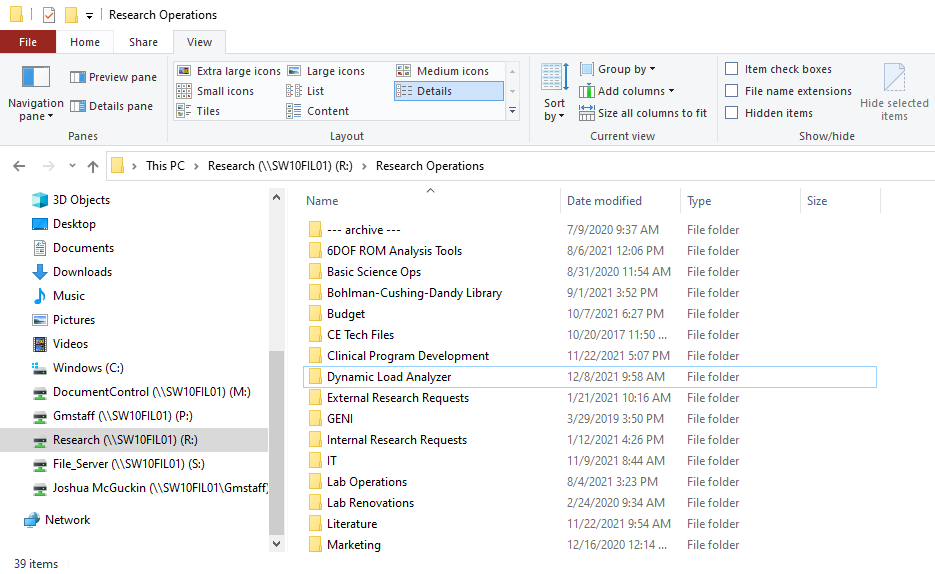
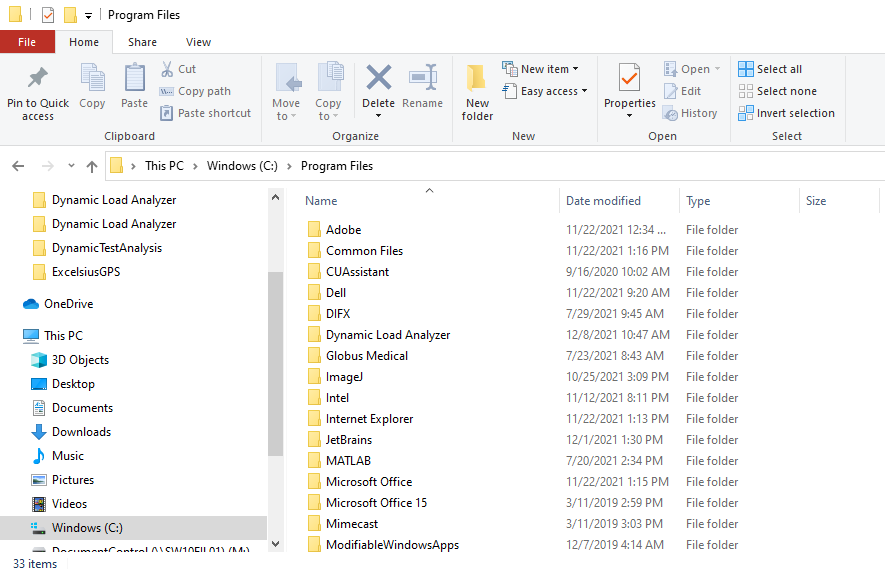
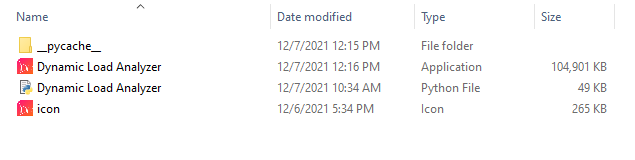
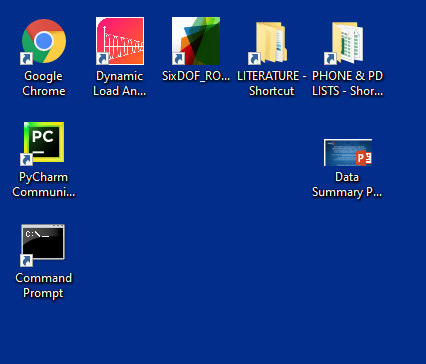
# To Install App (Dynamic Load Analyzer):

* Go to research server (R:)
* In the ‘Research Operations’ folder, open ‘Dynamic Load Analyzer’
  + As shown below:



* There is a subfolder also called ‘Dynamic Load Analyzer’
  + Copy this folder into C:\Program Files
  + Note: To make changes to your computer’s C: drive, you will need admin access. If you do not already have this access, you can instead save this folder to your desktop or another user-specific location

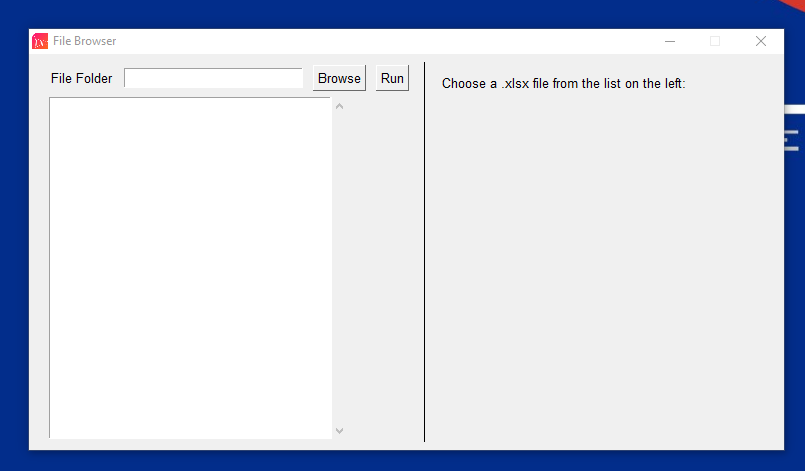


* Once copied, open this folder
* Right-click on the ‘Dynamic Load Analyzer’ application file and select ‘create a shortcut’ as shown below: 
* This will generate an icon on your desktop to access the application as shown below: 

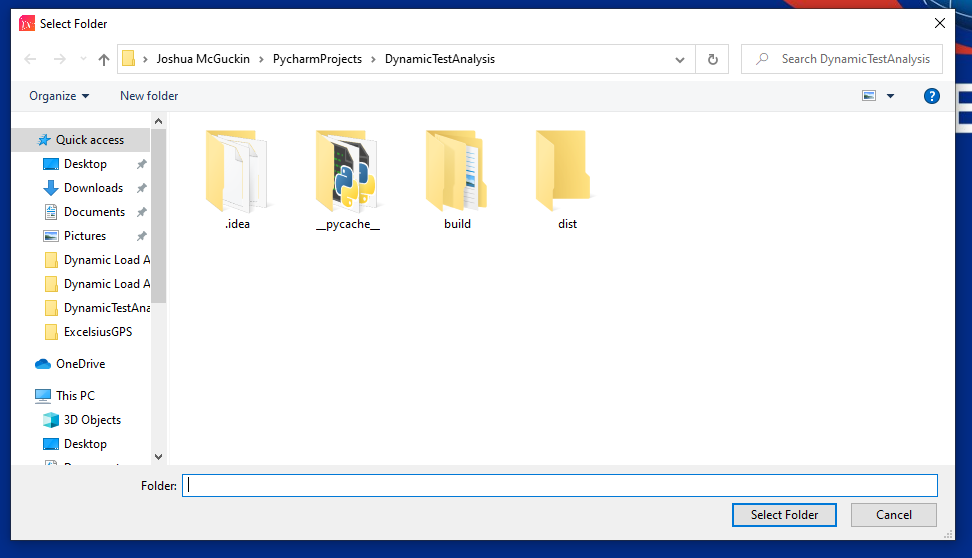
# To Run App (Dynamic Load Analyzer):

Select and Import File

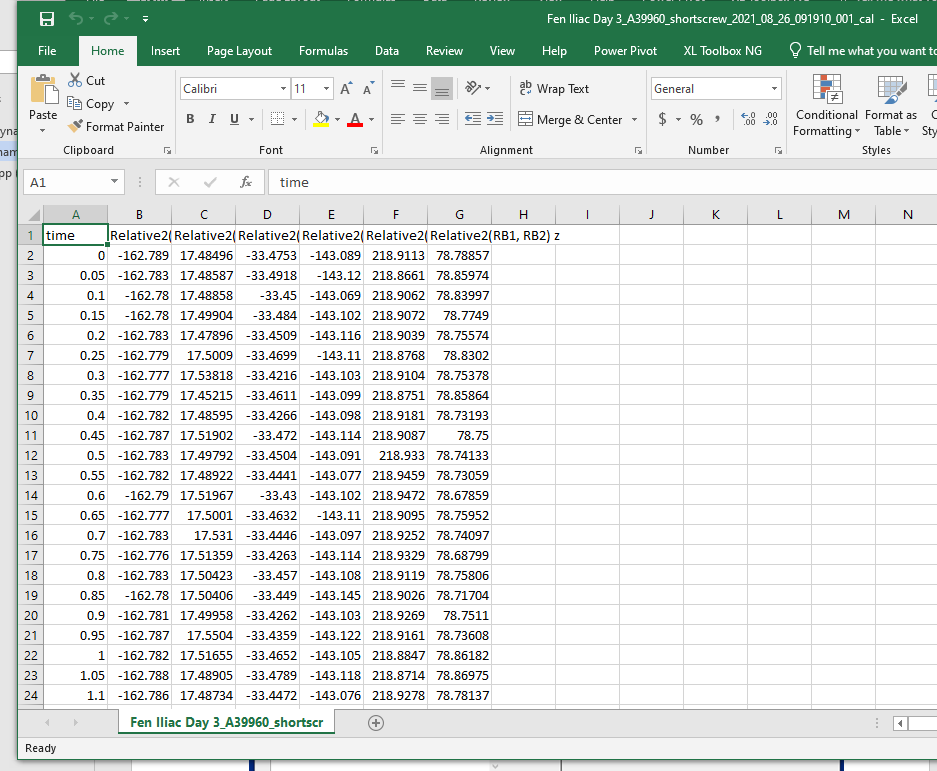
* Double click on the ‘Dynamic Load Analyzer’ icon
* In the popup menu, press ‘Browse’



