# Data preparation

**Normalization**: Normalize x and y axis using matlab code line 26-47 in LetterPlot.m.

1. Normalize date label in x axis into numbers with same interval
2. Eliminate repeated data points.

**Note:** This procedure should be applied to all the data sets before further processes.

# Letter shape recognition by segments(W, V)

**Algorithm (find median points within certain interval of total data set as abrupt points to segment data. Median point will not affect by outliers) Line 50-89 in LetterPlot.m:**

1. Set up threshold which is certain subtle intervals to look for median points to segment. In our case for now, generally, we set it up to ⅙ of total data set.
2. Loop through all prepared data points, eliminate NaN data points.
3. Find the median point in each subtle interval.
4. If needed, output each segmentation info. Now, we output json file include start and end points of each segment, and linear function formula of each linear segment, average residual error and number of total data points after data preparation.

Note: LetterPlot.m also allow user to slide window to get small area of curve fitting. Enter start value and end value when calling the main function.

**Code analysis:**

**Main function** : LetterPlot.m

Contents: 1. draw scatter plot using date as x axis and value of chemisty as y axis.

2. Normalization axises and eliminate redundant data points.

3. Remove NaN

4. Find median points in certain interval by looping through data points. Threshold = total data points/6.

5. Allow user to slide window to get small area of curve fitting. Enter start value and end value when calling the main function.

**Call script:** call.m

Call LetterPlot function, enter 4 parameters:

Chemistry column number in row data, row data name, start value, end value.

Note: If user wants the whole window of data, enter 0 as start value, 10000 or other roughly large number as end value.

Raw data uses matlab.mat format of data. Load the data into matlab and call the function would work.

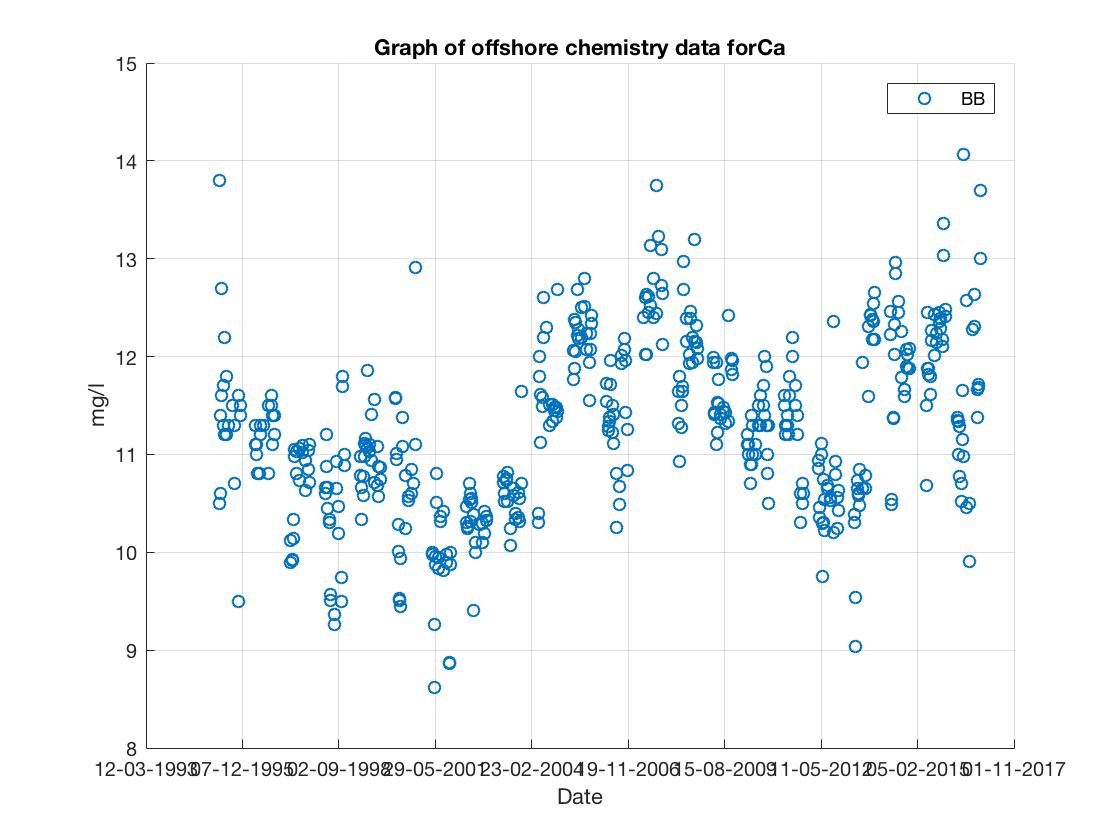
**Typical example graphs demonstration:**

For better understanding, line 72-90 in LetterPlot.m plot graph contains only median points we found, and then plot linear fit of each segments and connect together in another graph. Graphs are shown below.

