やっぱり

Version 15 06 2023

YAPPARI

YAPPARI stands for Yet Another Program for Analysis and Research in Impedance.
This program can be referenced as http://dx.doi.org/10.13140/RG.2.2.15160.83200
If you are using Windows 10, you can download and install YAPPARI v5 (or a previous, simpler version Yappari 4.2) from https://github.com/nitad54448.
YAPPARI-5 is designed to multiple datasets fitting of the impedance spectra of a user-made circuit.
You are encouraged to contribute to the help file, you can send it to me or fork it on Github.

Panels

The program has several graphic panel windows with a parameter list and several commands grouped in the right side of the window.

Zr, -Zi

This panel shows a Nyquist plot, which is a standard way to visualize impedance data. The scale on the graph will adjust automatically based on the data, with the same axis range for the imaginary part and real part. However, if you want to manually set a specific range, you can disable the Auto-axis feature by clicking on the graph, or directly changing the scale in the legend. Some other standard graph functions are available in the top left "palette" such as zoom in, out... etc. All graphic panels will plot experimental and simulated data (if any) of selected datasets. The change of the plot colors, style, etc.... in this graph will affect all the graphs.

Zr, Zi, ln R, theta

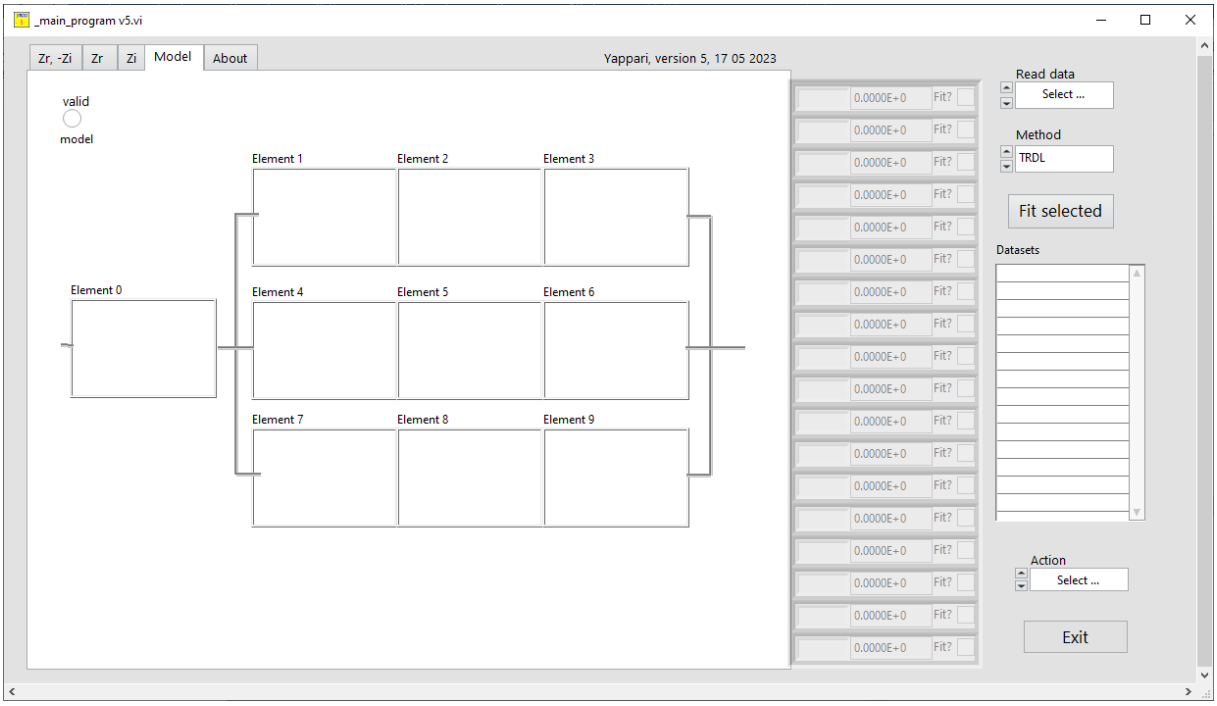
These panels will show the dependency of impedances (real, imaginary, modulus or phase) as a function of frequency and the differences between the calculated and experimental values (if any).

3D plot

This panel will show a 3D plot of selected datasets, either in Nyquist, Zr or Zi or their difference, as selected by the user. This is useful for many datasets, more than 20 I guess, it will allow the user to see tendencies or check systematic errors in the fits. You can right click on the graph to adjust plotting properties to your liking (3D Plot Properties) or change the size of the graph.

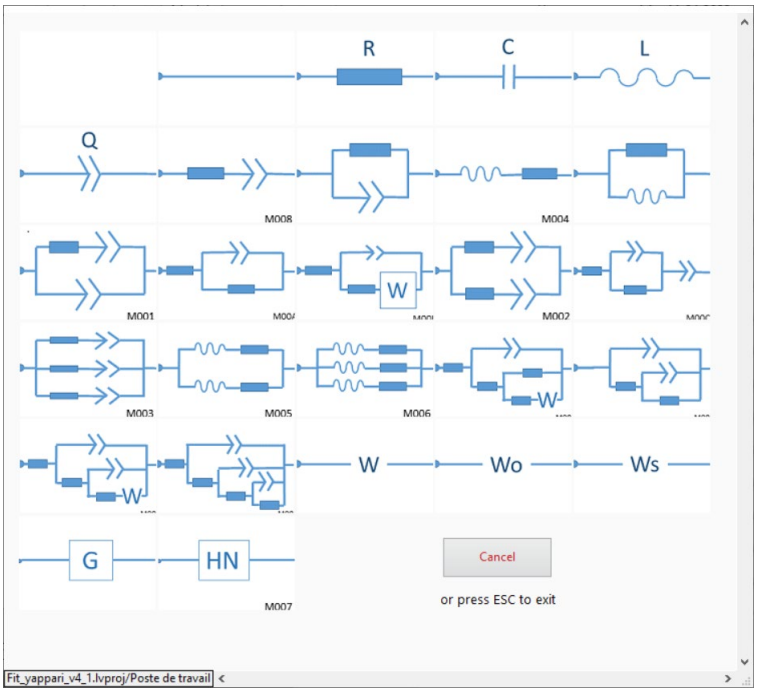
Model

A circuit can be created by the user by selecting circuits, on this page.



Up to ten elements can be added (obviously it is not realistic to fit such a circuit, unless you want to fit a crocodile). Only the first 18 parameters will be shown on the screen.

When you click on one of the ten available cases, a new window will appear where you can select the element you want to add. Simply click on the picture of the element you want to add and it will be added to the model. The available circuit elements include resistors, capacitors, inductors, and more complex elements such as constant phase elements or Warburg elements.



You can edit the png image files to your liking (just for aesthetics, the calculations are not affected), they are in the subdirectory /models. The size of the png files should be 150*100 pixels.
The elements used now (as of march 2023) are: Resistor, Capacitor, Inductor, CPE, Zarc, simple Randles circuit, Randles with kinetic and diffusion, Warburg (infinite diffusion, equivalent with a CPE with n=0.5 coefficient), Warburg short, Warburg Long, Gerischer; Havriliak-Negami and several compositions of these.
Warburg "infinite" diffusion coefficient σ is expressed here as:

$$Z_W = \sigma\omega^{-1/2} - j\sigma\omega^{-1/2}$$

The parameters obtained for Warburg in other programs are typically by fitting a CPE with n=0.5, you will get the same result but the Q parameter obtained is

$$\sigma = \frac{1}{\sqrt{2}\cdot Q_0}$$

The Warburg "open" describes the impedance of a finite-length diffusion with reflective boundary. The formula used here is

$$Z_{W_O} = \frac{A_W}{\sqrt{j\omega}}\coth\left(B\sqrt{j\omega}\right)$$

The Warburg "short" describes the impedance of a finite-length diffusion with transmissible boundary, with the expression:

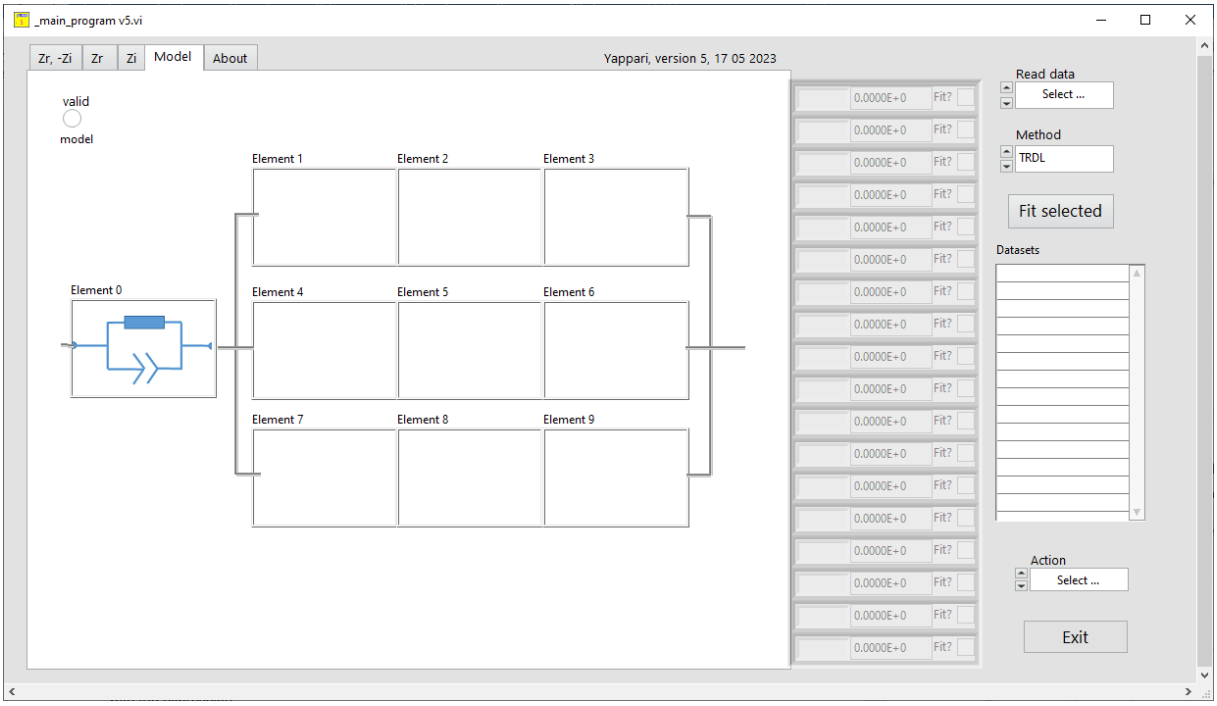
$$Z_{W_S} = \frac{A_W}{\sqrt{j\omega}}\tanh\left(B\sqrt{j\omega}\right)$$

Fitting a Warburg “short” will be very slow as it requires removal of some NANs for the high frequency values.

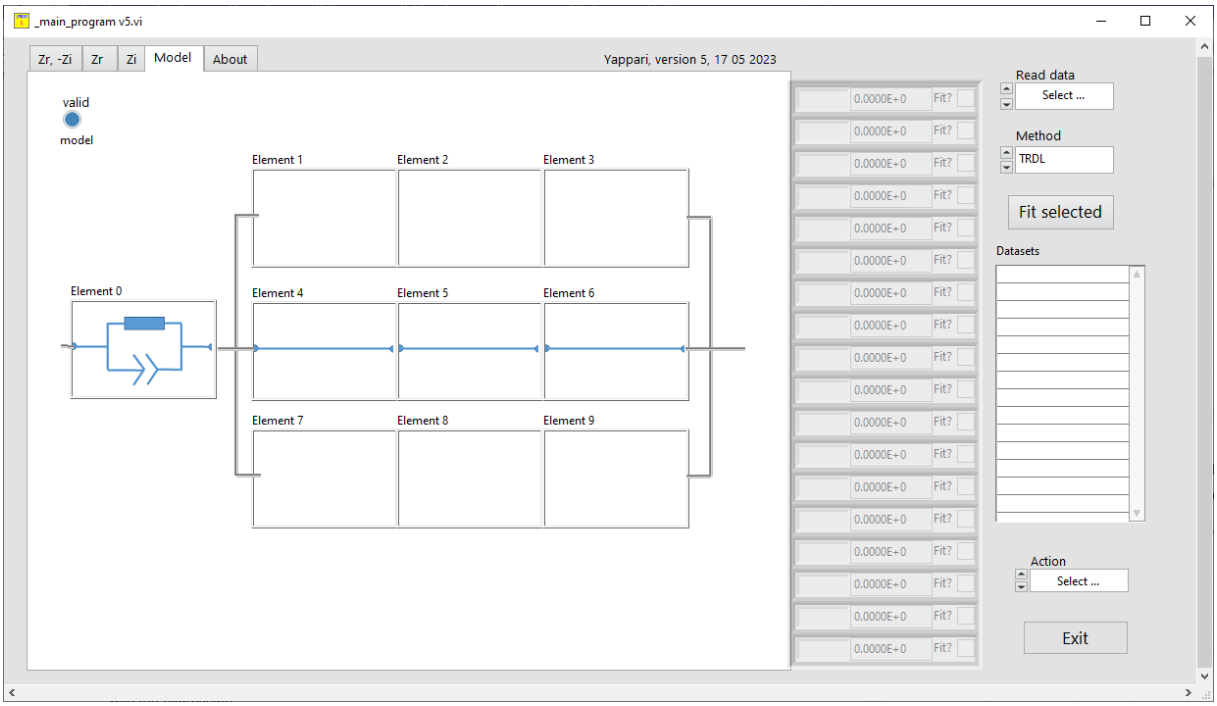
Some others can be added upon request, if I will have the time and if there is an interest for it.

When you create a circuit using the circuit editor, the circuit is not valid until you have properly connected all the elements together. Once the circuit is valid, a LED labeled "valid" will light up on the model panel, indicating that the circuit is ready for use, and you can see a list of all the parameters for each element of the circuit if you have at least one dataset.
Each parameter is labeled with a decimal, which indicates which element it belongs to. For example, the first element of the circuit will have parameters labeled as 0.x, the second element as 1.x, and so on.
When you add a parallel RQ element to the first element of the circuit (a Zarc, as element 0), you need to create "electrical contacts" in the next three elements (elements 1, 2, and 3, or 4,5 and 6, or 7, 8 and 9….) for the circuit to be complete.

Otherwise, the circuit will be open and no impedance can be calculated. In other words, all the elements of the circuit need to be properly connected for the circuit to be valid and for impedance calculations to be performed.