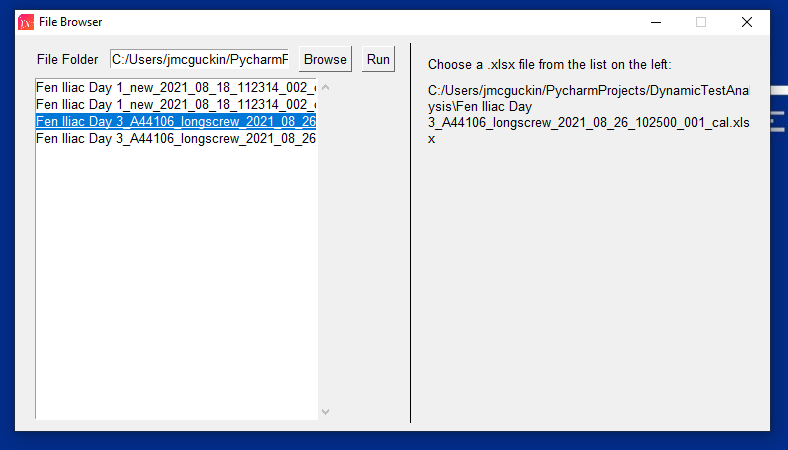
* Select the folder you’d like to work in



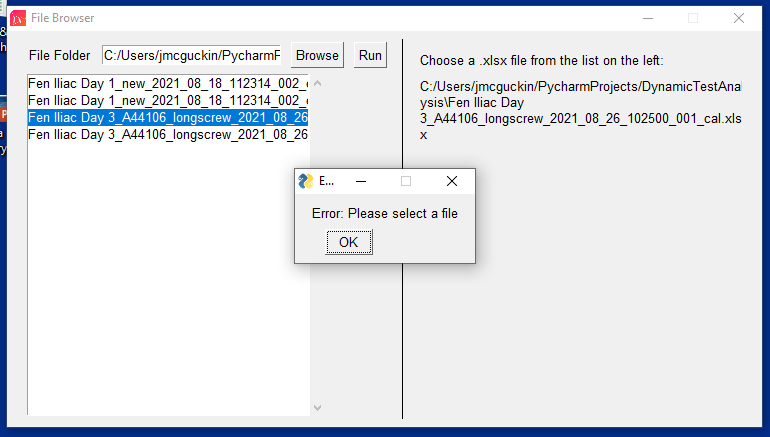
* A list of all .xlsx files in the selected folder will be shown in the left column
  + Note: The cal file you save from the 6DOF will be saved as a spreadsheet (.xls). Before running this application, you need to open the cal file and resave it as a workbook (.xlsx). Otherwise, it will not show up in this list.
  + Should look like the figure below:



* From the list, click on the file you’d like to work on
  + The path of the selected file will appear in the right column
* Then press ‘Run’ to continue

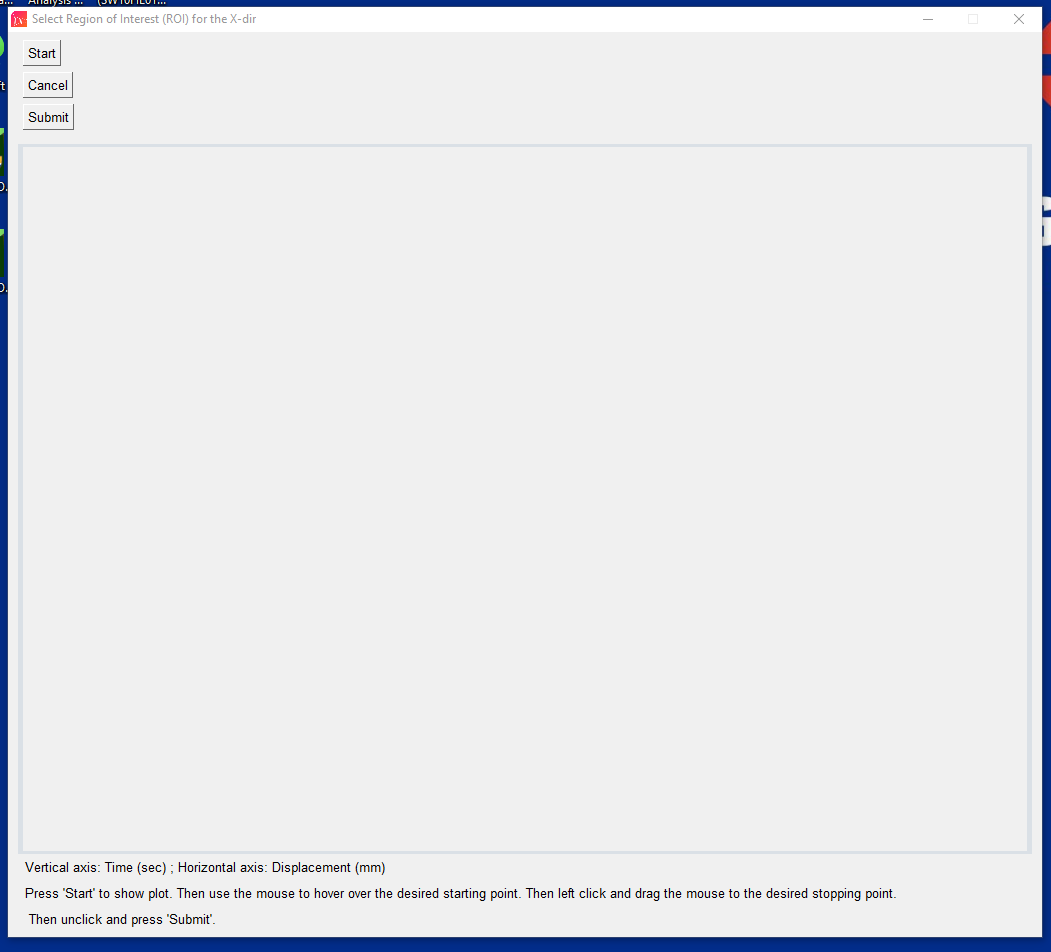


* If you try pressing ‘Run’ without clicking on an .xlsx file from the list, you’ll be prompted with an error message asking you to select a file

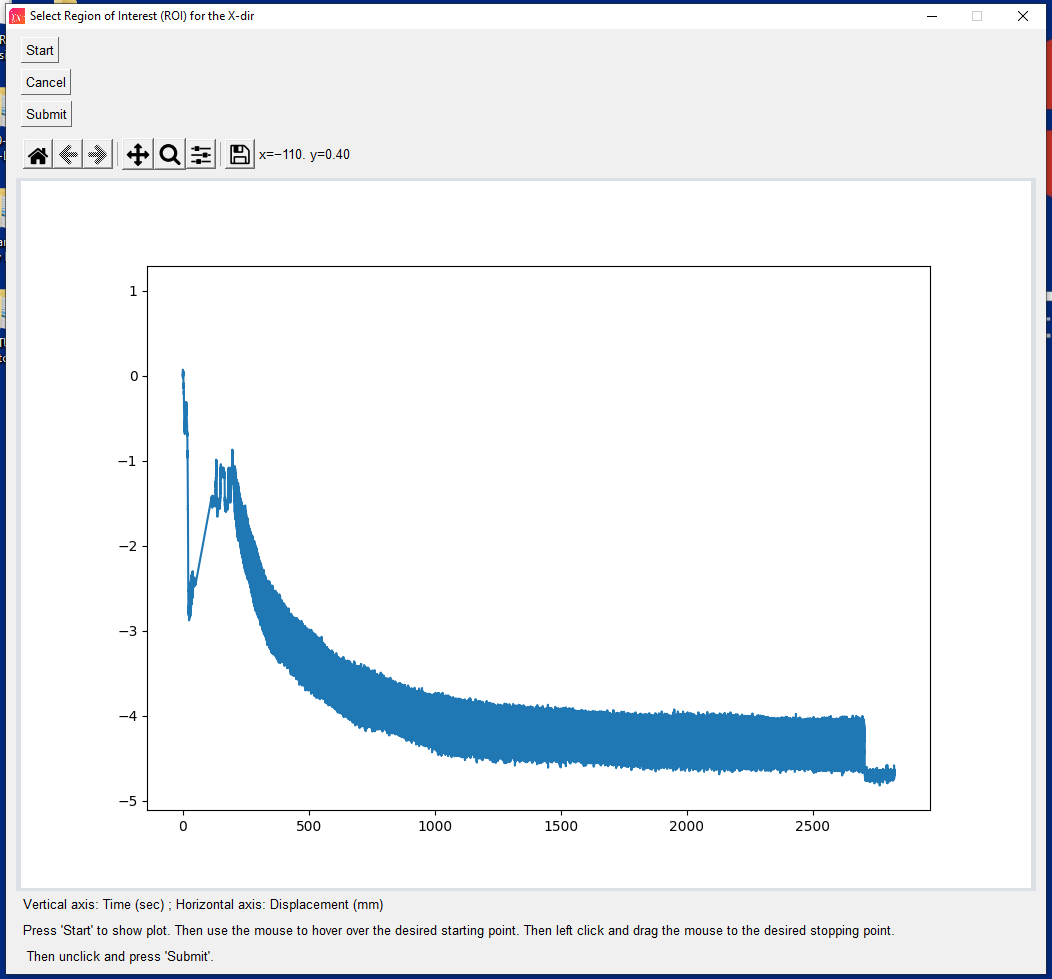


Select Region of Interest (ROI)

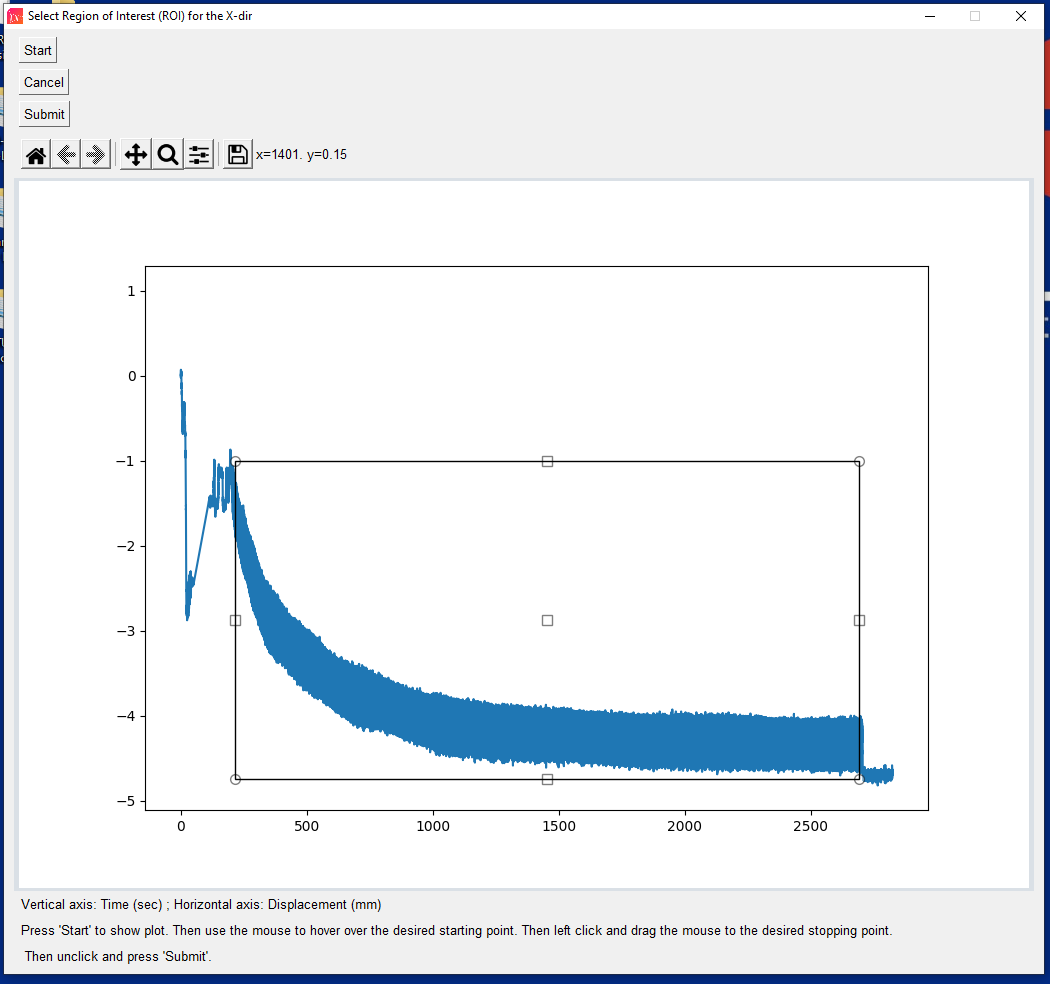
* A new popup window labeled ‘Select Region of Interest (ROI) for the X-dir’ will appear
* This window shows the dynamic displacement data captured from the Optotrak sensors in the X-axis
* Press ‘Start’ to continue