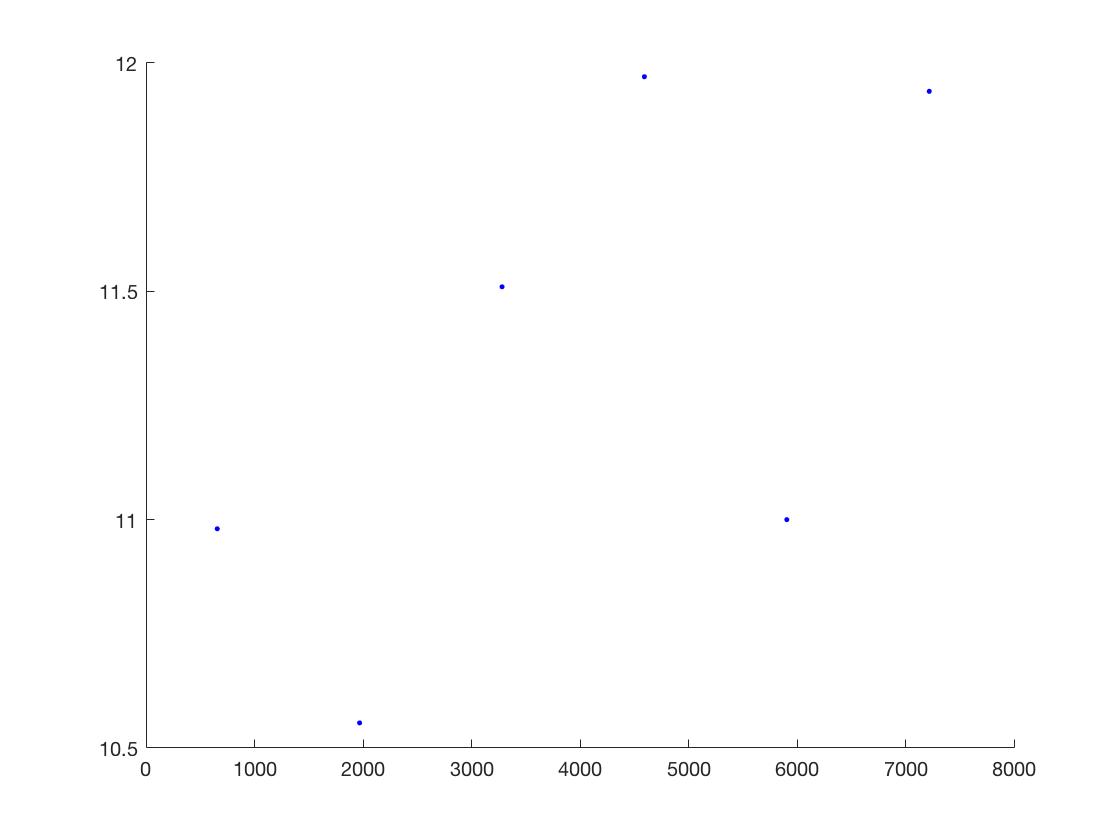
**Typical examples:**

Ca at BB:

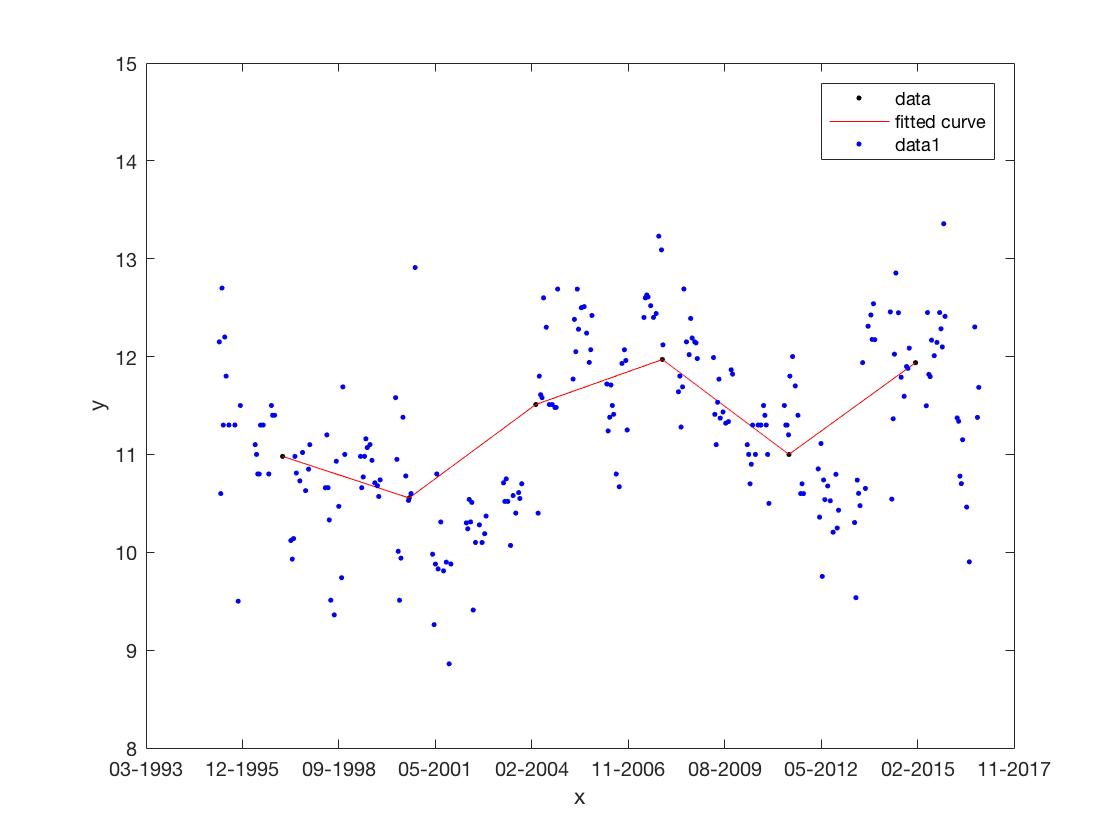
1. Original scatter plot before data preparation



