ZfitGUI

User Manual

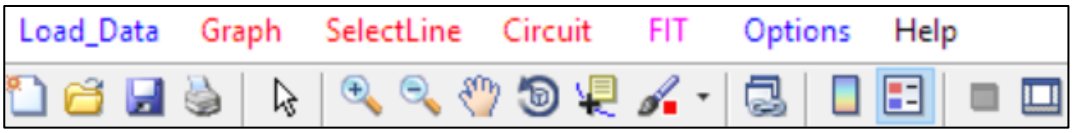
2020

Jean-Luc Dellis
Laboratoire de la Physique de la Matière Condensée, UFR des Sciences, Amiens, France

ZfitGUI is downloadable from the Mathworks 'file exchange'
and from my professionnal Web site :
https://sites.google.com/site/jeanlucdellis/ (programmes Matlab)

1 What ZfitGUI is for ?

ZfitGUI creates a figure with uimenus devoted to process impedance data.



The main features are:

Graphical.

The data can be plotted for each immitance types (Z, Y, C or M), in the complex plane, as well as their real, imaginary parts or their magnitude in function of the frequency (Log/Log scales).

Simulation.

The possibilities are almost infinite. Over the classical R, C, L electrical elements, one can use CPE (constant Phase Element) and Diffusion Impedances.

Fitting.

ZfitGUI makes use of the simplex algorithm. Parameter values may be fixed. To fit only portion of the spectra is user-friendly. Note that there exists the programmatic version of ZfitGUI named "Zfit" which can be found in the Matworks site file exchange.

1.1 The immitances (definitions)

Usually, the experimental data are gathered as impedance data: Z. The following definitions of the immitances (complex quantities computed from Z) apply in ZfitGUI :

ADMITTANCE: $Y = \frac{1}{Z}$

CAPACITANCE: $C = \frac{1}{Zj\omega}$

MODULUS(*): $M = \frac{1}{C} = Zj\omega$

(*) Care about the modulus. According to the definitions given by MacDonald, for instance in the article "Impedance Spectroscopy", Annals of Biomedical Engineering, Vol. 20, pp. 289 (1992) the modulus is the inverse of the complex dielectric constant (epsilon) which differs from the capacitance by a geometrical coefficient (Area/Thickness) and by the vacuum dielectric constant(epsilon zero). The definition used in ZfitGUI is made in order to not ask the geometrical coefficient to the user and to process the rough data only. Next computations to get the Epsilon complex or the complex conductivity for instance may be easily undertaken by the user with his own program. See also the MacDonald book : "Impedance spectroscopy: theory, experiment and applications".

1.2 Important note about fitting experimental data

Beware of the data which have small magnitude in impedance. Because the so-called "proportional weighting" method is used in fitting, these data may have heavy weights when the user could wrongly think they have not. That is important when these data are noisy or have a bias which might result in bad fitting. To solve this problem, choose a representation ‘z’ or ‘m’ or ‘c’ or ‘y’ where the spectrum shows well identified shapes, loop(s) or straight line(s). The user can also remove bad data before to process them ("Remove Outliers" menu).
On the other hand, when the fitting fails repeatedly, it is suggested to use the preceding result vector as an initial vector and to relaunch the fitting (it is the default state of the result figure option). If there is no success after several trials, maybe the Equivalent Circuit is not the right one. Restart with a simple p(R1,C1) or s(R1,C1) circuit then add components one by one. Also, it is recommended to read the present document and the hints given in the tutorial (in english) which can be found in https://sites.google.com/site/jeanlucdellis, repertory : programmes Matlab.

1.3 Limitations in using ZfitGUI

The user would not plot additional curves unless they were made in using ZfitGUI (Load Data or Simulation).