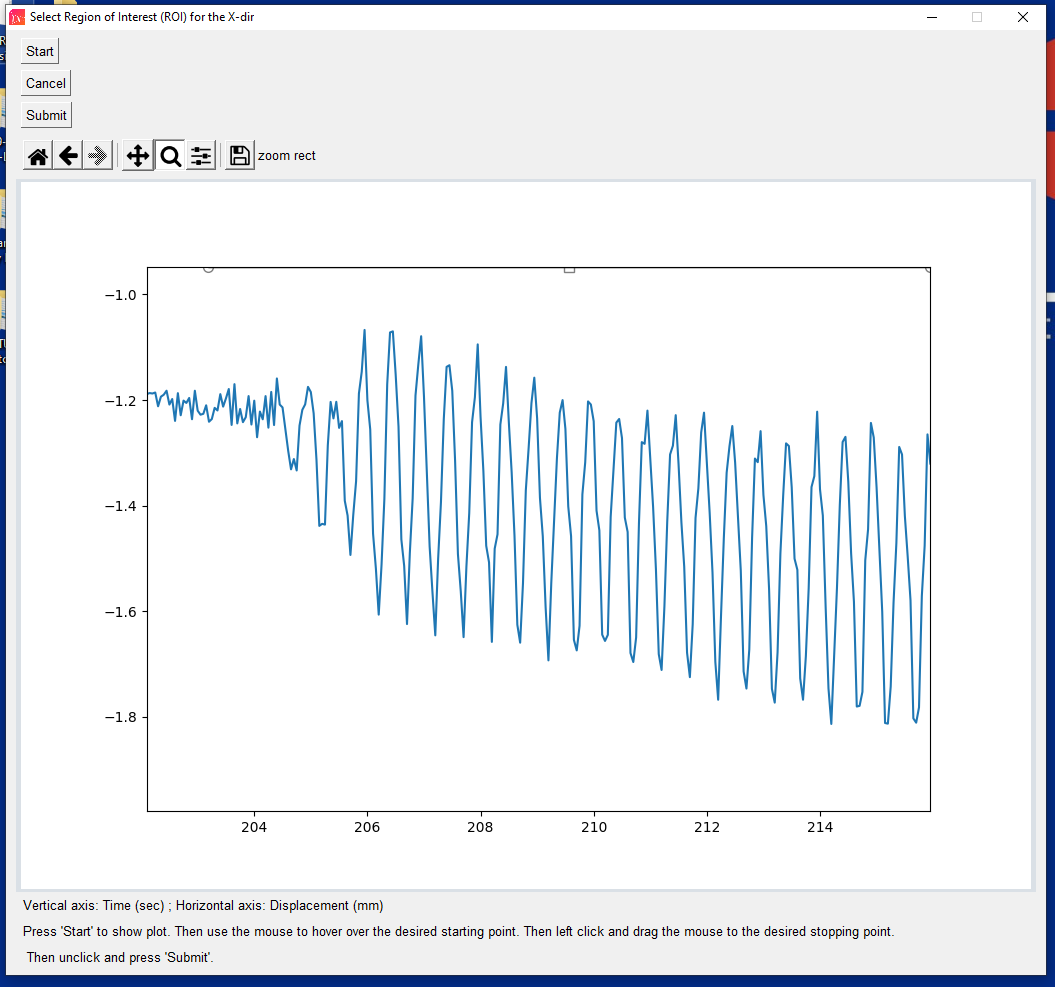
* A MATLAB plot of the time-dependent displacement data in the X-axis will be generated



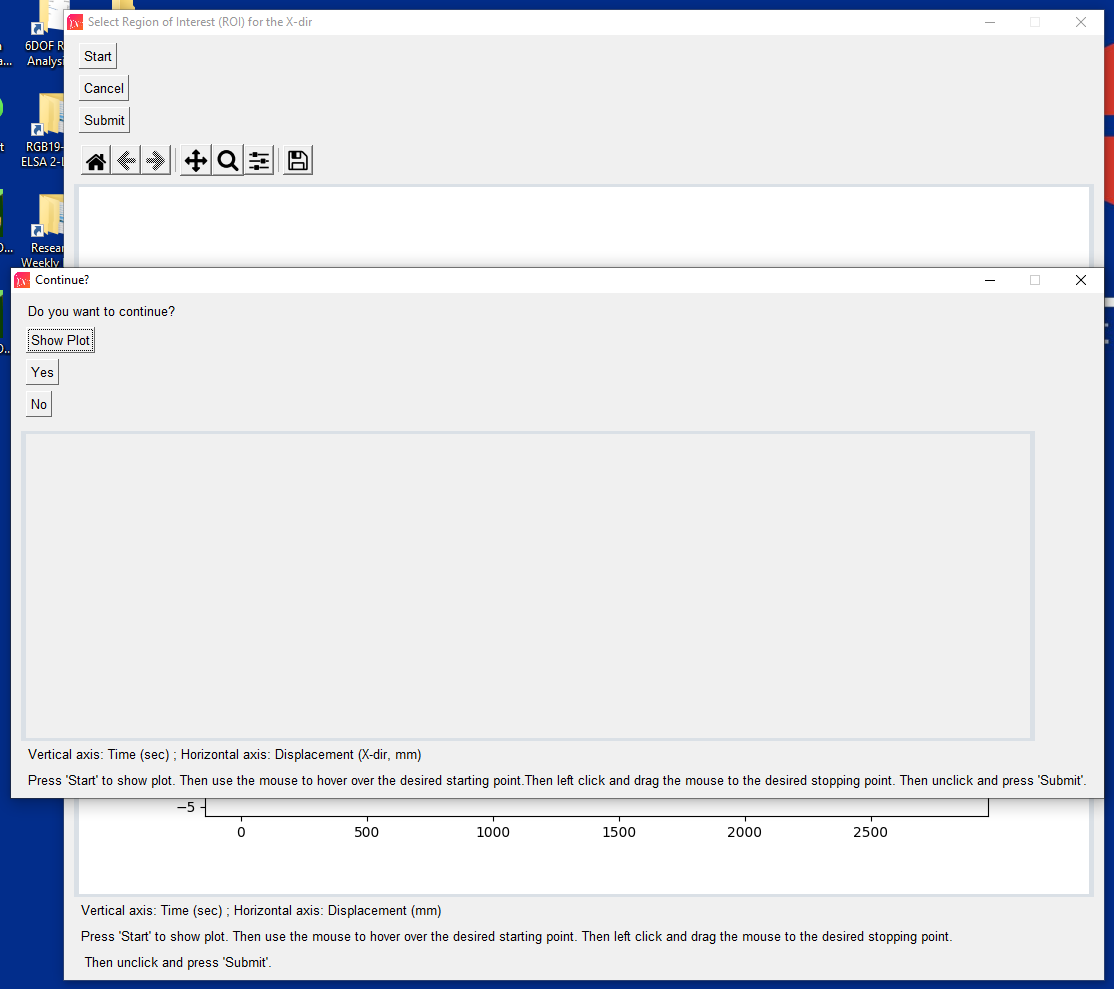
* Next, select the region of interest (ROI) where you would like to analyze the displacement data
* Do this by clicking on your desired start point, dragging your curser, and then releasing it on your desired stop point
  + This will generate a rectangle overlay that represents the ROI you selected
  + Note: Typically, the best ROI is starts where the cyclic displacement has become stable (i.e. peaks and valleys are not changing quickly), and stops right before the cycling displacement suddenly drops.
  + Note: You can draw the ROI as many times as you please. Only the last ROI will be used.
* Once you are satisfied with the ROI, press ‘Submit’ to continue



