AION

Reference Guide – Ver: 1.0.2

MAY 1, 2018

AUTHOR: HARRY TUNSTALL

Contents

Introduction

Aion has been designed and written to perform data analysis on calibration and unbuffered data generated by CHI potentiostats performing square wave voltammetry anodic scans, using functionalised boron diamond pH electrodes.

This code has been designed to handle erroneous data and filenames as best as possible, however not all cases may have been anticipated. Currently the list of exceptions anticipated for unbuffered data analysis are as follows:

- If the filename contains too few underscores to contain the correct amount of data, the file is skipped. Then the user is notified
- If the current data contained within the file contains a positive value, the data is flagged as potentially erroneous and the user is notified. Further action is required by the user to proceed.
- If the provided data is noisy, the correct 2nd derivative should be selected. All tested data, even with significant noise, has successfully selected the correct 2nd derivative. However, there may be a dataset which has not been accounted for.

Performance & Requirements

RAM

At most Aion should use no more than 150 MB of RAM, with general use sitting at just under 100 MB of RAM.

Performance

The provided test dataset has been tested on two systems:

• A low-performance system (LPS) running an:	AMD A10-8700P	@ 2.00GHz
• A high-performance system (HPS) running an:	AMD FX-8350	@ 4.61 GHz

Calibration calculations require very little time to run and, as such, have not had performance tests. Unbuffered calculations were performed on the test dataset both saving, and not saving the output PDF. The results are as follows:

	LPS	HPS
	Average time per file processed / s	
Saving PDF	1.73	1.07
Not saving PDF	0.78	0.44

If you wish to test the validity of Aion you can run the unbuffered calculation on this dataset (provided in the testing section) and compare the output to the already generated output.

Application Structure and Usage

Running Aion (the program)

Anaconda Distribution

This program was built using Anaconda3 (Python 3.6) and is dependent on the libraries provided with its instillation. If you need to install Anaconda3 the download can be found at this link: https://www.anaconda.com/download/

If you are installing Anaconda3, please tick the option "add anaconda to path".

Terminology Clarification

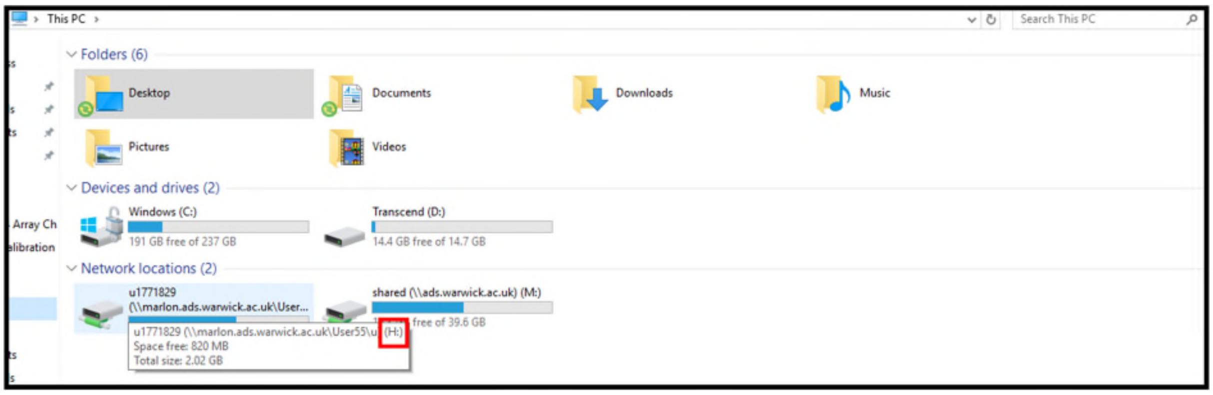
First, some clarification on nomenclature used for the rest of this documentation. The use of the tilde (~) symbol before a file path means any length of path before this. For example, "~\aion-master\aion" is the correct path for any of the following paths:

```
"H:\Desktop\aion-master\aion-master\aion"
"H:\Desktop\aion-master\aion-master\aion-master\aion"
"C:\Users\Harry\My Documents\aion-master\aion"
Etc.
```

Running aion.py

To run Aion run the 'aion.py' file. If you know how to run a python script from the command prompt (make sure it's the Anaconda3 command prompt – with the correct PATH variable), great! Go do it. If you don't it will be explained step by step here.

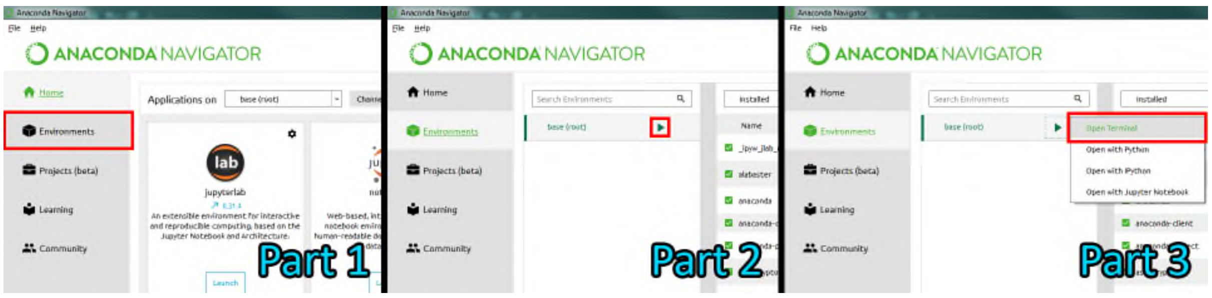
1. First locate the "~\aion-master\aion" folder, we'll need this path later.
    a. In this instance it is located on the desktop, in the location "H:\Desktop\aion-master\aion-master\aion". **N.B: Directory paths are case sensitive when using the command prompt.**
    b. The drive letter was found as shown in Walkthrough 1.



Walkthrough 1 – This shows you how to find the drive letter of the mounted network drive which holds the desktop.

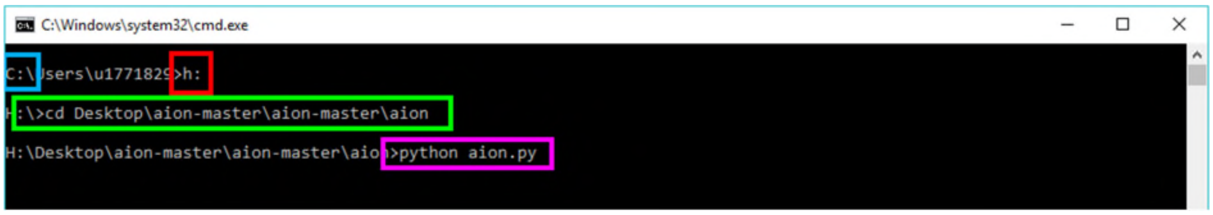
2. Open the Anaconda Navigator.
    a. This can be done by selecting the start menu and searching for "anaconda navigator".
3. Go into the Environments tab (Walkthrough 2 – Part 1).
4. Left click the button to the right of the "base (root)" option (Walkthrough 2 – Part 2).

