Tutorial of pHfit procedure

Supplementary Material of the work:

"Non-linear multivariate curve resolution analysis of voltammetric pH titrations"

J.M. Díaz Cruz*, J. Sanchís, E. Chekmeneva, C. Ariño and M. Esteban

Departament de Química Analítica. Universitat de Barcelona
Martí i Franquès, 1-11, E – 08028 – Barcelona (Spain)
E-mail: josemanuel.diaz@ub.edu

FILES INVOLVED

In order to be able to run **pHfit** and/or **peakmaker** from Matlab, the following files must be copied to a directory included in the Matlab path:

peakmaker.m	Matlab program for the visual initial estimation of pure voltammograms.
pHfit.m	Matlab program for the fitting of experimental data matrices by means of potential shift correction.
shiftcalc31.m **shiftcalc32.m** **cgolay.m** **genfactor.m** **golay.m** **grampoli.m** **negtozero.m** **polipes.m**	set of Matlab programs which are used by peakmaker and pHfit

SAMPLE DATA

The workspace Matlab file **pHfitsample.mat** contains original data obtained in the pH titration of the Cd(II)-PC_2 system at 1:2 ratio (for more details see Figure I of Supplementary Material). They can be used to practise some aspects of the pHfit procedure according to the present tutorial.

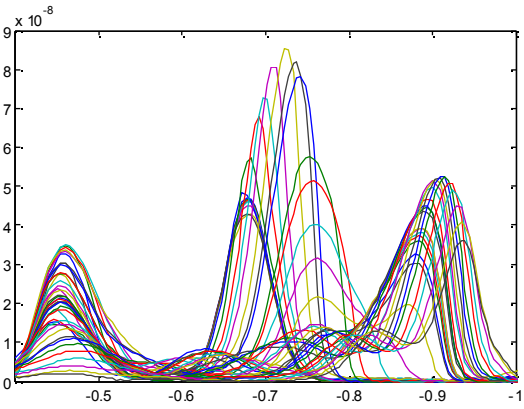
ANALYSIS OF SAMPLE DATA

When you load the file pHfitsample.mat in Matlab you can find the following matrices:

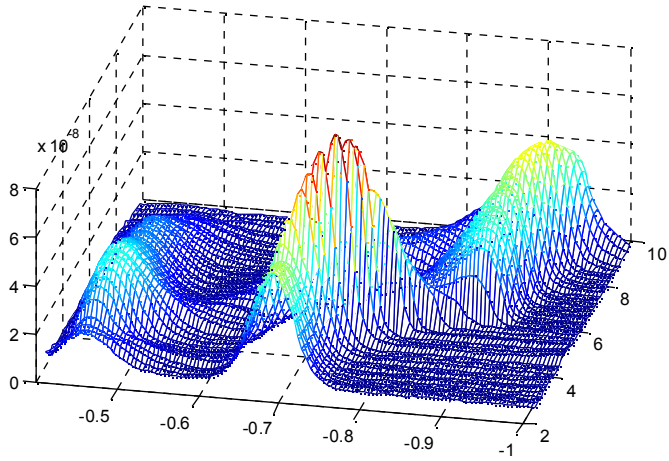
Iexp	(121 rows, 42 columns)	current data matrix obtained at 121 potentials and 42 pH values (42 DP voltammograms)
E	(121 rows, 1 column)	column vector with the applied potentials
pH	(42 rows, 1 column)	column vector with the pH values

You can look at the data matrix in both 2D and 3D ways by means of the commands

```
plot(E,Iexp)
```




```
mesh(pH,E,Iexp)
```



Before starting the pHfit analysis of the **Iexp** matrix, it is necessary to provide a reference pure signal for every component. This can be done in many ways. In the sample file, the reference signal matrix **esv** has been obtained from the visual inspection of the experimental matrix using the **peakmaker** program. This program can be invoked in the way:

```
esv=peakmaker(Iexp);
```

Then, you get the mask:

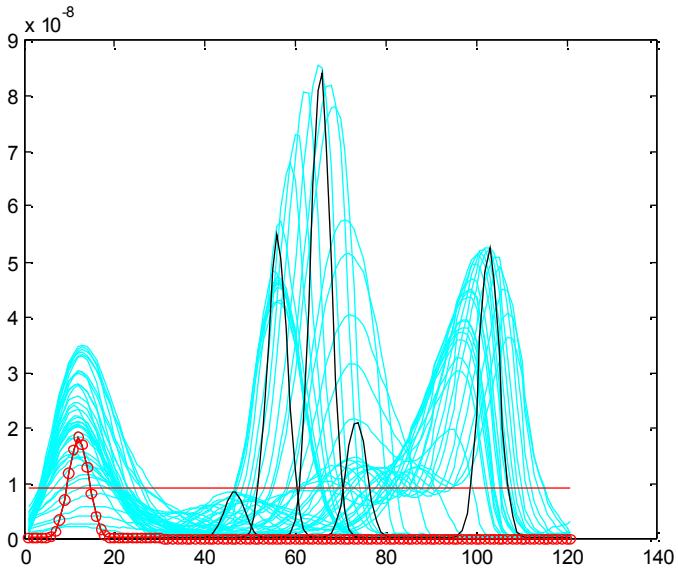
```
INITIAL ESTIMATION OF PEAK-SHAPED FUNCTIONS

How many peaks per species?
```

You have to answer 1 (and press enter), unless you have a component defined as a double peak, which is not usual

Later, you are asked about the symmetry of the peak (1 for symmetric, 0 for asymmetric) and after selecting it you get the message:

```
Symmetric peak? yes (1), no (0)    1

Select maxima of the peaks
with mouse and press enter
```

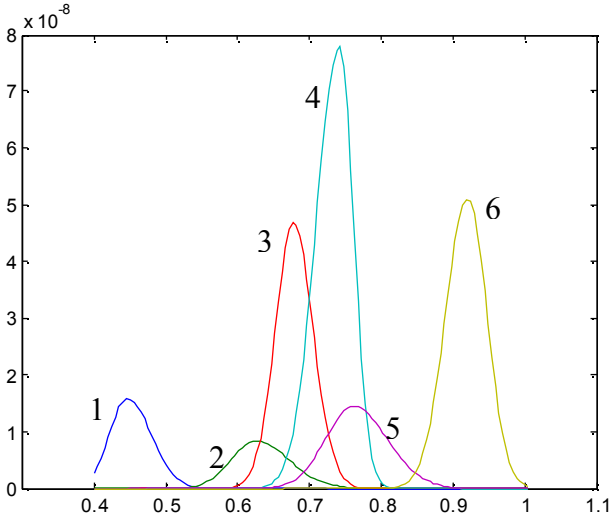
Simultaneously, you get a cyan 2D plot of the experimental matrix. Then, you have to click with mouse the top of every component, which defines both height and position of the peaks. Until you finish you will not see any change in the plot. When you have completed the selection, just press enter and a first estimation of all peaks will be displayed over the cyan-coloured plot of the matrix. In the case of the sample data, 6 components should be defined, which produces a graph more o less like this (it can change, depending on where you arbitrarily choose the position and the height of the peaks, and the best is to try in these parts of the experiment where the peaks are less overlapping to other signals):



In this graph, peaks are shown in black, except for the first one, which is red and denotes the experimental points with circles. A horizontal line is drawn at the half height of the peak. Then you have to select with mouse the x-axis position where you expect the peak to cross such line. In this way you are selecting the width of the symmetric peak, which is plotted now in black, while the red colour and the circles move to the next peak to repeat the same operation.

If you are working with asymmetric peaks, you have to click twice around every peak, to define the left and right boundaries and, hence, the asymmetry). At the end of the process, you should obtain an **esv** matrix similar to that provided in the file (**esv_reference**, with 121 rows and 6 columns):

```
plot(-E,esv)
```



The numbers indicate the order in which the components have been selected.

Besides the **esv** matrix, to run **pHfit** it is necessary to provide some information inside the **isshift** matrix. In the sample file, the elements of **isshift** are:

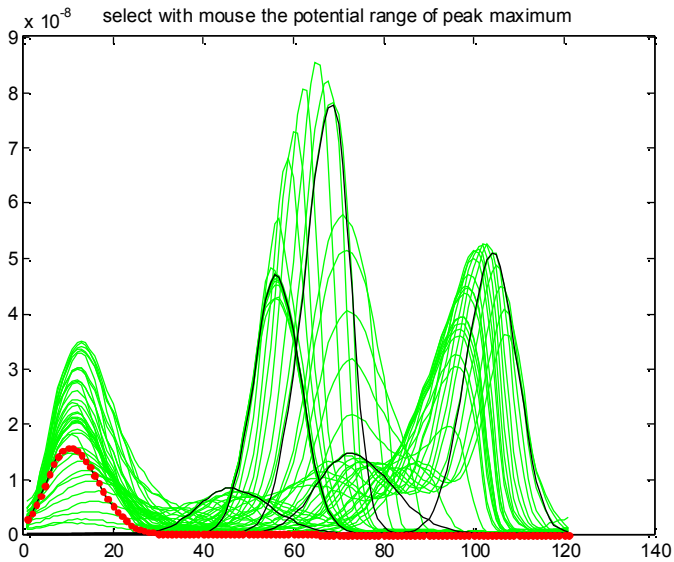
1	2	0	3	3	3
1	16	1	9	28	17
42	31	12	26	42	42
1	16	1	9	28	22
42	30	12	21	42	42

The first row indicates the type of movement of every peak (*e*.g., third component is immobile and is denoted with 0, whereas forth component is sigmoid and is denoted with 3; see Section 2 of the paper for more information). Inside the column corresponding to each component, the second and third rows define the existence range (i.e., the scan numbers when the peak appears and disappears) and the fourth and the fifth define the fitting range (i.e., the range of scan numbers where the potential shifts can be fitted to a linear or sigmoid shape). For instance, component 4 appears in voltammogram number 9 and disappears after voltammogram 26, but only the voltammograms inside the range 9-21 are convenient to make an initial estimate of the sigmoid evolution of potential shifts (in the rigorous fitting all the voltammograms inside the existence range will be considered). The numbers in the last four rows have to be deduced by a careful inspection of the data matrix, voltammogram by voltammogram.

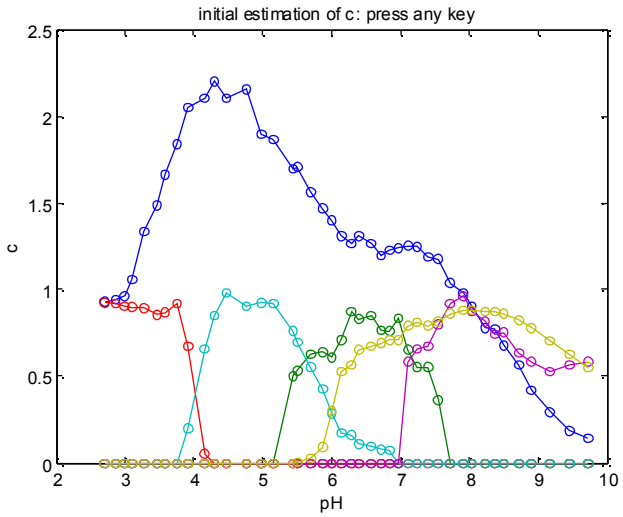
Once **esv** and **isshift** are ready, **pHfit** program can be launched with the command:

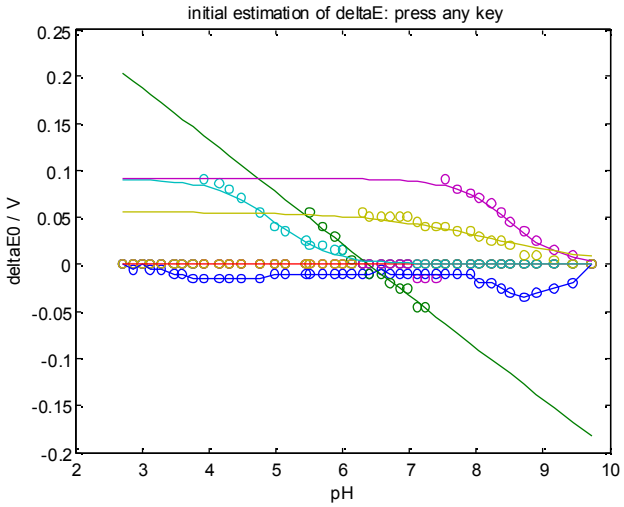
```
[deltaE,Icor,Irep,c,slope,infpoint]=pHfit(Iexp',esv,E,isshift,pH);
```

