Manual

for an automated evaluation of Δi_p

in voltammetry

Benedikt Piller

February 7, 2022

Contents

Link to the Python code: `www.github.com/BielSeb/voltammetry_meas_eval`

This code was established by Sebastian Bielmeier and improved by Benedikt Piller in the framework of a practical internship at the professorship of Electrobiotechnology at TUM Campus Straubing.

On www.github.org the complete python code is provided for open access, hopefully thinking it will be helpful for further research purposes. Some test CVs for different electrodes (Glassy Carbon Electrode, Flat ITO, Porous ITO) are provided as .txt-files to comprehend easier how the data file has to be structured.

For questions or ideas for further development send an e-mail to: benedikt.piller@tum.de.

1 Why is an automated evaluation of Δi_p needed?

In electrochemistry Linear Sweep Voltammetry (LSV) or Cyclic Voltammetry (CV) with different scan rates are often performed as standard tests. In several cases, the analysing value of the dataset is the forward peak current Δi_p. For an actual Δi_p, the background current of the neat liquid of the shape has to be recognized.

Usually, a CV without redox probe is performed to have an off-set data. Then the same CV is performed with the dissolved redox probe in the neat liquid. An overlay of both signals and the subtraction helps to find the actual value of Δi_p.

In some cases, it is not practicable to do this expenditure of doubled experiments. If there is only the data of CVs with redox probe, there is a helpful method to find nearly the same value for Δi_p: Put a baseline to the flat region (light blue) and add a vertical line (dark blue) from the peak to the baseline (see figure 1 right). Δi_p can be determined by calculating the difference between the peak current and the current of the extrapolated baseline . This method can be done manually with *OriginLab* or with other analysing software solutions.

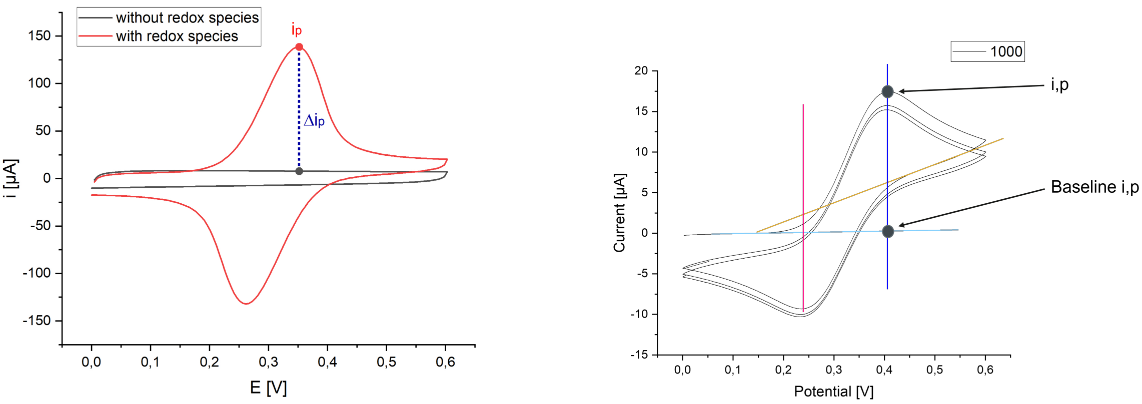

Figure 1: Left: overlay of a CV with redox species (red) and background signal without redox species (black);
Right: Manual determination of Δi_p with OriginLab; analogous for reverse Δi_p , if needed.

However, if there are many CV datasets or each dataset has many scan rates, the manual factor limits the analysis due to time and error potential. This analysing problem leads to the necessity of an automatic analysing method for Δi_p. An external supporter of this internship helped willingly on this issue and wrote a program code in Python which applies the same method as described. Especially, the analysing time is reduced to a minimum and also the error rate is lower.

2 Electrochemical software procedures

Various electrochemical experiments were performed by a potentiostat. A potentiostat is an electronic device which generates a change of either potential or frequency on its output electrodes and measures the current or impedance simultaneously for very short timescales (range of microseconds) [1].

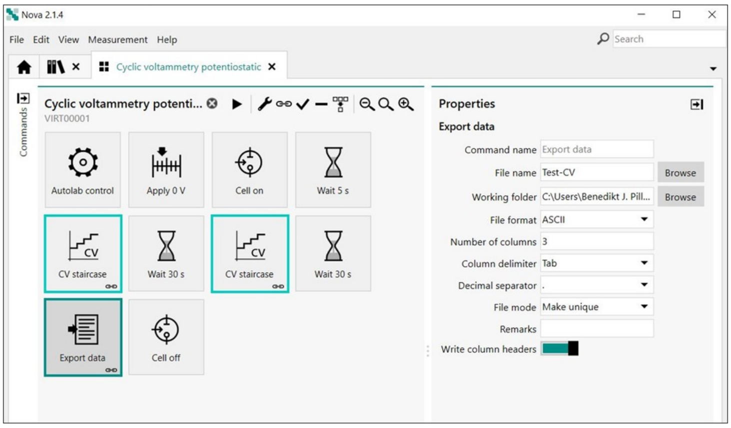

Figure 2: CV procedure is prepared: select each staircase and export data icon; link each other with the "chain" symbol