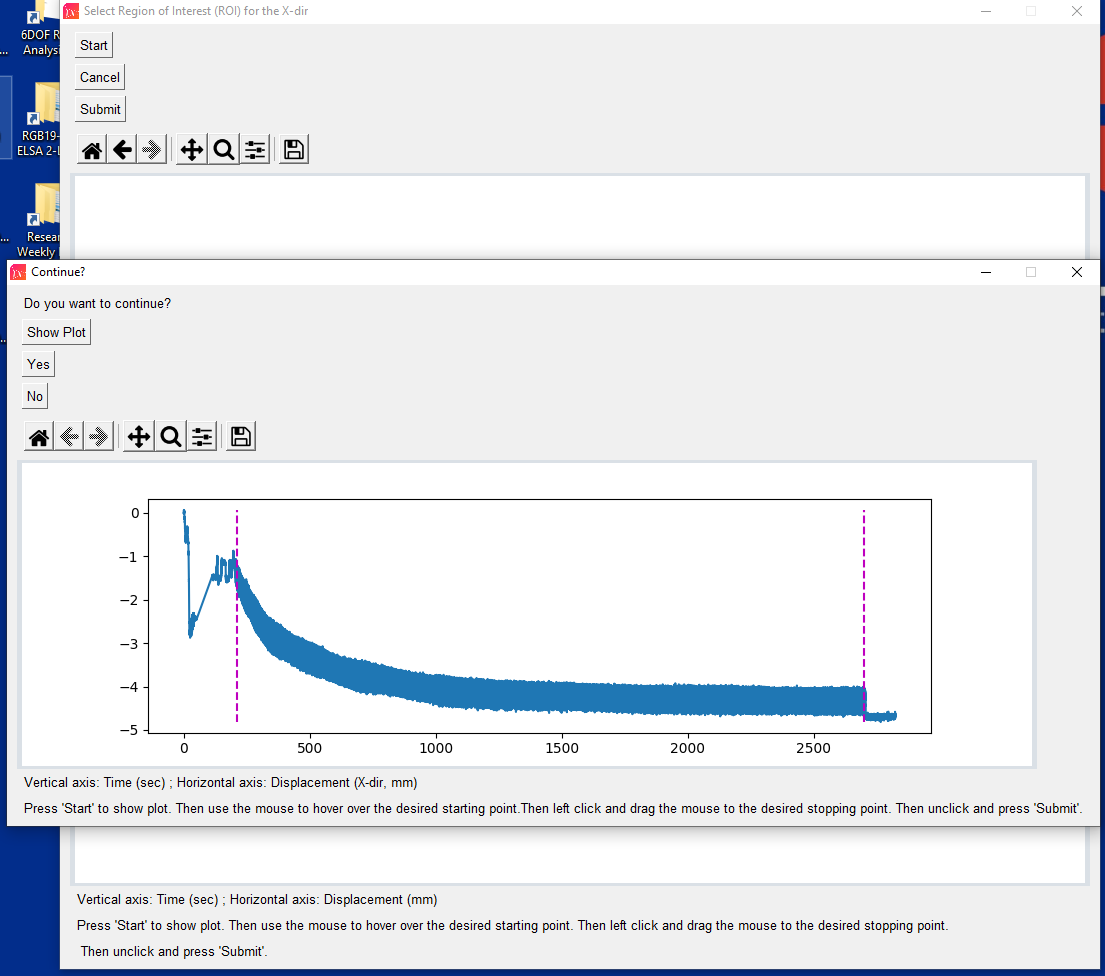
* Note: To help find the best start and stop points for the ROI, use the zoom button (**RED**) to zoom in on a selected area in the same manner as you drew the rectangle to select the ROI. Be sure that when you are ready to draw the ROI that you use the home button (**BLUE**) to re-size the plot so that you can capture the start and stop points.



* Another popup window will appear asking if you are satisfied with the ROI selected and would like to continue
* Press ‘Show Plot’ to see the plot of the time-dependent displacement data and the ROI



* If you are satisfied with the ROI, press ‘Yes’ to continue
* If you are unsatisfied with the ROI, press ‘No’ or close the window to reselect the ROI

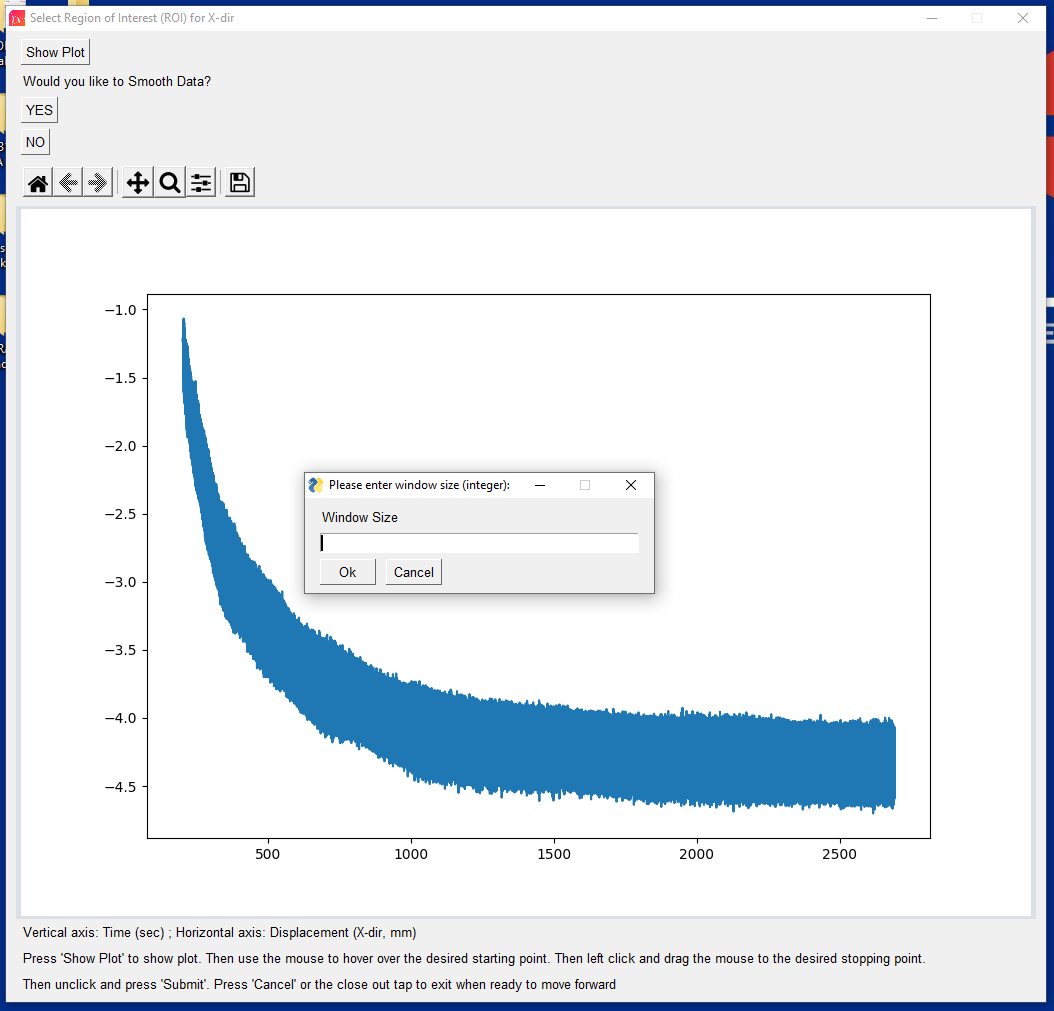


Smooth Data? - Y/N

* A new popup window will appear asking if you would like to smooth the data
  + Note: Smoothing data is typically done to reduce the noise associated with a signal to achieve a higher “signal-to-noise ratio”. This is done by combining neighboring points in a specified window size to make the data less noisy and more easily analyzed.
* If you think that smoothing the data might be necessary, press ‘Yes’ to continue
* Otherwise, press ‘No’ or close the window to continue



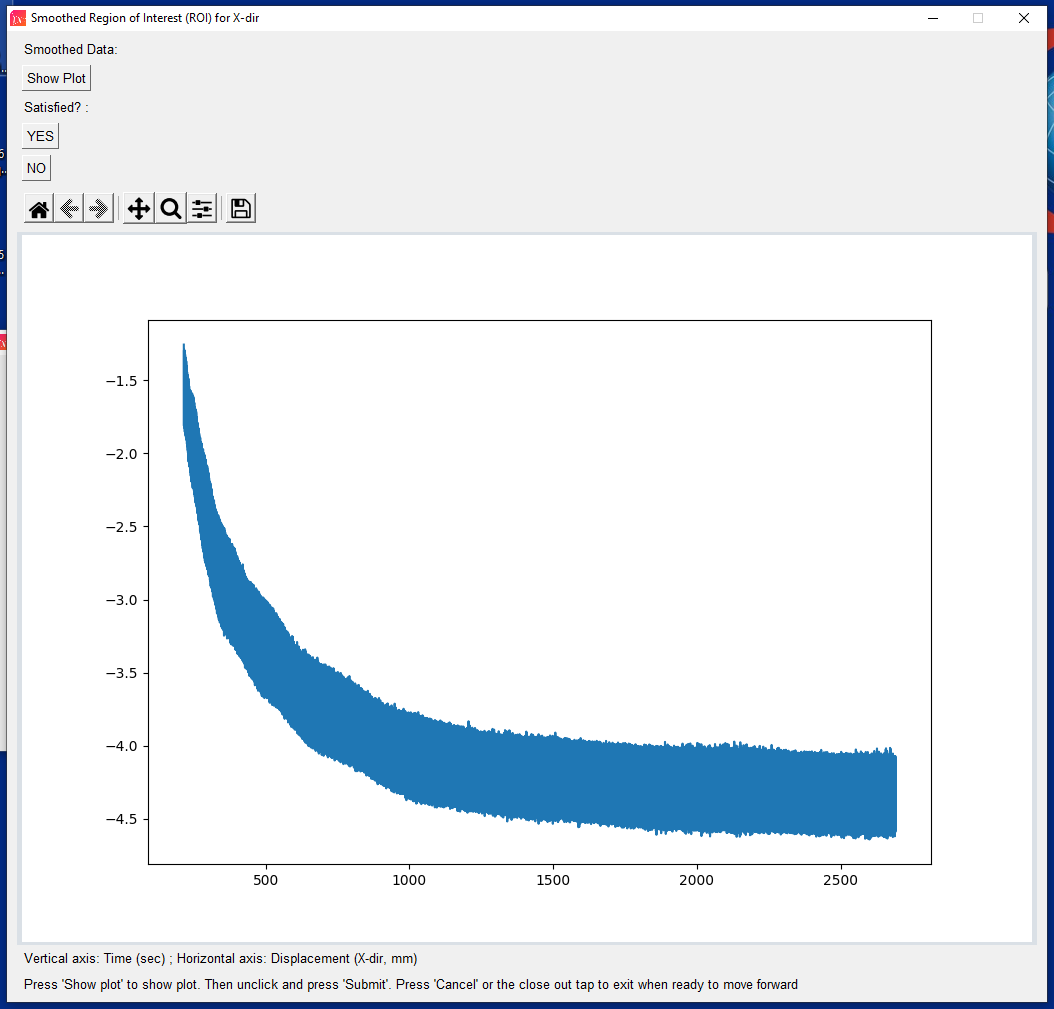
* If you pressed ‘Yes’, a popup window will appear asking for you to enter a window size.
* Enter an integer for your window size, and press ‘Ok’
  + Note: A window size is the number of adjacent points in the data that will be averaged to smooth the data. Typically, best practice is to minimize the window size to the smallest required to accurately capture the peaks and valleys of the data (e.g. window size = 2 or 3). This is because over-smoothening the data will distort the frequency and amplitude of peaks and ultimately can affect findings.



