**EClab\_CV Class**

Latest version: 1.0

Latest update: 14/09/19

Latest update personal: Amir

Version 1- 09/11/19 Amir

Main changes - establishing the base, no changes yet.

Bugs -

Future work -

Comments -

**Documentation:**

Every EClab\_CV Include the following fields:

E – The potential samples vector

I – the current samples

ImaxOx – the maximum value of I oxidative.

ImaxRed – the maximum value of I reduction

EmaxOx – the potential sampled which resemble ImaxOx

EmaxRed - ImaxRed

peaks – a table with the result of get peaks function in the following parameters: peak I (peakI) | peak E (peakE) | peak width (peakW) | peak prominance (peakP)

N\_cycles – the amount of cycles

Speed – scan rate

Operators:

‘-’ - minus sign. If a and b are CV objects. C=a-b, C is a CV object . C.E will inherit a.E and C.I=a.I-b.I. this operation is only if the potential range is has less than 10^-3 change between the 2 CVs, in this case the data is interpolated according to the shorter signal. This operator can be used element-wise on EClab\_CV object matrix if a and b are matrixes with the same number of elements.

‘+’ - plus sign. If a and b are CV objects. C=a+b, C is a CV object . C.E will inherit a.E and C.I=a.I+b.I. this operation is only if the potential range is has less than 10^-3 change between the 2 CVs, in this case the data is interpolated according to the shorter signal. This operator can be used element-wise on EClab\_CV object matrix if a and b are matrixes with the same number of elements.

Functions:

**plot**

Syntex:

plot(a)

plot(a,name,value)

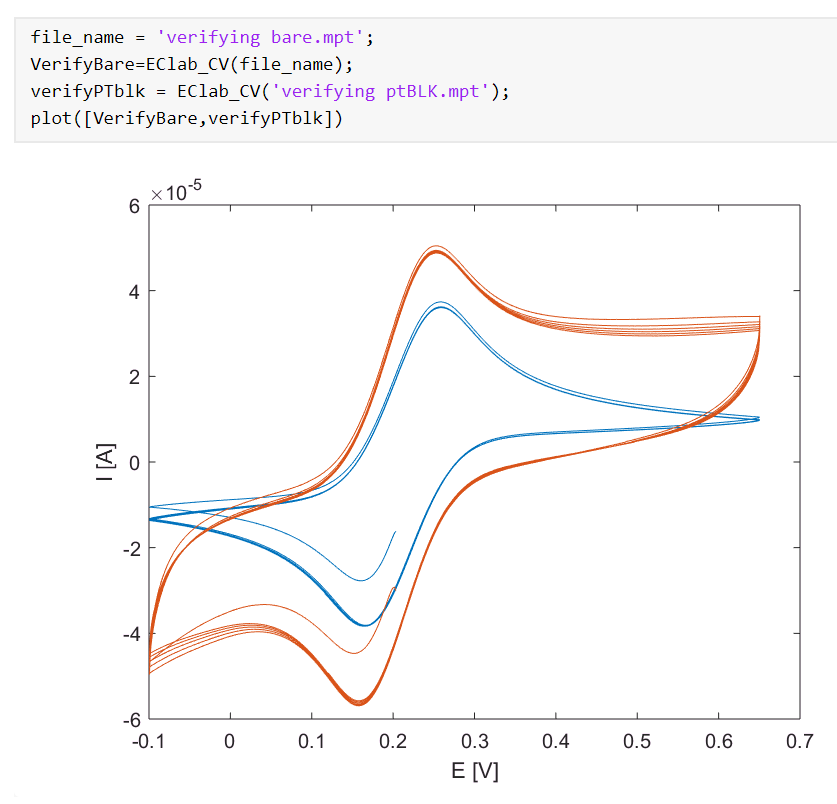
Description:

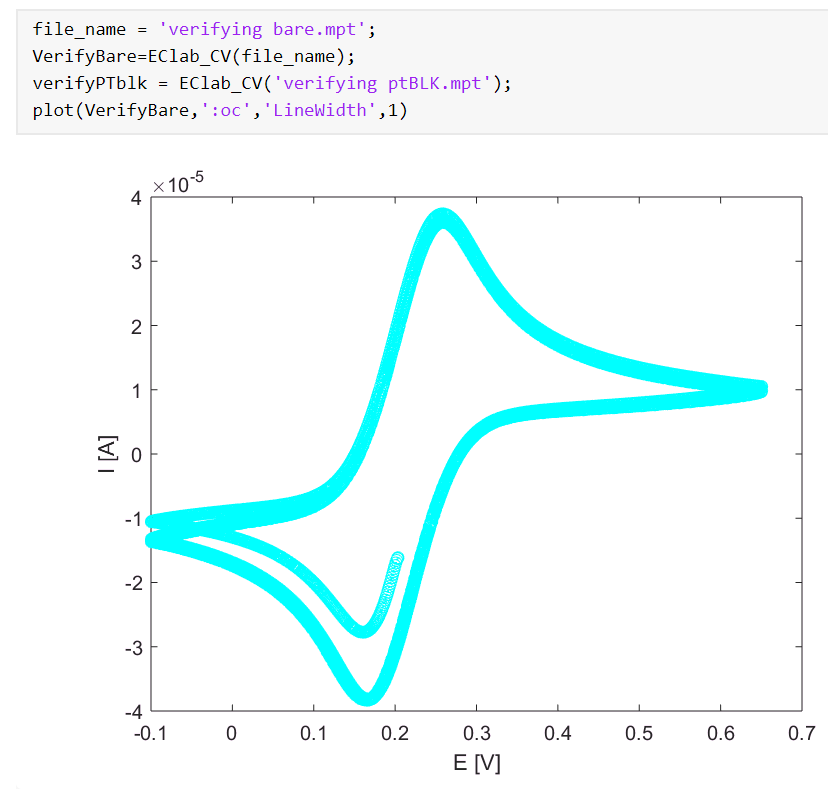
plot(a) creates a 2-D line plot of the data in a.I Vs the corresponding values in a.E.

* + - If a is DPV matrix plot(a) while create several lines as the number of elements in a

Plot(a,name,value) let the user define some line properties as motioned in the regular plot documentation.

Example:





**scatter**

Syntex:

scatter(a)

scatter(a,name,value)

Description:

scatter(a) creates a 2-D scatter plot of the data in a.I Vs the corresponding values in a.E.

* + - If a is DPV matrix scatter(a) while create several lines as the number of elements in a

scatter(a,name,value) let the user define some line properties as motioned in the regular scatter documentation.

**cycleSplit**

Syntex:

a=cycleSplit(b)

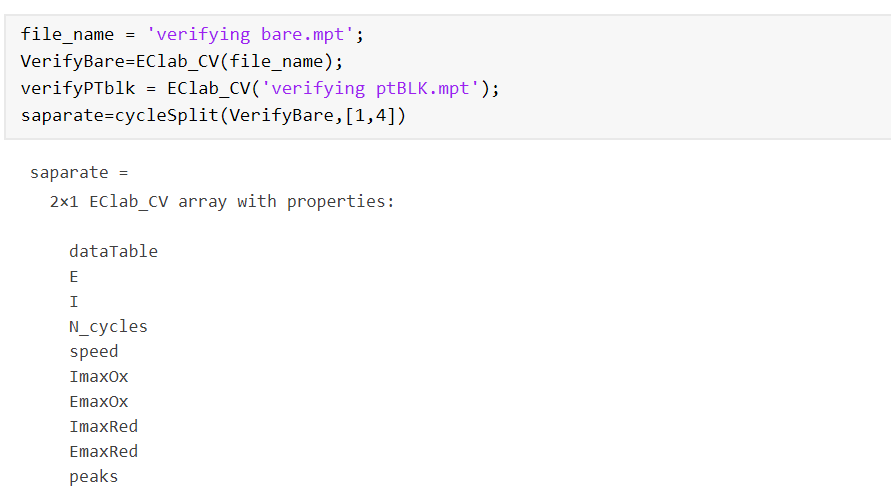
a=cycleSplit(b,[cycle numbers])