Dealing with the potentiostat, an adapted software is required. In this laboratory *Nova* is used as application (V. 2.1.4, made by Metrohm Autolab B.V.); for beginners this software is comfortable and comprehensive in workday practice. Nova enables to use default procedures by changing values.

Regarding to this method, the default procedure "Cyclic voltammetry potentiostat" is selected and adapted to an individual setting. The collected data can be saved as a .txt-file. Then the file is loaded in a data calculation program and the two columns "WE(1).Potential" and "WE(1).Current" are selected.

Figure 3: Prepare an automated data export for a CV procedure with several scan rates: first CV is linked to "WE(1).Potential" and WE(1).Current", next CV is only connected with "WE(1).Current".

If there is a series of scan rates, it is better to compress the desired data columns to one .txt-file. In figure 2 and 3 the preparation is shown how a CV procedure is linked with an automated data export for two scan rates within one procedure.
The Python code is made for multiple scan rates, e.g. from 1000 mV/s to 100 mV/s in intervals of 100 mV/s. These scan rates can be defined as a scan rate pattern (srp). In this work we used different scan rate patterns for each electrode type, which was used. Four scan rate pattern are preset in the code, but there is no limit for new a new srp.

3 Dealing with this code successfully

1. **Preparation .txt-file (with NOVA software)** (At first time use included test CV files)

   (a) Perform a CV and save the data as a .txt file in ASCII format;
   The .txt file will be structured in this order: first column is the list of potential values and all other columns are lists of current values in order to decreasing scan rates.

   (b) It is recommended to create an own CV procedure with automatic saving selected data. The course action is listed in steps **??** to 1d for the software NOVA. Otherwise, skip these steps.

   (c) Put the option 'Export data' at the end of your procedure; select the needed number of columns (n = number of CVs + 1); see figure 2.

   (d) Link each single CV staircase with the 'Export data' icon in this way: Column 1 is connected with "WE(1).Potential" of the first CV, column 2 is connected with "WE(1).Current", columns 3 is allocated to "WE(1).Current" of the next CV; see figure 3.

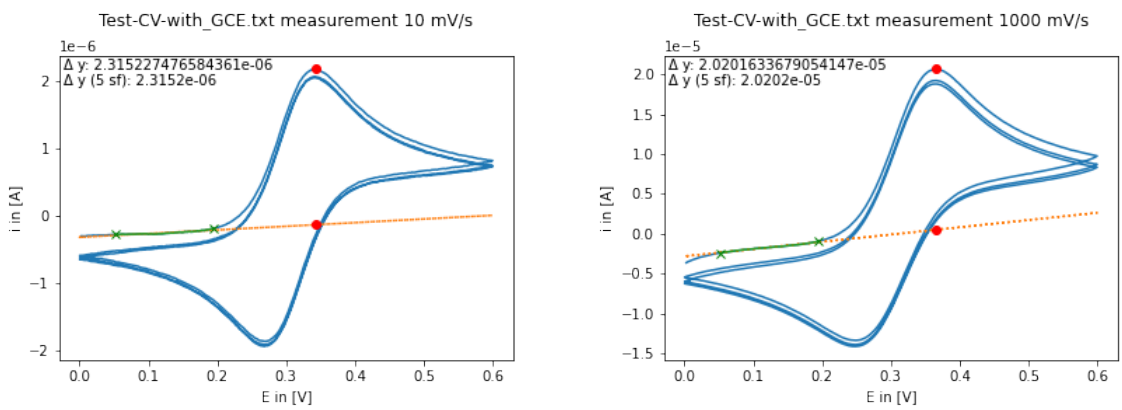

Figure 4: Plotted i-E-diagrams with a standard GCE, $v = 10\ mV/s$ (left) and $v = 1000\ mV/s$ (right); settings: start pos = 20, end pos = 80, srp = 2

2. **Python code settings**

(a) Launch the python code and include data by writing the **filename without file ending**.

(b) Set the **start** and **end position** (green x's in figures 4/5) of the desired baseline (hint: set default values 0 and 100, look at the plots, and change the position values if necessary).

(c) Choose one of the **default scan rate pattern** by type the number in the planned field. If there is not the right pattern available, add one line for your scan rate scheme.

(d) Start the program and look at the created i-E-plots if the baseline is set correctly for all scan rates (see figure 4).