5. Select “Open Terminal” (Walkthrough 2 – Part 3).



Walkthrough 2 – Left, step 3. Middle, step 4. Right, step 5.

6. In this terminal you need to navigate to the location of “~\aion-master\aion” (Walkthrough 3 – blue, red & green). If you know how to do this, please skip straight to step 7.
    a. First check if the drive letter is correct. In this instance it is not.
    b. To change the drive letter, type the required letter and then a colon (e.g “h:”), then hit enter.
    c. To change the location type “cd” then a space then the path we found earlier (without the drive letter) “Desktop\aion-master\aion-master\aion”.
7. Now to run Aion type “python aion.py” then hit enter (Walkthrough 3 – magenta).



```
C:\Users\u1771829>h:
H:\>cd Desktop\aion-master\aion-master\aion
H:\Desktop\aion-master\aion-master\aion>python aion.py
```

Walkthrough 3 – Blue, step 6a. Red, step 6b. Green, step 6c. Magenta, step 7.

Distribution

The Aion distribution uses sha-256 checksums to make sure that the distribution has not been corrupted when sent over the internet. This most likely should not be an issue, however if Aion fails to load this could be the cause. The distribution does involve a nested .zip architecture, please see below for more information.

When using Aion from a network location (of if you are moving it around on your computer), please use it from the final aion-master folder or higher. This aion-master folder should contain the four folders: aion, docs, logs & testing, and the 2 files: LICENCE.txt & README.txt

Folder 'logs'

This folder contains the log files generated by this program; these logs are time and date coded. The logs are generated by Aion, and are mainly used for debugging purposes, however they can be informative, as well as serve as a record of what has been done. For the development build, the log file shows all logging information not just information shown in Aion.

Input Paths (Buffered & Unbuffered)

When you provide a path to Aion it is taken and used as the working directory for Aion. Any output *.csv files or *.png files will be saved to that directory. This directory can include spaces as it is handled as a string. The path validity is checked before Aion continues.

File Name Format

Calibration

The filename format must conform to:
[defining_string]_pH[pH_number]_[repeat_number].txt

- The defining string can include any number of underscores
- The last two sections must be as shown above.

Example:
"2_G4_A2_pH4_1.txt"

Unbuffered

The filename format must conform to:
[defining_string]_pH[pH_number]_[repeat_number].txt

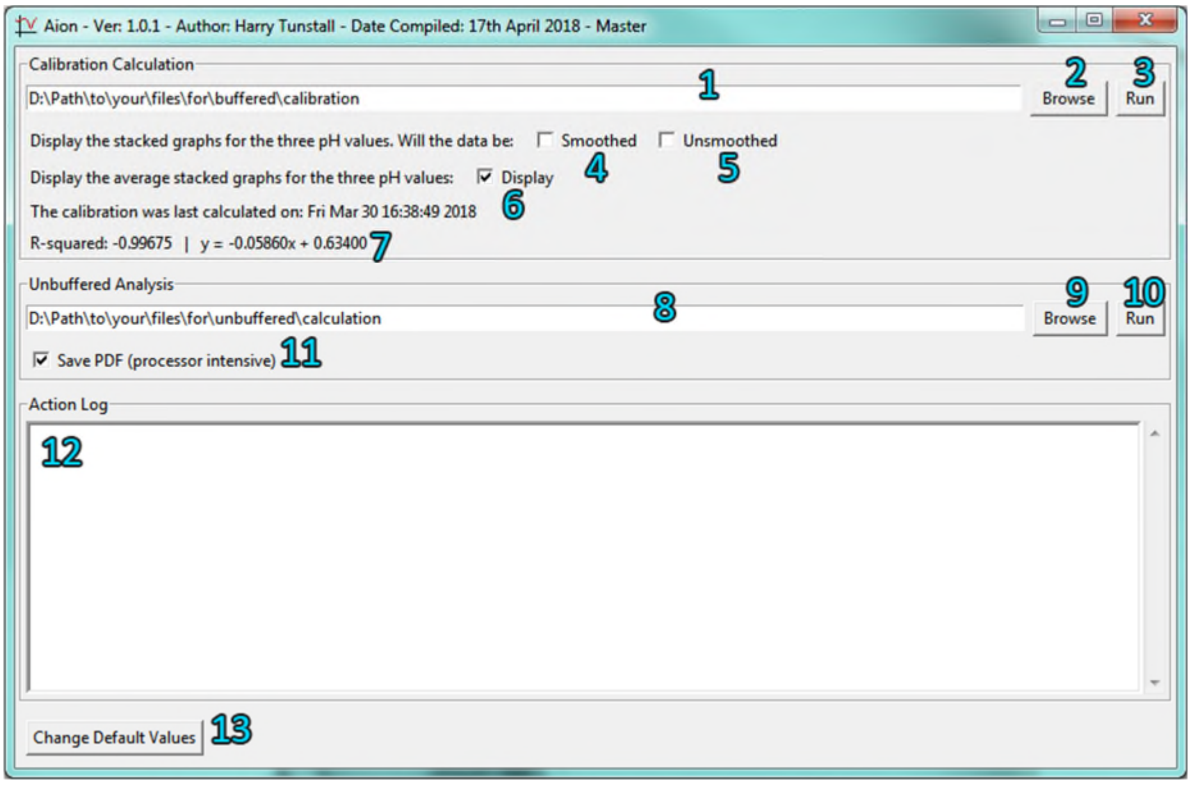
- The defining string can include any number of underscores
- The pH number must be the second to last item (as defined by underscores and shown above) and if a decimal point is required in the pH value a comma must be used
- The repeat number can include text such as "wider1" and the repeat number will be designated as 1. If no repeat number is given (the final underscore must still be present) then the repeat number will be listed as 0

Examples:

"1_E1_9,2_0118_pH4,07_.txt"	Repeat number = 0
"1_E1_9,2_0118_pH4,07_1.txt"	Repeat number = 1
"1_E1_9,2_0118_pH4,07_wider1.txt"	Repeat number = 1

Graphical User Interface (GUI)

Master


Aion - Ver: 1.0.1 - Author: Harry Tunstall - Date Compiled: 17th April 2018 - Master

Calibration Calculation

D:\Path\to\your\files\for\buffered\calibration

Display the stacked graphs for the three pH values. Will the data be: ☐ Smoothed ☐ Unsmoothed

Display the average stacked graphs for the three pH values: ☑ Display

The calibration was last calculated on: Fri Mar 30 16:38:49 2018

R-squared: -0.99675 | y = -0.05860x + 0.63400

Unbuffered Analysis

D:\Path\to\your\files\for\unbuffered\calculation

☑ Save PDF (processor intensive)

