DearEIS - A GUI program for analyzing impedance spectra

Ville Yrjänä [1]

1 Åbo Akademi University, Faculty of Science and Engineering, Johan Gadolin Process Chemistry Centre, Laboratory of Molecular Science and Engineering, Turku (Åbo), Finland

DOI: 10.21105/joss.04808

Software

- Review
- Repository
- Archive

Editor: Jeff Gostick

Reviewers:

- @Aslan-Kosakian
- @mdmurbach

Submitted: 19 September 2022
Published: 13 December 2022

Summary

Electrochemical impedance spectroscopy (EIS) is a technique that is widely used to characterize the properties of materials used in, e.g., batteries and ion-selective electrodes. Analyses of the recorded impedance spectra typically involve some preprocessing, data validation, and fitting of a suitable model to extract quantitative data. Analyses of the distribution of relaxation times may also be relevant and can help with choosing an appropriate model. DearEIS is a free, open source, cross-platform program developed for performing such analysis work. The primary audience for DearEIS is researchers and engineers. However, the program may also prove useful in the context of, e.g., teaching university-level courses on electrochemistry.

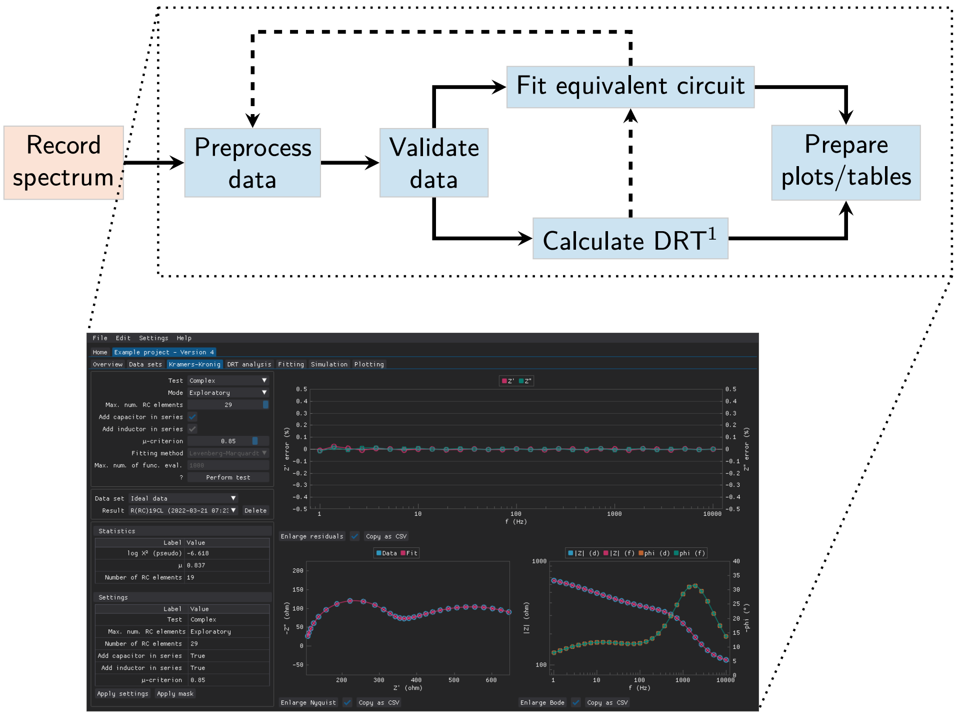

Figure 1: A flowchart of the steps that are typically involved in the process of obtaining and analyzing impedance spectra. The parts of the process that can be performed with DearEIS are highlighted. [1]Distribution of relaxation times.

Statement of need

DearEIS (Yrjänä, 2022a) aims to fill a niche that in the author's opinion is currently not filled by other software:

- free and cross-platform to lower the barrier to entry
- open source to maximize extensibility
- support for importing measurement data from multiple data formats
- capable of performing Kramers-Kronig testing (KK), distribution of relaxation times (DRT) analysis, and equivalent circuit fitting (ECF)
- graphical user interface (GUI) for ease of use
- application programming interface (API) to facilitate batch processing

The core functionality (circuits, KK, DRT, ECF, etc.) of DearEIS is implemented as a separate package called pyimpspec (Yrjänä, 2022b). DearEIS provides a GUI, which is implemented using Dear PyGui (Hoffstadt & Cothren, 2020), and an API wrapper for pyimpspec's API. The inclusion of both a GUI and an API makes DearEIS suitable for people of varying technical abilities and needs. These two interfaces also facilitate the use of hybrid workflows where tasks that are easier to do manually can be performed using the GUI (e.g., iterative development of equivalent circuits or basic composition of figures) while other tasks can be automated using Python scripts (e.g., batch processing results to generate publication-ready tables and/or figures). There are several examples (Table 1) of software capable of performing KK, DRT, and/or ECF but they all fail to fulfill at least one of the points listed above. Thus, there was an impetus to develop DearEIS.