(e) If the settings don't fit optimally, change some parameters as long as your baseline is good enough for each scan rate.

(f) Copy the value(s) of Δi_p for further uses, i.e. create a Δi_p-vs.-$\upsilon^{1/2}$-plot or a secondary plot ($\frac{\Delta i_p}{\upsilon^{1/2}}$ vs. $\upsilon^{1/2}$)

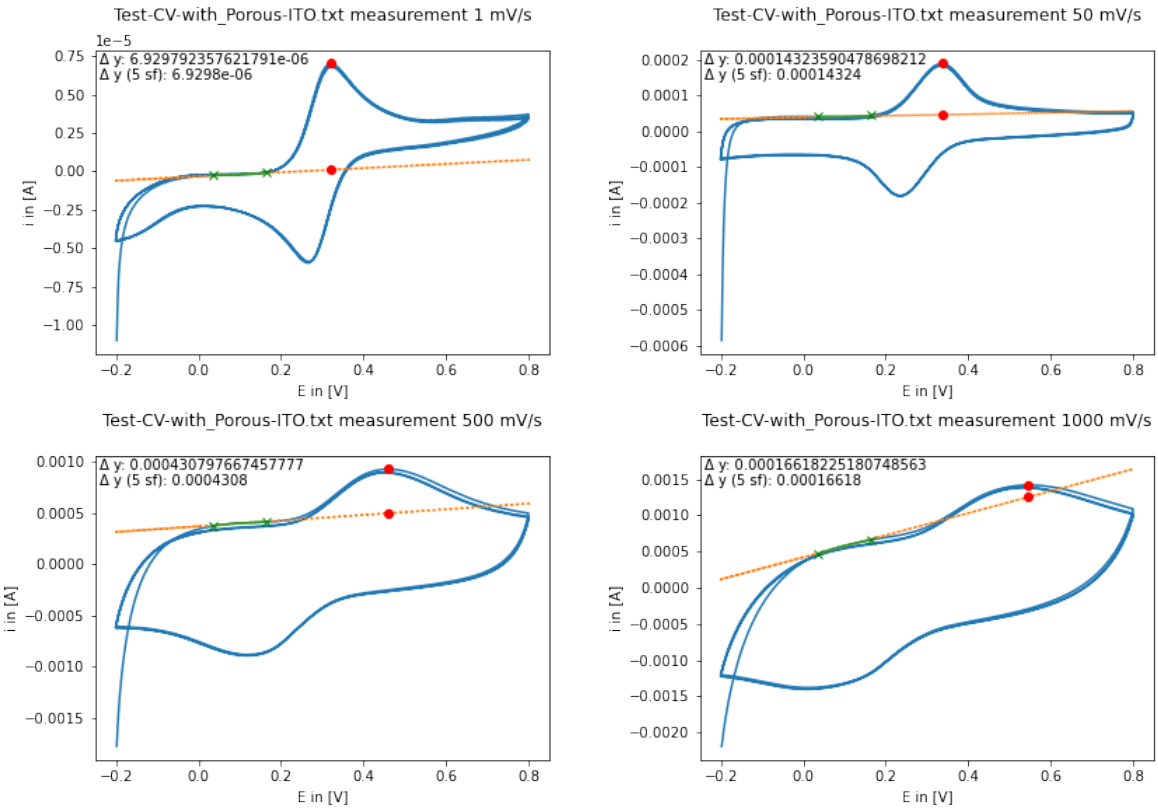

Figure 5: At the top: Porous ITO electrode with $\upsilon = 1\ mV/s, 50\ mV/s$ with expected values; at the bottom: CVs fail for higher scan rates than $\upsilon = 500\ mV/s$; settings: start pos = 95, end pos = 150, srp = 3

4 Troubleshooting

Nevertheless, wrong determined values are not excluded, especially for datasets with many scan rates as depicted in figure 5. Possible faults may cause, when the baseline points are not correctly set. The CV shape varies over scan rate range. That's why a look at each generated plot can recognize, if the code is failing or not. The baseline setting will lead to viable values at low scan rates (like 50 V/s), but for higher scan rates the baseline is gone bad (compare with figure 5 at the bottom). If the code does fail, start and end position will be adapted to an better purpose. Sometimes there will be an error, if the scan rate pattern is too short (`IndexError: list index out of range`. Doublecheck if the right scan rate pattern is chosen.

References

[1] Alex W. Colburn, Katherine J. Levey, Danny O'Hare, and Julie V. Macpherson. Lifting the lid on the potentiostat: a beginner's guide to understanding electrochemical circuitry and practical operation. Phys. Chem. Chem. Phys., 23:8100–8117, 2021. doi: 10.1039/D1CP00661D. URL http://dx.doi.org/10.1039/D1CP00661D.