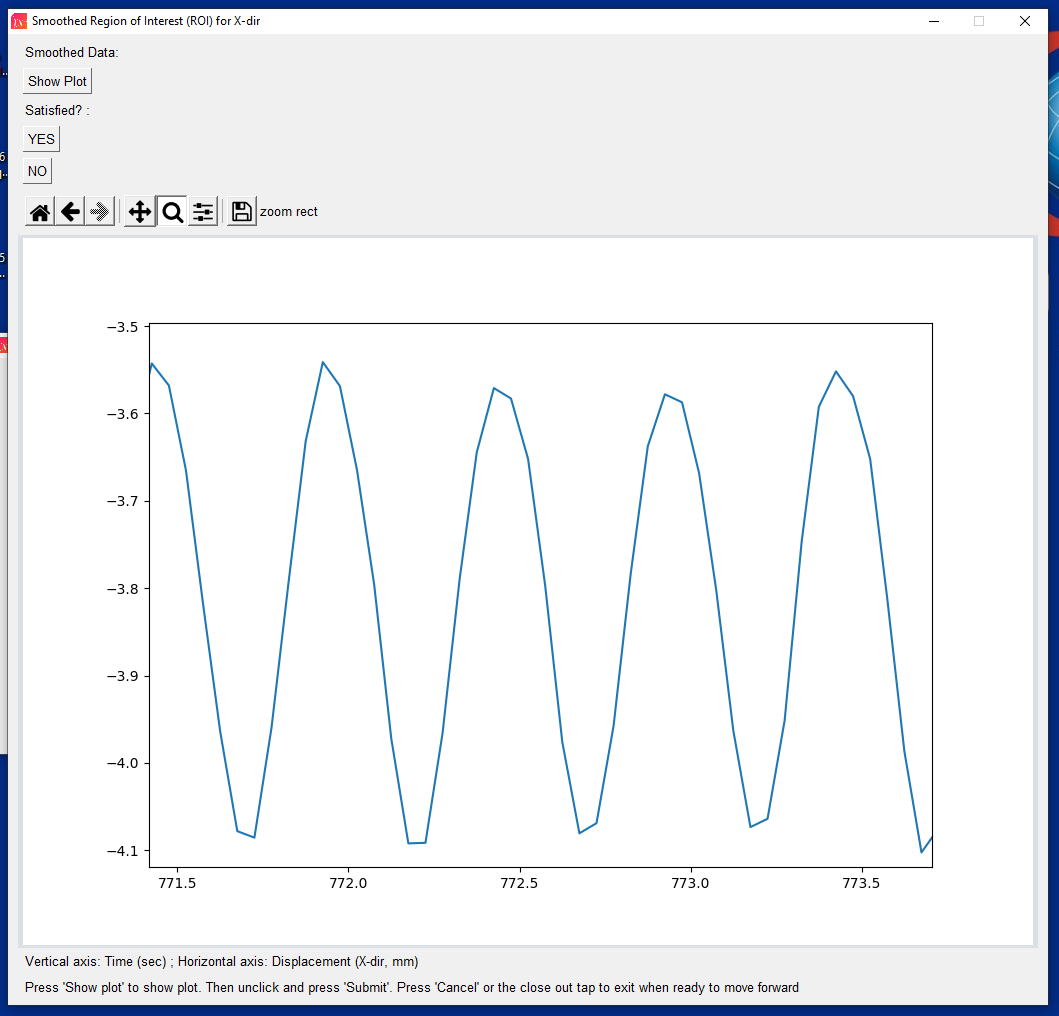
* After entering a window size, a new window will appear
* Press ‘Show Plot’ to see the smoothed time-dependent displacement data



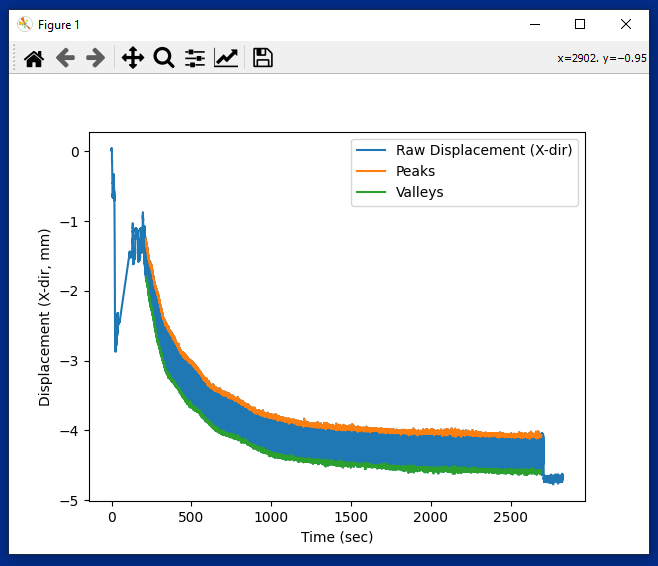
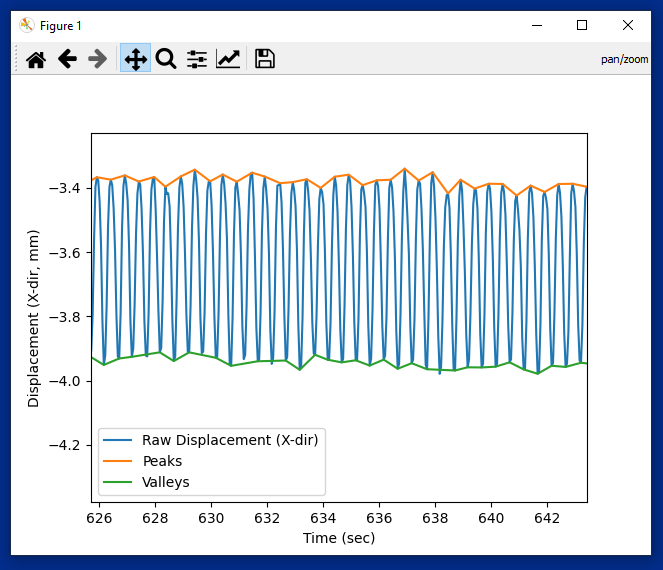
* Press the zoom button of the MATLAB toolbar and zoom in on a region of the data to inspect if there is any residual noise in the smoothed data



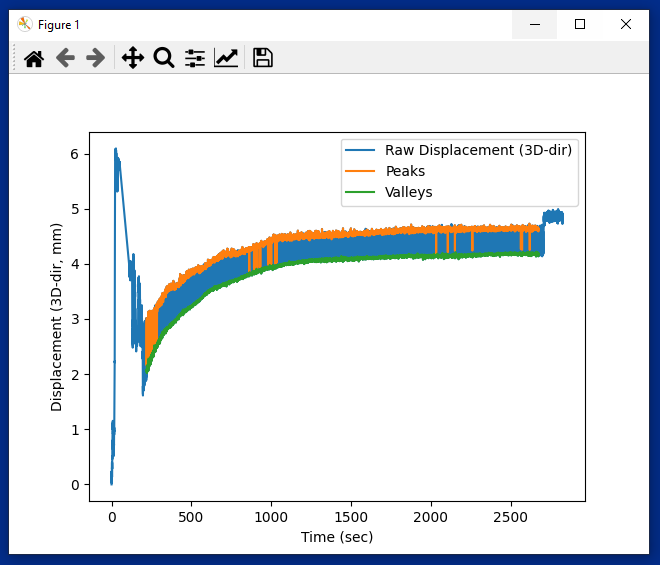
* If peaks and valleys are clearly identifiable (single points) then the data is suitable to move on
  + Note: This step can be critically important because the application assumes that if any point is greater/less than two points before and after it than it is a peak/valley.
* If you are satisfied by the smoothed data, press ‘Yes’ or close the window to continue
* If you are unsatisfied by the smoothed data and would like to re-smooth or not smooth at all, press ‘No’ and you will be returned to the previous window.

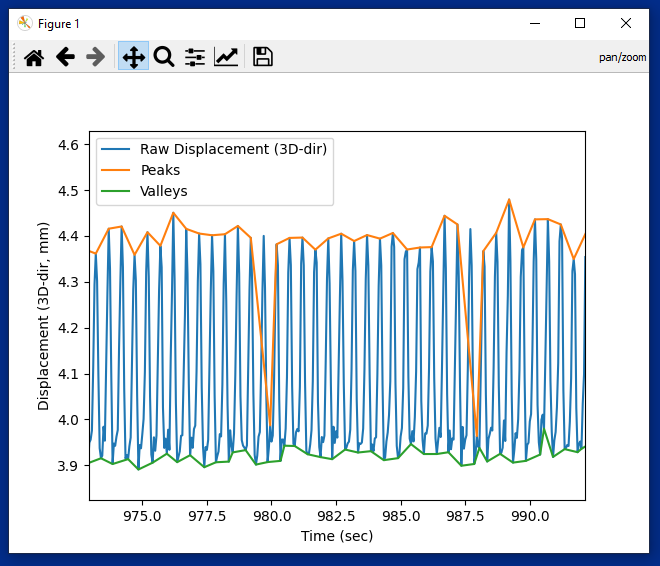


* Figures will be automatically generated showing the analyzed displacement data
* Each figure window much be closed to move on
* Note: The figure below is a great example of data that would not require smooth as the peaks and valleys are correctly identified.



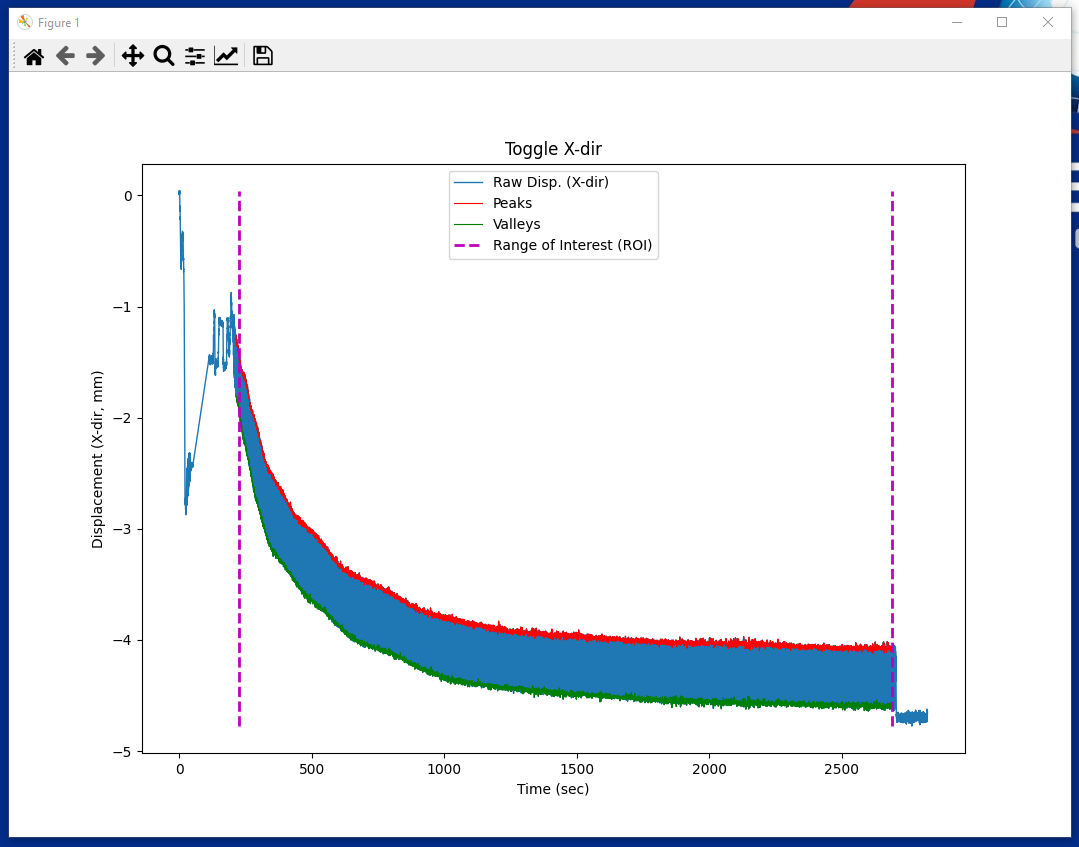
* The figure below is a great example of data that might see some benefit from smoothing