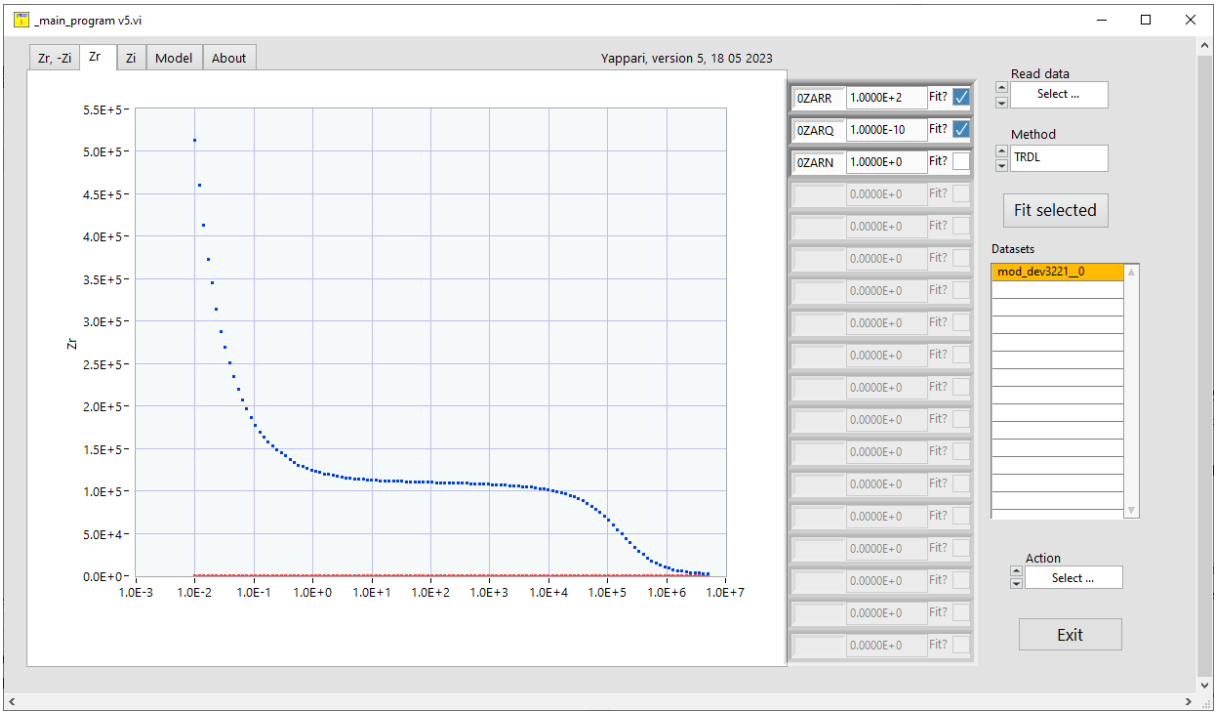
In the above figure you may notice the top left indicator, Valid, as OFF. The circuit is not closed so no calculation can be made as there is no flow of current. Let's make a valid circuit, as here:



Now, as the circuit is valid, with a Zarc in element 0 position, the led Valid is ON.

You need to have a dataset to view the parameters associated with it (note that it is better to add datasets *before* creating a circuit model).

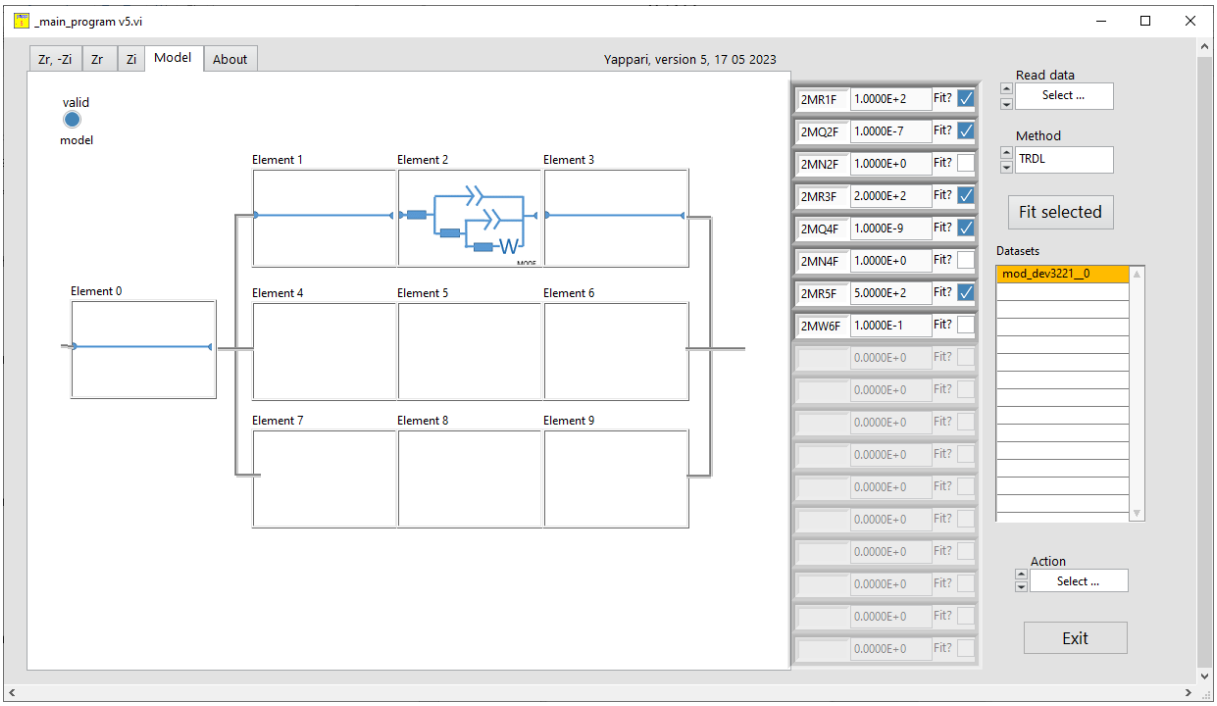
Let's add a dataset with the Read data command:



The experimental data, in blue here, the colors can be changed in latest version, and three parameters with the names: 0ZARR, 0ZARQ and 0ZARN and some dummy values. The names are rather self-explaining for the parameters describing a Zarc located in the position 0 of the circuit, with the three parameters describing a parallel RQ. You can use a RQ element and fix the N to 1 to obtain the equivalent RC circuit. The equivalent capacitance for a RQ circuit is C=((RQ)^1/n)/R.

The program will calculate the impedance spectra of a circuit with the values listed in the right side window. You can use the wheel of the mouse to evaluate the change in the output impedance in real time.

For a more complex circuit, like in the following image:



On the right side of the screen, you can see names such as 2MR1D, 2MQ2D, 2MN2D, 2MR3D, 2MR4D, 2MR5D, and 2MW6D. The first number, "2", indicates which element case the device is in.

The letters "M" and "D" are internal notations that are used by the program to identify the device type, but they are not important for the user. The type of device is listed after the "M" notation, such as "R" for resistor or "W" for Warburg.
The numbering of the devices goes from left to right and top to bottom. For example, the first device is a resistor and can be described by the parameter "2MR1D". The second device in the circuit is a Zarc, which is a combination of a constant phase element (CPE, or Q) in parallel with a resistor. This device is described by the parameters "Q2" and "N2", the “2’ meaning is the second element.

Overall, the notation is quite straightforward once you become familiar with the conventions used, but you cannot change these names

About

Brief help listing the version of the program.

Commands

Read data

This command opens a menu with several options as of now, designing which type of file to read. Note that reading a new file will just add more data wihtout losing the previous ones. You can remove some of the datasets with the command Erase selected datasets. You will need to select one or more datasets in order to plot them.

MFLI csv This is a ';' or ',' separated values file as obtained from MFLI/MFIA, a Zurich Instruments impedance analyzer. As in the Zview text file, multiple data sets can be saved or read from this file. In the data folder that is provided with this installer you can find such a file containing 34 measurements of the same sample. It would be boring and useless to fit all these 34 datasets one by one. Yappari-5 can handle such multiple data sets. You can select the separator string to be used in the Parameters page.

Versa Studio par This type of file contains data delimited by and >/Segments>. I did not exenisively checked this type of file, an example is given in the /data folder.

Zview txt This is a Zview file, also an ASCII type, that can hold multiple data sets. Yappari will read all datasets it finds in this file and insert them in the datasets listing, with a name taken from the file name and a suffix indicating the position in the file : the first datasets will have__0__, then 1, .. and so on. The separator setting will not be used in reading this file.

3 cols This reads a three-column ASCII file, which should be separated by tabs, ; or , and contain frequency in Hz, Zr, and Zi. The separator used can be selected in the Parameters page. It is important to note that for French users (and some others), the separator value should be a dot “.” and not a comma “,”. If the reading is successful, the dataset will be inserted in the first position with a name taken from the filename open. This name can be changed by the user. Only one dataset can be read with this command. The first line of these files can be a text (the program will try to detect and discard a comment in the first line; if it fails, just remove all comments from the file and try again).

In the /data folder you will find some datafiles, experimental or simulated with other impedance programs.

_test_data.txt : is a three columns datafile, separated by Tabs with a simulated circuit R(RQ)(RQ)

dev3221_imps_0_sample_0000 is a measurement of a sample (including more than 20 datasets) as obtained with a MFLI/MFIA from Zurich Instruments; separator for this file is ';'

mod_dev3221_imps_0_sample_0000 is a Zview txt file, as obtained with MFLI instrument (one dataset)

R(CR)(CR)W.txt is a simulated dataset with a Warburg element in a three columns separated by tabs

R(CR)O.txt is a simulated dataset with a Warburg "short" element (three columns separated by tabs)

R(CR)T.txt is a simulated dataset with a Warburg "open" element (three columns separated by tabs)

Method

This command allows the user to select the method used for nonlinear fitting. There are three methods available: Trusted Region Dog Led algorithm (TRDL), Constrained Levenberg Marquardt, and Levenberg Marquardt. The user can choose any of these methods, and if the model is robust, they should obtain the same results.

For TRDL and Constrained LM, the fit is constrained to certain intervals. For example, resistors are limited to the range of 1 mOhm to 1 TOhm, capacitors are between 10^-4 and 10^-14, and so on. It is recommended to start with TRDL and then make a final fit with the standard (unconstrained) LM method.

Fit selected

This command is used to fit the set of parameters that describes the circuit, if the circuit is valid (i.e., there are parameters to fit on the right side of the window). The user can select which parameters to fit and it is recommended to start with a few parameters first, ensuring that the initial values are close to the expected values. The simulated spectrum will be updated with every change in the parameters, and the user can perform manual adjustments as necessary.
For large datasets, if the data are described by the same model circuit, I suggest to select one measurement, adjust the parameters manually to be close to solution, then fit. After fit you can “Clone” these parameters to all other datasets and select all datasets, then Fit all selected in a go.

The fitting can be performed using different methods, which are discussed before, although there is not much difference in the output of these methods. The fitting process involves up to 8000 cycles for a dataset, and multiple iterations may be necessary, particularly if the initial values are far from the actual values.

The quality of the fit is evaluated using the R^2 statistical parameter and the chi^2 value. However, the use of the chi^2 value as a statistical parameter is debatable, as discussed in the paper "Dos and don'ts of reduced chi-squared" by Andrae et al. (https://arxiv.org/abs/1012.3754). The chi^2 value reported here is calculated as (Sum ((Z_obs-Z_calc)^2/Z_calc))/DOF. The degree of freedom (DOF) is considered as Nr_of_points - nr_of_fitted_params.

Datasets

This list box shows all the datasets in memory. You can select one or more datasets. The parameters listed are those of the dataset selected (or the first selected dataset if you have more than one selection). The datasets label can be edited.

Action

This button can trigger several commands:

Clone these parameters to all

Copy the listed parameters to all datasets. Useful for bulk fitting.

Save all parameters

Save all parameters names and values for all datasets in a file.

Save active exp datasets

This command allows you to save the active experimental data, that means the selected ones, to a single file in a specific format. The format is three columns, separated by the string you selected in the Parameters page, with frequency in Hz, Zr, and Zi. This is useful for simulating impedance spectra for a given model. All the datasets will be saved in a single file, each data subsequently added, with its name, to the same file.

Save active calc datasets

This command allows you to save the active calculated data to a file in a three columns format, separated by the string you selected in the Parameters page.

Save active exp and calc datasets

This command allows you to save the experimental data and calculated data, in a 5 columns ASCII file. Note that all datasets are saved in a single file, each dataset will be separated by the label of the set. This might be used to plot nicer graphs.

Average active datasets

This command will calculate the mean of Zr and Zi for selected datasets. This function can be applied to datasets measured at the same frequencies.

Erase active datasets

Irreversible action removing one or more datasets and all related parameters from memory (by active one should understand “selected”)

Report all

This command generates an HTML report containing information about the model used, the parameters used, the fitted parameters, and their standard deviation. It also includes images of the fit as well as all experimental and calculated data. The report is saved in your temporary directory and automatically opened in a browser. You can use the data in the report to create your own graphs or to check for any discrepancies. If you find any errors in the calculations, please report them so they can be corrected.

Z-Hit active datasets

This option will provide a Z-HIT simulation (which is a Hilbert transform of the phase into the real part of the impedance) for one or more datasets. The procedure, when and why to use it, is described here. In this implementation I am using the corrections including the 5th derivative of the phase as described in the link given here. This is a procedure similar to the better known Kramer-Kronig test.

Help

This will open this help file in a pdf format.

Exit

No need for explications on what this command does.

Author

For questions or comments:
Nita DRAGOE
Université Paris-Saclay
ICMMO/SP2M
91400 Orsay, France
github.com/nitad54448

Tutorial

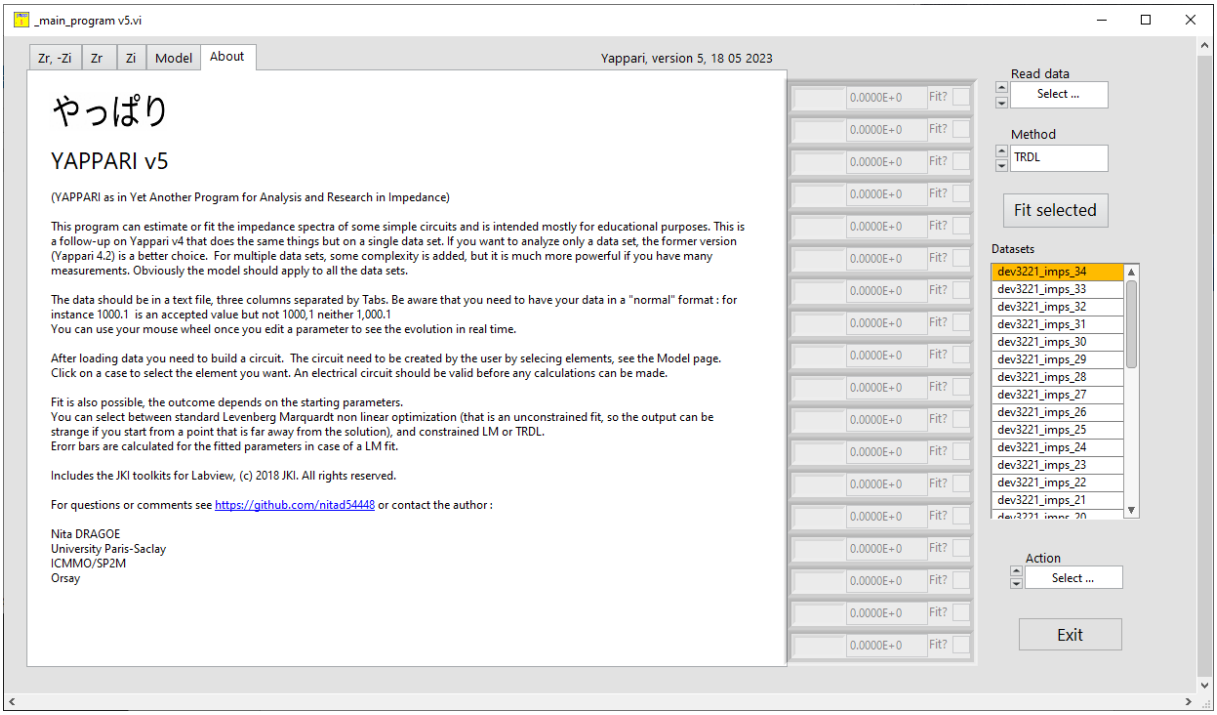
Let's make a fitting of some data to understand basic functions.

The first thing to do is select the separator (tab, ;, , or space) which is used in the data file (ths applies to MFLI CSV and 3 columns files). This parameter can be selected on the "Parameters page".

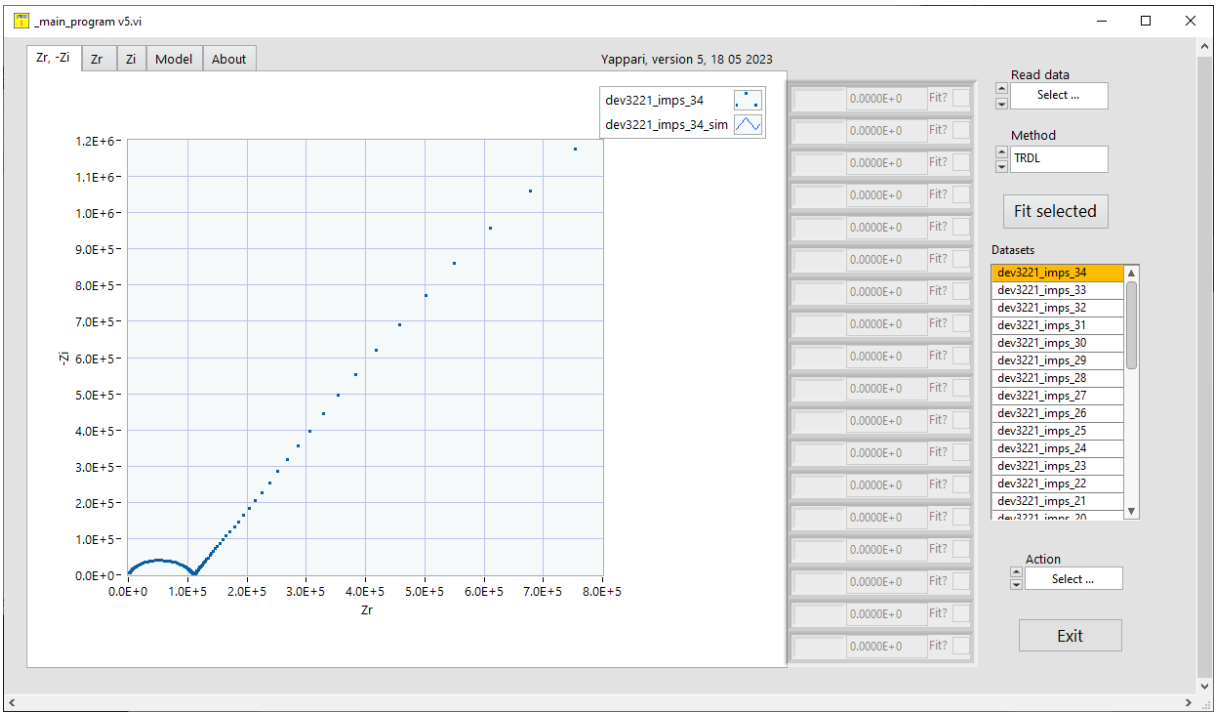
Then, the first thing is to read a data file.

In the data folder supplied with the installer there is a file dev3221_imps_0_sample_00000.csv, which is a ceramic measured over a weekend with a Zurich Instruments impedance analyzer. In this device there is an option Auto Save, who just adds all the data to a single file.

We can open this file with Read data, by selecting CSV format and select which data set to see (one or more datasets an be selected)

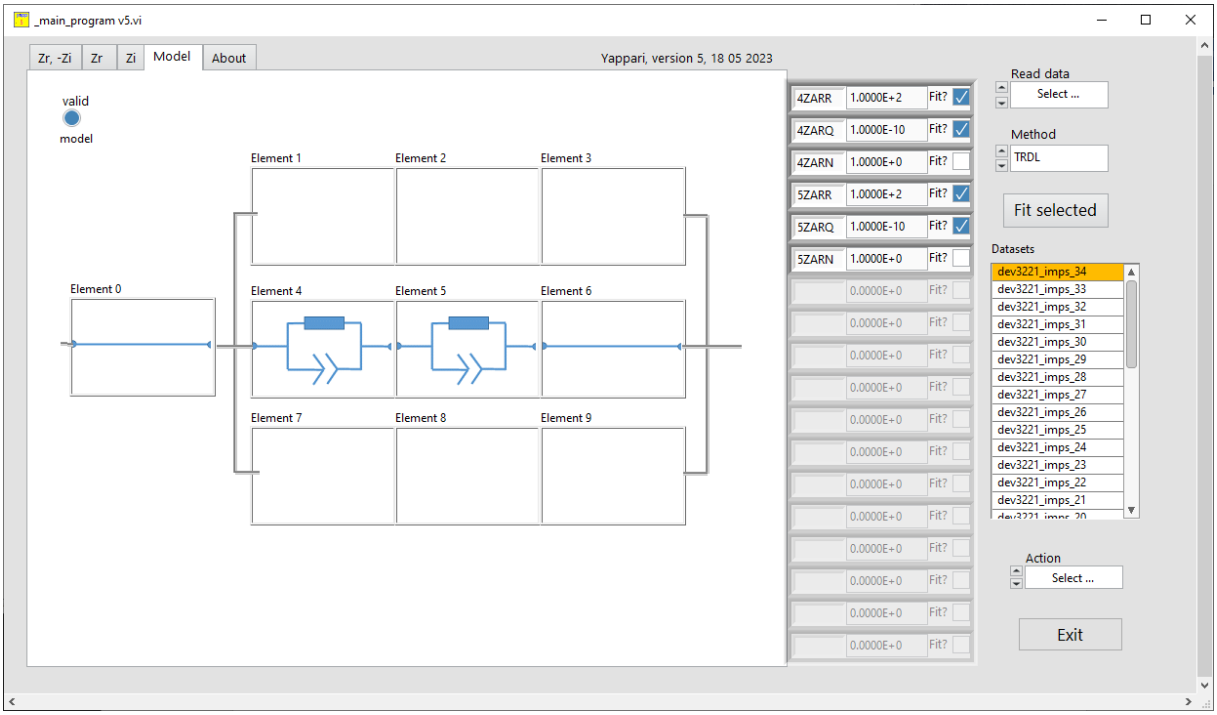


There are 35 measurements in this file, from 0 to 34 (the last measurement in the file appears as the first one in the datasets list). if we look at the Nyquist plot, the experimental data and simulated ones for all selected datasets are plotted:

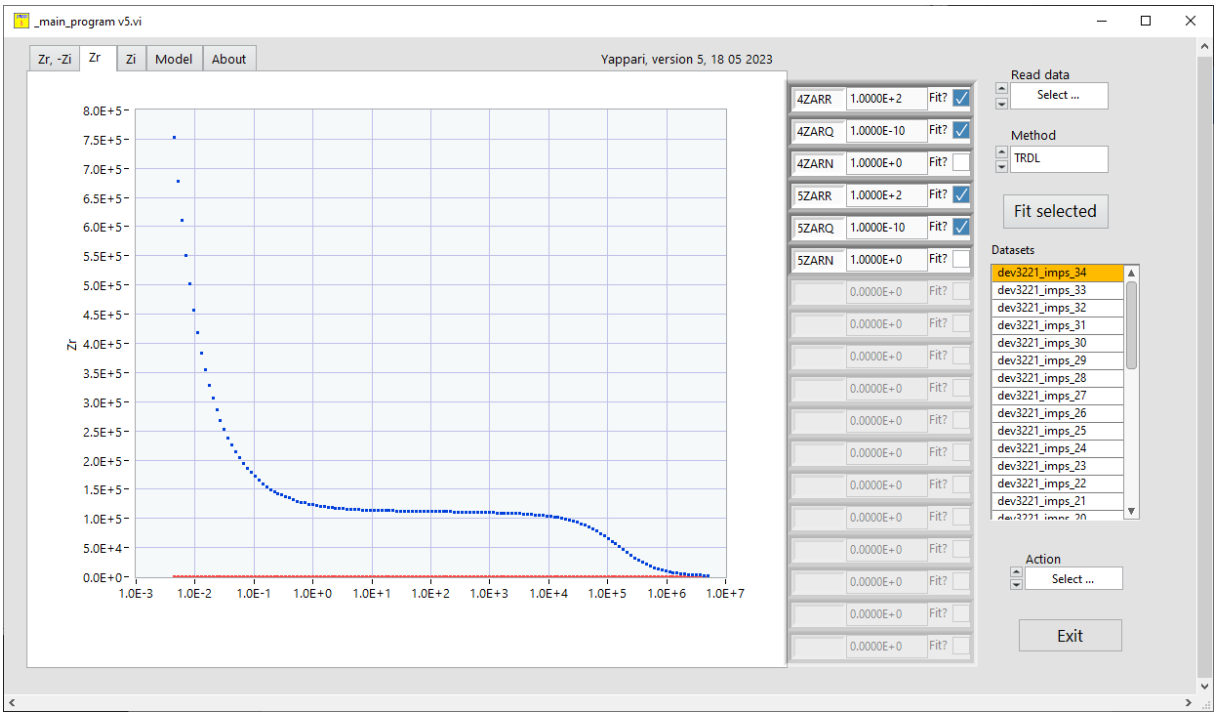


Here there is no simulation hence only one graph will appear. For calculations we'll need a model. Let's make one, I'll just make two Zarcs, to show how to fit (this is not an advice on what model to fit for this spectrum!)

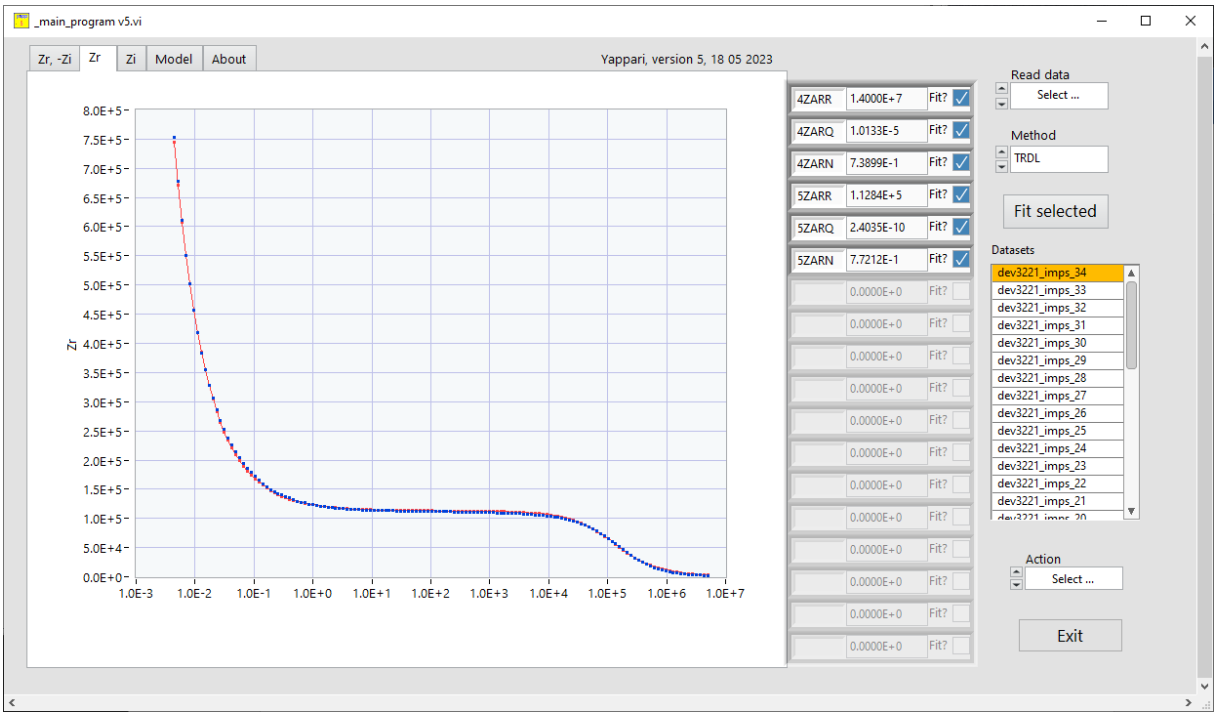
If I make a model like this



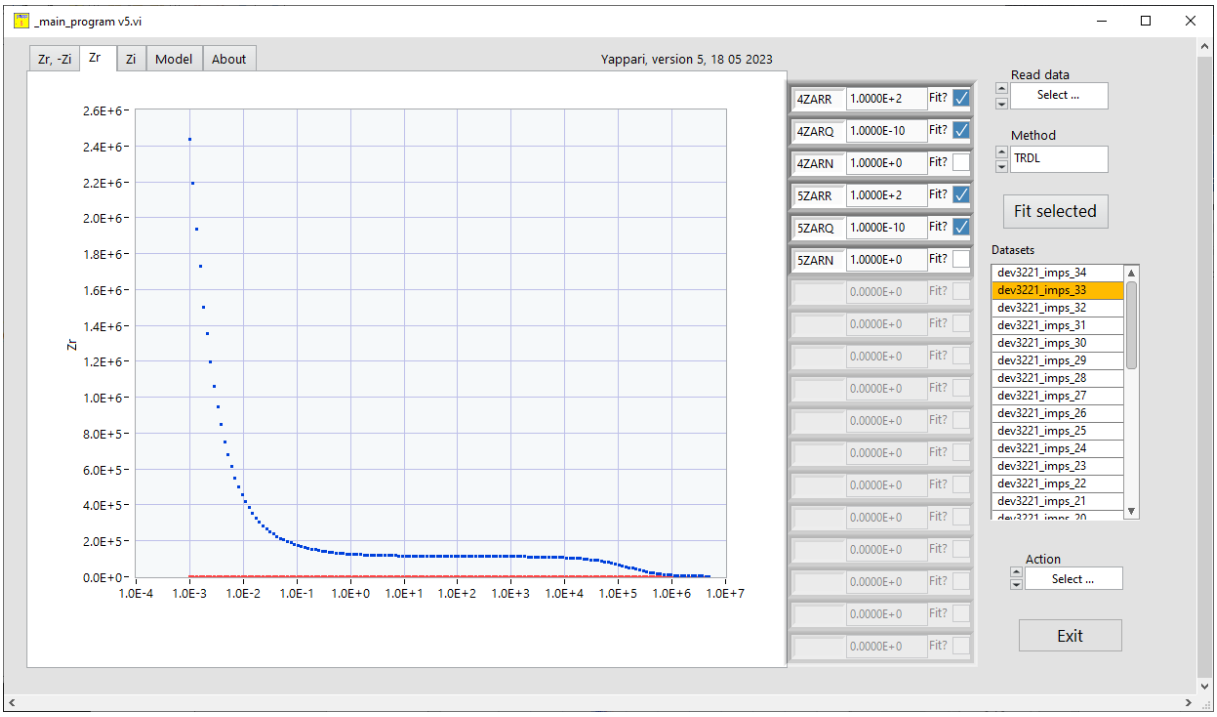
Six parameters, describing the two Zarcs, will appear with some dummy values. The calculated and experimental data are now plotted (the initial values are very far from the what they should be, the initial parameters are some defaults).



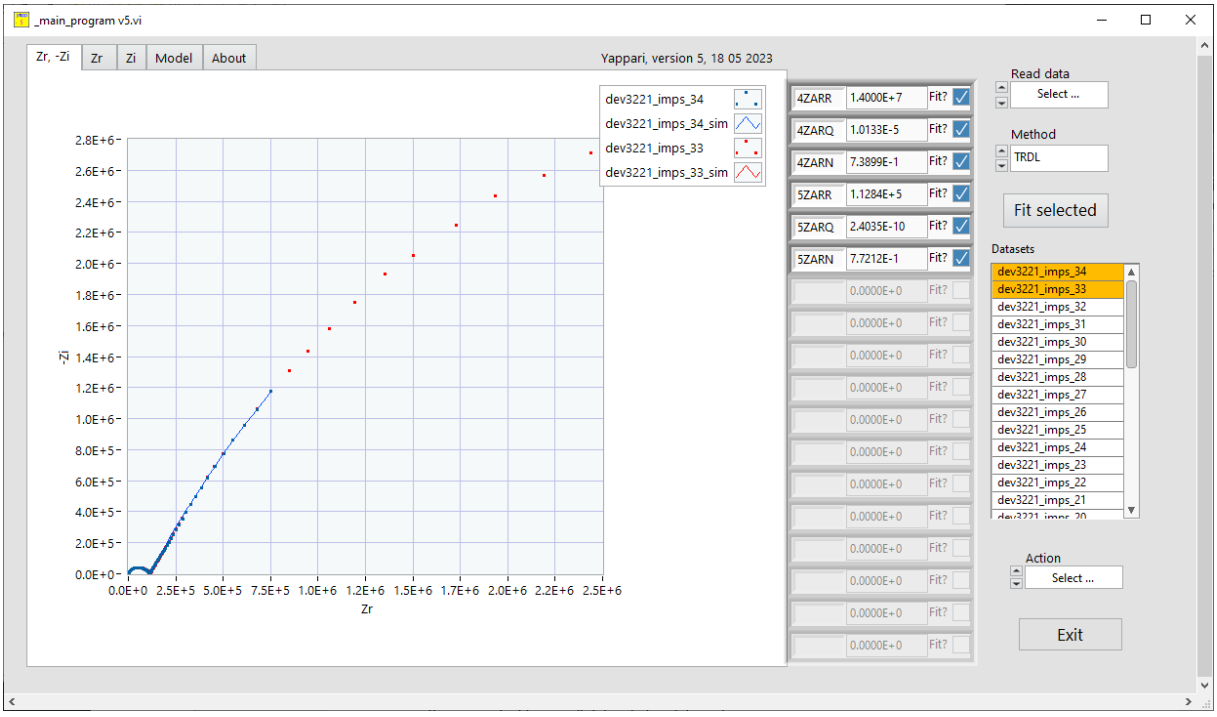
After adjusting manually and doing a couple of TRDL fits, we can get this which looks pretty good



We fitted only the first dataset, so if you select the second data you'll see the same initial parameters since we did not fit them.



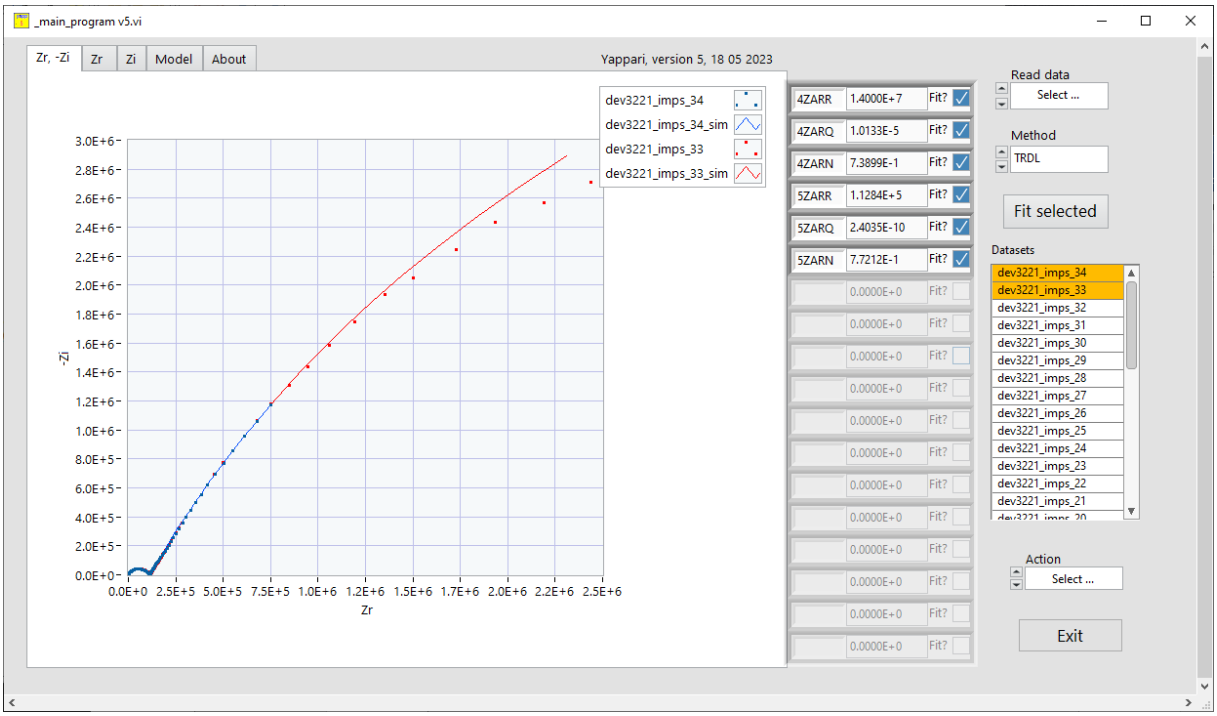
You can select one or more (or ALL) datasets for plots and comparison, let's look at Nyquist plot for the first 2 data sets:



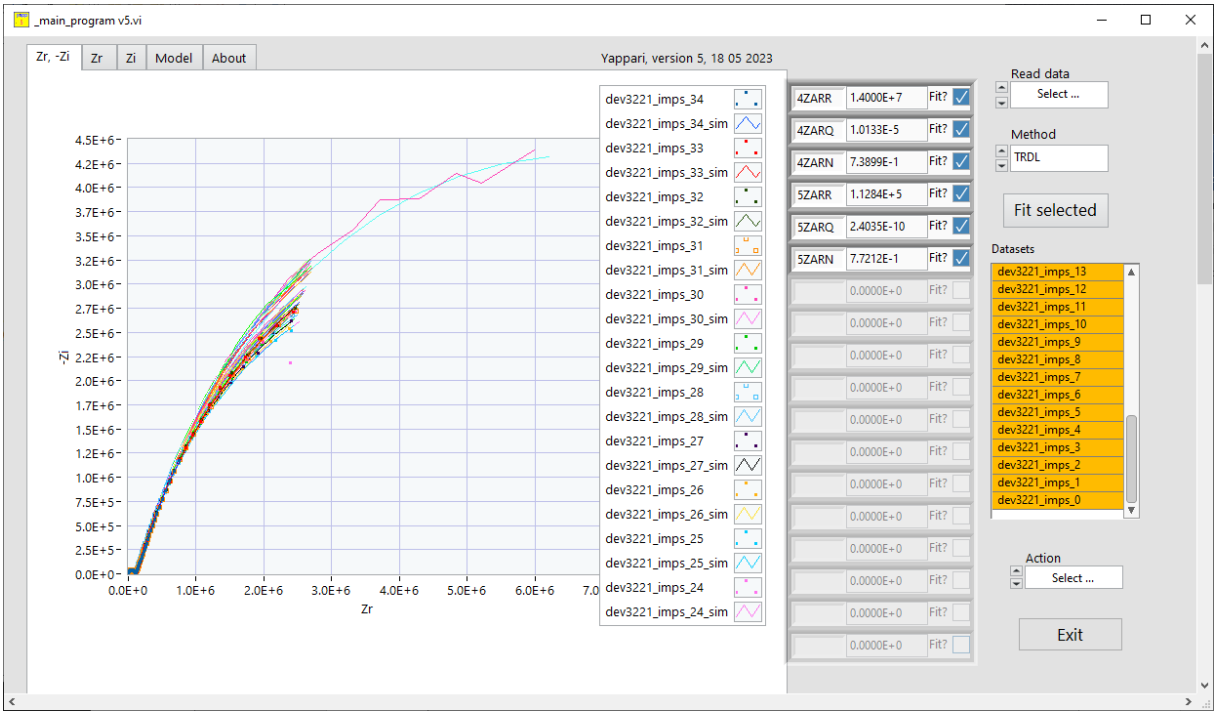
The two datasets have different frequency ranges, we can see the first set is fitted well, not the second (the legend on the right side of the graphs allows to change points, colors, and so on).

