eistoolbox

for

MathWorks® MATLAB

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

User Guide

Juan J. Montero-Rodríguez

Version 0.1-54

Contents

Chapter 1

Introduction

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

Currently it is **alpha software**, and it will evolve over time.

1.1 Impedance Spectroscopy

Electrochemical Impedance Spectroscopy (EIS) measures the complex impedance of a sample as a function of the frequency. The experimental results are stored in two possible formats: polar coordinates (magnitude and phase) or rectangular coordinates (real and imaginary).

$$Z(f) = R + jX \tag{1.1}$$

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:

$$R = \frac{\rho L}{A} \tag{1.2}$$

$$C = \frac{\epsilon A}{D} \tag{1.3}$$

The capacitive reactance is given by

$$X_C = \frac{1}{2\pi f C} \tag{1.4}$$

Substituting (1.3) and (1.4) into (1.1) results in the following equation, which describes the impedance in terms of the resistivity and permittivity of the sample between parallel electrodes:

$$Z(f) = \frac{\rho L}{A} + \frac{1}{2\pi f}\frac{D}{\epsilon A} \tag{1.5}$$

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

Chapter 2

Quick Start Guide

Steps for batch fitting impedance data curves:

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

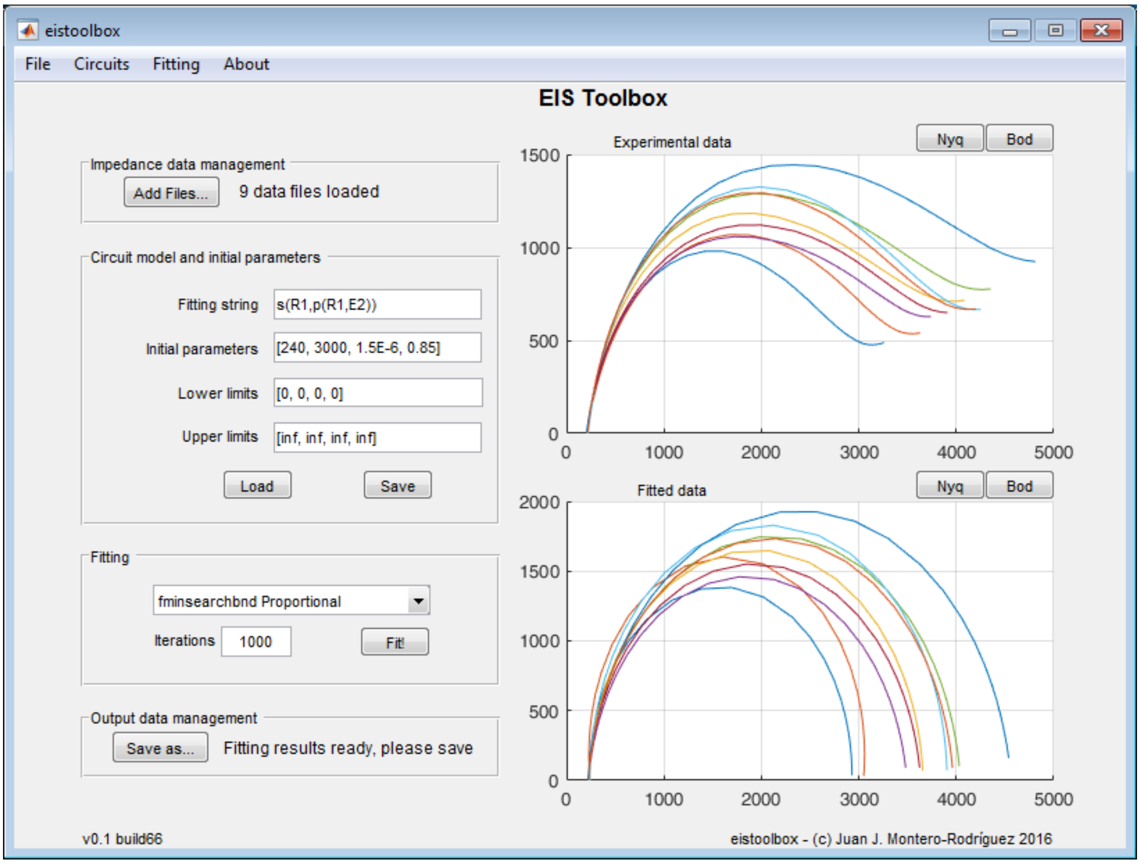
2. Write the circuit string and fitting parameters (circuit string formatting, see Chapter 3)