2. Median points

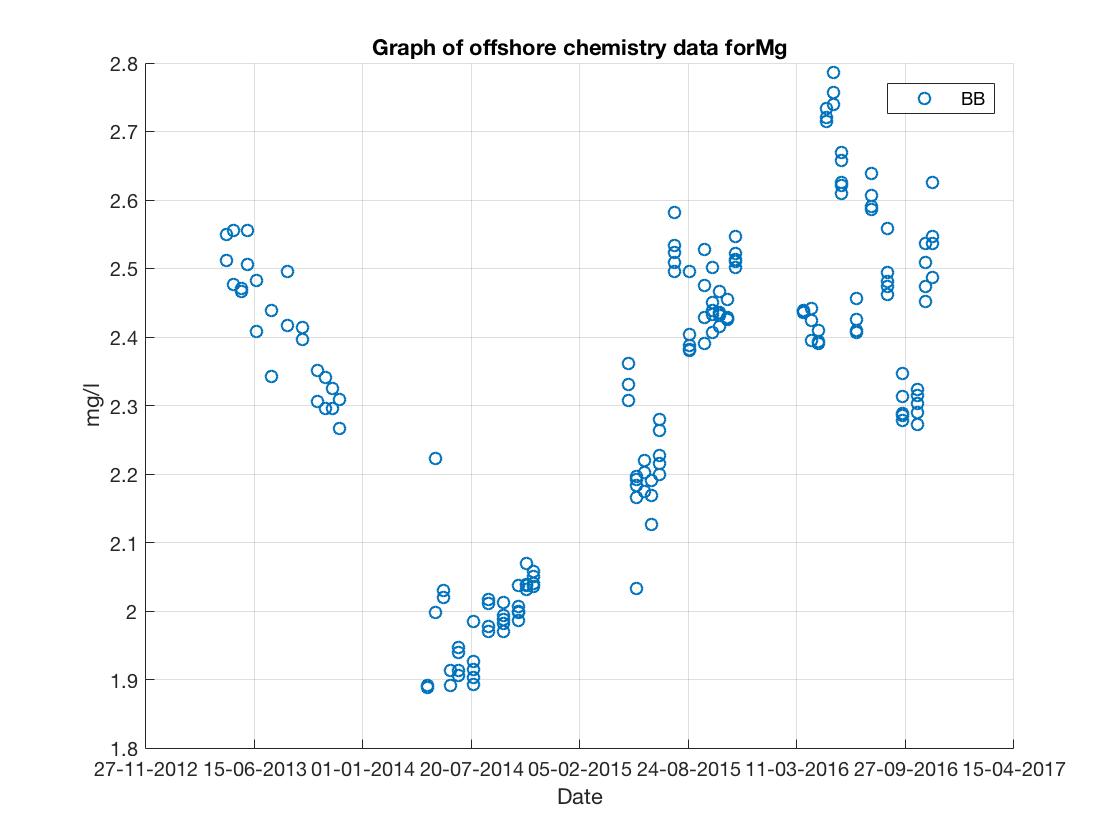


3. Curve fit after connect all the median points. We can see a rough W shape pattern

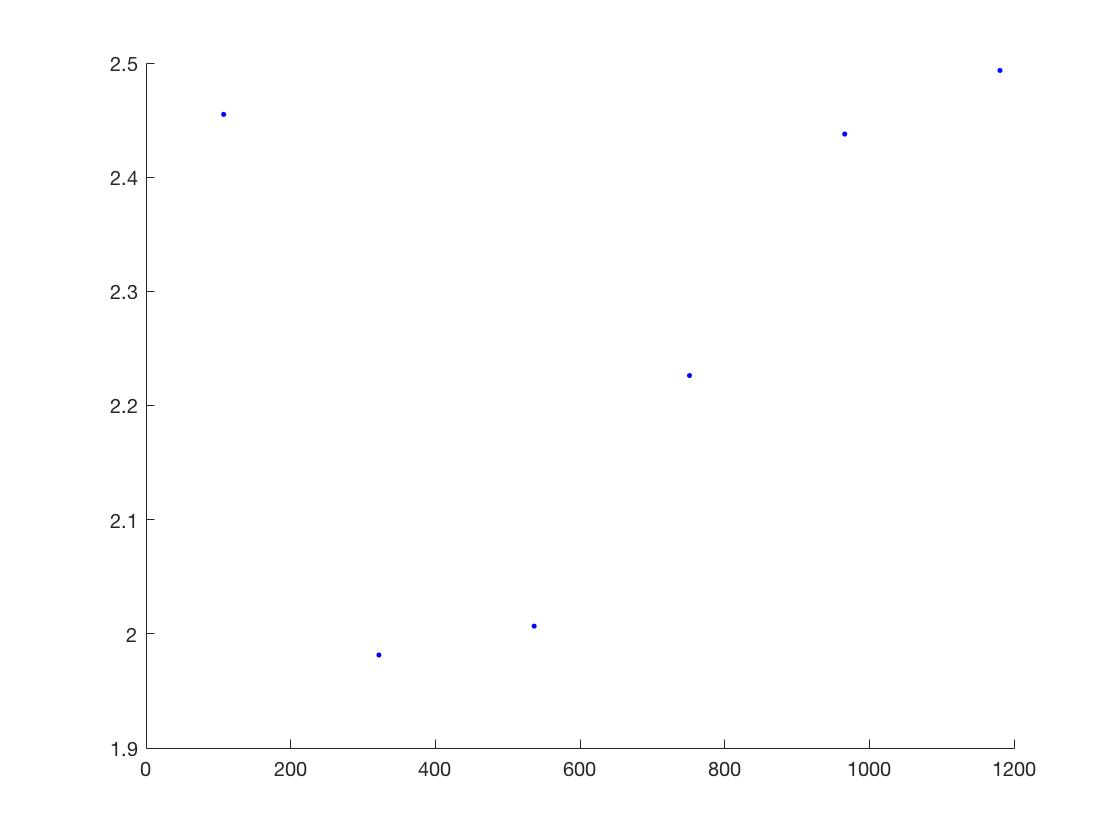


Mg at AN:

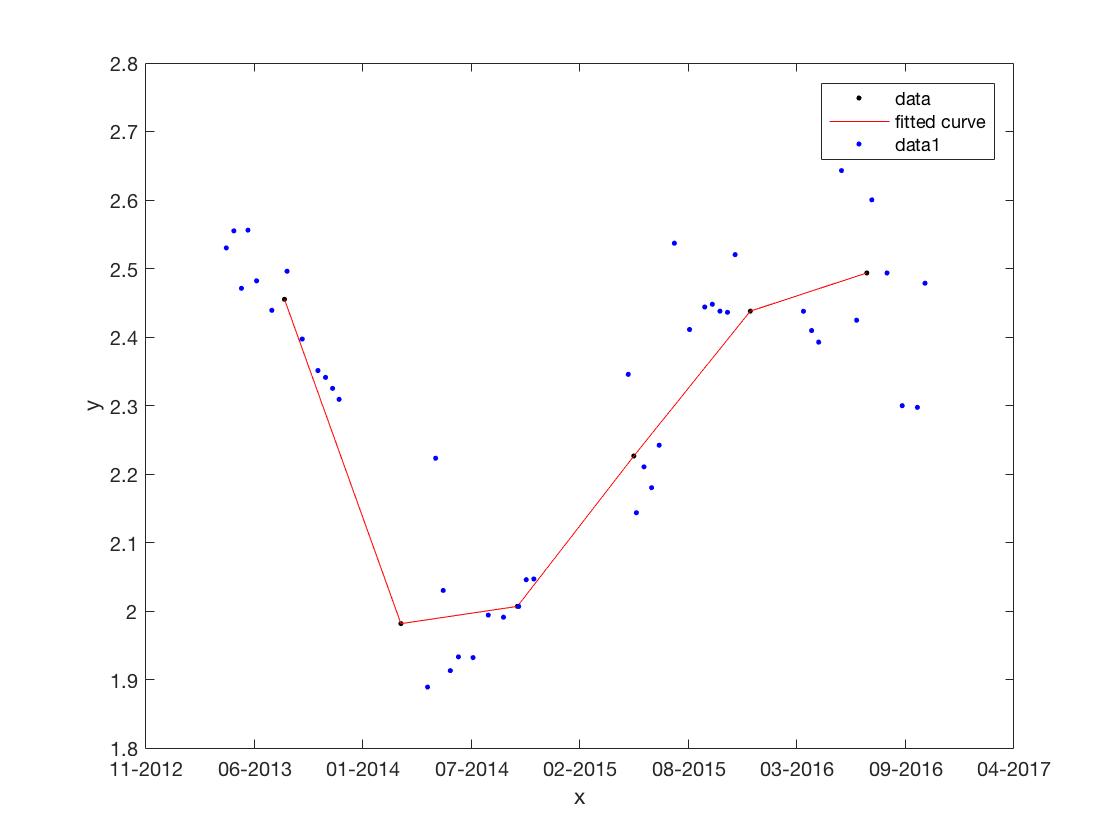
1. Original scatter plot before data preparation



2. Median points

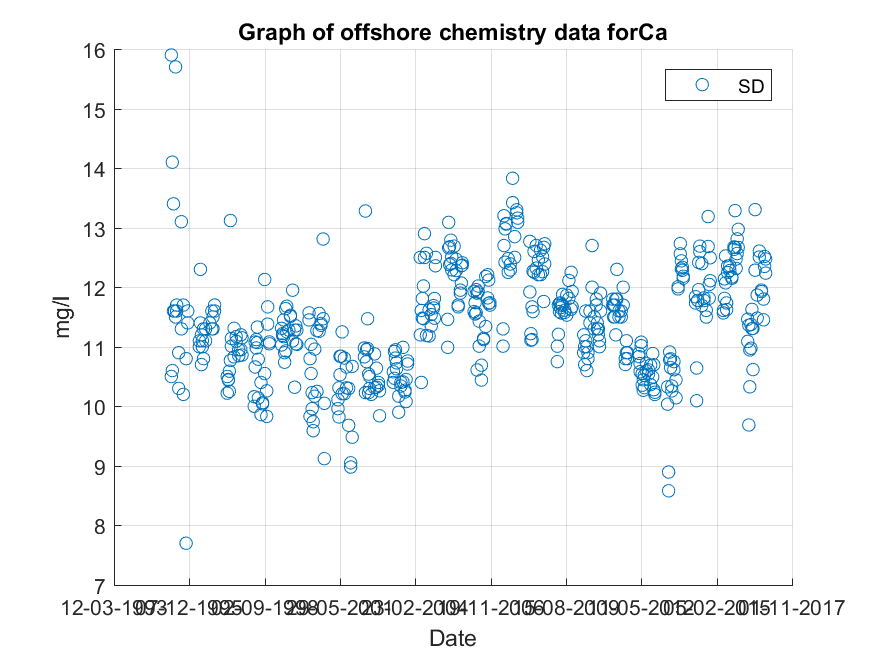


3. Curve fit after connect all the median points. We can see a rough V shape pattern



# Findchangepts (Matlab package - Not preferred but might help if needed)

**example:**



Method: main function (findchangepts)