Next step is to copy the parameters from the first dataset to the 34 others by using *Action : Clone these parameters to all*

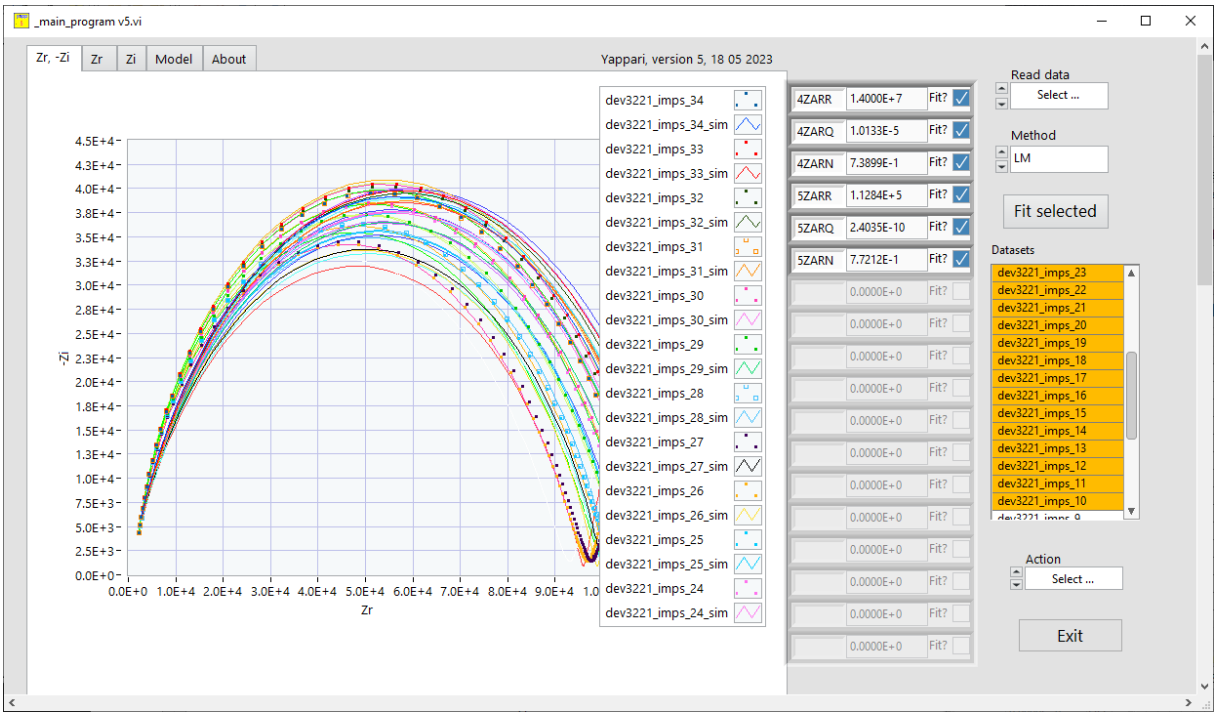
The parameters for the first set were copied to the other datasets and the calculations show that they are not that far



Let's fit them all, by selecting all datasets and use Fit selected. It will take a minute on a regular computer and you'll get something like



You can adjust the range for viewing the graph,

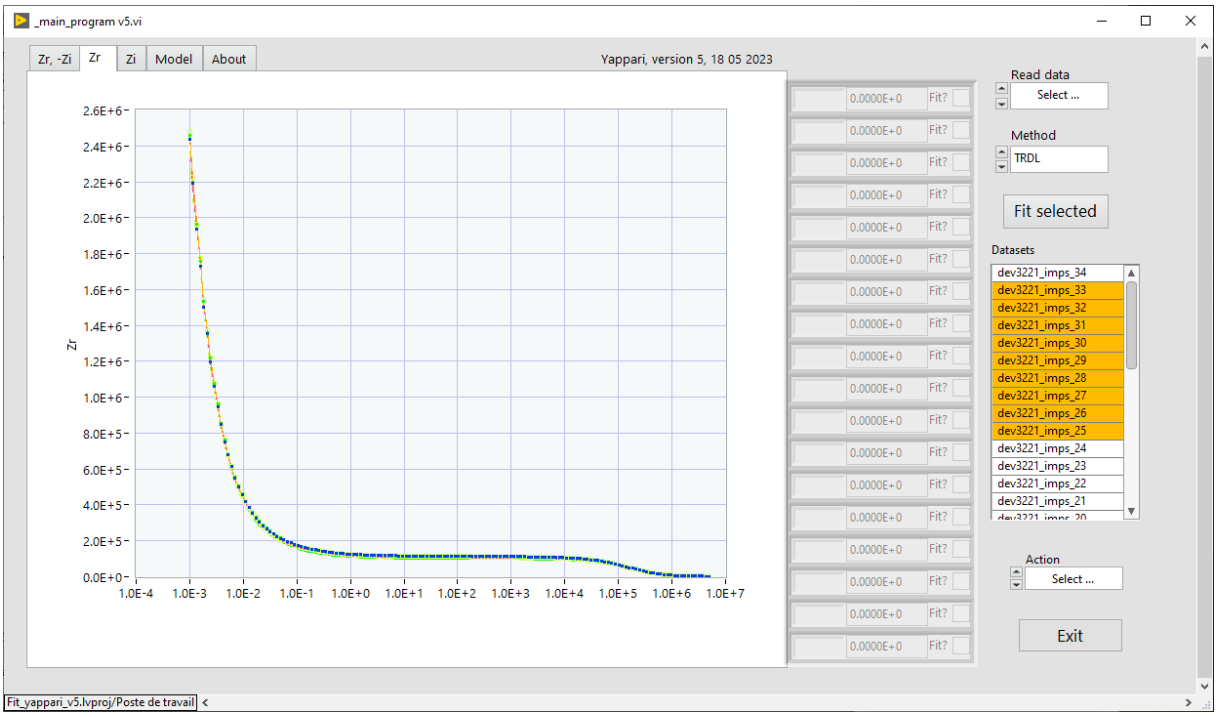


You can save all data to ASCII files for making your own graphs.

Also you can save all the parameters to a file, they are separated by Tabs. This file contains R2 and chi2 together with esd for the fits.

```
Yappari v5 - parameters saved : 18/05/2023 13:22:49
Dataset R2      chi2    4ZARR   esd     4ZARQ   esd     4ZARN   esd     5ZARR   esd     5ZARQ   esd
        5ZARN   esd
dev3221_imps_34         9.998103E-1     8.181934E+1     1.399964E+7     7.536805E+3     1.013275E-5
        1.091587E+4     7.389936E-1     1.470069E+4     1.128353E+5     3.494140E+3     2.403539E-10
        2.052499E+4     7.721182E-1     2.043698E+4
dev3221_imps_33         9.998613E-1     1.622669E+2     1.055534E+7     1.479720E+4     1.057738E-5
        3.342599E+4     7.580726E-1     3.927529E+4     1.165388E+5     7.403273E+3     2.721206E-10
        4.373066E+4     7.630881E-1     4.355314E+4
dev3221_imps_32         9.998856E-1     1.453812E+2     1.063659E+7     1.356263E+4     1.041905E-5
        3.057688E+4     7.568210E-1     3.595415E+4     1.157916E+5     6.784430E+3     2.731784E-10
        4.007398E+4     7.629034E-1     3.991107E+4
dev3221_imps_31         9.998839E-1     1.668674E+2     1.028405E+7     1.354980E+4     1.049484E-5
        3.059828E+4     7.588725E-1     3.592125E+4     1.146898E+5     6.817636E+3     3.021129E-10
        4.010963E+4     7.561332E-1     3.993908E+4
dev3221_imps_30         9.998849E-1     1.611541E+2     1.013578E+7     1.340489E+4     1.053283E-5
        3.026210E+4     7.591503E-1     3.551858E+4     1.117707E+5     6.753842E+3     3.067345E-10
        3.980872E+4     7.554907E-1     3.963927E+4
dev3221_imps_29         9.998852E-1     1.619192E+2     1.009246E+7     1.325226E+4     1.065630E-5
        2.993482E+4     7.593231E-1     3.513282E+4     1.086678E+5     6.674182E+3     3.220609E-10
        3.935585E+4     7.525943E-1     3.918562E+4
dev3221_imps_28         9.999379E-1     1.286392E+2     1.006806E+7     9.629417E+3     1.080187E-5
        2.175068E+4     7.593830E-1     2.553488E+4     1.047259E+5     4.833760E+3     2.868025E-10
        2.879782E+4     7.610694E-1     2.868093E+4
```

Another useful function (use with care, the measurement frequencies should be the same) is averaging a set of measurements. Select the datasets you want to average:



Then use *Action: Average selected dataset* command

An additional set appears with the name average (you can change it)