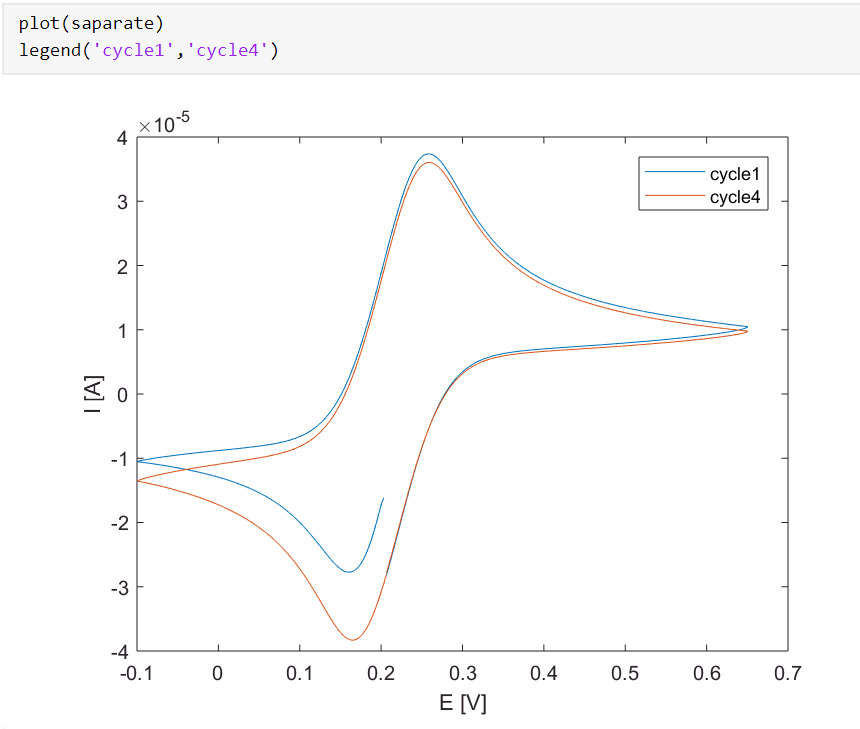
Description:

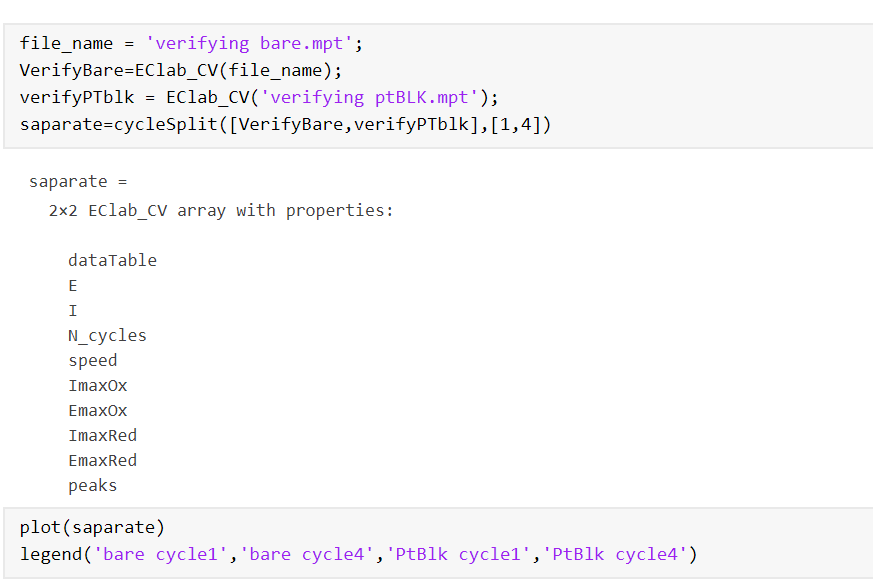
cycleSplit split CV object to several CV objects each contain one cycle.

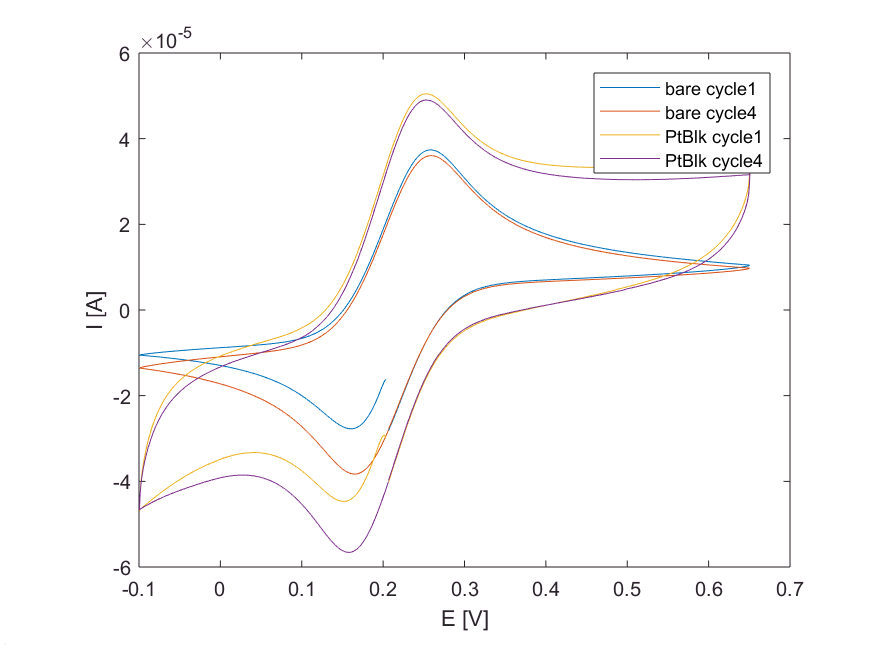
* if b is CV contain 5 cycles, a=CycleSplit(b), than a will be 1X5 EClab\_CV matrix contain 5 different cycle separately.
* if b is CV contain 5 cycles, a=CycleSplit(b,[3:5]), than a will be 1X3 EClab\_CV matrix contains 3 different cycle from 3 to five.
* If b is CV array with 3 components, a=CycleSplit(b), than a will be 3X5 EClab\_CV matrix each column contain 5 different cycle separately.