2 Load Data

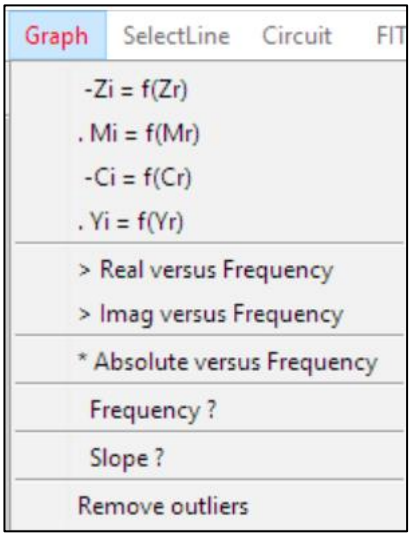
opens an UI tool to navigate to and open data files. The data must be numerical matrix with at least 3 columns. The first column must be the frequencies, the second, the real parts of the impedance data and the third, the imaginary parts.

$$\begin{matrix} f_1 & Z_{r1} & Z_{i1} \\ f_1 & Z_{r2} & Z_{i2} \\ . & . & . \\ . & . & . \\ f_{N-1} & . & . \\ f_N & Z_{rN} & Z_{iN} \end{matrix}$$

LoadData may open *.mat as well *.txt and *.xls files.
If no data are loaded, the user may simulate data in pressing the “Circuit” button.

3 Graph

GRAPH offers several tools :



3.1 -Zi = f(Zr), Mi = f(Mr), -Ci = f(Cr) and Yi = f(Yr)

display the immitances Z, complex capacitance (C), admittance (Y) or Modulus (M) in the complex plane. When the sign “minus” appears, it is to plot in the superior quadrant because the imaginary part is usually negative for conductive and capacitive systems.

3.2 Real, Imag and Absolute versus Frequency

displays the $\log_{10}$ of the real, the imaginary parts or the magnitude of the last plotted immittance in function of the $\log_{10}$ of frequency.

3.3 Frequency ?

displays the frequency of one or several points. Press the return key to deactivate the tool.

3.4 Slope ?

computes and displays the slope defined by 2 clicked points in any representations.

3.5 Remove outliers

allows to remove outliers. Press the return key to deactivate the tool.

3.6 Deleting graphic objects

All the graphic objects may be deleted by using the "Edit Plot" arrow of the Matlab figure. Activate the arrow, select the object with it, press the “del” key.

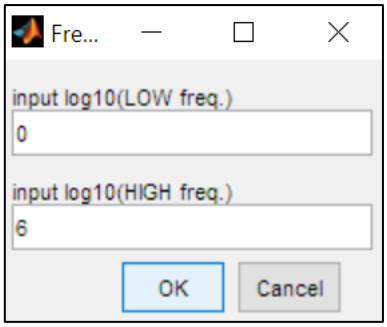
4 Select Line

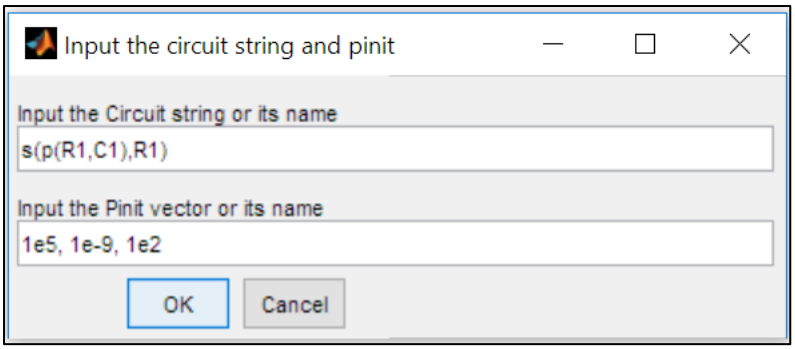
Because you might load several lines, you must select one before any fit.

5 Circuit

5.1 Frequency and the circuit string input

Enter a circuit string which describes the equivalent circuit to fit experimental data or to simulate impedance data when no lines are detected in the figure. In this last case, you are prompted to enter the logarithm frequency domain for simulations. If a line was selected, its frequency vector is used for the simulation and the fitting.