Info: <https://www.mathworks.com/help/signal/ref/findchangepts.html>

Steps:

1. Data preparation as mentioned above.
2. Remove outliers if necessary (Depends on the graph you pick)
3. findchangepts(exclude,'Statistic','linear','MaxNumChanges',max\_num\_change);

Note: findchangepts is the major function in the section. Five parameters are (i) data after prepared. (ii) Mode of this function. We just need ‘Statistic’ here. (iii) Model name. Different models find major abrupt points based on different things. Models can be found in <https://www.mathworks.com/help/signal/ref/findchangepts.html>. (iiii) reference to find abrupt points. (V) number of change points you want to find.

In our case, we just need to change number of abrupt points.

1. If needed, output each segmentation info. Now, we output json file include start and end points of each segment, and linear function formula of each linear segment.

“Linear” model uses mean and slope to find major abrupt points by entering how many change points we want.

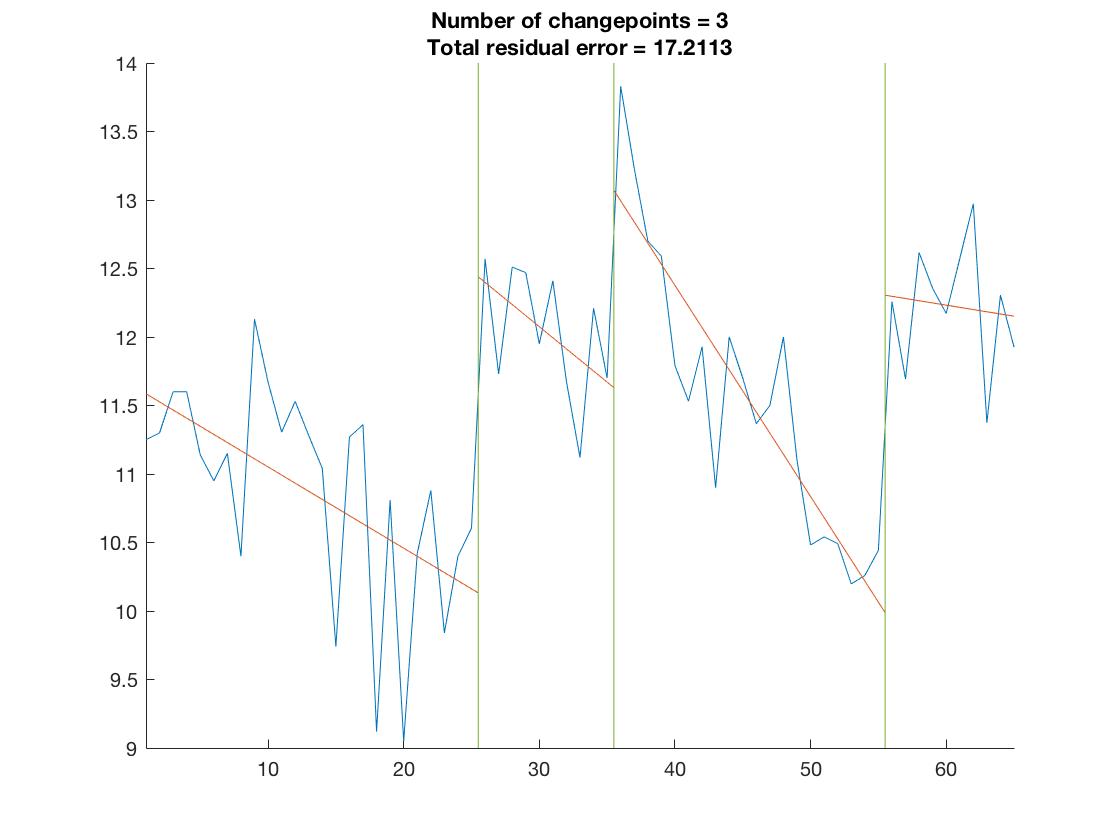
Note: remove NaN and outlier would make the segmentation works better.

**Problem:** 1. Complex graph with lots of data points can’t use this method.

2. By the mean and slope segmentation, hard to determine accurate letter shape.

3. Not able to automate optimal number of abrupts points. The more you pick, the better the residual error.

4. The resulting graph will become the following. Hard to control exact location of change points.



**Code analysis:**

**Main function** : Plot.m

Contents: 1. draw scatter plot using date as x axis and value of chemisty as y axis.

2. Normalization axises to avg data points and eliminate redundant data points.

3. Remove NaN and outliers

4. Using findchangepts to output each segmentation x and y value, linear function approximation for each segmentation and output as json file.

Note: The corresponding x and y value is not the graph after findchangepts curve fit. The values correspond to the x axis after normalization. plot(tu,tv,'.b'); Allows users to see the graph after normalization.

5. Allow user to slide window to get small area of curve fitting. Enter start value and end value when calling the main function.

**Call script:** call.m

Call Plot function, enter 4 parameters:

Chemistry column number in row data, row data name, start value, end value.