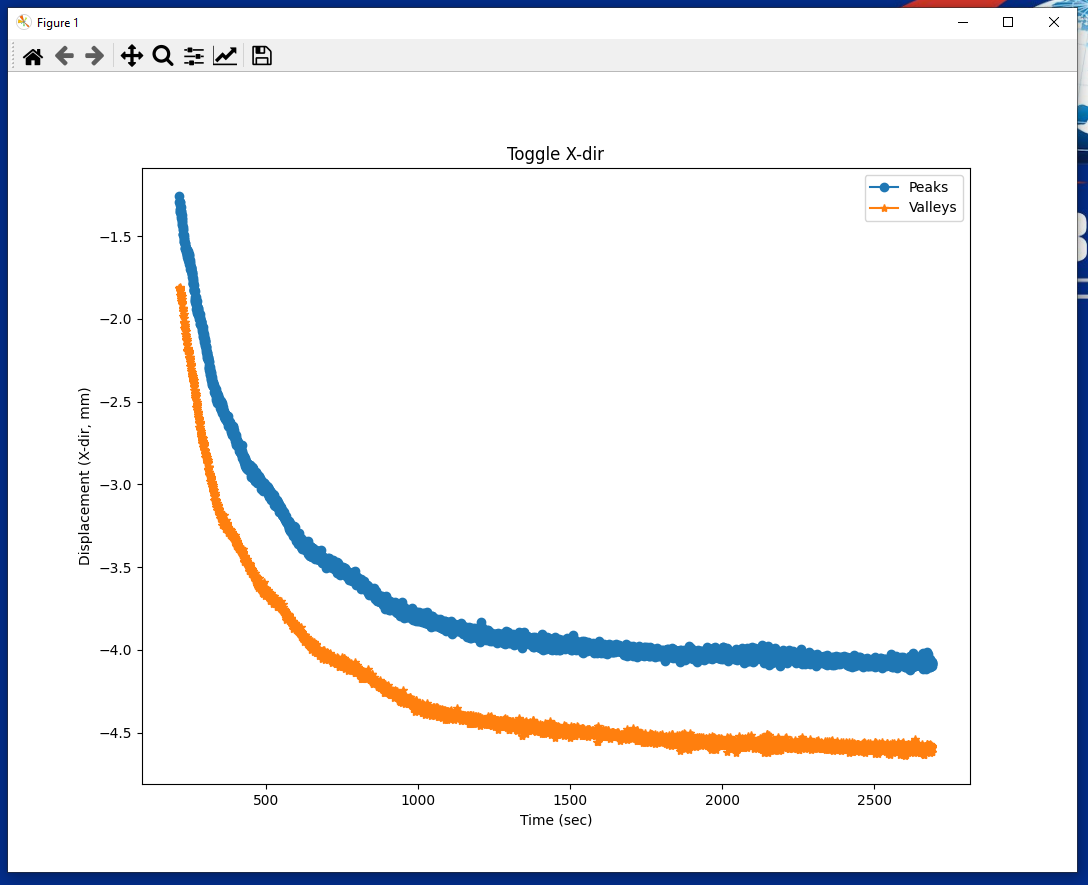


Auto-populated Plots

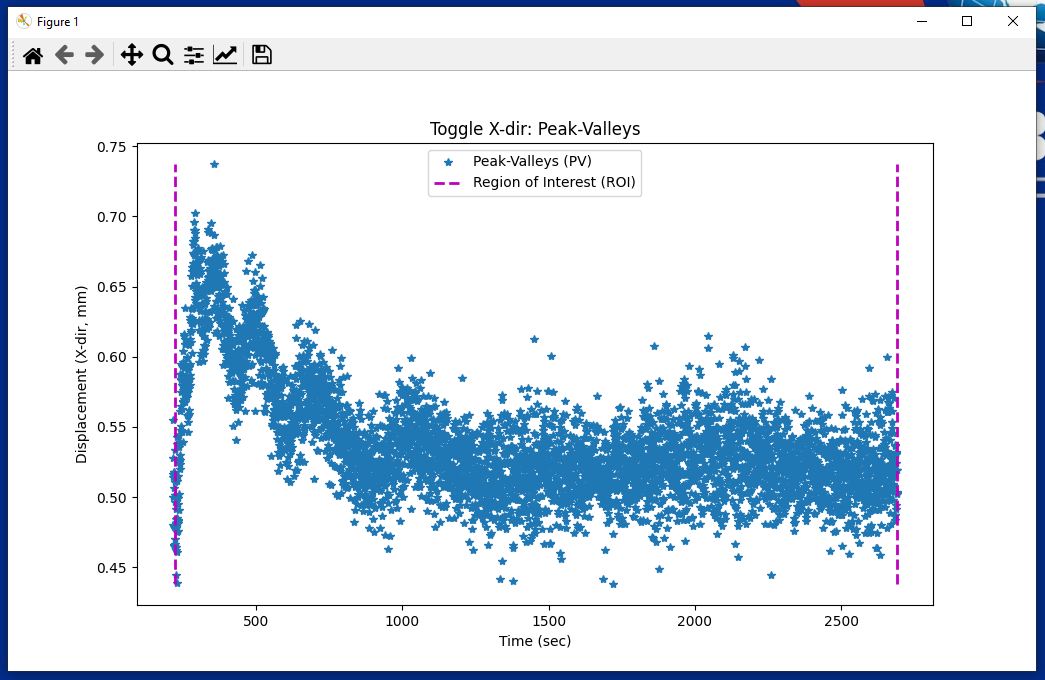
* These plots show the data as it is analyzed for each direction (x, y, z) and for 3D displacement
* In order to continue, close each window after it pops up
* All figures will automatically be saved in the same folder as the .xlsx file used



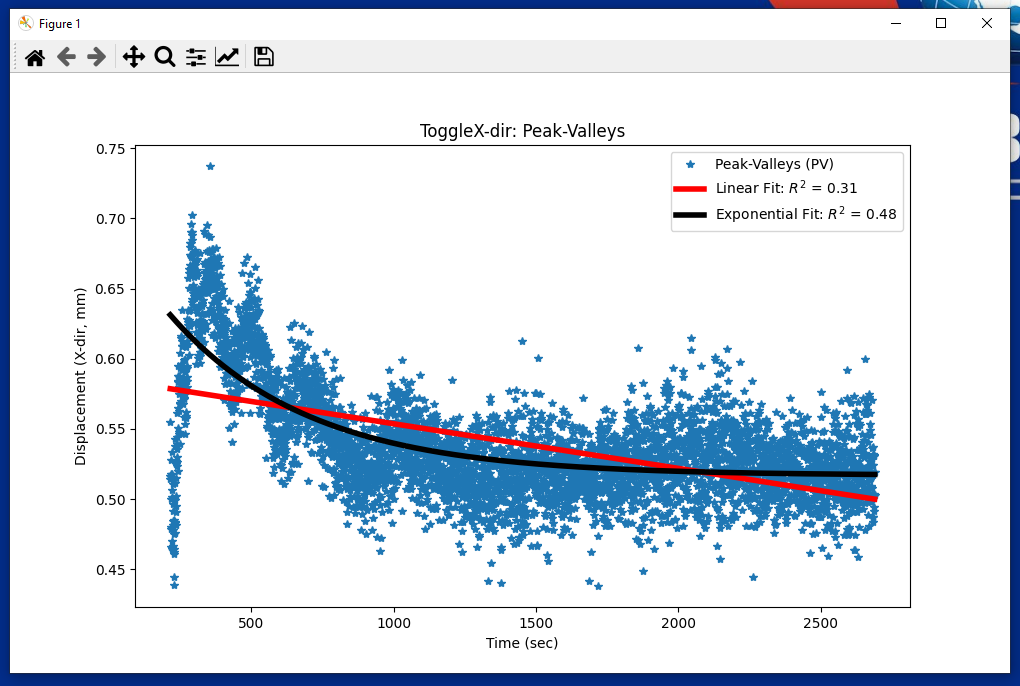
Plot showing time-dependent displacement data, peaks, valleys, and ROI. This figure is very useful to determine if smoothing was necessary and adequate.



Plot showing the time-dependent peaks and valleys extracted from the displacement data.



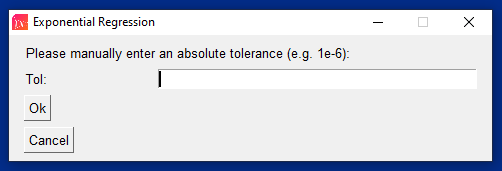
Plot showing the difference between adjacent peaks and valleys for each cycle (PV).



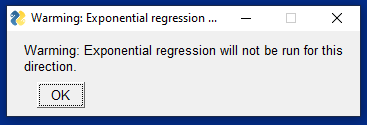
Plot showing the trends of PV over time. The application automatically uses regression to fit linear and decaying exponential models to the PV data. The R-squared values are shown to indicant model accuracy.

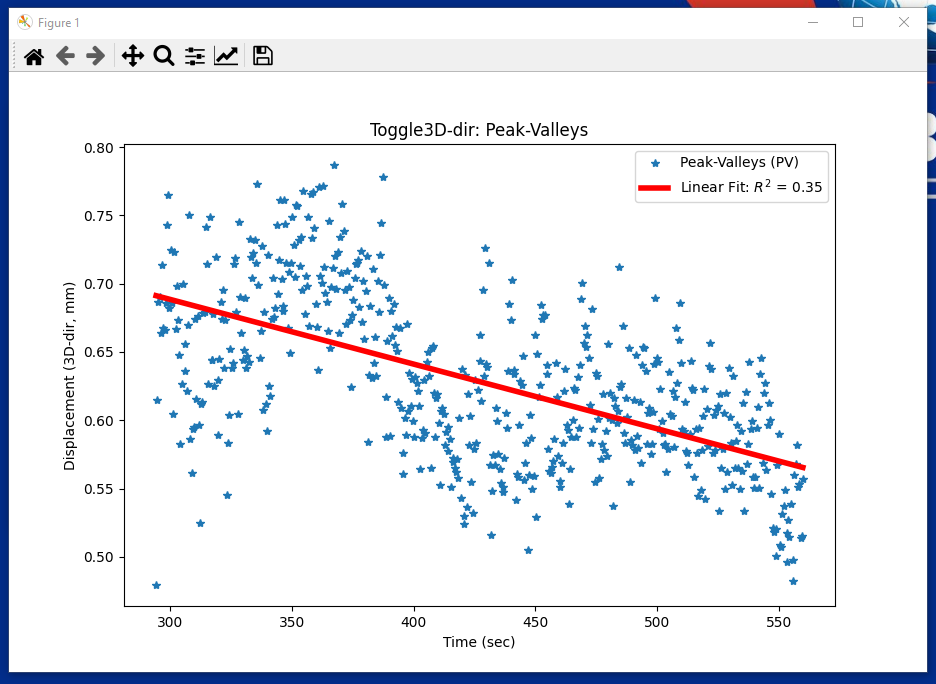
Potential Errors with Exponential Regression

* In certain cases, the regression algorithm might run into issues in trying to find converging coefficients that describe the data as a decaying exponential
* This occurs because the regression algorithm requires a set absolute tolerance to optimize the coefficients of the fitted exponential model
* If the coefficients simply cannot converge to meet this tolerance, then you will be prompted to change this tolerance until convergence can be achieved.
  + Note: It is possible that no tolerance will allow for convergence.