Table 1: Examples of software that are available for analyzing impedance spectra. [1]Requires MATLAB and the Optimization Toolbox. [2]pyimpspec is a dependency of DearEIS.

Name	Reference or company
Aftermath	Pine Research Instrumentation, Inc.
DearEIS	**Yrjänä (2022a)**
DRT-python-code	Kulikovsky (2020)
DRTtools[1]	Wan et al. (2015)
EC-Lab	BioLogic Science Instruments SAS
Echem Analyst	Gamry Instruments, Inc.
EIS Spectrum Analyser	Bondarenko & Ragoisha (2005)
Elchemea Analytical	Koch et al. (2021)
impedance.py	Murbach et al. (2020)
IviumSoft	Ivium Technologies BV
Kramers-Kronig Test	Boukamp (1999)
LEVM/LEVMW	Macdonald (2015)
Lin-KK Tool	Schönleber et al. (2015)
Nova	Metrohm AG
PSTrace	PalmSens BV
pyDRTtools	Wan et al. (2015)
PyEIS	Knudsen (2019)
pyimpspec[2]	Yrjänä (2022b)
RelaxIS	rhd instruments GmbH & Co. KG
Zahner Analysis	Zahner-Elektrik GmbH & Co. KG
ZView	Scribner Associates, Inc.

Some of the other software listed in Table 1 are available for free though they are not necessarily also open source (Table 2). For example, the source code is not publicly available for EIS Spectrum Analyzer, Kramers-Kronig Test, and Lin-KK Tool. The source code for LEVM/LEVMW is included with the binaries, but the source code is not distributed under an open source license. DRTtools requires the user to have licenses for MATLAB and its Optimization Toolbox package, which adds a financial barrier to its use. Fortunately, a Python-based port called pyDRTtools is available. Several of these free software support multiple platforms.

Table 2: Comparison of source code availability, applicable open source license(s) if the source code is available, and supported major platforms of the software included in Table 1. [1]The Apache License version 2.0 (APLv2), the GNU General Public License version 3.0 (GPLv3), and the MIT license (MIT). [2]May also work on other platforms with the help of a compatibility layer or an emulator. [3]pyimpspec is a dependency of DearEIS. [4]Separate Python package that interacts with the main program (Zahner-Elektrik GmbH & Co. KG, 2022).

Name	Source available	License[1]	Platform(s)[2]
Aftermath			Windows
DearEIS	**Yes (Python)**	**GPLv3**	**Linux, MacOS, Windows**
DRT-python-code	Yes (Python)	GPLv3	Linux, MacOS, Windows
DRTtools	Yes (MATLAB)	MIT	Linux, MacOS, Windows
EC-Lab			Windows
Echem Analyst			Windows
EIS Spectrum Analyser			Windows
Elchemea Analytical	Yes (Perl)	GPLv3	Linux
impedance.py	Yes (Python)	MIT	Linux, MacOS, Windows
IviumSoft			Windows
Kramers-Kronig Test			Windows
LEVM/LEVMW	Yes (Fortran)	Not specified	MS-DOS/Windows
Lin-KK Tool			Windows
Nova			Windows
PSTrace			Windows
pyDRTtools	Yes (Python)	MIT	Linux, MacOS, Windows
PyEIS	Yes (Python)	APLv2	Linux, MacOS, Windows
pyimpspec[3]	Yes (Python)	GPLv3	Linux, MacOS, Windows
RelaxIS			Windows
Zahner Analysis	Yes (Python)[4]	MIT[4]	Linux, MacOS, Windows
ZView			Windows