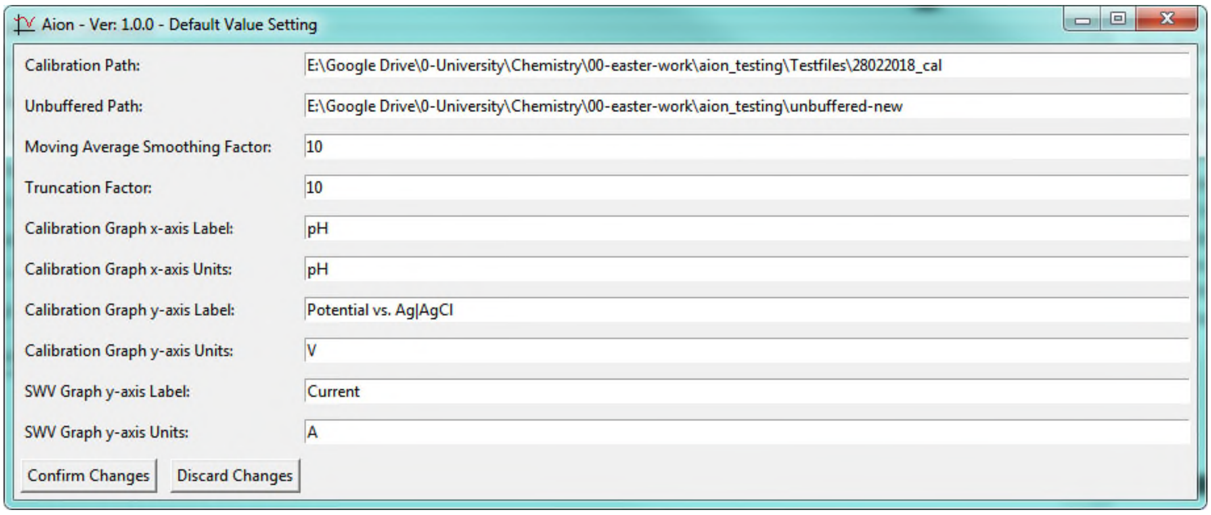
Action Log

Figure 1 - Above is master window labelled. This window appears when Aion is executed.

1. This is the location of the directory containing the files you wish to use to calibrate. It also becomes the current working directory for any output files that are created using the run button (3).
2. Brings up a browse file dialogue to select a path. Equally the path can be copied & pasted (must be ctrl+v to paste, the right click context menu will not work) into (1).
3. This button starts the calibration calculation progress. The date, R^2, slope & intercept values will be calculated, displayed (6) and saved for later use in Aion.
4. If checked this will display the ph 4, 7 & 10 graphs off all repeats, and the average, for the smoothed data. If unchecked, the graphs will still be saved, just not displayed to the user. The saved graphs are in the calibration directory, labelled accordingly.
5. See above, but for unsmoothed data
6. If this is checked a graph showing the average ph 4, 7 & 10 datasets will be displayed.
7. As mentioned in (3), this is the data from the calculated calibration line, displayed to the user. The date the calibration was performed is also shown.
8. This is the location of the directory containing the files you wish to use for unbuffered calculations. It also becomes the current working directory for any output files that are created using the run button (10).
9. Brings up a browse file dialogue to select a path. Equally the path can be copied & pasted (must be ctrl+v to paste, the right click context menu will not work) into (8).
10. This button started the unbuffered pH calculation process. Once the calculations have been completed a further GUI (Unbuffered Plot) will be opened, this is explained below.

11. If checked, then a PDF containing intermediate graphs will be created. This is highly recommended and is explained in the Appendix - Section: Unbuffered Second Derivative Analysis.
12. The log box keeps the user updated as to what Aion is currently doing. It is timestamped for ease of looking back. This data is also saved to the log file, however since this is a dev build much of the debugging data is also saved to this log file. The log box information is saved with the prefix of ":INFO:" in the log file (see figure 4 – log section).
13. Clicking this button enables the user to change some of the default values that Aion retains between executions.

Update Default Values


Aion - Ver: 1.0.0 - Default Value Setting

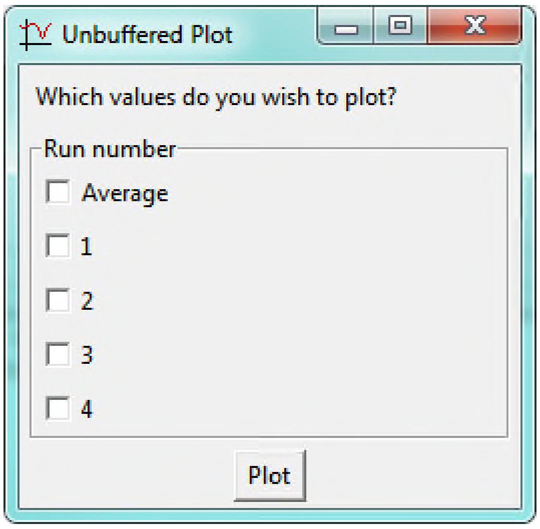
Calibration Path:	E:\Google Drive\0-University\Chemistry\00-easter-work\aion_testing\Testfiles\28022018_cal
Unbuffered Path:	E:\Google Drive\0-University\Chemistry\00-easter-work\aion_testing\unbuffered-new
Moving Average Smoothing Factor:	10
Truncation Factor:	10
Calibration Graph x-axis Label:	pH
Calibration Graph x-axis Units:	pH
Calibration Graph y-axis Label:	Potential vs. Ag|AgCl
Calibration Graph y-axis Units:	V
SWV Graph y-axis Label:	Current
SWV Graph y-axis Units:	A

Figure 2 - The GUI that loads when button "Change Defualt Values" (13) is pressed.

This is a simple GUI enables the user to change some of the values that are retained between executions of Aion. The bulk of the GUI is quite self-explanatory, the value is listed on the right inside an editable field, with the descriptor located on the left.

After changes have been made confirm changes must be clicked. Either closing out the window or clicking discard changed will **not** save the changes.

Unbuffered Plot


Unbuffered Plot

Which values do you wish to plot?

Figure 3 – Above, this dialogue is given to the user after the calculation is completed for the unbuffered files. When plot button is clicked a graph is produced with the data from the checked run numbers.

This GUI is used to give the user the option to plot only specific repeat numbers from the entire selection of files within the given directory. This GUI is dynamically generated based on the input data. Repeat numbers are pulled from the filenames, for example a file ending "_1.txt", "_wider1.txt" & "_random1lettersxyzabc.txt" would all be registered as repeat number 1. Only the values [0-9], as defined by regular expressions, are counted.

Output Files

Calibration

When the run button is used to calculate the calibration line multiple output files are generated in the given working directory. A total of 9 *.png graphs are created & 7 *.csv output files are created.

A graph (*.png) and *.csv file is created for each pH for both smoothed and unsmoothed data. Then the calibration line graph is created and saved ("calibration_line.png"), as well as a list of all the pH peaks 2nd derivatives referenced by their filename and pH value. Two other graphs are also saved, the smoothed and unsmoothed average stacked pH 4, 7 & 10 datasets.

Unbuffered

When the run button is used to calculate the calibration line multiple output files are generated in the given working directory. A maximum total of 3 files are created: one "ub_data.csv" file, one "unbuffered_scatter.png" & one "all_graphs.pdf" but only if save PDF (button 11 – Figure 1) is checked.

The file "ub_data.csv" contains all the data calculated from the input *.txt files. It is effectively a saved version of the data structure used by Aion.

"unbuffered_scatter.png" is a saved version of the displayed output graph.