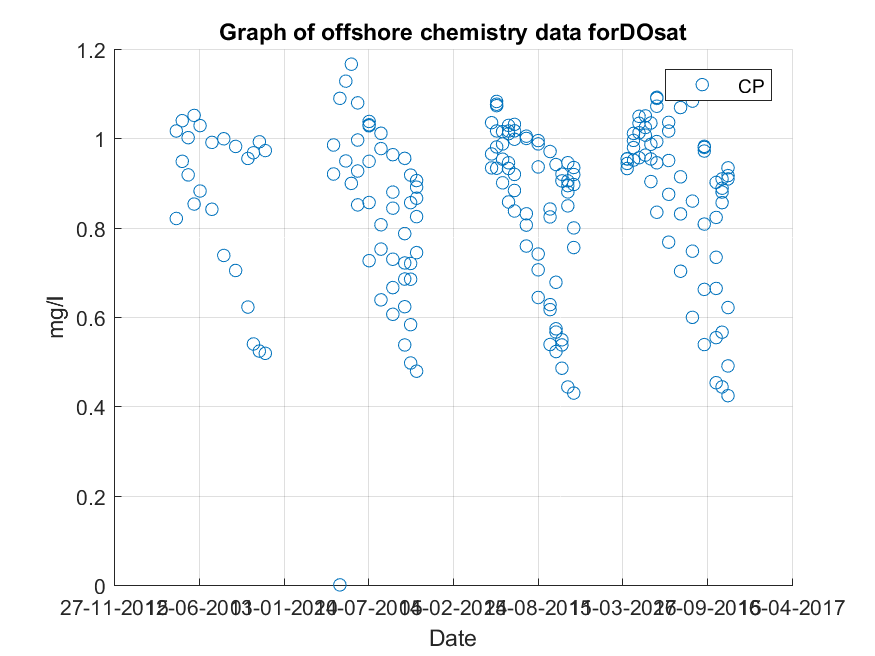
Note: If user wants the whole window of data, enter 0 as start value, 10000 or other roughly large number as end value.

Raw data uses matlab.mat format of data. Load the data into matlab and call the function would work.

# CFTOOL usage demonstration:

# (Matlab package used to do different sorts of function curve fit.)

Example:



After normalization, two methods can be used to better fit this type of graph to make it easy to describe.

Method 1: Using interpretation. Open CFTOOL in matlab, load sorted data inside and choose interpolation-shape preserving (or other mode for interpolation)

CFTOOL provides goodness of fit on the left side. For interpolation, the goodness of fit is perfect.

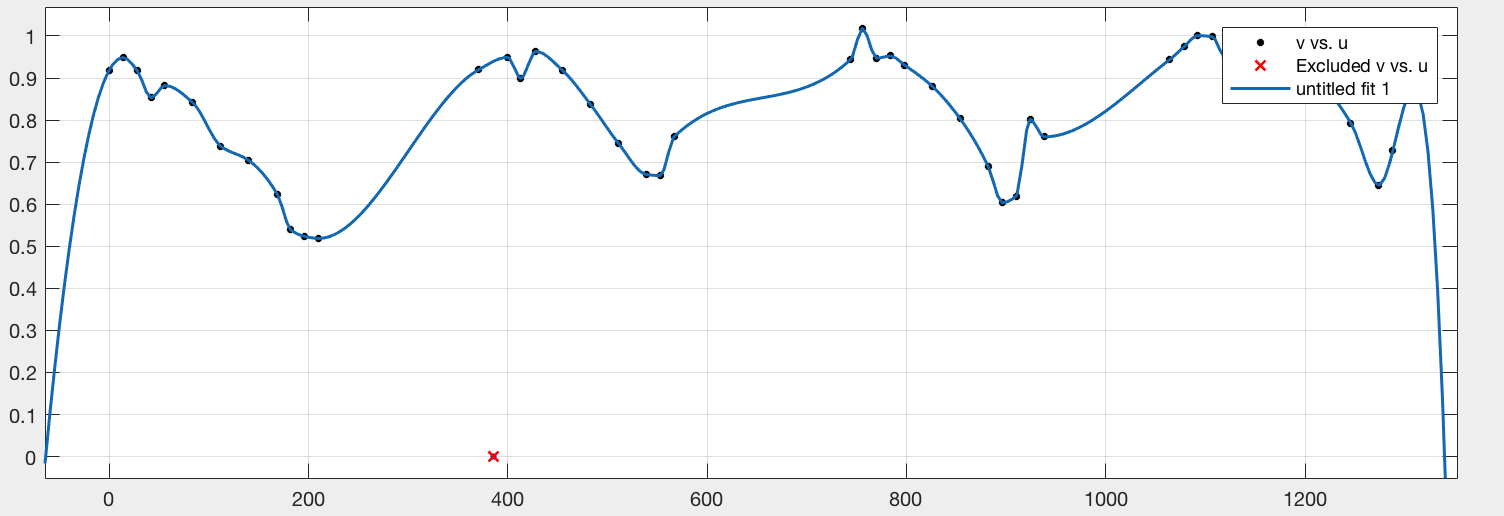
NOTE: Using interpolation requires x value to be unique. Could use exclude outliers to smooth curve

(In PLOT.m, the normalization and avg parts of code gives unique x value and eliminate redundant data points.)

