eistoolbox

for

MathWorks® MATLAB

A toolbox for batch fitting Electrochemical Impedance Spectroscopy data to equivalent circuit models

User Guide

M.Sc. Juan J. Montero-Rodríguez

Version 0.2-124

Contents

Chapter 1

Introduction

eistoolbox is a toolbox for MATLAB® used for batch fitting Electrochemical Impedance Spectroscopy (EIS) data to equivalent circuits models.

In EIS the **impedance** of a sample is recorded at different frequencies, by applying a small AC voltage signal to the sample, and measuring the current during the experiment. The impedance is stored as a complex number, with the magnitude expressed in Ohms (Ω) and the phase expressed in radians (rad).

As an example, consider the following experiment. An ionic solution of $CaCl_2$ is produced by diluting the salt in water, and stored in a 1 cm^3 cubic container. This chamber has two parallel-plate, stainless-steel walls. The impedance is measured from 0.1 Hz to 1 MHz. The experiment is repeated for concentrations of 600, 400, 200, 100, 50, 25, 12, 6, 3, 1.5, 0.8, 0.4, 0.2 mM and then water as reference. In the following magnitude plot, the top turquoise curve is the reference, and the bottom red curve is 600 mM.

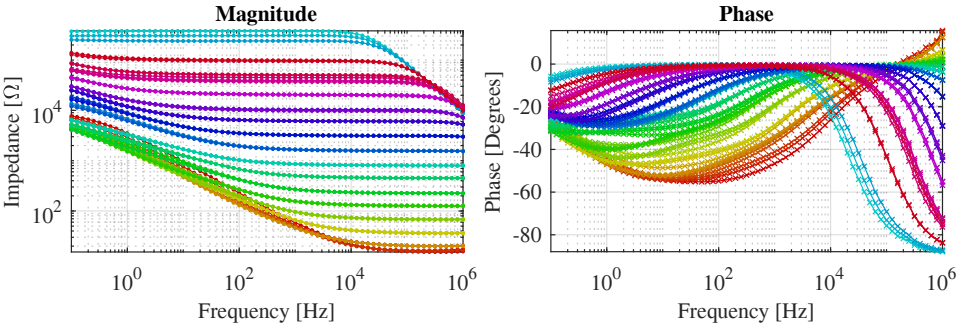


Figure 1.1: Impedance spectrum of $CaCl_2$ with different molar concentrations.

From the impedance curves of Figure 1.1 one can observe that the ionic concentration affects the impedance and the phase of the sample. For high ionic concentrations, the impedance magnitude is low since abundant ions contribute to the conduction. Low concentration of ions produces a higher impedance since the conductivity is lower.

1.1 Circuit modeling

This impedance is commonly fitted to circuit models, selected depending on the physical characteristics of the measured system.

For the $CaCl_2$ example, the experimental chamber can be described as a parallel-plate capacitor. The ionic solution has a resistance R_S, and the capacitance of the parallel plates is C_p. Two additional components have to be included, since the ions in the solution form a double-layer in the proximity of the electrodes. The double-layer is typically modeled by a charge-transfer resistance R_{ct} and a double-layer capacitance C_{dl} which dominate the low-frequency part of the impedance spectrum.

The result of the modeling process is a list of **circuit elements** such as resistors, capacitors and inductors, and a list of parameter values or **unknowns**.

The problem is how to calculate these parameters. This is often achieved using **nonlinear optimization**, and some of the most famous algorithms for fitting EIS data are based on the Complex Nonlinear Least Squares (CNLS) algorithm such as the Levenberg-Marquardt. However there are many algorithms, and they are difficult to implement as they require advanced mathematics.

The present **eistoolbox** is a collection of algorithms for nonlinear optimization, and a graphical user interface for batch fitting of multiple EIS curves.