Finally, "all_graphs.pdf" contains all 5 intermediate graphs for each *.txt file given and metadata from the *.txt file header.

Code Structure

Directory Structure

```
./aion-master
./aion-master/aion                       # Aion files & logic
./aion-master/aion/classes
./aion-master/aion/classes/classes.py
./aion-master/aion/data_minip
./aion-master/aion/data_minip/ph_script.py
./aion-master/aion/data_minip/unbuffered.py
./aion-master/aion/file_handling
./aion-master/aion/file_handling/conf_values.py
./aion-master/aion/log
./aion-master/aion/log/functions.py
./aion-master/aion/ui
./aion-master/aion/ui/aion.ico
./aion-master/aion/ui/master.py
./aion-master/aion/ui/plot_unbuffered.py
./aion-master/aion/ui/settings.py
./aion-master/aion/aion.py               # Run this to start Aion
./aion-master/aion/default.ini           # Saves settings between executions
./aion-master/docs                       # Documentation folder
./aion-master/logs                       # location of saved log files
./aion-master/testing                    # location of testing files
./aion-master/testing/calibration
./aion-master/testing/calibration_expected_output
./aion-master/testing/unbuffered
./aion-master/testing/unbuffered_expected_output
```

aion-master/aion

This is the location for Aion, and all its dependencies to be executed. This section will go into further detail about each of the python files.

aion.py

This script starts Aion. The role of this file is to get the default values from the default.ini file and save then to active variables used throughout Aion. After this the master GUI is called.

classes/classes.py

The default values used throughout this program is a single variable defined by this class. This is because when looking though the code it is easier to understand what variable contains what value if it is called from an object. Also, when passing through multiple procedures, passing only one variable is very helpful.

data_minip/ph_script.py

This script is largely unchanged from its source. It was originally the script “zoes_ph_script.py” created by Jonathan Duncan for use to calculate calibration lines. The core logic has not been modified, apart from removing the calculation of derivatives for the unsmoothed data, which was later not used.

The modifications consist of creating a copy of the relevant segments of data and saving them to a 3D sparse array. In python terms it is a list of a 6-membered tuple with 3 elements of that tuple also being either lists or arrays.

This data structure is then saved to the relevant output *.csv files through calling of additional procedures.

data_minip/unbuffered.py

This script contains procedures procured and modified from “ph_script.py”. Any procedures that did not require changing was simply called from the “ph_script.py” file. The logic behind finding the turning point (crossing point) was completely changed. However, the logic behind smoothing and calculating the 2nd derivative is unchanged, as these are implemented to imitate the Origin Pro graphing software.

Please see the appendix (Section: Unbuffered Second Derivative Analysis) for full documentation of this section, mainly the method behind how the final algorithm is designed.

There are two called procedures plot(*args) which is called by the “plot_unbuffered.py” GUI, and unbuffered_calculation(*args) which is called by “master.py” when the run button for unbuffered is clicked.

The main logic is in the “find_zero_crossing_point_mod” procedure and its sub routines. These have been commented with more detail. The comments in conjunction with the appendix (section: Unbuffered Second Derivative Analysis) should give enough detail about the working of this file.

The definition of the start of the data is to look for “Potential/V” and any lines after this is the dataset. If CHI were to change this in future versions this string is located in the procedure “find_file_start_row_headder” in unbuffered.py, at around line 550.

file_handling/conf_values.py

This handles the saving and retrieval of values to and from the default.ini file. This is done by calling either get(), set() or set_list(). Values will only be saved to the default.ini file if a set procedure is called.

log/functions.py

This handles the use of the logging feature. When aion.py is called the init() procedure is called. This created a log file with the relevant file name (defined below – logs section). Whenever something is to be logged the log() procedure is called. When called, and given a message, the message is saved and displayed in the relevant places.

ui/master.py

This contains most of the logic for Aion since it is the main/master form. This is called by aion.py into the show() procedure. The GUI is built and then each button is set to a procedure defined lower in the script. The logic in these procedures is the core of Aion referencing all the other *.py files to use their procedures.

The addition of unbuffered calculations, however, I see requiring another GUI. This GUI will be called as a toplevel and will get defined within a new *.py file.

ui/plot_unbuffered.py

This handles purely the GUI for plotting the graph output for the unbuffered procedure. The unbuffered.py file calls a toplevel window which gets built by the show() procedure.

ui/settings.py

This handles purely the GUI for changing the default values in the default.ini file and Aion. The master GUI calls a toplevel window which gets built by the show() procedure.

aion-master/docs

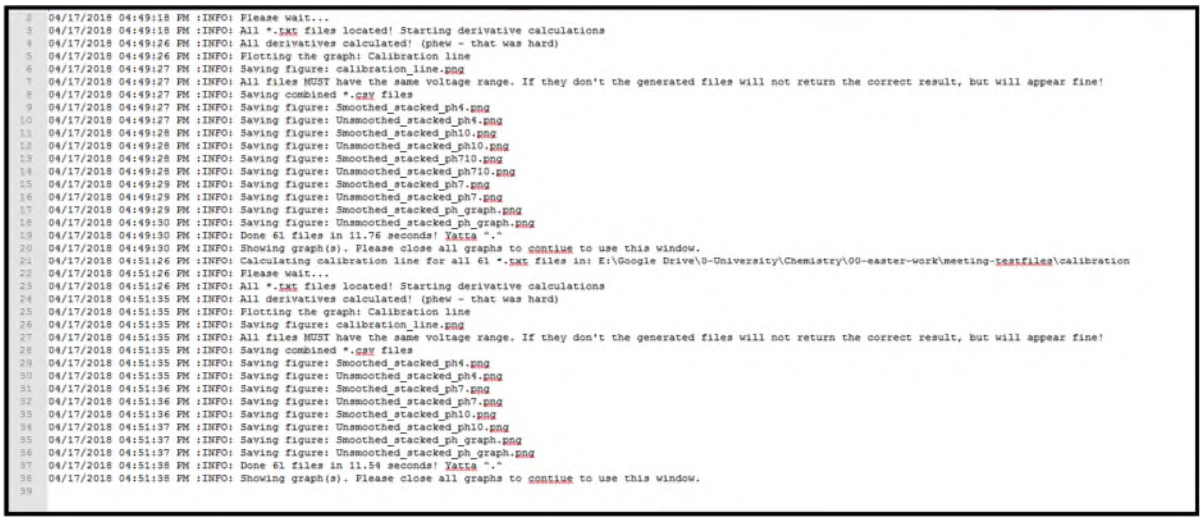
This is the location of this file, and any other supporting information.

aion-master/logs

This folder contains the log files generated by Aion. The generated log files are date-time stamped at the start of Aion runtime in the format:

YYYY-MM-DD--HH-MM-SS

These log files contain the information printed to the log box in the master GUI. While the program is in development the log file contains debugging information too. Below (figure 3) is an example of a log file whilst in development mode, the log box information can be found by the prefix ":INFO:".



```
04/17/2018 04:49:18 PM :INFO: Please wait...
04/17/2018 04:49:18 PM :INFO: All *.txt files located! Starting derivative calculations
04/17/2018 04:49:26 PM :INFO: All derivatives calculated! (phew - that was hard)
04/17/2018 04:49:26 PM :INFO: Plotting the graph: Calibration line
04/17/2018 04:49:27 PM :INFO: Saving figure: calibration_line.png
04/17/2018 04:49:27 PM :INFO: All files MUST have the same voltage range. If they don't the generated files will not return the correct result, but will appear fine!
04/17/2018 04:49:27 PM :INFO: Saving combined *.csv files
04/17/2018 04:49:27 PM :INFO: Saving figure: Smoothed_stacked_ph4.png
04/17/2018 04:49:27 PM :INFO: Saving figure: Unsmoothed_stacked_ph4.png
04/17/2018 04:49:28 PM :INFO: Saving figure: Smoothed_stacked_ph10.png
04/17/2018 04:49:28 PM :INFO: Saving figure: Unsmoothed_stacked_ph10.png
04/17/2018 04:49:28 PM :INFO: Saving figure: Smoothed_stacked_ph710.png
04/17/2018 04:49:28 PM :INFO: Saving figure: Unsmoothed_stacked_ph710.png
04/17/2018 04:49:29 PM :INFO: Saving figure: Smoothed_stacked_ph7.png
04/17/2018 04:49:29 PM :INFO: Saving figure: Unsmoothed_stacked_ph7.png
04/17/2018 04:49:29 PM :INFO: Saving figure: Smoothed_stacked_ph_graph.png
04/17/2018 04:49:30 PM :INFO: Saving figure: Unsmoothed_stacked_ph_graph.png
04/17/2018 04:49:30 PM :INFO: Done 61 files in 11.76 seconds! Yatta ^.^
04/17/2018 04:49:30 PM :INFO: Showing graph(s). Please close all graphs to contiue to use this window.
04/17/2018 04:51:26 PM :INFO: Calculating calibration line for all 61 *.txt files in: E:\Google Drive\0-University\Chemistry\00-easter-work\meeting-testfiles\calibration
04/17/2018 04:51:26 PM :INFO: Please wait...
04/17/2018 04:51:26 PM :INFO: All *.txt files located! Starting derivative calculations
04/17/2018 04:51:35 PM :INFO: All derivatives calculated! (phew - that was hard)
04/17/2018 04:51:35 PM :INFO: Plotting the graph: Calibration line
04/17/2018 04:51:35 PM :INFO: Saving figure: calibration_line.png
04/17/2018 04:51:35 PM :INFO: All files MUST have the same voltage range. If they don't the generated files will not return the correct result, but will appear fine!
04/17/2018 04:51:35 PM :INFO: Saving combined *.csv files
04/17/2018 04:51:35 PM :INFO: Saving figure: Smoothed_stacked_ph4.png
04/17/2018 04:51:35 PM :INFO: Saving figure: Unsmoothed_stacked_ph4.png
04/17/2018 04:51:36 PM :INFO: Saving figure: Smoothed_stacked_ph7.png
04/17/2018 04:51:36 PM :INFO: Saving figure: Unsmoothed_stacked_ph7.png
04/17/2018 04:51:36 PM :INFO: Saving figure: Smoothed_stacked_ph10.png
04/17/2018 04:51:37 PM :INFO: Saving figure: Unsmoothed_stacked_ph10.png
04/17/2018 04:51:37 PM :INFO: Saving figure: Smoothed_stacked_ph_graph.png
04/17/2018 04:51:37 PM :INFO: Saving figure: Unsmoothed_stacked_ph_graph.png
04/17/2018 04:51:38 PM :INFO: Done 61 files in 11.54 seconds! Yatta ^.^
04/17/2018 04:51:38 PM :INFO: Showing graph(s). Please close all graphs to contiue to use this window.
```

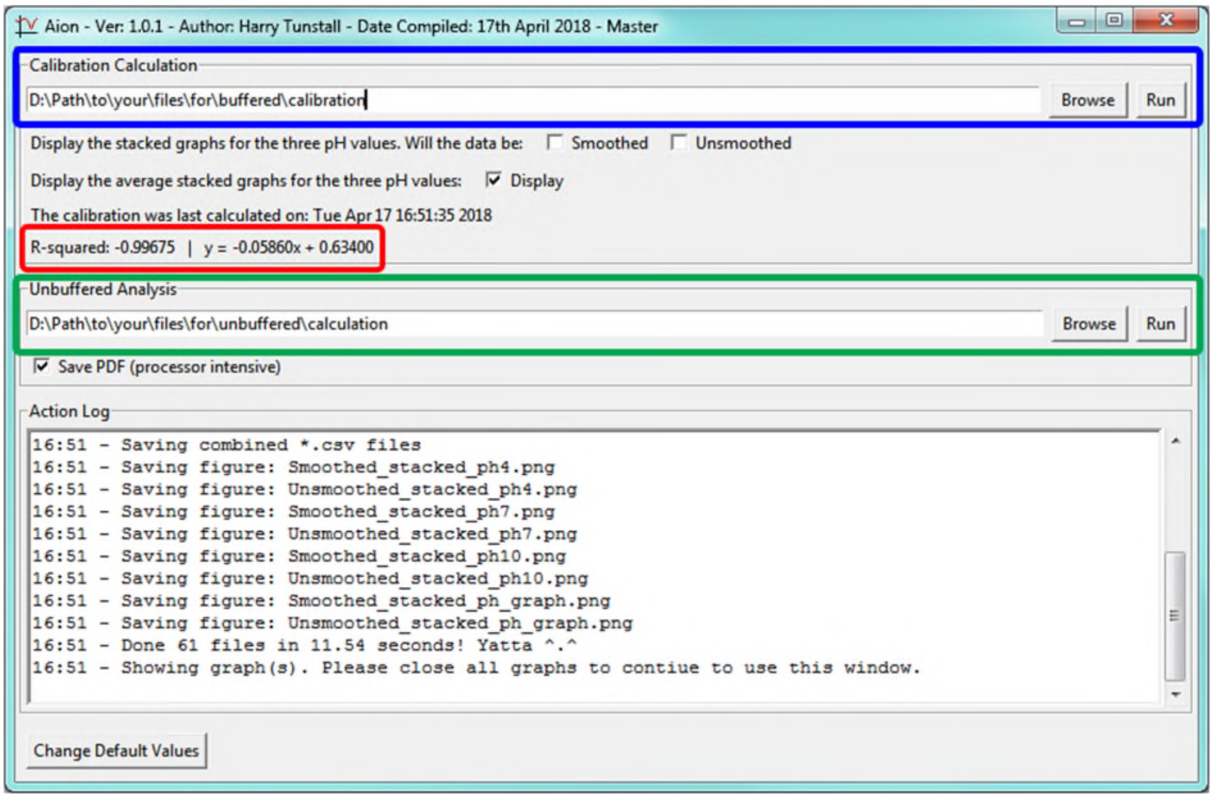
Figure 4 – An example log file generated by Aion 1.0.1

aion-master/testing

This folder contains the files for calibration & unbuffered testing. It also contains folders with the expected output files. To run the test please see the outlined steps below.