3. Click the "Fit" button

4. Save the results using the "Save..." button

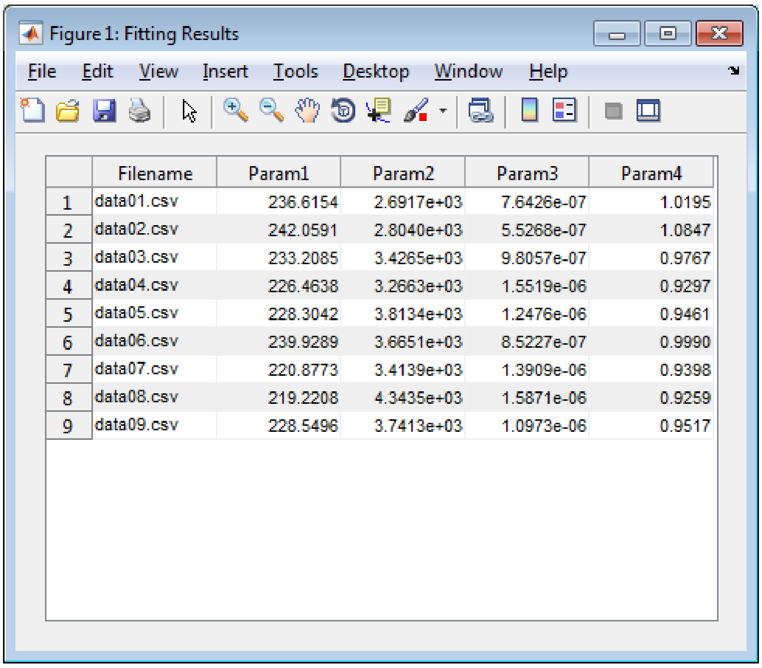
2.1 Graphical User Interface

2.1.1 Main Window



2.1.2 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	Param4
1	data01.csv	236.6154	2.6917e+03	7.6426e-07	1.0195
2	data02.csv	242.0591	2.8040e+03	5.5268e-07	1.0847
3	data03.csv	233.2085	3.4265e+03	9.8057e-07	0.9767
4	data04.csv	226.4638	3.2663e+03	1.5519e-06	0.9297
5	data05.csv	228.3042	3.8134e+03	1.2476e-06	0.9461
6	data06.csv	239.9289	3.6651e+03	8.5227e-07	0.9990
7	data07.csv	220.8773	3.4139e+03	1.3909e-06	0.9398
8	data08.csv	219.2208	4.3435e+03	1.5871e-06	0.9259
9	data09.csv	228.5496	3.7413e+03	1.0973e-06	0.9517