Chapter 2

Theory

2.1 Impedance Spectroscopy

Electrochemical Impedance Spectroscopy (EIS) measures the complex impedance of a sample as a function of the frequency. An input voltage (2.1) is applied to the sample, and the current (2.2) is measured for different frequencies.

$$V_i = V \sin(\omega t) \tag{2.1}$$
$$I_i = I \sin(\omega t - \theta) \tag{2.2}$$

The impedance is calculated by means of the Ohm's Law by dividing (2.1) by (2.2)

$$Z(\omega) = \frac{V_i}{I_i} \tag{2.3}$$
$$Z(\omega) = \frac{V \sin(\omega t)}{I \sin(\omega t - \theta)} \tag{2.4}$$

Rewriting (2.4) in complex notation, using polar coordinates

$$Z(\omega) = \frac{V \angle 0^\circ}{I \angle -\theta} = |Z| \angle \theta \tag{2.5}$$

Equation (2.5) describes the impedance of the sample at all possible frequencies. The magnitude of the impedance is dependent of the frequency, since the current is measured independently at each frequency point. The phase is also frequency-dependent.

2.2 Dielectric Spectroscopy

The impedance of the sample, shown in (2.5), can also be rewritten in rectangular format, by converting the magnitude and phase to real and imaginary parts:

$$R = |Z| \cos\theta \tag{2.6}$$
$$X = |Z| \sin\theta \tag{2.7}$$
$$Z(\omega) = R + jX \tag{2.8}$$

where R is the resistance and X is the reactance of the sample.

The real part of the impedance is proportional to the resistivity, and the imaginary part is proportional to the permittivity. Both parameters can be calculated directly from the measurements, considering the exact geometry of the electrodes and measurement setup. For parallel plate electrodes, the following equations apply:

$$R = \frac{\rho L}{A} \tag{2.9}$$
$$C = \frac{\epsilon A}{D} \tag{2.10}$$

The capacitive reactance is given by

$$X_C = \frac{1}{\omega C} \tag{2.11}$$

Substituting (2.9), (2.10) and (2.11) into (2.8) results in the following equation (2.12), which describes the impedance in terms of the resistivity and permittivity of the sample between parallel electrodes:

$$Z(\omega) = \frac{\rho L}{A} + \frac{1}{\omega}\frac{D}{\epsilon A} \tag{2.12}$$

Impedance Spectroscopy is also referred as Dielectric Spectroscopy, because it gives information about the dielectric properties of the measured sample.

Chapter 3

Quick Start Guide

Steps for batch fitting impedance data curves:

1. Add files using the "Add file..." button (supported .CSV or .DTA)

Result: A plot of the input files is shown immediately after the files are loaded. Click on the plot buttons to obtain a Nyquist, Bode or Re/Im plot. Save this plot by clicking the save button.

2. Write the circuit string and fitting parameters (or load a .ckt file)

3. Select an algorithm and max iteration number

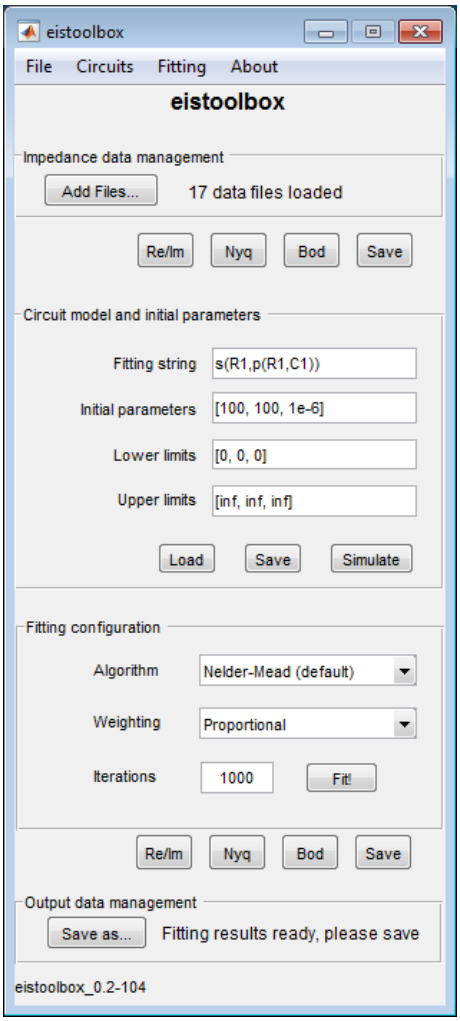
4. Click the "Fit" button

Result: A plot of the fitted results is shown immediately after the fit process. Click on the plot buttons to obtain a Nyquist, Bode or Re/Im plot. Save this plot by clicking the save button.

5. Save the results using the "Save..." button

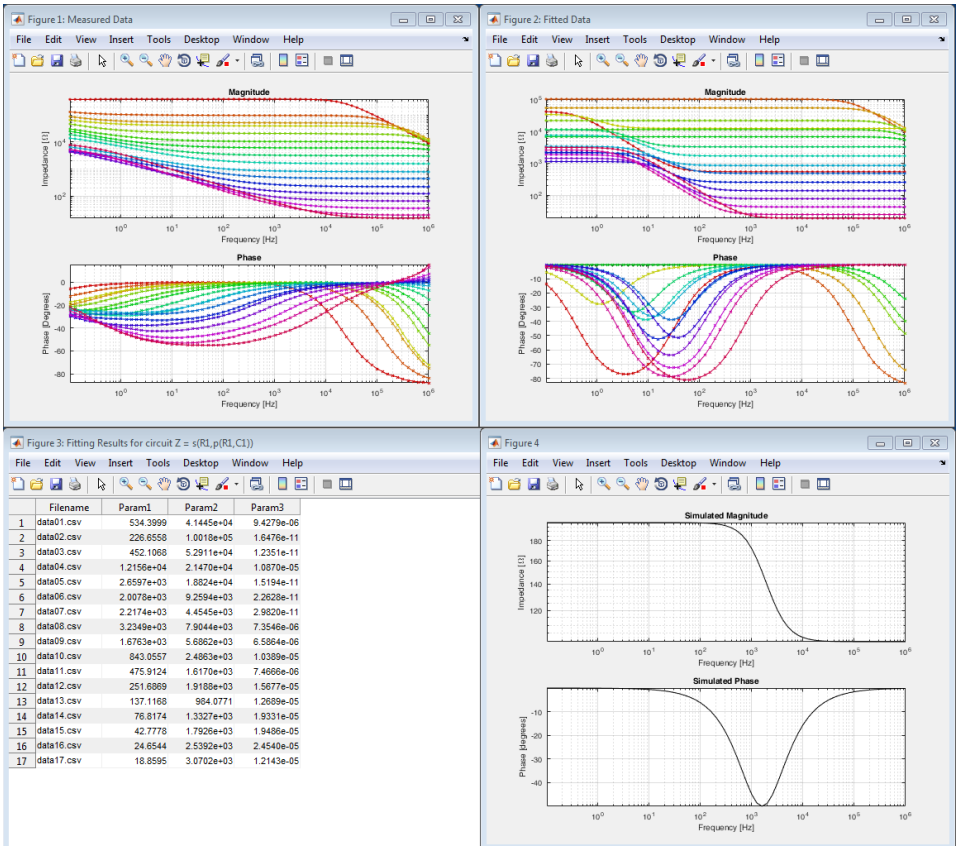
3.1 Graphical User Interface

3.1.1 Main Window



3.1.2 Results Window

The output parameters after the fitting are displayed as a table, in the same order as they appear in the circuit string.



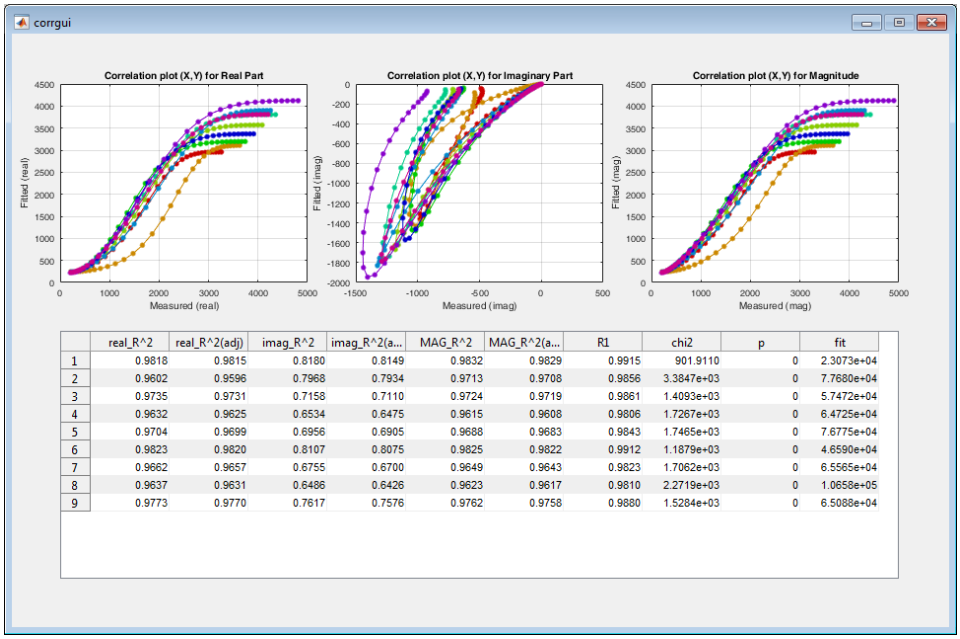
	Filename	Param1	Param2	Param3
1	data01.csv	534.3999	4.1445e+04	9.4279e-06
2	data02.csv	226.6558	1.0018e+05	1.6476e-11
3	data03.csv	452.1068	5.2911e+04	1.2351e-11
4	data04.csv	1.2156e+04	2.1470e+04	1.0870e-05
5	data05.csv	2.6597e+03	1.8824e+04	1.5194e-11
6	data06.csv	2.0078e+03	9.2594e+03	2.2628e-11
7	data07.csv	2.2174e+03	4.4545e+03	2.9820e-11
8	data08.csv	3.2349e+03	7.9044e+03	7.3546e-06
9	data09.csv	1.6763e+03	5.6862e+03	6.5864e-06
10	data10.csv	843.0557	2.4863e+03	1.0389e-05
11	data11.csv	475.9124	1.6170e+03	7.4666e-06
12	data12.csv	251.6869	1.9188e+03	1.5677e-05
13	data13.csv	137.1168	984.0771	1.2689e-05
14	data14.csv	76.8174	1.3327e+03	1.9331e-05
15	data15.csv	42.7778	1.7926e+03	1.9486e-05
16	data16.csv	24.6544	2.5392e+03	2.4540e-05
17	data17.csv	18.8595	3.0702e+03	1.2143e-05

3.1.3 Correlations Window

This window is useful to validate the fitting results.

There are three correlations implemented: real, imaginary and magnitude. For a perfect fit, the three plots are straight lines where y(x)=x.

Additionally, the fitting results are compared with the original data files by using the Pearson's Chi-square Test of Goodness-of-Fit. This information includes the following parameters: R, chi2, p, fit. These parameters are further explained in Chapter 5.



	real_R^2	real_R^2(adj)	imag_R^2	imag_R^2(a...	MAG_R^2	MAG_R^2(a...	R1	chi2	p	fit
1	0.9818	0.9815	0.8180	0.8149	0.9832	0.9829	0.9915	901.9110	0	2.3073e+04
2	0.9602	0.9596	0.7968	0.7934	0.9713	0.9708	0.9856	3.3847e+03	0	7.7680e+04
3	0.9735	0.9731	0.7158	0.7110	0.9724	0.9719	0.9861	1.4093e+03	0	5.7472e+04
4	0.9632	0.9625	0.6534	0.6475	0.9615	0.9608	0.9806	1.7267e+03	0	6.4725e+04
5	0.9704	0.9699	0.6956	0.6905	0.9688	0.9683	0.9843	1.7465e+03	0	7.6775e+04
6	0.9823	0.9820	0.8107	0.8075	0.9825	0.9822	0.9912	1.1879e+03	0	4.6590e+04
7	0.9662	0.9657	0.6755	0.6700	0.9649	0.9643	0.9823	1.7062e+03	0	6.5565e+04
8	0.9637	0.9631	0.6486	0.6426	0.9623	0.9617	0.9810	2.2719e+03	0	1.0658e+05
9	0.9773	0.9770	0.7617	0.7576	0.9762	0.9758	0.9880	1.5284e+03	0	6.5088e+04

Chapter 4

Equivalent Circuits

Experimental data is fitted to equivalent circuit models. The models are designed to describe the interfaces, chemical processes and boundaries of the measured setup.

4.1 Circuit Elements

The elements present in the software are described in Table 4.1:

Table 4.1: Equivalent circuit elements and their MATLAB implementation.

Symbol	Element	Equation
R1	Resistor	$Z(f) = R$
C1	Capacitor	$Z(f) = 1/j2\pi fC$
L1	Inductor	$Z(f) = j2\pi fL$
E2	Constant Phase Element	$Z(f) = 1/p_1(j2\pi f)^{p_2}$

The Warburg element can be obtained with a Constant-Phase Element by setting $p_2 = 0.5$

4.2 Circuit String Syntax

Circuits can be built using series and parallel combinations of the elements in Table 4.1, using the series and parallel operators s() and p(). These operators can contain any number of elements, separated by commas.

The number next to the element letter is the number of free parameters for this

element. For a capacitor (C1) the only free parameter is the capacitance. For the constant-phase element (E2) the free parameters are p1 and p2.

Common mistake: Do not write the circuit elements as s(R1,R2,R3,C4...). The elements cannot be written as labels. Instead, write s(R1,R1,R1,C1...).

4.3 Circuit Files (.ckt)

The circuit string, initial parameters and boundary conditions can be stored and loaded from a circuit file with the .ckt extension. This file is read by MATLAB line-to-line. The file should include only four lines with the following content:

Line 1: circuit string

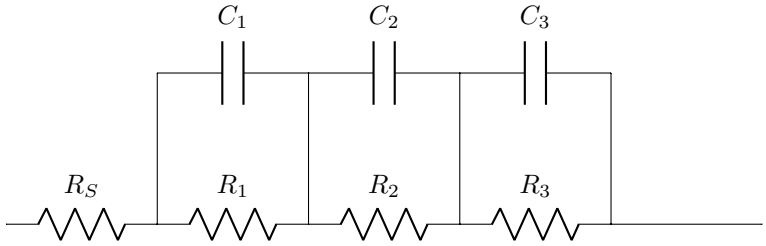
Line 2: initial parameters, sorted in the same order as they appear in the circuit string

Line 3: lower boundary conditions (LB)

Line 4: upper boundary conditions (UB)

Check the folder 'examples_circuits' for more examples.

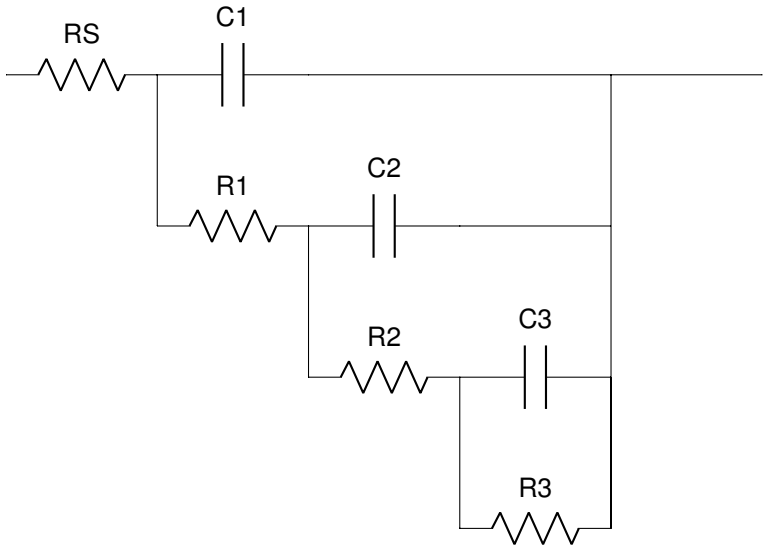
Example 1: Voigt model in the form R+R//C+R//C+R//C



Recommended circuit file
s(R1,p(R1,C1),p(R1,C1),p(R1,C1))
[100,100,1e-6,100,1e-6,100,1e-6]
[0,0,0,0,0,0,0]
[inf,inf,inf,inf,inf,inf,inf]

Notes
The elements appear in the circuit string in the following order: RS,R1,C1,R2,C2,R3,C3
The initial parameters, LB and UB have the exact same order

Example 2: Ladder circuit in the form: ((((R//C)+R)//C)+R)//C+R



Recommended circuit file

s(p(s(p(s(p(R1,C1),R1),C1),R1),C1),R1)
[100,1e-6,100,1e-6,100,1e-6,100]
[0,0,0,0,0,0,0]
[inf,inf,inf,inf,inf,inf,inf]

Notes

The elements appear in the circuit string in the following order: R3,C3,R2,C2,R1,C1,RS
The initial parameters, LB and UB have the exact same order

Chapter 5

Algorithms

This chapter includes the mathematical description of all the algorithms implemented in the program.