Interpolation-shape preserving

In matlab code: ’ PCHIP’ model curve fit

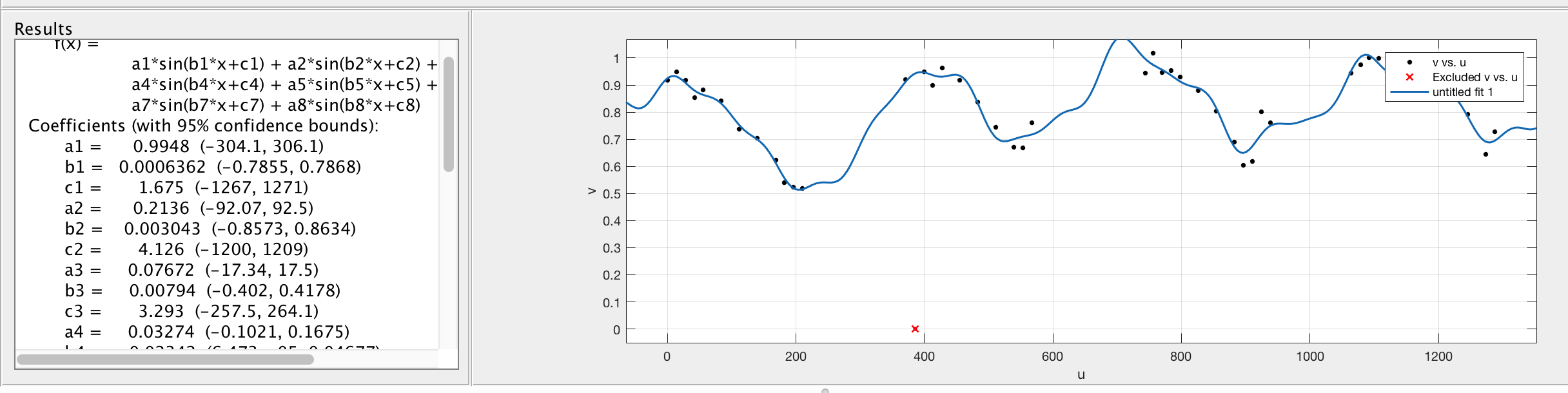
When you select 'shape-preserving interpolant' for fitting a curve, MATLAB uses a piecewise cubic Hermite interpolation (PCHIP) for fitting the data. Therefore, in the resulting fit, each pair of consecutive points is connected by a different cubic polynomial described with 4 coefficients.



Method 2: Open CFTOOL in matlab, load sorted data inside and choose cos or sin function, choose your own function to fit a roughly small residual error curve fit.

On the results of CFTOOL, it provides function data to use if needed. (It is not preferred)

Both method 1 and 2 do not generate output other than visualizations



5. Find the major peaks:

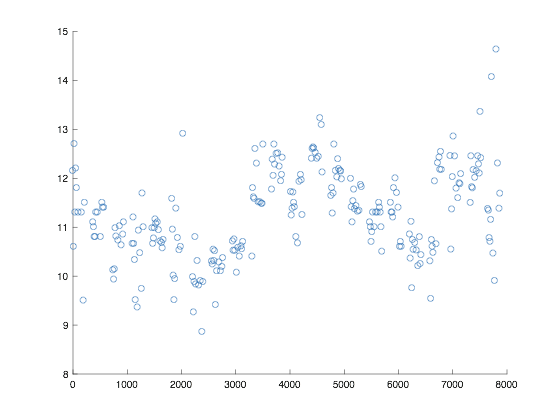
**What is it?**

The function **Peaks\_Finding(kt,lakedata\_desend)** is a function that takes in the **chemistry type** and **lake chemistry data table** and output the diagrams for original dot diagram, major peaks(both down and up) and store the *maximum chemistry data and date, minimum chemistry data and date, standard deviation, mean value for the chemistry data, rapid increase date, rapid decrease date, and major peaks* in a Json file.

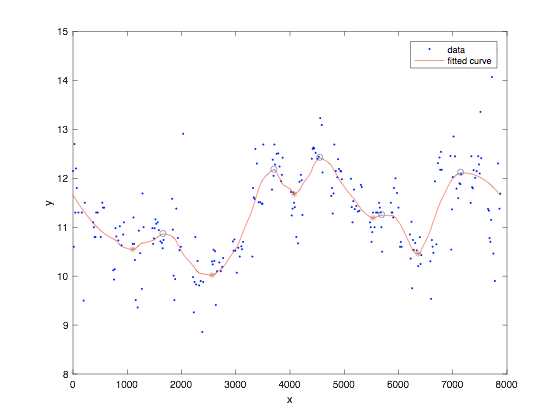
**Output example:**

By using Peaks\_Finding(19,lakedata) - 19 represents Ca.

**The original dot diagram:**



**The Major Peaks:**



“O” represents the up major peaks

“\*” represents the down major peaks

The Jason file:



**How to use it?**

Package needed: savejason.m -- this function is part of JSONLab toolbox (http://iso2mesh.sf.net/cgi-bin/index.cgi?jsonlab)

The way to use it: Peaks\_Finding(kt,lakedata\_desend)

For example: Peaks\_Finding(19,lakedata\_desend)

19 represents the Ca,

**The Chemistry name with index:**



A-Z which means 1-26

The lakedata\_desend can be found in the share folder which called “Chemistry\_Date.mat”

**The code analysis:**

1. Import the data from Chemistry\_data.mat
2. Put the date values and chemistry data from mat file to local variables.
3. Unique the data by average repeated points’ values and excluding the Invalid values.
4. Draw the original dot figure.
5. Smooth the data to let the major peaks show up. (Line-79): v2 = smooth(u,v,0.2,'loess'); the 0.2 is the degree of the smooth and loess is the method used on the smoothing, and there are other potential method talked in the rest of documentation.
6. Using the smoothed data to do curve fitting by the method: “pchip”. (Line-85): fitline = fit(u2,v2,'pchip'); the ‘pchip’ is the method used on the curve fitting, and there are other potential method talked in the rest of documentation.
7. Using findpeaks() to get all potential peaks. (Line-93): [ypk,idx,~,prm] = findpeaks(yfitted); ‘prm’ is the variable to show the degree of how significant the peak is.
8. Using the prm which indicates the degree of the significance for the peak from findpeaks() to sort out the most “major” peaks. (line 94): [~,i] = sort(prm,'descend'), i holds the major peaks’ index by descend.
9. Get the number of major peaks you want. (Line- 106): uppeak = v(idx(i(1:5))), if we need more or less peaks, change the number from 5 to the number we want.