2.1.3 Correlations Window

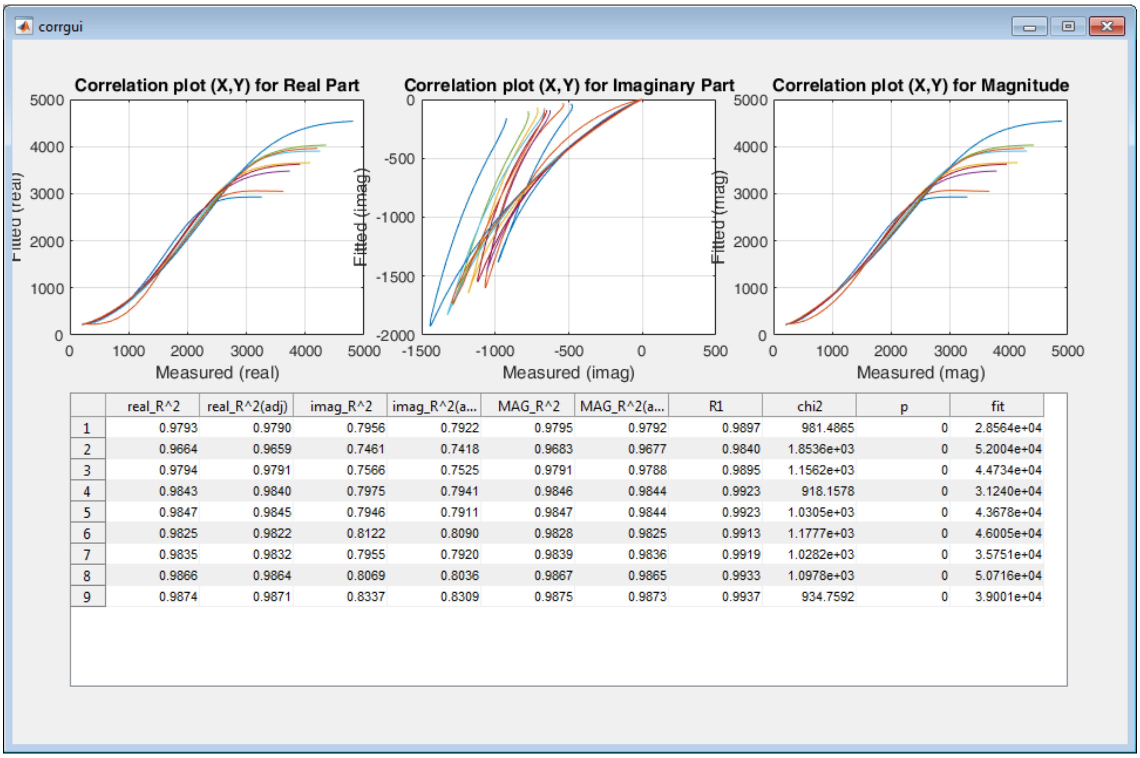
This window is useful to validate the fitting results.

There are three correlations implemented: real, imaginary and magnitude.

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

The plots themselves are presented as individual figures. 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	0.9793	0.9790	0.7956	0.7922	0.9795	0.9792	0.9897	981.4865	0	2.8564e+04
2	0.9664	0.9659	0.7461	0.7418	0.9683	0.9677	0.9840	1.8536e+03	0	5.2004e+04
3	0.9794	0.9791	0.7566	0.7525	0.9791	0.9788	0.9895	1.1562e+03	0	4.4734e+04
4	0.9843	0.9840	0.7975	0.7941	0.9846	0.9844	0.9923	918.1578	0	3.1240e+04
5	0.9847	0.9845	0.7946	0.7911	0.9847	0.9844	0.9923	1.0305e+03	0	4.3678e+04
6	0.9825	0.9822	0.8122	0.8090	0.9828	0.9825	0.9913	1.1777e+03	0	4.6005e+04
7	0.9835	0.9832	0.7955	0.7920	0.9839	0.9836	0.9919	1.0282e+03	0	3.5751e+04
8	0.9866	0.9864	0.8069	0.8036	0.9867	0.9865	0.9933	1.0978e+03	0	5.0716e+04
9	0.9874	0.9871	0.8337	0.8309	0.9875	0.9873	0.9937	934.7592	0	3.9001e+04

Chapter 3

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.

3.0.1 Circuit Elements

The elements present in the software are described in Table 3.1:

Table 3.1: Equivalent circuit elements and their MATLAB implementation.

Symbol	Element	Equation	MATLAB expression
R1	Resistor	$Z(f) = R$	`z=p*ones(size(f))`
C1	Capacitor	$Z(f) = 1/j2\pi fC$	`z=1./(1i*2*pi*f*p)`
L1	Inductor	$Z(f) = j2\pi fL$	`z=1i*2*pi*f*p`
E2	Constant Phase Element	$Z(f) = 1/p_1(j2\pi f)^{p_2}$	`z=1./(p(1)*(1i*2*pi*f).^p(2))`

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

3.0.2 Circuit String Syntax

Circuits can be built using series and parallel combinations of the elements in Table 3.1, 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.

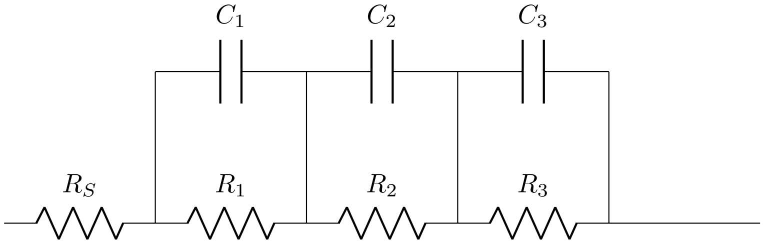
Common mistake: Do not write the circuit elements as s(R1,R2,R3,C4...). The elements cannot be written as labels. Instead, write s(R1,R1,R1,C1...).