5.1 Weighting types

The fitting algorithms of this toolbox reduce the following distance function (Andrzej Lasia 2014).

$$dist = S = \sum_{i=1}^{n} \left(w_i'[Z_i' - Z_{i,calc}']^2 + w_i''[Z_i'' - Z_{i,calc}'']^2 \right) \tag{5.1}$$

The parameters w_i' and w_i'' are the weighting values, computed by one of the following methods:

Unit weighting	$w_i' = w_i'' = 1$
Modulus weighting	$w_i' = w_i'' = 1/\lvert Z\rvert^2$
Proportional weighting	$w_i' = 1/(Z')^2$ and $w_i'' = 1/(Z'')^2$
Statistical weighting	$w_i' = 1/(\sigma_i')^2$ and $w_i'' = 1/(\sigma_i'')^2$ (not implemented yet)

5.2 Minimization algorithms

5.2.1 Nelder-Mead (Simplex)

The toolbox uses the **fminsearchbnd** algorithm from John D'Errico.

5.2.2 Genetic Algorithm

ga()

5.2.3 Simulated Annealing

simannealbnd()

5.2.4 Constrained Minimization

fmincon()

5.2.5 Other Algorithms

Not implemented yet

- Levenberg-Marquardt
- BFGS
- Powell

5.3 Statistics

The overall quality of the fitting can be determined by comparing the original measured data file (expected values) to the simulated values using the fit results (observed values).

Currently the software implements two methods for this comparison: linear regression and Pearson's chi-square test of goodness-of-fit.

5.3.1 Linear Regression

Linear regression is used to compare the original input data with the fitted data.

- The real correlation coefficient (rsq_real) is obtained by linear regression of the fitted vs. measured real part of the impedance.
- The imaginary correlation coefficient (rsq_imag) is obtained by linear regression of the fitted vs. measured imaginary part of the impedance.
- The magnitude correlation coefficient (rsq_MAG) is obtained by linear regression of the fitted vs. measured magnitude of the impedance.

This is achieved with the following steps:

Linear fit using polyfit

$$p = polyfit(experimental, fitted, 1) \tag{5.2}$$

evaluate the line to get measured points

$$yfitted = polyval(p, experimental) \tag{5.3}$$

Residual values

$$y_{resid} = experimental - fitted \tag{5.4}$$

square the residuals and get the residual sum of squares

$$SS_{resid} = \sum (y_{resid})^2 \tag{5.5}$$

compute the total sum of squares by multiplying variance by n-1

$$SS_{total} = (n-1) * var(experimental) \tag{5.6}$$

compute R^2

$$R^2 = 1 - \frac{SSresid}{SStotal} \tag{5.7}$$

compute adjusted R^2 to account for degrees of freedom

$$rsq_{adj} = 1 - \frac{SSresid}{SStotal} * \frac{n-1}{n-p-1} \tag{5.8}$$

Where p is the degree of the polynomial used for the regression (in this case p=1: linear)

5.3.2 Chi-square Goodness-of-Fit

The Chi-square parameter is calculated with the following equation (Andrzej Lasia 2014).

$$\chi^2 = \sum_{i=1}^{n} \left(\left[\frac{Z'_i - Z'_{i,calc}}{\sigma'_i} \right]^2 + \left[\frac{Z''_i - Z''_{i,calc}}{\sigma''_i} \right]^2 \right)$$

This parameter depends on the number of points.

Usually it is divided by the number of degrees of freedom $\nu = 2N - m$

$$\chi^2_v = \frac{chi^2}{\nu} = \frac{chi^2}{2N-m}$$

N is the number of frequencies: there are 2N measured impedance points (N real and N imaginary)