Example:









**getOx**

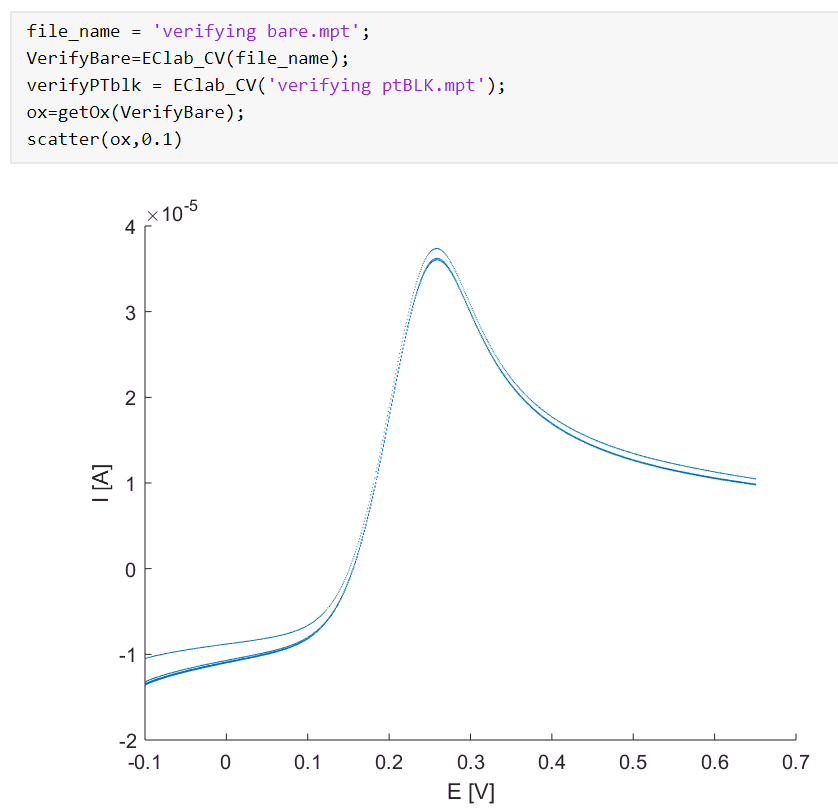
Syntex:

ox=getOx(a)

Description:

This function takes only the oxidation part of CV object. If a is a matrix of CVs ox will be matrix of CVs containing only the oxidation part

Example:



**getRed**

Syntex:

red=getRed(a)

Description:

This function takes only the reduction part of CV object. If a is a matrix of CVs red will be matrix of CVs containing only the oxidation part

Example:

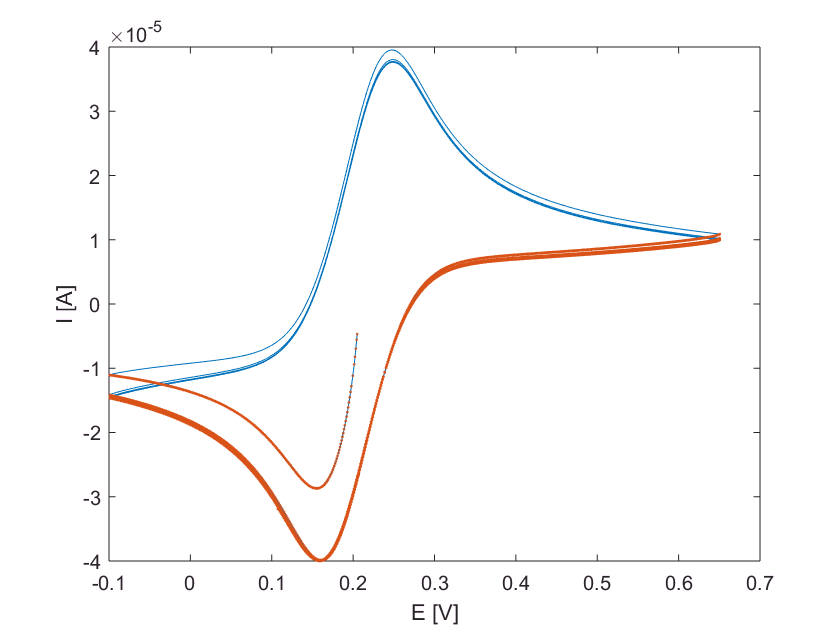
CV=EClab\_CV('1.mpt');

