

impedance.py: A Python package for electrochemical impedance analysis

Matthew D. Murbach¹, Brian Gerwe², Neal Dawson-Elli³, and Lok-kun Tsui⁴

1 Hive Battery Labs **2** University of Washington, Department of Chemical Engineering **3** PayScale, Inc. **4** University of New Mexico, Center for MicroEngineered Materials

DOI: [10.21105/joss.02349](https://doi.org/10.21105/joss.02349)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Monica Bobra](#) ↗

Reviewers:

- [@ma-sadeghi](#)
- [@EricaEgg](#)

Submitted: 20 May 2020

Published: 03 August 2020

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

`impedance.py` is a community-driven Python package for making the analysis of electrochemical impedance spectroscopy (EIS) data easier and more reproducible. `impedance.py` currently provides several useful features commonly used in the typical impedance analysis workflow:

- *preprocessing*: functions for importing data from a variety of instruments and file types
- *data validation*: easy-to-use methods for checking measurement validity
- *model fitting*: a simple and powerful interface for quickly fitting models to analyze data
- *model selection*: parameter error estimates and model confidence bounds
- *visualization*: interactive and publication-ready Nyquist and Bode plots

Background

Electrochemical impedance spectroscopy (EIS) is a powerful technique for noninvasively probing the physicochemical processes governing complex electrochemical systems by measuring the relationship between voltage and current (i.e. the impedance) as a function of frequency (Orazem & Tribollet, 2008). Analysis of EIS spectra often involves the construction and fitting of models to represent the system response. Equivalent circuit models, a popular method for interpreting impedance spectra, represent processes in an electrochemical system with elements in a circuit allowing quantitative characterization of the data. EIS inherently requires the system response is linear, causal and stable over the measurement time. As such, researchers must also ensure their data complies with these assumptions – often through some implementation of Kramers-Kronig analysis. Following the approach outlined above, workers have applied EIS to problems ranging from measuring the dielectric properties of biological systems (Semenov et al., 2001) to corrosion rates of coated metals (Bonora, Deflorian, & Fedrizzi, 1996). More recently, EIS has been widely adopted for studying processes governing batteries (Rodrigues, Munichandraiah, & Shukla, 2000; Tröltzsch, Kanoun, & Tränkler, 2006) and fuel cells (Springer, Zawodzinski, Wilson, & Gottesfeld, 1996; Wagner, Schnurnberger, Müller, & Lang, 1998).

Statement of Need

To date, typical impedance analysis solutions have relied on either instrument-specific, proprietary software or ad hoc, lab-specific code written for internal use. These proprietary or home-grown solutions lack cross-platform support, and users of Linux or MacOS are unable to perform analysis on their computers. In addition to making reproducible analysis difficult, these solutions can be restrictive to defining custom circuit elements and processing large datasets. By providing an open-source, cross-platform (Windows, MacOS, and Linux),

community-driven package for the full impedance analysis pipeline from data management to parameter extraction and publication-ready figures, `impedance.py` seeks to encourage reproducible, easy-to-use, and transparent analysis. In addition to decades of electrochemical research, many new methods for validating (Schönleber, Klotz, & Ivers-Tiffée, 2014) and analyzing (Buteau & Dahn, 2019; Murbach & Schwartz, 2018) impedance spectra have been developed by researchers. By lowering the barrier to use tried-and-true methods along side cutting-edge analytical techniques via a consistent interface, `impedance.py` also serves to grow as a community repository of best-practices while facilitating the adoption of new techniques.

Current `impedance.py` functionality

Preprocessing

Once you've collected your data, the last thing you want to do is worry about how to turn the files generated by the instrument into results you can learn from. `impedance.py` offers easy-to-use functions for importing data from BioLogic, CH Instruments, Gamry, PARSTAT, VersaStudio, and ZView files. Don't see your instrument yet? [Create an issue with a sample file](#) and help contribute to the project!

Data validation

Ensuring that the system under measurement is linear, stable, and causal is an important, but often overlooked part of impedance analysis. `impedance.py` provides several methods (measurement models, Lin-KK algorithm (Schönleber et al., 2014)) for data validation as a part of the same easy-to-use package.

Equivalent circuit fitting

`impedance.py` equivalent circuit fitting combines an extremely flexible circuit definition scheme with a range of available circuit elements – from simple L, R, C, and constant-phase elements to electrochemistry specific elements such as Warburg, Gerischer (Boukamp & Bouwmeester, 2003), and transmission-line models (Paasch, Micka, & Gersdorf, 1993). Custom elements can be easily added to a single users installation or to the overall project by [creating an issue](#). Additional model fitting features include parameter bounds, holding circuit elements constant, weighted fitting, and even saving/loading models to a human readable .json file. Fitting is performed by non-linear least squares regression of the circuit model to impedance data via `curve_fit` from the `scipy.optimize` package (Virtanen et al., 2020).

Model selection

One of the biggest challenges in extracting parameters from EIS spectra is quantitatively evaluating the fit of different models. `impedance.py` returns estimated parameter error bars with every fit and comes with a built in method for calculating the confidence bounds of a fit model.

Data visualization

EIS is often qualitatively interpreted by examining the shape of a spectrum which relies on properly defined axes. `impedance.py` eliminates the tedium of manually formatting plots by

providing built-in methods to generate clean, publication-ready figures including interactive plots via Altair (VanderPlas et al., 2018) and customizable Nyquist or Bode plots via matplotlib (Hunter, 2007).

An example of the simple model API

The documentation for `impedance.py` contains [a guide on getting started](#) and several examples of what a typical analysis workflow might look like using the package. Here we show how importing data, defining and fitting an equivalent circuit model, and visualizing the results can be done with just a handful of lines in `impedance.py`:

```
# 1. loading in data:
from impedance.preprocessing import readFile
f, Z = readFile('exampleData.csv')

# 2. remove positive imaginary impedance data:
from impedance.preprocessing import ignoreBelowX
f, Z = ignoreBelowX(f, Z)

# 3. importing and initializing a circuit:
from impedance.models.circuits import CustomCircuit
initial_guess = [1e-8, .01, .005, .1, .9, .005, .1, 200, .1, .9]
circuit = CustomCircuit('L_0-R_0-p(R_1,CPE_1)-p(R_2-Wo_1,CPE_2)',
                        initial_guess=initial_guess)

# 4. fitting the circuit to the data:
circuit.fit(f, Z)
print(circuit)
```

```
Circuit string: L_0-R_0-p(R_1,CPE_1)-p(R_2-Wo_1,CPE_2)
Fit: True
```

```
Initial guesses:
    L_0 = 1.00e-08 [H]
    R_0 = 1.00e-02 [Ohm]
    R_1 = 5.00e-03 [Ohm]
    CPE_1_0 = 1.00e-01 [Ohm^-1 sec^a]
    CPE_1_1 = 9.00e-01 []
    R_2 = 5.00e-03 [Ohm]
    Wo_1_0 = 1.00e-01 [Ohm]
    Wo_1_1 = 2.00e+02 [sec]
    CPE_2_0 = 1.00e-01 [Ohm^-1 sec^a]
    CPE_2_1 = 9.00e-01 []
```

```
Fit parameters:
    L_0 = 1.28e-07 (+/- 4.92e-08) [H]
    R_0 = 1.49e-02 (+/- 8.31e-04) [Ohm]
    R_1 = 8.31e-03 (+/- 2.24e-03) [Ohm]
    CPE_1_0 = 1.13e+00 (+/- 8.07e-01) [Ohm^-1 sec^a]
    CPE_1_1 = 6.85e-01 (+/- 1.27e-01) []
    R_2 = 7.92e-03 (+/- 1.51e-03) [Ohm]
    Wo_1_0 = 1.37e-01 (+/- 9.03e-02) [Ohm]
    Wo_1_1 = 1.23e+03 (+/- 1.61e+03) [sec]
```

```
CPE_2_0 = 3.69e+00 (+/- 3.09e-01) [Ohm^-1 sec^a]
CPE_2_1 = 9.79e-01 (+/- 6.02e-02) []
```

```
# 5. visualize the results:
circuit.plot(f_data=f, Z_data=Z)
```

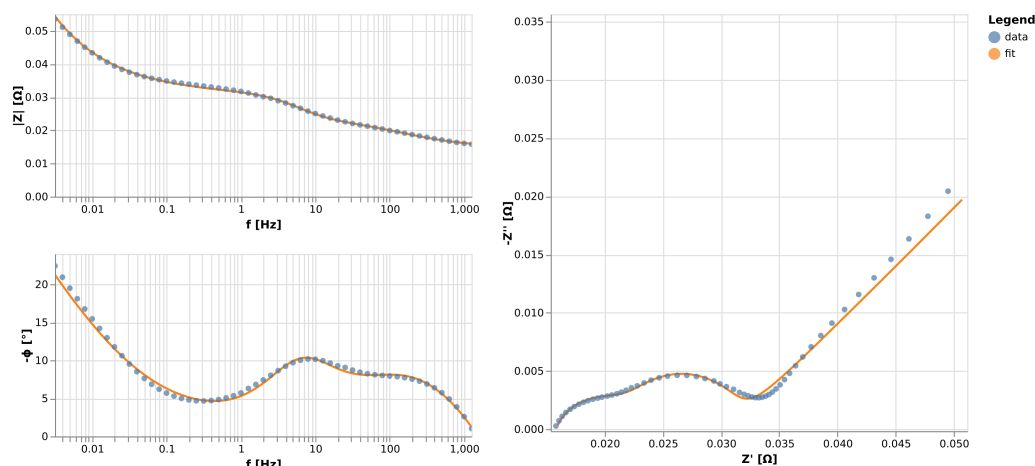


Figure 1: Interactive impedance plots are as easy as `.plot()`!

Acknowledgements

We thank participants on the 2018 Electrochemical Society (ECS) Hack Week team in Seattle, WA as well as Dan Schwartz and David Beck for their guidance. An up-to-date [list of contributors can be found on GitHub](#). Example data is from (Murbach et al., 2018).