Information for Running the Unbuffered Testing

1. Point the unbuffered path to ~\aion-master\testing\calibration (Figure 4 – blue)
2. Run the calibration calculation (Figure 4 – blue)
3. Check the expected output and the actual output conform with each other (Figure 4 – red).
   > Further checks can be made with the expected output located:
   ~\aion-master\testing\calibration_expected_output
4. Point the unbuffered path to ~\aion-master\testing\unbuffered (Figure 4 – green)
5. Run the unbuffered calculations (Figure 4 – green)
6. When prompted about the invalid dataset, select "yes". The data is invalid.
7. When prompted to select which data points to plot, plot: Average, 1 & 5
8. Compare the output, located in the unbuffered folder, to the expected output. The expected output is located ~\aion-master\testing\unbuffered_expected_output


Aion - Ver: 1.0.1 - Author: Harry Tunstall - Date Compiled: 17th April 2018 - Master

Calibration Calculation

D:\Path\to\your\files\for\buffered\calibration

Display the stacked graphs for the three pH values. Will the data be: ☐ Smoothed ☐ Unsmoothed

Display the average stacked graphs for the three pH values: ☑ Display

The calibration was last calculated on: Tue Apr 17 16:51:35 2018

R-squared: -0.99675 | y = -0.05860x + 0.63400

Unbuffered Analysis

D:\Path\to\your\files\for\unbuffered\calculation

☑ Save PDF (processor intensive)

Action Log

```
16:51 - Saving combined *.csv files
16:51 - Saving figure: Smoothed_stacked_ph4.png
16:51 - Saving figure: Unsmoothed_stacked_ph4.png
16:51 - Saving figure: Smoothed_stacked_ph7.png
16:51 - Saving figure: Unsmoothed_stacked_ph7.png
16:51 - Saving figure: Smoothed_stacked_ph10.png
16:51 - Saving figure: Unsmoothed_stacked_ph10.png
16:51 - Saving figure: Smoothed_stacked_ph_graph.png
16:51 - Saving figure: Unsmoothed_stacked_ph_graph.png
16:51 - Done 61 files in 11.54 seconds! Yatta ^.^
16:51 - Showing graph(s). Please close all graphs to contiue to use this window.
```

Figure 4 – Above is an annotated version of the master Aion GUI. Highlighted in blue is where the expected input is for calibration calculation is required, along with the associated run button. Highlighted in red is the expected output once Calibration has been completed. Please verify this is identical. Highlighted in green is the entry fields for unbuffered analysis.

Appendix

Unbuffered Second Derivative Analysis

This code has gone through multiple stages of development, the key stages were:

- Initial Development From "ph_script.py"
- Advanced Data Manipulation
- Final Algorithm

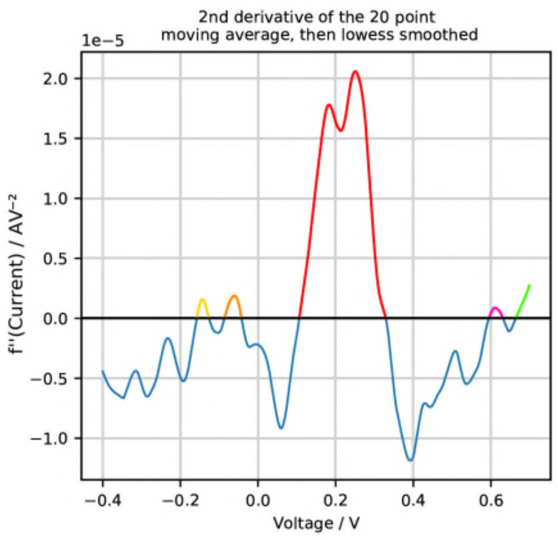
Initial Development From "ph_script.py"

The changed logic was in the "find_zero_crossing_point_mod" function. Before, this procedure found the smallest negative value and iterated from there to find the turning point. However, this was bugged for some of the input files for unbuffered due to noise. The initially discovered edge cases in this instance were two potential scenarios: (the array y is an array of the y-axis' second derivative)

1. The array y would start negative (as expected) however a pseudo turning point (negative to positive value change) would occur earlier, flicking to positive for about 20 indexes. The array y would return to negative value until the true crossing point was reached.
2. The array y would start positive, for about 50 indexes, before turning negative (as expected). The true crossing point was then reached at the next change from negative to positive.

The solution initially implemented to solve both these problems is as follows:

- Two lists are generated – one for the indexes of y[i] which relate to a positive value, and one for the indexes of negative y[i] values.
- In the positive value list, we now have all the indexes that correspond to the positive values in the y[i] array. Shown below in Appendix 1 by the coloured sections not in blue. The zeroth item from this list would return the index just after the crossing point if only the red section was present.

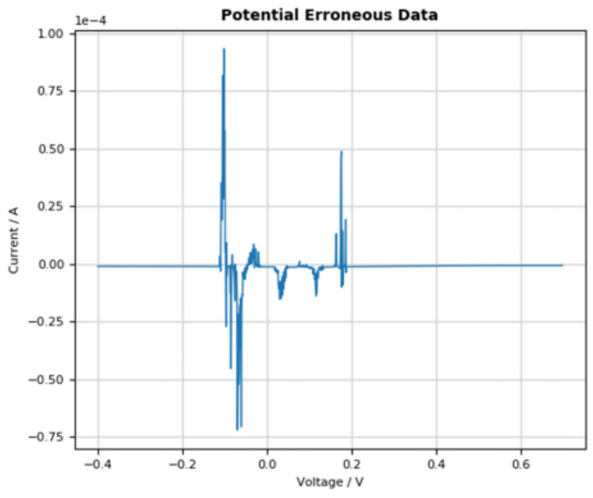


Appendix 1 – Above, a plot of the 2nd derivative, after lowess smoothing. The 2nd derivative was calculated from a 20-point moving average applied to the collected data. The sections shown in: green, magenta, red, orange & yellow are the sections saved to the positive index list. In the next section this list is broken into sublists shown by the respective colours.

- To solve both problems with the same solution, Aion simply looked for a sequential number of indexes. Initially this was set to 100, then 200. The choses number had to be large enough to notice issue 2, but not too large as to over extend the dataset. If a sublist of this positive index array is all sequential then we can deduce that the initial index of that sublist is indeed the index just after the true crossing point.
    - Further information on the sequential number (100, or 200) – The program would occasionally error out or skip a file. This would be due to the logic for determining the turning point failing (this is an edge case and hard to account for in this implementation).
- If, however, the sublist was not sequential then the sublist is shifted one index down the list of positive indexes and is tested again.
- Eventually, this sublist will be sequential. When this is the case the sublists zeroth index is the true crossing point.
    - To visualise this implementation, see Appendix 1. The list would contain the indexes of the green, magenta, then red etc. (left to right) sections. The sequential number of 200 would be longer than the green and magenta sections. Therefore, you could only have a fully sequential sublist if the list started where the red section started.

Advanced Data Manipulation

The previous implementation was poor coding. The use of arbitrary numbers is never good. Also, it was discovered that sometimes completely invalid datasets were provided, see Appendix 2 for the dataset in question. A simple solution for this was to flag datasets that contained positive values for current. All flagged datasets are shown to the user asking to verify if the dataset is indeed invalid.