3.0.3 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.

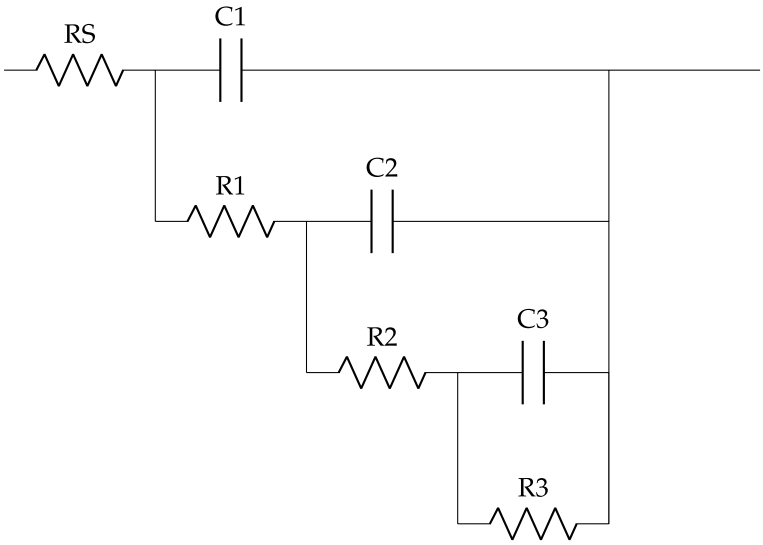
Check the folder ’examples_circuits’ for more examples.

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)) % circuit string
[100,100,1e-6,100,1e-6,100,1e-6] % initial parameters
[0,0,0,0,0,0,0] % lower boundary conditions
[inf,inf,inf,inf,inf,inf,inf] % upper boundary conditions

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) % circuit string
[100,1e-6,100,1e-6,100,1e-6,100] % initial parameters... order: R3,C3,R2,C2,R1,C1,RS
[0,0,0,0,0,0,0] % lower boundary conditions
[inf,inf,inf,inf,inf,inf,inf] % upper boundary conditions

Chapter 4

Fitting Algorithms

The fitting algorithms of this toolbox reduce the following distance function:

```
function dist=distance(param)
    ymod=feval(handlecomputecircuit,param,circuitstring,freq);
    if isequal('fitNP',fitstring)
        dist=sum(sum((ymod-zrzi).^2));
    else
        dist=sum(sum(((ymod-zrzi)./zrzi).^2));
    end
end
```

For **non-proportional fitting** the algorithm minimizes the squared difference between fitted (observed) and measured (expected) data.

$$d = \sum (o-e)^2$$

For **proportional fitting** the algorithm calculates the squared difference between fitted (observed) and measured (expected) data, divided by the measured (expected) data, in a way similar to the Pearson's chi-square test:

$$d = \chi^2 = \sum \frac{(o-e)^2}{e}$$

If the experimental impedance curve matches exactly the simulated curve, the distance function would have a value of zero.

To minimize the distance function, the toolbox uses the **fminsearchbnd** algorithm.

4.1 Other algorithms

The following optimization algorithms are not yet implemented in this toolbox. These are some of the most used algorithms for fitting EIS data to equivalent circuit models.

- Levenberg-Marquardt
- Nelder-Mead
- BFGS
- Powell

Chapter 5

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.

5.1 Linear regressions

5.1.1 Real of fitted vs Real of measured

5.1.2 Imag of fitted vs Imag of measured

5.1.3 MAG of fitted vs MAG of measured

5.2 Chi-square goodness of fit

$$\chi^2 = \sum_i^n \frac{(Observed_i - Expected_i)^2}{Expected_i}$$

Observed= fitted data

Expected= measured data

5.3 Error estimates for individual parameters

ToDo

Chapter 6

Licenses for included software

6.1 Zfit

The original file was released in 2005 and it is available here:

https://de.mathworks.com/matlabcentral/fileexchange/19460-zfit

6.2 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 https://de.mathworks.com/matlabcentral/fileexchange/8277-fminsearchbnd--fminsearchcon.

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.