References

- Bonora, P. L., Deflorian, F., & Fedrizzi, L. (1996). Electrochemical impedance spectroscopy as a tool for investigating underpaint corrosion. *Electrochimica Acta*, 41(7-8), 1073–1082. doi:[10.1016/0013-4686\(95\)00440-8](https://doi.org/10.1016/0013-4686(95)00440-8)
- Boukamp, B. A., & Bouwmeester, H. J. M. (2003). Interpretation of the Gerischer impedance in solid state ionics. *Solid State Ionics*, 157(1-4), 29–33. doi:[10.1016/S0167-2738\(02\)00185-6](https://doi.org/10.1016/S0167-2738(02)00185-6)
- Buteau, S., & Dahn, J. R. (2019). Analysis of Thousands of Electrochemical Impedance Spectra of Lithium-Ion Cells through a Machine Learning Inverse Model. *Journal of The Electrochemical Society*, 166(8), A1611. doi:[10.1149/2.1051908jes](https://doi.org/10.1149/2.1051908jes)
- Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science Engineering*, 9(3), 90–95. doi:[10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55)
- Murbach, M. D., Hu, V. W., & Schwartz, D. T. (2018). Nonlinear Electrochemical Impedance Spectroscopy of Lithium-Ion Batteries: Experimental Approach, Analysis, and Initial Findings. *Journal of The Electrochemical Society*, 165(11), A2758–A2765. doi:[10.1149/2.0711811jes](https://doi.org/10.1149/2.0711811jes)

- Murbach, M. D., & Schwartz, D. T. (2018). Analysis of Li-Ion Battery Electrochemical Impedance Spectroscopy Data: An Easy-to-Implement Approach for Physics-Based Parameter Estimation Using an Open-Source Tool. *Journal of The Electrochemical Society*, 165(2), A297. doi:[10.1149/2.1021802jes](https://doi.org/10.1149/2.1021802jes)
- Orazem, M. E., & Tribollet, B. (2008). *Electrochemical impedance spectroscopy*. The Electrochemical Society series. Hoboken, N.J: Wiley. ISBN: [978-0-470-04140-6](https://doi.org/10.1002/9780470041406)
- Paasch, G., Micka, K., & Gersdorf, P. (1993). Theory of the electrochemical impedance of macrohomogeneous porous electrodes. *Electrochimica Acta*, 38(18), 2653–2662. doi:[10.1016/0013-4686\(93\)85083-B](https://doi.org/10.1016/0013-4686(93)85083-B)
- Rodrigues, S., Munichandraiah, N., & Shukla, A. K. (2000). A review of state-of-charge indication of batteries by means of a.c. Impedance measurements. *Journal of Power Sources*, 87(1-2), 12–20. doi:[10.1016/S0378-7753\(99\)00351-1](https://doi.org/10.1016/S0378-7753(99)00351-1)
- Schönleber, M., Klotz, D., & Ivers-Tiffée, E. (2014). A Method for Improving the Robustness of linear Kramers-Kronig Validity Tests. *Electrochimica Acta*, Electrochemical Impedance Spectroscopy, 131, 20–27. doi:[10.1016/j.electacta.2014.01.034](https://doi.org/10.1016/j.electacta.2014.01.034)
- Semenov, S. Y., Simonova, G. I., Starostin, A. N., Taran, M. G., Souvorov, A. E., Bulyshev, A. E., Svenson, R. H., et al. (2001). Dielectrical model of cellular structures in radio frequency and microwave spectrum. Electrically interacting versus noninteracting cells. *Annals of Biomedical Engineering*, 29, 427–435. doi:[10.1114/1.1366675](https://doi.org/10.1114/1.1366675)
- Springer, T. E., Zawodzinski, T. A., Wilson, M. S., & Gottesfeld, S. (1996). Characterization of polymer electrolyte fuel cells using AC impedance spectroscopy. *Journal of The Electrochemical Society*, 143(2), 587–599. doi:[10.1149/1.1836485](https://doi.org/10.1149/1.1836485)
- Tröltzsch, U., Kanoun, O., & Tränkler, H.-R. (2006). Characterizing aging effects of lithium ion batteries by impedance spectroscopy. *Electrochimica Acta*, 51(8-9), 1664–1672. doi:[10.1016/j.electacta.2005.02.148](https://doi.org/10.1016/j.electacta.2005.02.148)
- VanderPlas, J., Granger, B., Heer, J., Moritz, D., Wongsuphasawat, K., Satyanarayan, A., Lees, E., et al. (2018). Altair: Interactive Statistical Visualizations for Python. *Journal of Open Source Software*, 3(32), 1057. doi:[10.21105/joss.01057](https://doi.org/10.21105/joss.01057)
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., et al. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17, 261–272. doi:[10.1038/s41592-019-0686-2](https://doi.org/10.1038/s41592-019-0686-2)
- Wagner, N., Schnurnberger, W., Müller, B., & Lang, M. (1998). Electrochemical impedance spectra of solid-oxide fuel cells and polymer membrane fuel cells. *Electrochimica Acta*, 43(24), 3785–3793. doi:[10.1016/S0013-4686\(98\)00138-8](https://doi.org/10.1016/S0013-4686(98)00138-8)