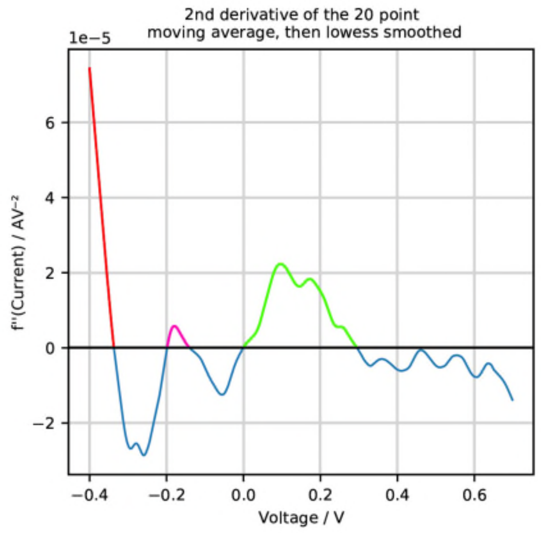


Appendix 2 – This is the plot given by Aion when it encounters potentially erroneous data. The criteria for flagging a dataset is the presence of positive current values.

To abstract the problem to a more algorithmic approach. As seen from Appendix 1 multiple turning points (located to the right of each coloured section) can be present in a dataset.

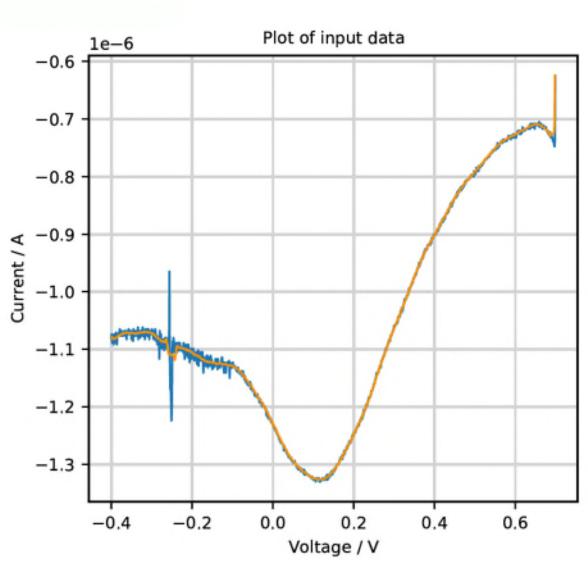
First, the list of all the positive indexes was split into a list of each sublist. Each section: green, magenta, red, orange & yellow were saved as separate lists contained within a list structure.

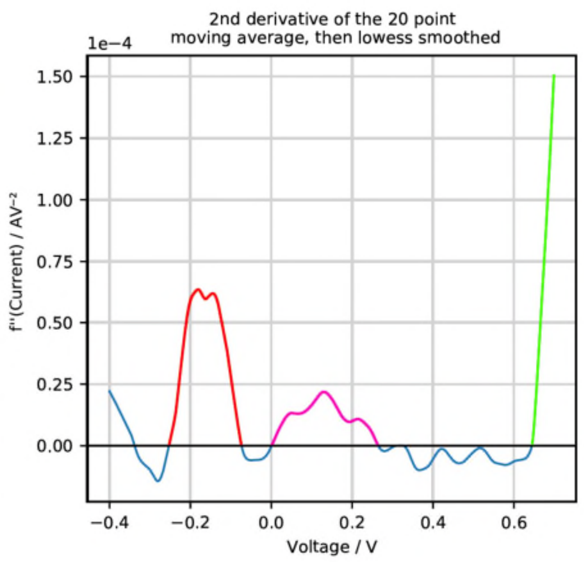
From observation of the graph in Appendix 1 (red) & 3 (green – ignoring sublists adjacent to the scan limits) it was noted that the true turning point would be in the section with the largest maximum value for the 2nd derivative.



Appendix 3 – Above, another example where the sublist containing the largest value of the 2nd derivative is the true turning point. When any sublists adjacent to the scan limits are ignored.

Something else of note was that if positive sections started or ended at the maximum index then these sections could be ignored. This was implemented, as shown in Appendix 4 right, because the spikes at the start or end of a dataset would be higher than the true turning point peak.



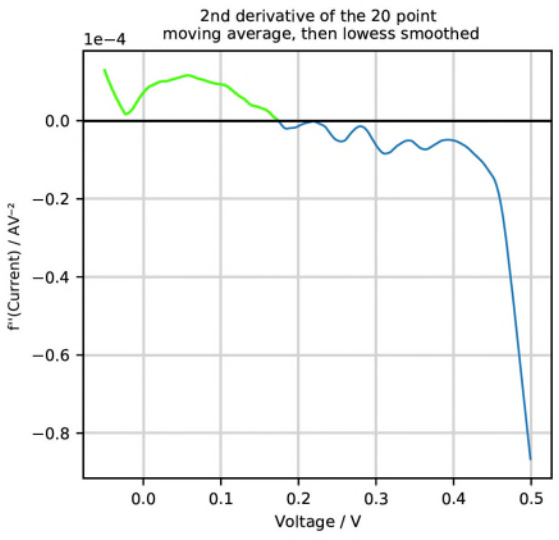


Appendix 4 – Above left, the raw input data for the file "18_G1_pH6,01_1.txt" (blue), with the 20-point moving average also shown (orange). Above right, shows both the above described starting peak (green) and the issue with noise creating a false peak (red), and the true turning point section (magenta).

The described algorithmic logic worked for most cases, however it was destroyed by a large instance of noise (Appendix 4, left). This noise was significant enough to make a larger 2nd derivative spike (Appendix 4, right - red) than the true turning point (Appendix 4, right - magenta).

Final Algorithm

The final solution was to smooth further smooth the 20-point moving average smoothed data (Appendix 4, left - orange) by a large amount. It was also discovered that some datasets had the true turning point in the edge spikes (Appendix 5). A solution to this was also achieved.

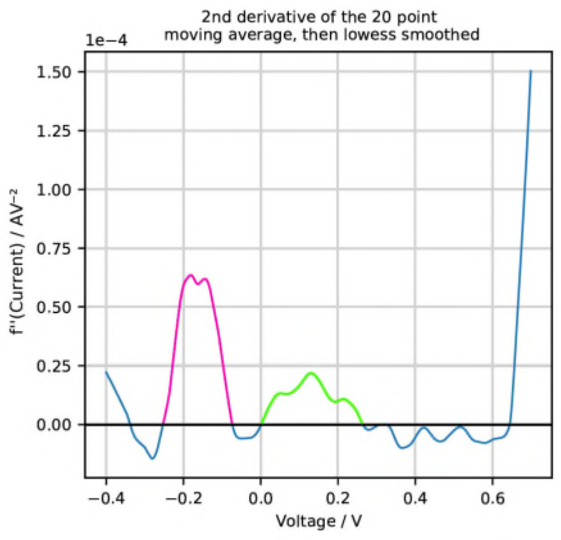


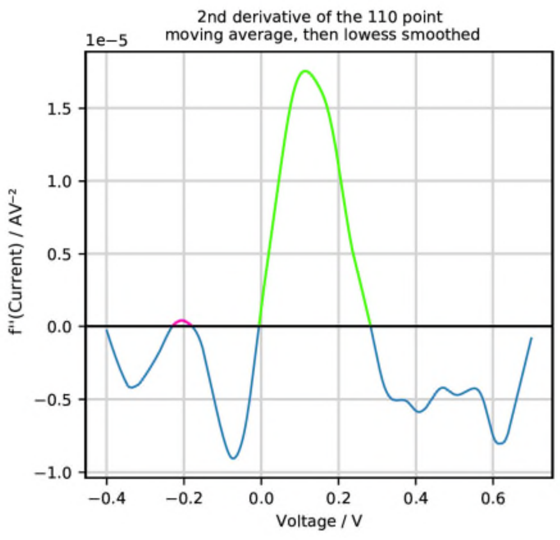
Appendix 5 – Above, shows an example where the true turning point is in a section which touches the edge of the scan region.