m is the number of adjustable parameters in the model

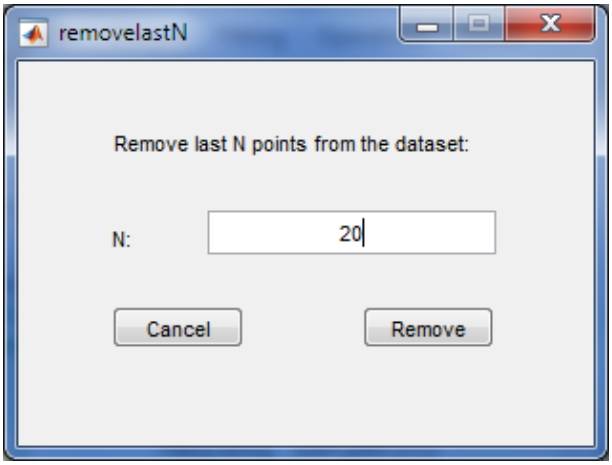
Chapter 6

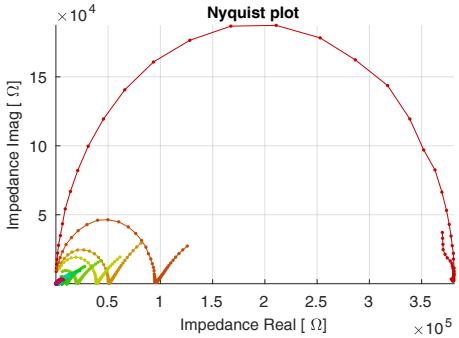
Operations menu

This allows to perform data operations on the RAM memory without modifying the original files. The user can remove any number of points from the end of the files, remove data points higher than a threshold, or plot the data at a specific frequency without fitting.

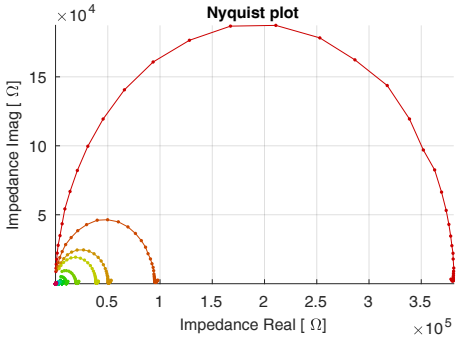
6.1 Remove last N points

This removes N points from the low-frequency part of the spectrum.





(a) Original data without modifications.



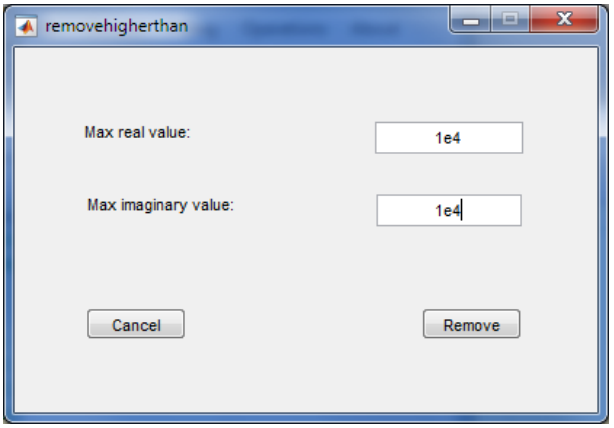
(b) After removing 20 points.

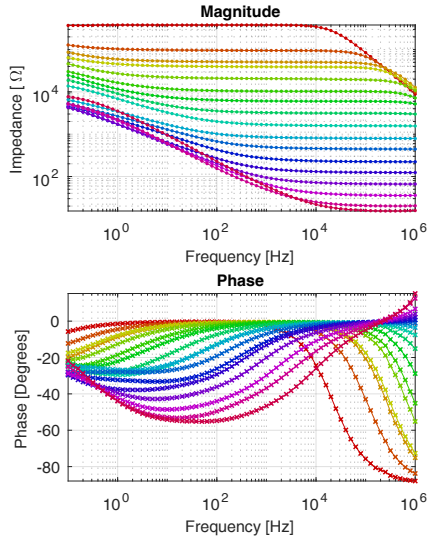
6.2 Remove values higher than

This enables to remove points with very high impedance. This can happen when using an Agilent 4284A or similar devices with the GPIB interface, where callibration errors result in high-impedance data points.

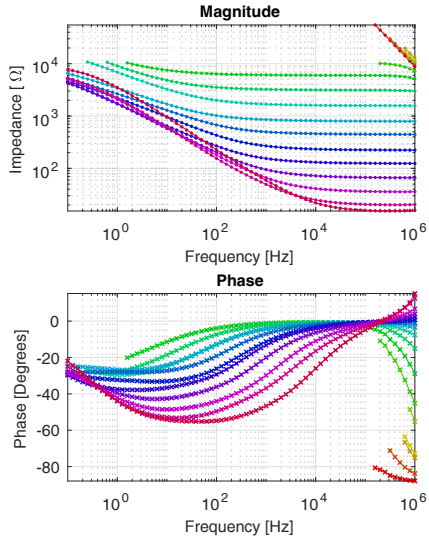
The user should be careful not to remove ALL data points accidentally. This can happen if the threshold is too low.

Applying this to the sample files provided:





(a) Original data without modifications.



(b) After removing points $>$ 10e4.

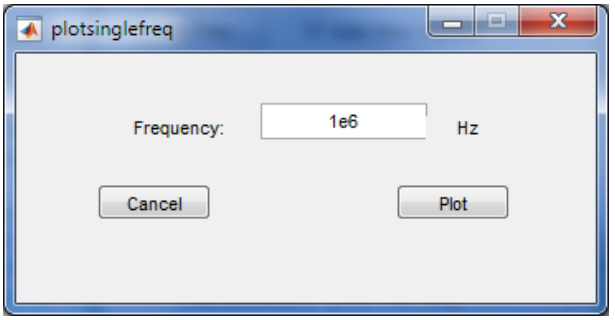
6.3 Plot data at single frequency

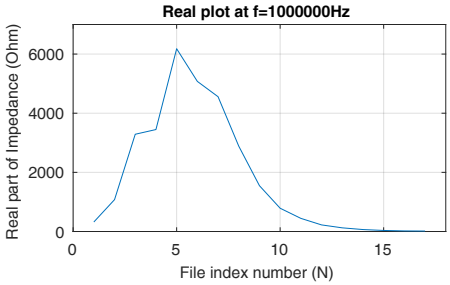
This option plots the data at a single frequency.

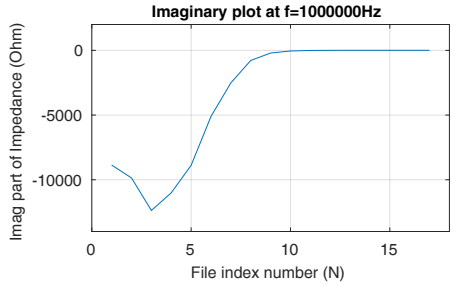
The user can input the desired frequency, and data is plotted at the nearest found frequency point.

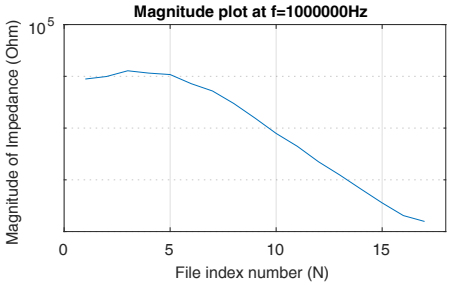
Notice that if some data was removed earlier, this plots the next closest value.

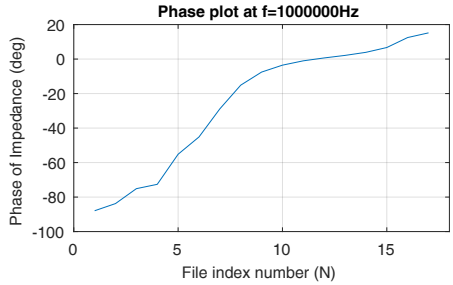
For the files provided as example, plotting the data at a frequency of 1 MHz:




Real plot at f=1000000Hz
Real part of Impedance (Ohm)
6000
4000
2000
0
0
5
10
15
File index number (N)


Imaginary plot at f=1000000Hz
Imag part of Impedance (Ohm)
0
-5000
-10000
0
5
10
15
File index number (N)


Magnitude plot at f=1000000Hz
Magnitude of Impedance (Ohm)
10^5
0
5
10
15
File index number (N)


Phase plot at f=1000000Hz
Phase of Impedance (deg)
20
0
-20
-40
-60
-80
-100
0
5
10
15
File index number (N)

Chapter 7

Licenses for included software

7.1 Zfit

The original script is available at [1].

7.2 fminsearchbnd

This function was written by John D'Errico and published on MathWorks MATLAB Central under an open-source license. The original file can be downloaded at [2]. The function is based on fminsearch and includes the possibility of using boundary conditions, such as the lower and upper limits for the individual circuit parameters.

7.3 export_fig

This tool is used to save plots as PDF.

Bibliography

[1] Jean-Luc Dellis. Zfit: function which can PLOT, SIMULATE and FIT impedance data, 2010.

[2] John D'Errico. Bound constrained optimization using fminsearch, 2012.