Most instrument manufacturers bundle their instruments with software that can be used to analyze impedance spectra and a few third-party companies also sell software for this purpose (Table 2). These two types of software are typically closed source and not publicly available for download (e.g., requiring the purchase of a license or registration of an instrument when making an account). Trial versions with some limitations (e.g., an inability to save results) may be publicly available in some cases. Restrictive licenses and/or digital rights management technologies (e.g., the USB dongle or the online verification required by RelaxIS) may limit or even prevent distribution of the software to, e.g., colleagues or students. The commercial software in Table 2 only support Windows, with the exception of Zahner Analysis

In terms of key functionality, RelaxIS is the most similar alternative to DearEIS with LEVM/LEVMW as a close second (Table 3). However, RelaxIS is a closed-source, commercial product, which introduces a financial barrier to entry, and it officially only supports Windows. LEVM/LEVMW is, as was mentioned earlier, a free though not truly open source alternative. The software officially supports MS-DOS/Windows but may also work natively on other platforms provided that a compatible Fortran compiler is available. There is a rather steep learning curve associated with using LEVM and its CLI directly, particularly regarding the process of preparing the input files, despite the rather comprehensive manual that is included. LEVMW greatly simplifies this process, though some may find its GUI not intuitive to use by modern standards.

Table 3: Comparison of some key features currently available in the software included in Table 1. [1]Kramers-Kronig testing. [2]Distribution of relaxation times analysis. [3]Equivalent circuit fitting. [4]Application programming interface (API), command-line interface (CLI), or graphical user interface (GUI). [5]Script that temporarily displays figures. [6]Web interface to local or remote instance. [7]pyimpspec is a dependency of DearEIS. [8]Separate Python package that interacts with the main program (Zahner-Elektrik GmbH & Co. KG, 2022).

Name	KK[1]	DRT[2]	ECF[3]	Interface(s)[4]
Aftermath	Yes		Yes	GUI
DearEIS	**Yes**	**Yes**	**Yes**	**GUI, API**
DRT-python-code		Yes		GUI[5]
DRTtools		Yes		GUI
EC-Lab	Yes		Yes	GUI
Echem Analyst	Yes		Yes	GUI, API
EIS Spectrum Analyser	Yes		Yes	GUI
Elchemea Analytical			Yes	GUI[6], CLI
impedance.py	Yes		Yes	API
IviumSoft	Yes		Yes	GUI
Kramers-Kronig Test	Yes			GUI
LEVM/LEVMW	Yes	Yes	Yes	CLI/GUI
Lin-KK Tool	Yes			GUI
Nova	Yes		Yes	GUI
PSTrace			Yes	GUI, CLI, API
pyDRTtools		Yes		GUI
PyEIS	Yes		Yes	API
pyimpspec[7]	Yes	Yes	Yes	API
RelaxIS	Yes	Yes	Yes	GUI, API
Zahner Analysis	Yes		Yes	GUI, API[8]
ZView	Yes		Yes	

Some of the software listed in Table 3 focus on a single aspect of data analysis while most can do both KK and ECF. DearEIS includes implementations of the linear Kramers-Kronig tests (Boukamp, 1995) and an algorithm for automatically choosing the number of parallel RC circuits (Schönleber et al., 2014). An alternative implementation of the latter is also included to help with identifying false negatives. DearEIS uses lmfit (Newville et al., 2014) to perform complex non-linear least-squares fitting, which enables the use of different fitting methods (Levenberg-Marquardt, Nelder-Mead, etc.). A few different weighting options are also available (modulus, proportional, etc.). A specific combination of fitting method and weighting can be chosen or multiple combinations can be tried in parallel to find a combination that provides the best fit. The following circuit elements are currently implemented: resistor, capacitor, constant phase element, inductor, modified inductor, Gerischer, Havriliak-Negami, de Levie, Warburg (semi-infinite, finite space, and finite length). Additional elements may be implemented in the future. Lower and upper limits can be defined or omitted for any parameters that a circuit element may have. The parameter values can also be defined as constant values. Equivalent circuits can be constructed using either a graphical, node-based editor or a circuit description code (CDC). DearEIS supports a basic CDC syntax as described by Boukamp (Boukamp, 2001) and an extended syntax that can be used to also define parameter values, parameter limits, and labels for the circuit elements.

Support for DRT analysis is not very common as can be seen in Table 3 even though DRT results can be useful in the development of an appropriate equivalent circuit by revealing the number of time constants present in an impedance spectrum. The shapes of the peaks in the DRT plots can also provide some information about the type of circuit element that could be appropriate to include in an equivalent circuit (e.g., a sharp, symmetrical peak is typical for the type of slightly non-ideal capacitance that a constant phase element is often used for). However, calculating the DRT from an impedance spectrum is an ill-posed problem, which means that some care must be taken when interpreting the results. The wrong combination of impedance spectrum, method, and method parameters can result in, e.g., peaks that are actually just artifacts, or broad peaks that should actually be two or more separate peaks. DearEIS currently includes support for a few methods for performing DRT analyses:

- Tikhonov regularization and radial basis function (or piecewise linear) discretization with or without Bayesian credible intervals, which was implemented in DRTtools and described by Wan et al. (2015), Ciucci & Chen (2015), and Effat & Ciucci (2017)
- Bayesian Hilbert transform, which was implemented in DRTtools and described by Liu et al. (2020)
- Tikhonov regularization and non-negative least-squares fitting, which was implemented in DRT-python-code as an alternative to the original approach that used projected gradient descent and was described by Kulikovsky (2020)
- multi-(RQ)-fit, which was described by Boukamp (2015) and Boukamp & Rolle (2017)

Software that is bundled with instruments often does not support loading measurement data from many file formats, which may limit its use for the analysis of results obtained with other manufacturer's instruments unless plain-text files with character-separated values are supported such as in the case of IviumSoft. On the other hand, third-party software like RelaxIS and ZView do support loading measurement data from many different file formats including the formats used by many well-known instrument manufacturers. DearEIS currently supports importing experimental data from file formats used by some instrument manufacturers (e.g., BioLogic, Gamry, and Ivium). Experimental data that is stored as character-separated values in spreadsheets (`.xlsx` and `.ods`) or plain-text files can also be imported into DearEIS.

Many of the software included in Table 3 have functionality beyond those included in that table (e.g., Mott-Schottky analysis is included in RelaxIS). Even some of the closed-source software have APIs that can be used to, e.g., batch process analyses (e.g., Zahner Analysis) or implement new circuit elements (e.g., RelaxIS). DearEIS has several features in addition to those that are included in Table 3:

- disabling data points to, e.g., remove outliers
- making corrections to a data set by subtracting a fixed value, an equivalent circuit's impedance response, or another impedance spectrum
- simulating the impedance response of a circuit
- overlaying data sets and/or analysis results in plots, which can also be exported using matplotlib (Hunter, 2007) as the backend
- copying various data as plain-text to the system clipboard (e.g., the data used to generate a plot as character-separated values, tables of fitted parameters as Markdown, or circuit diagrams as Scalable Vector Graphics)
- an API that can be used for, e.g., batch processing
- new circuit elements can be added by extending pyimpspec

Ultimately, DearEIS provides a unified GUI for various analytical functions that in some cases were previously only available as separate, smaller programs that may or may not have had a GUI. The barrier to entry should therefore be quite low since DearEIS works on multiple platforms, is freely available, and does not require a highly technically proficient user. Simultaneously, DearEIS does facilitate more advanced use cases by providing an API. The open source nature of DearEIS will hopefully ensure that the project can be extended and used far into the future.

Acknowledgments

Some parts of pyimpspec and, by extension, DearEIS are based on code from DRT-python-code, Elchemea Analytical, impedance.py, and pyDRTtools. See the LICENSES folders in the GitHub repositories for more information. The author would like to thank Wenyang Xu for providing feedback and help with testing during development. The financial support of the author's doctoral studies during the development of pyimpspec and DearEIS by Svenska Litteratursällskapet i Finland, Åbo Akademi University, and Suomen Kulttuurirahasto is gratefully acknowledged.

References

Bondarenko, A. S., & Ragoisha, G. A. (2005). *Progress in Chemometrics Research* (A. L. Pomerantsev, Ed.; pp. 89–102). Nova Science Publishers, New York. http://www.abc.chemistry.bsu.by/vi/analyser/

Boukamp, B. A. (1995). A linear Kronig-Kramers transform test for immittance data validation. *Journal of the Electrochemical Society*, *142*(6), 1885–1894. https://doi.org/10.1149/1.2044210

Boukamp, B. A. (1999). *Kramers-Kronig Test*. https://www.utwente.nl/en/tnw/ims/publications/downloads/