10. Output all information in a Json file.

(Line-118):savejson(‘’, output variable, ‘filename’);

**The important variables in the function:**

x = lakedata\_desend{5457:5951,3};

data\_t = lakedata\_desend{5457:5951,kt}

X represents the date range for site bb.

Data\_t represents the Chemistry data for site bb.

If we want to use the function on other sites, the only thing need to do is changing the range(here is 5457:5951).

**Other sites:**

Site T: 83:1300

Site SD: 1300:1794

Site S: 1794:2678

Site R: 2678:3498

Site N: 3498:3606

Site F: 3606:4426

Site D: 4426:5249

**The important built-in functions used:**

yy = smooth(y,span,*method*) sets the span of *method* to span. For the loess and lowess methods, span is a percentage of the total number of data points, less than or equal to 1. For the moving average and Savitzky-Golay methods, span must be odd (an even span is automatically reduced by 1).

**Method:**

I have used “loess” with degree “0.2” in the function.

The following is the list of all potential method we can use in smooth function.

|  |  |
| --- | --- |
| 'moving' | Moving average (default). A lowpass filter with filter coefficients equal to the reciprocal of the span. |
| 'lowess' | Local regression using weighted linear least squares and a 1st degree polynomial model |
| 'loess' | Local regression using weighted linear least squares and a 2nd degree polynomial model |
| 'sgolay' | Savitzky-Golay filter. A generalized moving average with filter coefficients determined by an unweighted linear least-squares regression and a polynomial model of specified degree (default is 2). The method can accept nonuniform predictor data. |
| 'rlowess' | A robust version of 'lowess' that assigns lower weight to outliers in the regression. The method assigns zero weight to data outside six mean absolute deviations. |
| 'rloess' | A robust version of 'loess' that assigns lower weight to outliers in the regression. The method assigns zero weight to data outside six mean absolute deviations. |

**Curve fitting method:**

fitobject = fit(x,y,fitType)

In the function I use phcip which is Piecewise Cubic Hermite Interpolating Polynomial.

The following is the list of the all potential fit types we can use.

Fit type:

|  |  |
| --- | --- |
| 'poly1' | Linear polynomial curve |
| 'poly11' | Linear polynomial surface |
| 'poly2' | Quadratic polynomial curve |
| 'linearinterp' | Piecewise linear interpolation |
| 'cubicinterp' | Piecewise cubic interpolation |
| 'smoothingspline' | Smoothing spline (curve) |
| 'lowess' | Local linear regression (surface) |

4. Rapid grow and down:

What is it?

Rapid grow and down function will output the rapid increase date with the chemistry data and rapid decrease date with the chemistry data

How to use it?

function [start,tend] = rapid\_grow(x,y)

X is the date values, and Y is the chemistry values.

The output is the index of the start point and end point.

Code analysis:

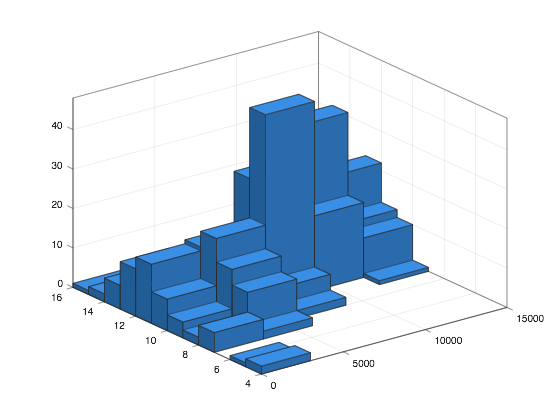
The function will use a loop to get the difference between any two chemistry data and find the biggest difference and record the points caused the difference.

5. Density

What is it:

A built-in method to draw bar plot for numeric data that group the data into 2-D bins.

Example:



From the figure we can easily find out the density relationship.

How to use it:

After using the unique method from the peaks\_finding function, input the unique x values(date) and y values(chemistry data) in histogram2(x,y) then the figure will be drawn out automatically.

6. Generate PLA Graph

Generates a Piecewise Linear Approximation Graph from the CSV of a single data variable from the Lake George dataset.

How to use:  
1. From Offshore Chemistry data 1980 – 2016, create a CSV using two columns:

* Date
* The column of whatever variable you wish to make a graph of (e.g. CA, NA, COND)

2. Clean the CSV such that the year portion of each date is the full year (e.g. 2016 instead of 16)

3. In generate\_PLA\_graph.m, change the variable ‘filename’ to the full path of the CSV created above.

4. Change plot variables at bottom of function as necessary (Axis titles, plot name, etc.).

5. Run function and generate PLA Graph.

Tasks:

1. Better Piecewise Linear Approximation (not w/ set segment lengths)

\* Item 1., 6., 5.1

2. Better shape matching to graph (hardest one?)

\* Item 2., 5.1

3. Automatically generate segment CSV and meta .json files

\* Item 6.

4. Using Density or major peak matching to describe graph

\* Item 5.5

Please write your name next to the task you’d like to take.