The circuit string (with no spaces) may be composed of the operators "s" and/or "p" which puts Z-elements in series or in parallel.
Attention : when copying/pasting strings or values in the dialog boxes, from Excel to ZfitGUI for instance, it could happen there is one or several spaces or other invisible characters pasted and arising to an error when the ZfitGUI code checks the string validity… In case of non-validity when all seems ok, remove all the spaces, place comas if needed, …

Examples :

p(R1,C1)	a resistor R and a capacitor C are put in parallel.
s(R1,p(R1,C1)	a R//C parallel circuit is put in series with a resistor.
s(p(R1,C1),p(R1,C1))	2 R//C circuits are put in series.
p(R1,C1,E2)	a Constant Phase Element is put in parallel with a R//C circuit.
C1	a capacitor alone (no operator used in the circuit string)

Notice the numeral after the letter : it gives the number of parameters used in the involved Z-element definition. Below the impedance definitions are given with *p(i)*, the parameters :

R1 resistor, $z = p(1)$
C1 capacitor, $z = \frac{1}{p(1)\ j\omega}$
L1 inductor, $z = p(1)\ j\omega$
E2 constant phase-angle element, $z = \frac{1}{p(1)\ (j\omega)^{p(2)}}$
G2 Finite Length Warburg with SHORT Circuit, $z = \frac{1}{p(1)\sqrt{j\omega}}\ tanh\left(p(2)\sqrt{j\omega}\right)$
H2 Finite Length Warburg with OPEN Circuit, $z = \frac{1}{p(1)\sqrt{j\omega}}\ \frac{1}{tanh\left(p(2)\sqrt{j\omega}\right)}$

So R, C and L will always have an "one" attached to them, when E, G and H has to be followed by the numeral 2. These components have subfunctions which compute their impedances.

5.2 The circuit initial parameters

The "circuit" uimenu also defines “Pinit”, the parameters *p(1), p(2),* ... values before fitting or simulating. The user has to input commas-separated values. These values have to come as their Z-elements come in the circuit string from left to right.
For instance, the circuit string s(p(R1,C1),p(R1,C1)) could be companioned with the parameter vector : p=[1e4, 2e-9, 3e4, 4e-8].
Each time "FIT" is clicked, the minimizer will start with these values or with those given in the ‘Results’ figure created from a preceding fit if this option is selected (see ‘Fit’ below). When difficulties are encountered for the convergence, it is suggested to use the preceding result vector as an initial vector.

5.3 User-defined function and how to set fixed parameters

User-defined function names may also be input. Here comes an example with an user-defined component "U3" described lower :

Circuit string = 'p(R1,U3)'
Initial parameter = [1e5,1e-10,0.98,1e4]

U corresponds to an user-defined function (whose name must be made of ONLY one letter, different of R, C, L, E, G or H already used internally). The function must have the format given here now. The function named here U is a m-file saved apart and must have 2 inputs : the vector of parameters p which has 3 elements in this example and the vector of frequencies :

```
function z=U(p,f)
z=1./(p(1)*1j*2*pi*f).^p(2)+p(3);
end
```

Because z needs 3 parameters to be computed, the name U must be followed by the numeral "3" in the circuit string (but not in the m-file name).
To end, the User-defined function is the way to handle “anchored” parameter value. Imagine you have a model with R, C and E all in parallel. Unfortunately, the fitting converged for the CPE to an exponent α different of the one you expected from theorical considerations. You want to fit the data with the expected, for instance, $\alpha = 0.75$ fixed. You might write a function which looks like :

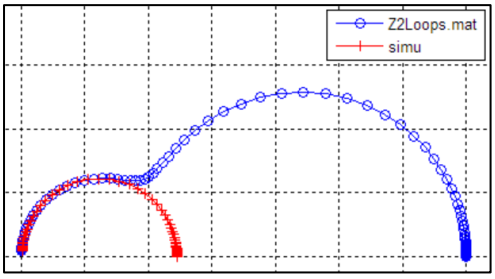
```
function z=U(p,f)
alpha=0.75;
z = 1/(p(1)*(1j*2*pi*f).^alpha)
end
```

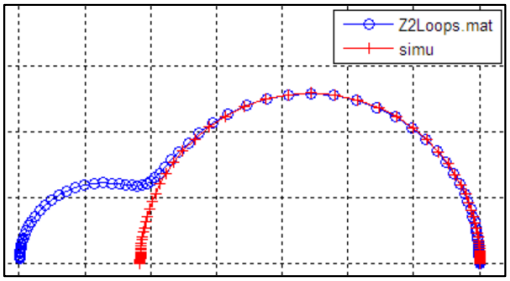
and the circuit string which would be : p(p(R1,C1),U1)

6 Fit

A line must be selected before to use 'Fit'. The user selects a portion of the spectrum or of the curve displayed in the figure by clicking twice on it at the chosen boundaries. That is allowing to fit only a part of the data in order to set up a model step by step. The weighting method described below, is proportional. 'Fit' is putting the excluded-points weights to ZERO. After the fit, the weights are automatically reset to their initial values.

Below, fitting result for few points on the left, then for few points on the right side of the spectrum :





The minimization engine searches the values of the parameters which minimizes the sum :

$$Khi2 = \sum \{weightmatrix * (model - exper)^2\}$$

Where :

$model = [Real(zmod), Imag(zmod)]$
$exper = [Real(zexp), Imag(zexp)]$
$zmod$ are the values returned by the modeling equivalent circuit,
$zexp$ are the experimental data which have to be fitted,
$weightmatrix = \left[\frac{1}{Real(zexp)}, \frac{1}{Imag(zexp)}\right]$

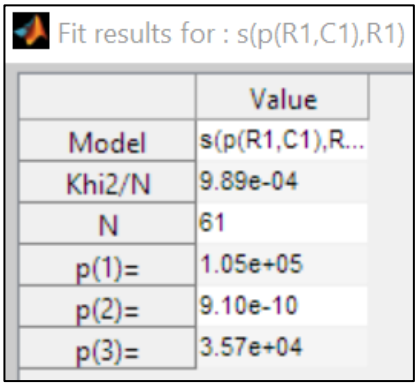
The matrix $weightmatrix$ is the 2-columns matrix of the weight coefficients associated with each point. It is the custom in impedance spectroscopy to put the weight coefficients equaled to $\frac{1}{zexp}$ in order that the high-valued points have the same weight than the low-valued points. This is the so-called proportional weighting.

After the process, ZfitGUI generates a figure with the results :

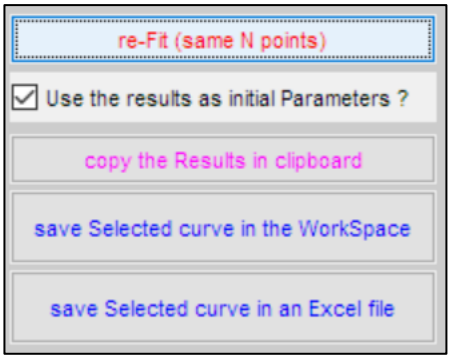
The circuit string is reminded.

The normalized Khi2/N is given as well as the current best parameters (N = amount of points processed).

You may modify the p(i) values to use them as initial parameters for the next fit trial.



- "re-Fit" launches again the fit on the same points previously framed using the current Pinit.
- When checked, the above p(i) are re-used for the next fit process.
- You may copy them in clipboard for any usage.
- Also, the data values of the current selected curve may be saved in an excel file or in the WorkSpace.

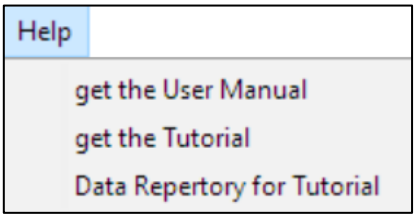


7 Options

There is only one option which results in displaying a legend when selected. The legend displays the file names of the current data. Note that the simulated data are labeled by the word ‘simu’.

8 Help

Clicking on the items make the PC to open the corresponding Web page to download documents or data.