Boukamp, B. A. (2001). *The CIRCUIT DESCRIPTION CODE explained*. https://www.utwente.nl/en/tnw/ims/publications/downloads/

Boukamp, B. A. (2015). Fourier transform distribution function of relaxation times; application and limitations. *Electrochimica Acta*, *154*, 35–46. https://doi.org/10.1016/j.electacta.2014.12.059

Boukamp, B. A., & Rolle, A. (2017). Analysis and application of distribution of relaxation times in solid state ionics. *Solid State Ionics*, *302*, 12–18. https://doi.org/10.1016/j.ssi.2016.10.009

Ciucci, F., & Chen, C. (2015). Analysis of electrochemical impedance spectroscopy data using the distribution of relaxation times: A bayesian and hierarchical bayesian approach. *Electrochimica Acta*, *167*, 439–454. https://doi.org/10.1016/j.electacta.2015.03.123

Effat, M. B., & Ciucci, F. (2017). Bayesian and hierarchical bayesian based regularization for deconvolving the distribution of relaxation times from electrochemical impedance spectroscopy data. *Electrochimica Acta*, *247*, 1117–1129. https://doi.org/10.1016/j.electacta.2017.07.050

Hoffstadt, J., & Cothren, P. (2020). Dear PyGui: A modern, fast and powerful GUI framework for Python. In *GitHub repository*. GitHub. https://github.com/hoffstadt/DearPyGui

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Knudsen, K. B. (2019). *kbknudsen/PyEIS: PyEIS: A Python-based electrochemical impedance spectroscopy simulator and analyzer*. Zenodo. https://doi.org/10.5281/zenodo.2535951

Koch, S., Graves, C., Vels Hansen, K., & DTU Energy. (2021). *Elchemea Analytical*. https://www.elchemea.com/

Kulikovsky, A. (2020). PEM fuel cell distribution of relaxation times: A method for the calculation and behavior of an oxygen transport peak. *Phys. Chem. Chem. Phys.*, *22*, 19131–19138. https://doi.org/10.1039/D0CP02094J

Liu, J., Wan, T. H., & Ciucci, F. (2020). A bayesian view on the hilbert transform and the kramers-kronig transform of electrochemical impedance data: Probabilistic estimates and quality scores. *Electrochimica Acta*, *357*, 136864. https://doi.org/10.1016/j.electacta.2020.136864

Macdonald, J. R. (2015). *LEVM/LEVMW*. https://jrossmacdonald.com/levmlevmw/

Murbach, M. D., Gerwe, B., Dawson-Elli, N., & Tsui, L. (2020). impedance.py: A Python package for electrochemical impedance analysis. *Journal of Open Source Software*, *5*(52), 2349. https://doi.org/10.21105/joss.02349

Newville, M., Stensitzki, T., Allen, D. B., & Ingargiola, A. (2014). *LMFIT: Non-Linear Least-Square Minimization and Curve-Fitting for Python*. https://doi.org/10.5281/zenodo.11813

Schönleber, M., Goyal, R., & Ivers-Tiffée, E. (2015). *Lin-KK Tool*. Karlsruhe Institute of Technology, Institute for Applied Materials - Electrochemical Technologies. https://www.iam.kit.edu/et/Lin-KK.php

Schönleber, M., Klotz, D., & Ivers-Tiffée, E. (2014). A method for improving the robustness of linear kramers-kronig validity tests. *Electrochimica Acta*, *131*, 20–27. https://doi.org/10.1016/j.electacta.2014.01.034

Wan, T. H., Saccoccio, M., Chen, C., & Ciucci, F. (2015). Influence of the discretization methods on the distribution of relaxation times deconvolution: Implementing radial basis functions with DRTtools. *Electrochimica Acta*, *184*, 483–499. https://doi.org/10.1016/j.electacta.2015.09.097

Yrjänä, V. (2022a). DearEIS: A GUI program for analyzing, simulating, and visualizing impedance spectra. In *GitHub repository*. GitHub. https://vyrjana.github.io/DearEIS

Yrjänä, V. (2022b). pyimpspec: A package for parsing, validating, analyzing, and simulating impedance spectra. In *GitHub repository*. GitHub. https://vyrjana.github.io/pyimpspec

Zahner-Elektrik GmbH & Co. KG. (2022). zahner_analysis. In *GitHub repository*. GitHub. https://github.com/Zahner-elektrik/Zahner-Analysis-Python