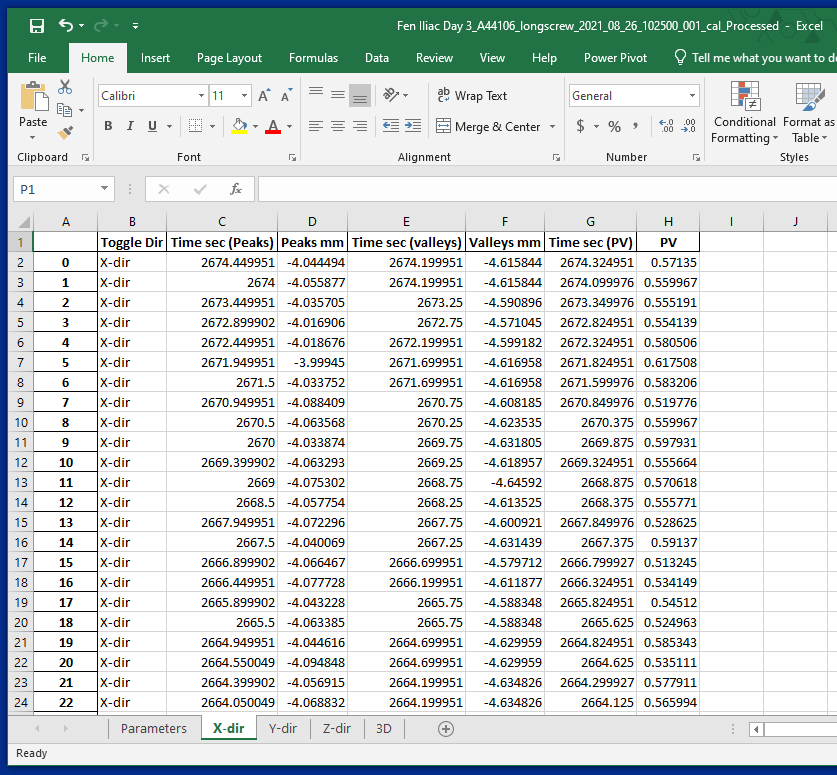
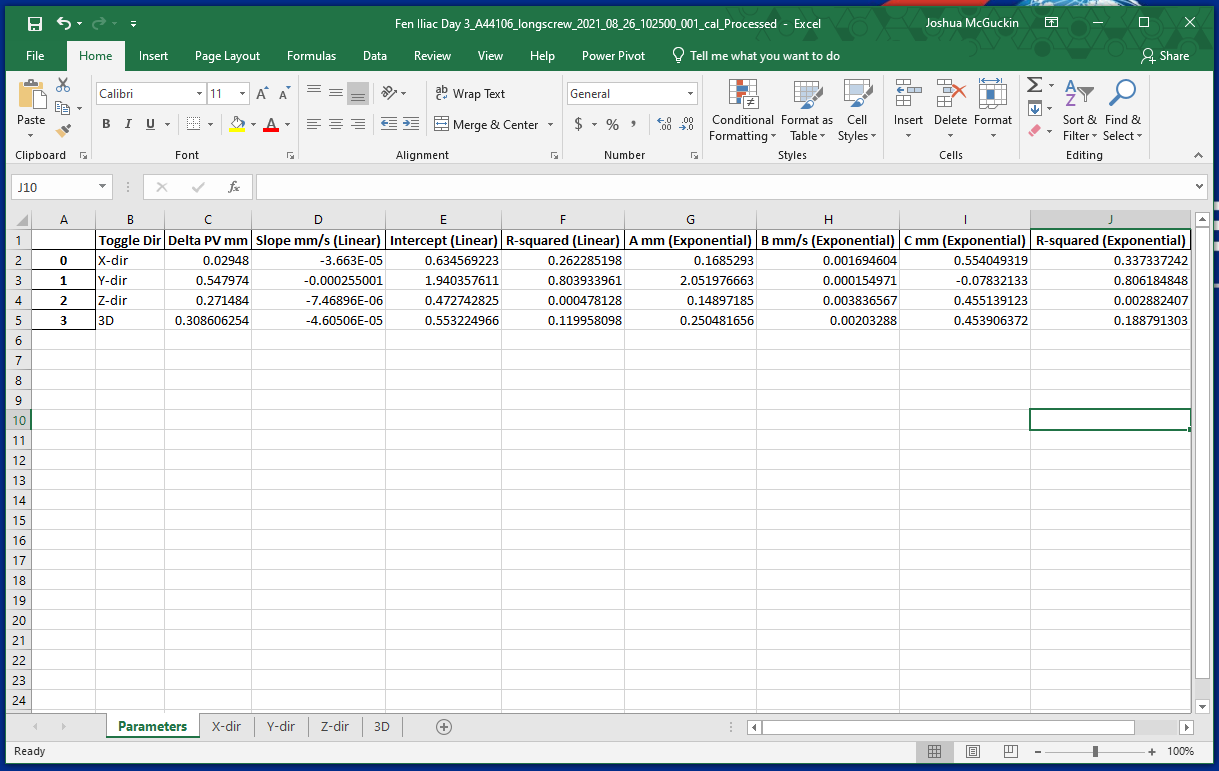
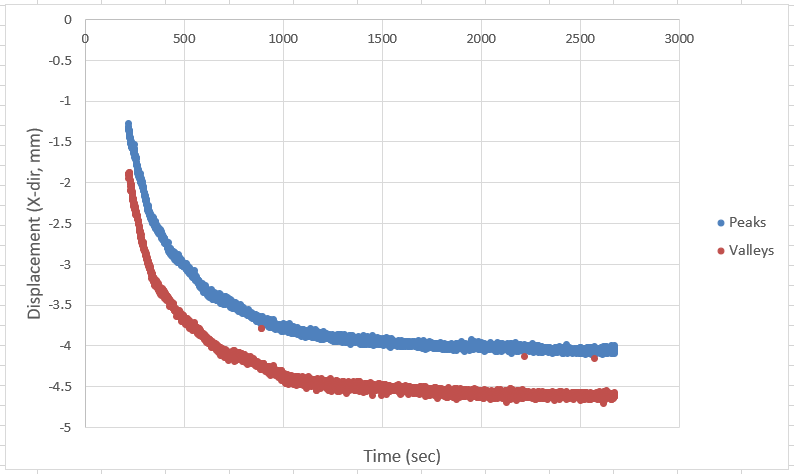


* If you press ‘Cancel’ or close the ‘Exponential Regression’ window, the following warning will appear to inform you that exponential regression will not be run for this direction.
  + Note: In the output .xlsx file, this value will be ‘NaN’ as it is nonconvergent.

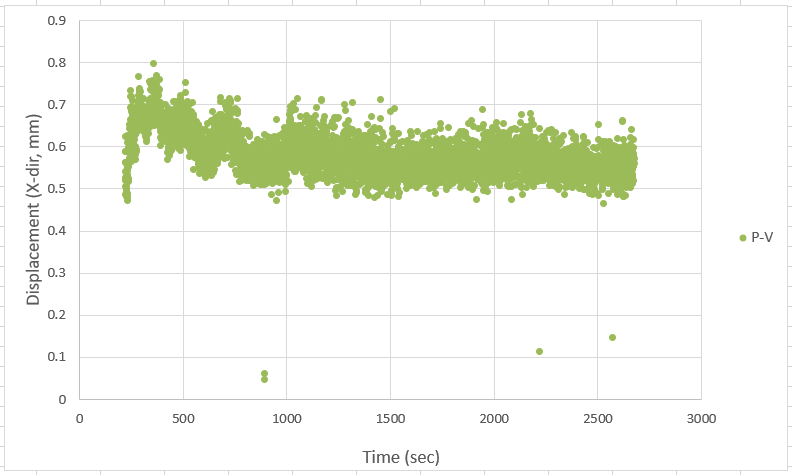




Regression plot only showing linear trend for ΔPV as exponential regression was nonconvergent.

Output of Processed Results 

* An excel workbook (.xlsx) will automatically be saved in the same folder as the .xlsx file used
* It will have key time-independent parameters (ΔPV, linear slope, y-int, R-squared, exponential time constant, etc.) as well as the time-dependent data (peaks, valleys, PV) for all 3 directions (x, y, z) and 3D displacement

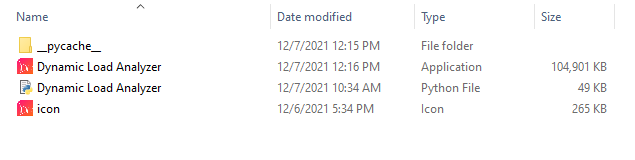


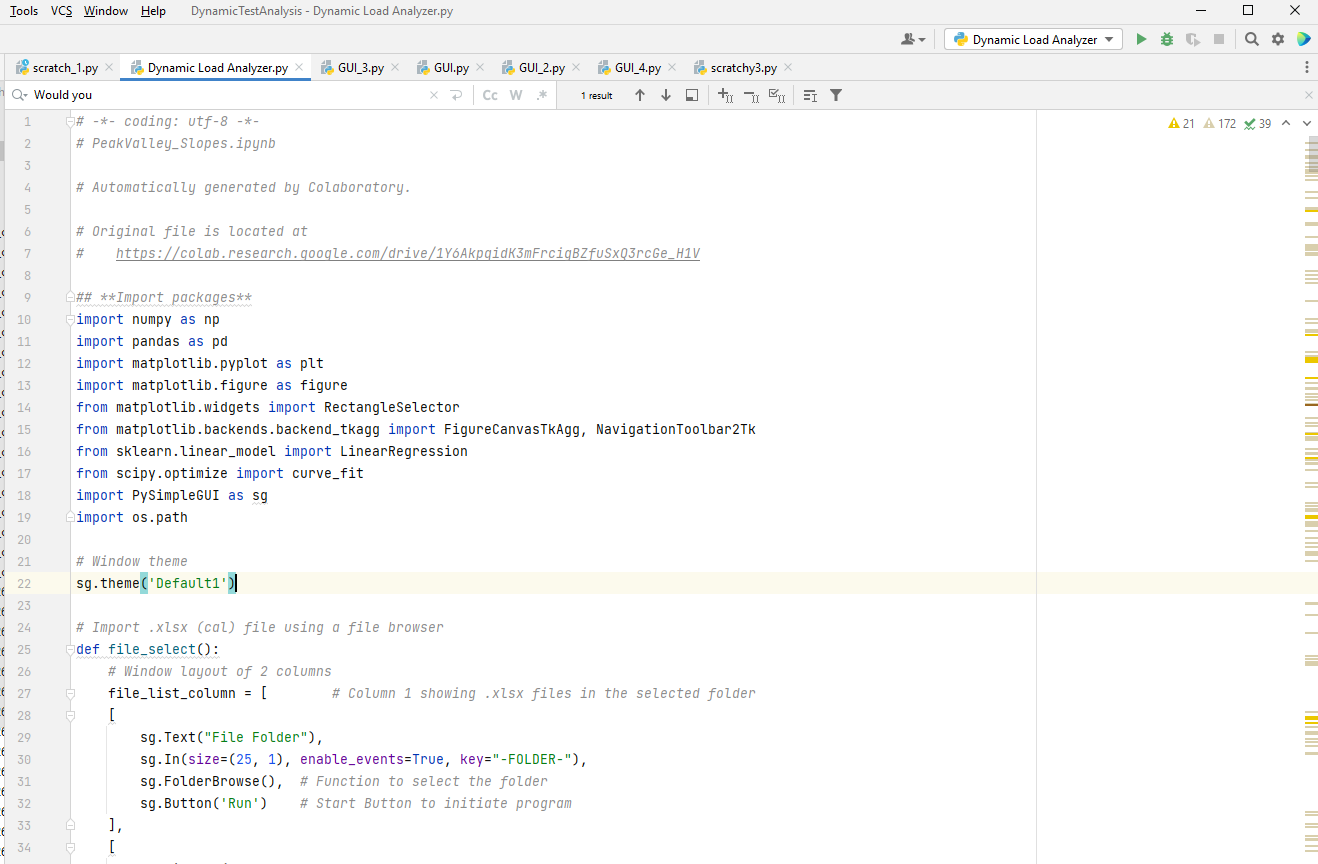
* The time-dependent data can be used to recreate the plots output by the application or to identify outliers (excessively high/low peaks/valleys) that were omitted by the application

# To Modify the App (Dynamic Load Analyzer):

Install Python, IDE, and Open Source Code

* In the event you would like to edit the application, you will first need to install Python 3.7 (64-bit)
  + [Download this version of python](https://www.python.org/downloads/release/python-370/)
    - Note: To install this, you might need admin access. In this case, you can typically email IT and they can do it remotely.
  + Run the setup wizard (e.g. python-3.7.8-amd64) and follow the installation steps
    - Note: Keep track of where you save the Python .exe file as you will need this to set up your IDE to edit and run the code.
* You will also need to download and install a compiler and IDE that’s compatible with Python
  + Examples of Popular Open Source IDEs:
    - [Spyder](https://www.spyder-ide.org/)
    - [PyCharm](https://www.jetbrains.com/pycharm/) (This is the IDE used by the developer – Joshua McGuckin)
    - [Thonny](https://thonny.org/)
    - [Atom](https://atom.io/)
    - [Microsoft Visual Studio](https://visualstudio.microsoft.com/)
* In your chosen Python IDE, you need to first create a new project
  + This creates a new folder to save your new versions of the application code
* Next, specify your Python interpreter for the virtual environment
  + Choose the pathname for the Python .exe file on your computer as you will be using Python 3.7
    - Ex. C:\Users\jmcguckin\AppData\Local\Programs\Python\Python37\python.exe
* Once setup, install the following Python packages
  + - numpy
    - pandas
    - matplotlib
    - sklearn
    - scipy
    - PySimpleGUI
    - os
* Note: Make sure that each of these packages successfully installs. The code will not be able to run otherwise.
* Return to where you have the ‘Dynamic Load Analyzer’ Folder on your computer (e.g. C:\Program Files) and open the Python file (.py) in your IDE

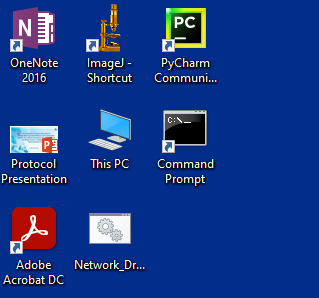




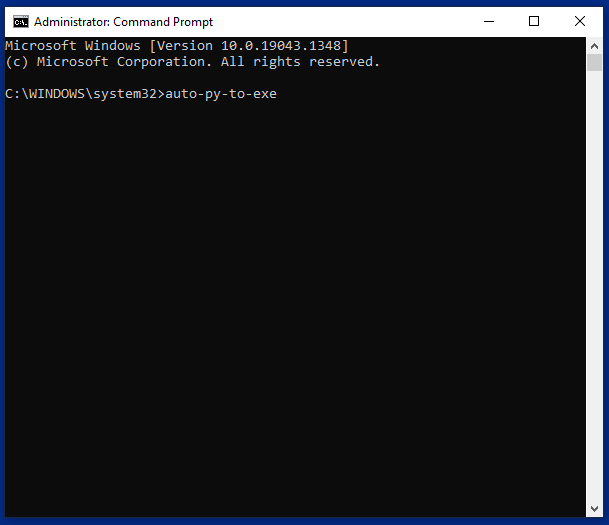
* From here you can edit your code and save it in the project folder.

Exporting Python Script (.py) as an App (.exe)

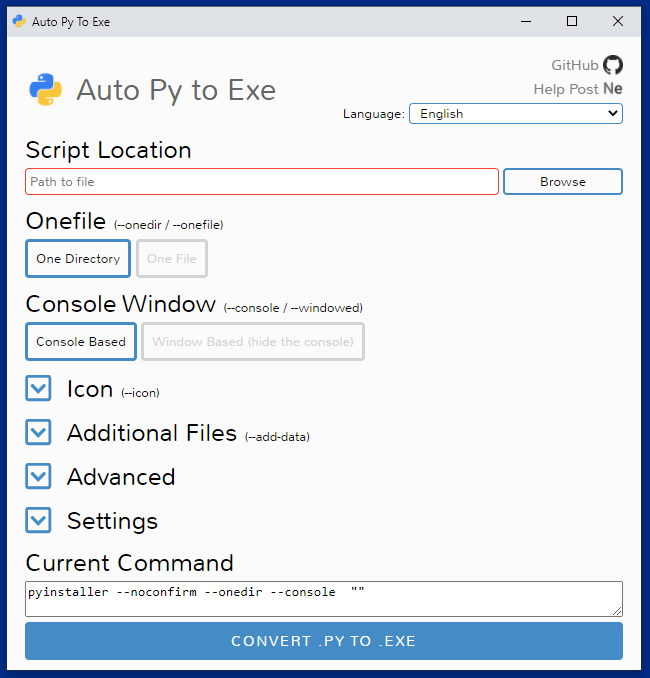
* After making your edits, move the updated Python code (.py) to the folder of the existing application and [download py2exe](https://pypi.org/project/auto-py-to-exe/)
* Follow the installation steps from the setup wizard
* Open a **command prompt as an administrator** (Should be in C: drive)
  + Note: It is crucial that you do this as an administrator because otherwise, you will not be able to run py2exe and turn the code into a new version of the application.



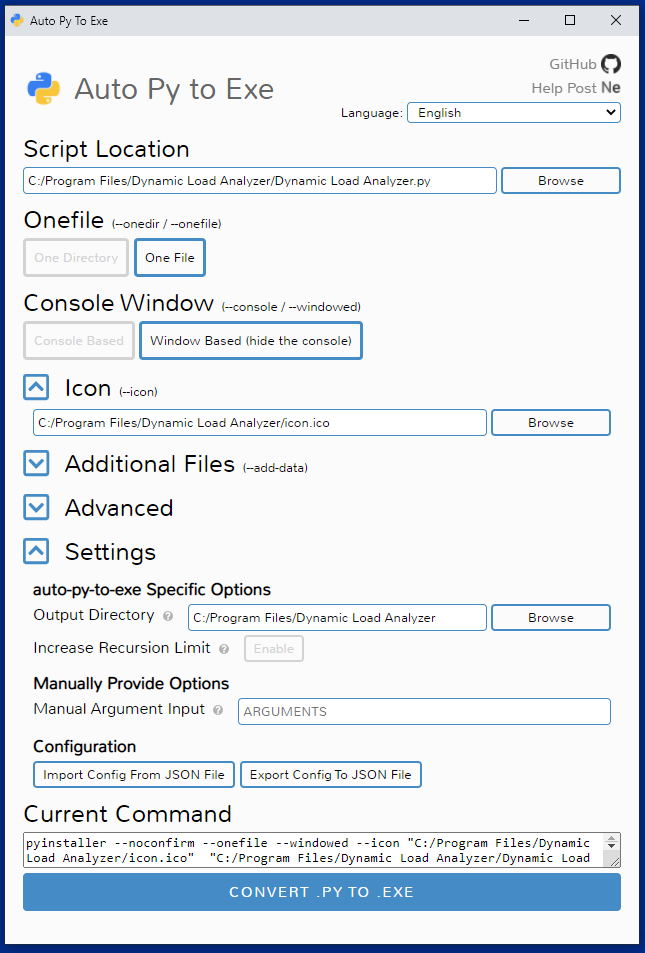
* In the command prompt, type ‘auto-py-to-exe’ and press enter



* The py2exe window GUI will pop up



* In the ‘Script Location’ section, press ‘Browse’ and select the Python file (.py) that has your edited code
* In the ‘Onefile’ section, select ‘One File’
* In the ‘Console Window’ section, select ‘Window Based (hide the console)’
* In the ‘Icon’ section, press ‘Browse’ and select the icon file (.ico) in the folder that had the original application’s Python code
* In the ‘Settings’ section, set the same folder as the ‘Output Directory’ and press ‘enable’ next to ‘Increase Recursion Limit’
* Then press ‘CONVERT .PY TO .EXE’



* Once the conversion is complete, open the folder with the new version of the application and double click on it to make sure it works
* If you are satisfied with your changes, you can create a new shortcut on your desktop for this new version of the code
* Finally, create a new folder titled ‘Dynamic Load Analyzer v.# - Date made’ in ‘R:\Research Operations\Dynamic Load Analyzer’
* Save the updated the code (.py) and application (.exe) as well as the icon file (.ico) in the new folder