To solve the problem shown in Appendix 5, rather than completely ignore the sublists which touch the limits of the scan range, a section is taken off the end of the sublist then the maximum derivative is found. The size of the section is calculated using a scale factor. This scale factor can be adjusted by the user by changing the default values if need be, however it is recommended not to change it unless necessary. The scale factor is currently set to 10, so of a 550 data point dataset, 55 points will be excluded from the end of a section if it touches the bounds of the scan region.

Something of note, when implementing this - if the final section was less than the number of data points being removed, then the sublist was deleted. If it was not deleted an empty sublist was left which caused errors within the code.

When calculating the further smoothed dataset, a similar implementation was used to the above problem; a scale factor was used. The scale factor is currently also set to 10, however they are independent variables. For example, if a 1100 point dataset is used then a 110-point moving average is applied to the 20-point moving average.

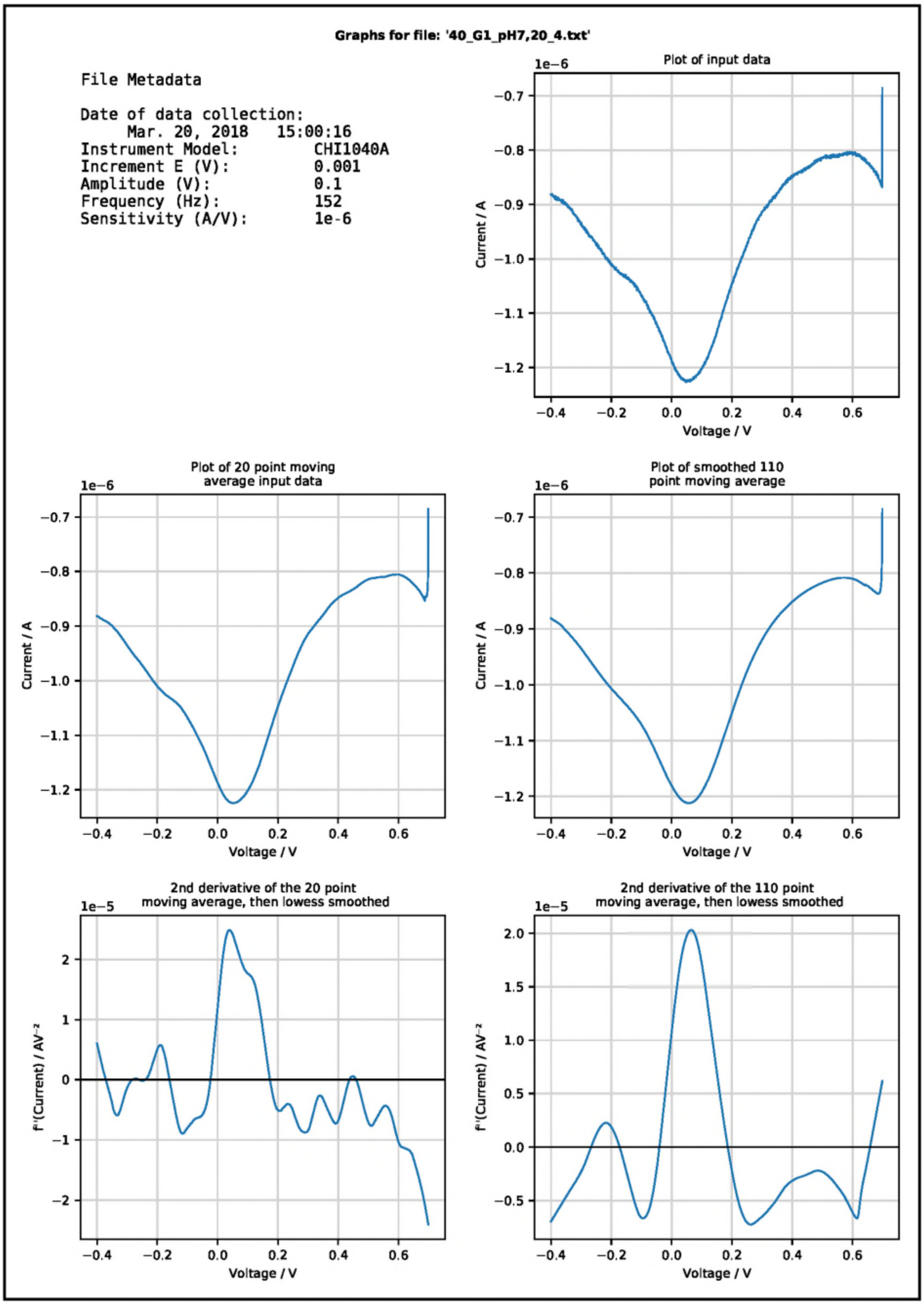




Appendix 6 – Above, these graphs are both on the same dataset as Appendix 4. Left, the standard 20-point moving average 2nd derivative. Right, the further smoothed 110-point moving average, 2nd derivative. Both 2nd derivatives were then lowess smoothed. Both the left and right graphs use the same colour for each sublist. The true turning point is in the green sublist. The false turning point, created by noisy data, is in the magenta sublist.

Using the further smoothed dataset, the same logic is applied which was described above. The largest value for the 2nd derivative will be the sublist which contains the true turning point. As shown in Appendix 6 the green section can be correctly selected using this algorithm.

After this implementation was complete it was decided that a PDF should be saved with all the relevant data used to calculate the 2nd derivative, and hence, the pH value for the solution. A check box was added to the GUI to allow the choice of whether this file is created. Appendix 7 is an example page from the created document.


Graphs for file: '40_G1_pH7,20_4.txt'

```
File Metadata

Date of data collection:
     Mar. 20, 2018   15:00:16
Instrument Model:          CHI1040A
Increment E (V):           0.001
Amplitude (V):             0.1
Frequency (Hz):            152
Sensitivity (A/V):         1e-6
```

*Appendix 7 – Above, this is an example of one of the pages of the generated PDF file. Each page of the PDF is for a datafile, listed in bold at the top. Each graph has their respective title, and in the top left information about the experiment is captured from the header of the *.txt file and saved to the PDF.*