red=getRed(CV);

plot(CV)

hold on

scatter(red,1)



**evenup**

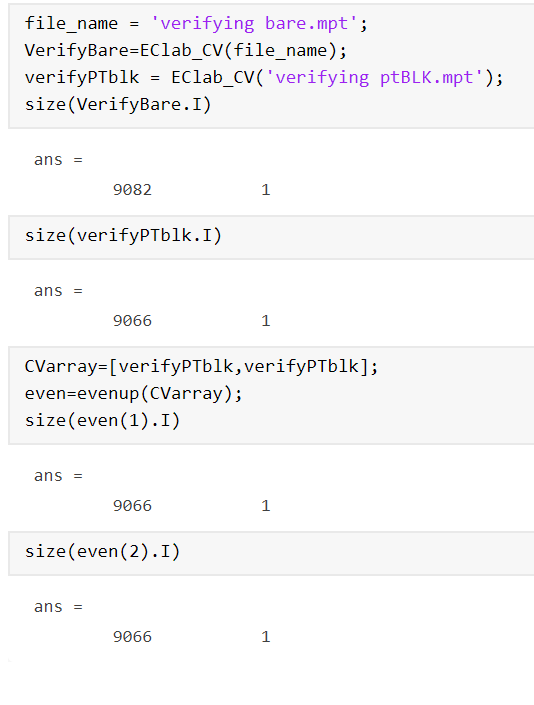
Syntex:

array=evenup(a)

Description:

This function interpolate CV array so all the objects in the array will have the same number of dots. It uses 1st order interpolation and take the smallest number of dots in the array.

Example:



**Iarray**

Syntex:

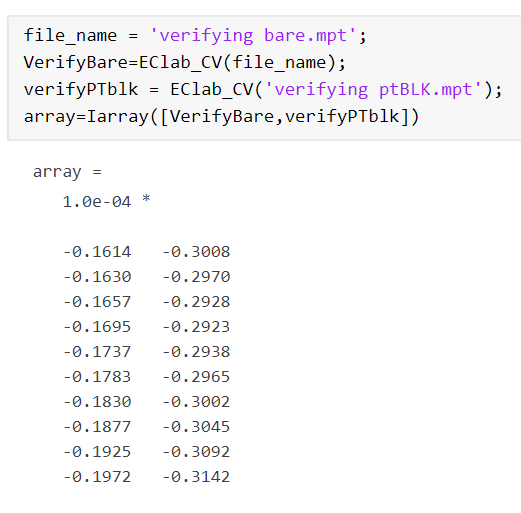
I\_mat=Iarray([a,b])

Description:

This function transform CV array to I matrix.

each column contain the I from the same CV object. Notice that the function uses evenup function so that the I vector will be in the same length.

Example:



**Mean**

Syntex:

M=mean([a,b])

Description:

This function finds the mean current in each potential for several CV object

Example:



**Std**

Same as mean but with std

**RSD**

Syntex:

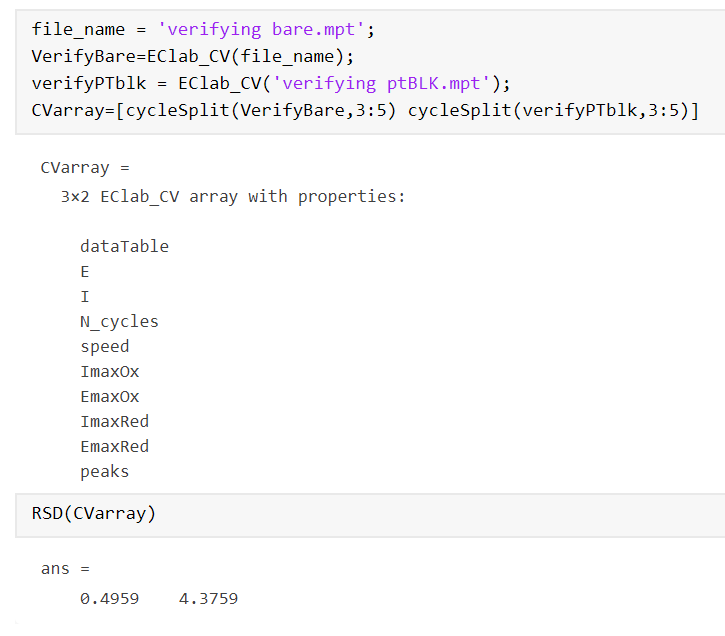
RSD(A)

Description:

This function get the mean RSD between all the I points for an column array of CV objects.

If A is a matrix RSD(A) while return the mean RSD for each column in the matrix

Example:



**getExtrimom**

Syntex:

getExtrimom(obj)

Description:

This function built mainly as a utility function, after doing some changes in the data there’s a need to update extrimoms (for example after doing c=a-b the minus function use getExtrimom function to update c.EmaxRed, c.EmaxOx, c.ImaxRed, c.ImaxOx, c.peaks)

**half\_cycle\_baseline**

Syntex:

[base, slope, intercept]=half\_cycle\_baseline(single\_ox)

Description:

This function is used as an auxiliary function. It get an helf cycle and retuen the base line I value , the slope and the intercept.

Example:

CV=EClab\_CV('1.mpt');

red=cycleSplit(getRed(CV),3);

[base, slope, intercept]=half\_cycle\_baseline(red);

disp(['baseI = ' num2str(slope) '\*E + ' num2str(intercept)])

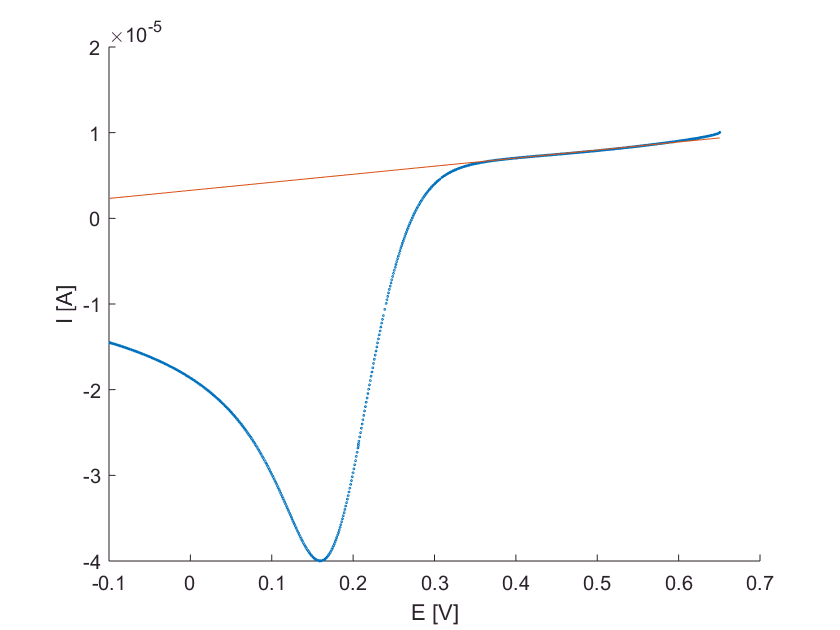
baseI = 9.411e-06\*E + 3.2644e-06

figure

scatter(red,1)

hold on

plot(red.E,base)



**Baseline\_Est**

Syntex:

[baseMat]=Baseline\_Est(obj)

[baseMat,noBase]= Baseline\_Est(obj)

[baseMat,noBase,Base]= Baseline\_Est(obj)

Description:

This function estimates the base line of CV object.

The function estimate the baseline for each half cycle separately.

[baseMat]=Baseline\_Est(obj) will return CV obj mat in the following shape:

|cycle 1| |baseline cycle1 ox | baseline cycle1 Red|

|cycle 2| |baseline cycle2 ox | baseline cycle 2 red|

etc…

[baseMat,noBase]= Baseline\_Est(obj) will return in addition CV array with the CV object subtracted by his baseline

[baseMat,noBase,Base]= Baseline\_Est(obj) will return in addition the baseline of the oxidation and reduction combined

Example: