917–1927.

2. P. Agarwal, O. D. Crisalle, M. E. Orazem, and L. H. García-Rubio, "Measurement Models for Electrochemical Impedance Spectroscopy: 2. Determination of the Stochastic Contribution to the Error Structure," *Journal of the Electrochemical Society*, **142** (1995) 4149–4158.

3. P. Agarwal, M. E. Orazem, and L. H. García-Rubio, "Measurement Models for Electrochemical Impedance Spectroscopy: 3. Evaluation of Consistency with the Kramers-Kronig Relations," *Journal of the Electrochemical Society*, **142** (1995) 4159–4168.

4. P. Agarwal, O. C. Moghissi, M. E. Orazem, and L. H. García-Rubio, "Application of Measurement Models for Analysis of Impedance Spectra," *Corrosion*, **49** (1993) 278–289.

5. M. E. Orazem, P. Agarwal, C. Deslouis, and B. Tribollet, "Application of Measurement Models to Electro-Hydrodynamic Impedance Spectroscopy," *Journal of the Electrochemical Society*, **143** (1996) 948–960.

6. M. E. Orazem, T. E. Moustafid, C. Deslouis, and B. Tribollet, "The Error Structure of Impedance Spectroscopy Measurements for Systems with a Large Ohmic Resistance with Respect to the Polarization Impedance," *Journal of the Electrochemical Society*, **143** (1996) 3880–3890.

7. M. A. Membrino, M. E. Orazem, E. Scott, and J. B. Phipps, "Electrochemical Impedance Measurements For Characterization of Ion Transport Pathways In Skin," in *Minutes: Transdermal Administration: A Case Study, Iontophoresis* (Paris, France: Éditions de Santé, 1997) 313.

8. I. Frateur, C. Deslouis, M. E. Orazem, and B. Tribollet, "Modeling of the Cast Iron/Drinking Water System by Electrochemical Impedance Spectroscopy," *Electrochimica Acta*, **44** (1999) 4345–4356.

9. M. Orazem, P. Agarwal, A. N. Jansen, P. T. Wojcik, and L. H. García-Rubio, "Development of Physico-Chemical Models for Electrochemical Impedance Spectroscopy," *Electrochimica Acta*, **38** (1993) 1903–1911.

10. M. Durbha, M. E. Orazem, and L. H. García-Rubio, "Spectroscopy Applications of the Kramers-Kronig Relations: Implications for Error Structure Identification," *Journal of the Electrochemical Society*, **144** (1997) 48–55.

11. J. Dygas and M. Breiter, "Variance of Errors and Elimination of Outliers in the Least Squares Analysis of Impedance Spectra," *Electrochimica Acta*, **44** (1999) 4163–4174.

12. J. R. Macdonald, "Some New Directions in Impedance Spectroscopy Data Analysis," *Electrochimica Acta*, **38** (1993) 1883–1890.

13. J. R. Macdonald and V. I. Piterbarg, "On the Transformation of Colored Random Noise by the Kronig-Kramers Integral Transforms," *Journal of Electroanalytical Chemistry*, **428** (1997) 1–9.

14. S. L. Carson, *Application of Complex Spectroscopic Techniques to the Characterization of particles in Suspension*, Ph.D. dissertation, University of Florida (1999).

15. S. L. Carson, M. E. Orazem, O. D. Crisalle, and L. H. García-Rubio, "Influence of Instrumentation on the Error Structure of Impedance Measurement," in *Electrochemical Approach to Selected Corrosion and Corrosion Control Studies*, P. L. Bonora and F. Deflorian, editors (London: The Institute of Materials, 2000) 344–360.