9 Go deeper in the Outputs

A figure is created to display the fitting results. It is described in the ‘Fit’ paragraph above. About the internal informations, they are written in graphic objects, lines or figures using the Matlab setappdata. To get them and others, you might write at the Matlab prompt :

```
>> H_main=findobj('tag','main'); getappdata(H_main)
```

Which could result in :

```
cmyz: 'z'                                   % type of plot
filenamechoice: 1                           % display a legend
frsimu: [100x1 double]                      % simulation frequency vector
Scircuit: 's(p(R1,C1),R1)'                  % circuit string
Spinit: '1.03e+05,9.37e-10,3.68e+04'        % pbest string
zfg: [1x1 struct]                           % structure
```

The zfg structure fields contain the results displayed in the results figure and handles used by the code.

You then may try :

```
>> zfg=getappdata(H_main,'zfg')
```

to get something like :

```
weightmatrix: [85x2 double]                 % see the “Fit” paragraph
weightmatrix_0: [85x2 double]               %
circuit: 'p(s(R1,C1),R1)'                   %
pbest: [9.8335e+03 1.8657e-08 8.6716e+06]   %
zbest: [85x3 double]                        % [Fr, Zr, Zi] of the simulated line
chi2: 0.2019                                % see the “Fit” paragraph
exitflag: 1                                 %
output: [1x1 struct]                        % given by fminsearch
results: 91.0021                            % handle of the result figure
reuseP: 92.0022                             % handles
copyP: 93.0024                              %
saveSimu: 94.0024                           %
```

They are also informations stored in the lines. For instance, try :

```
>> H_selected=findobj('tag','selected'); getappdata(H_selected)
```

to get :

```
frzrzi: [85x3 double]                       % [Fr, Zr, Zi] of the experimental line
filename: 'maruse.mat'                      % the experimental line string to legend
```

10 History

janvier 2010
first release

avril 2012
- disable the uimenus when one is working
- improve the removeoutlier tool
- improve the way the user inputs the model
- add the uimenu 'Remove Element' to allow an identifying method of the equivalent circuit.

mai 2012
- improve the help and some other details
- outputs the structure ZFG which contains the fit results

mars 2015
- the Finite Length Warburg with SHORT-Circuit and with OPEN-Circuit boundaries have been added as available default elements.

janvier 2016
- some details corrected in the graph titles
- put usual values as default values for the elements R, C, E and G when prompting the user to simulate a circuit
- a filter remove possible "nans" in the input data which gave correct plots but turned the fitting to fail (typically, NAN can come from experimental data when the digital communication between the device and the computer has bugs).

fevrier 2016
- the submenu 'Fit a portion only' has been added. Now, the way is straightforward. The user may fit a portion of the spectrum defined by clicking twice on it to define the portion boundaries. That is putting the excluded-points weights to ZERO.

juillet 2018
- some sub-functions code has been simplified by using the native GINPUT function.
- Cleaning up the bugs indicated by the code analyzer.
- Solving some uimenu bugs in using their "enable" property.

décembre 2018
- many changes and simplifications to increase the efficiency. For instances, add a link to this document and a tutorial, put hints in the usages, put filenames in legend, use uigetfile to load data, use only a ‘fit portion’ method to simplify, the fitting results are now displayed in a figure with an uitable where they may be modified and used as initial parameters, code lines have been clean up.

février 2019
- an uicontrol list type is now used to load and check impedance data from the Workspace;
- the hints on “How to” are displayed on the figure.

décembre 2019
- the last loaded data are automatically selected for further fitting.
- the “Slope” tool from the “graph” menu has been improved in returning in the WorkSpace the slope and the intercept of the best line of selected data points.

Janvier 2020
- An important improvement concerning the way to define the model strings have been brought. Now the operators “s” and “p” may handle more than 2 elements as arguments. For instance the string

“p(R1,E2,C1,E2)” with 4 impedances in parallel is allowed. I would like to warmly thank Hazrul Shabri who offered the code in the File exchange of Mathworks.