The experimental matrix must have the potential in columns and the pH in rows, so that the previous **Iexp** has to be transposed (**Iexp'**) to have 42 rows and 121 columns.
After executing this command, a figure shows the data matrix (in green) and the unit voltammograms of **esv**:

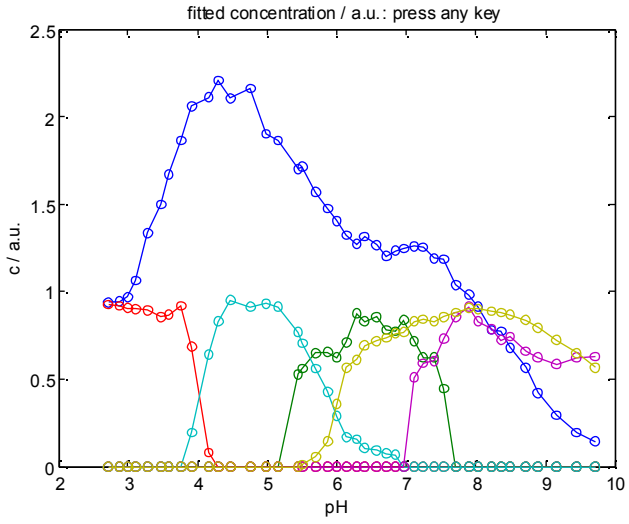


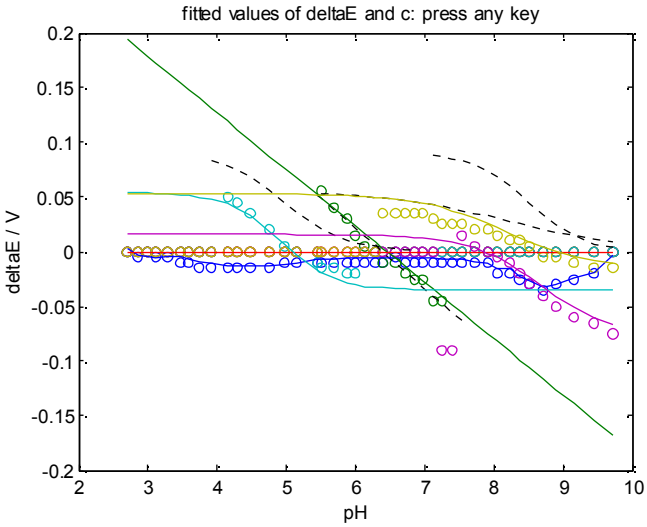
The graph denotes with red points the first unit voltammogram and asks for the potential range where you expect to find the maximum as it moves (and to help in this task, it shows in green the experimental voltammograms inside the existence range of the component considered). The left and right limits of the potential range have to be selected with mouse. Afterwards, the red points move to the next component to repeat such operation. In the case of immobile peaks (type 0), the program asks only to press enter. Finally, a preliminary estimation of the concentration profiles is shown and, after you press enter again, the initial estimation of potential shifts appears:



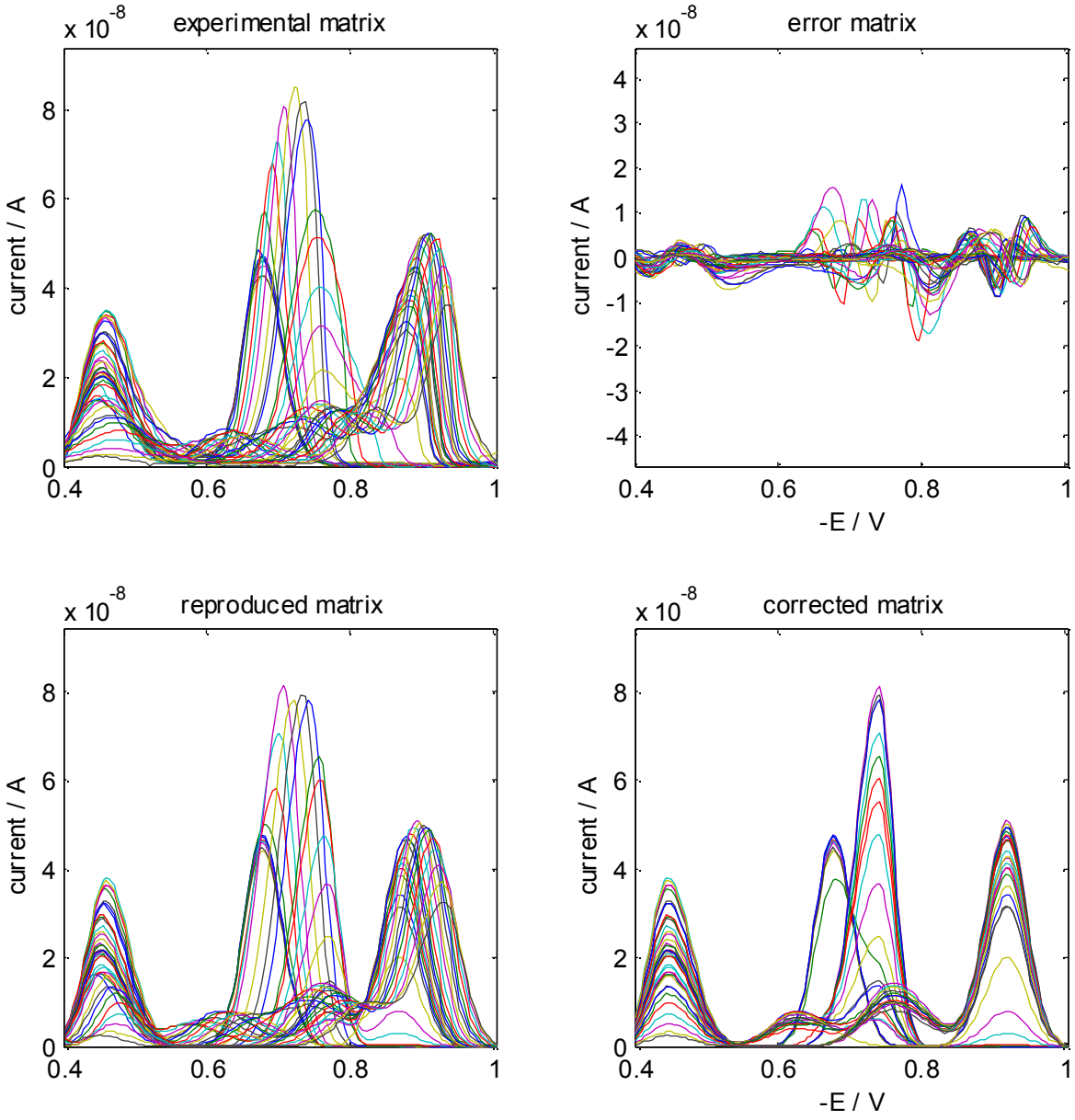


In the case of deltaE, circles denote a rough estimation of potential shifts by measuring the position of the maxima of overlapping peaks (inside the potential range of the component) and solid lines denote a first estimation of lines or sigmoids from these points (which are computed only in the fitting range). If a component is set to unconstrained fitting, the solid line is just joining the points. When you press enter, the iterative process starts to optimize all potential shift functions taking into account the full set of voltammograms and, after 60 - 120 seconds (it depends on the computer), you get several figures (after pressing enter several times) with both fitted potential shifts (**deltaE** matrix) and concentration profiles (**c** matrix):





In the case of potential shifts, the fitted functions are denoted with solid lines and they are compared to the initial estimations (circles and dashed lines).



The program generates the following matrices:

deltaE (42 x 6): potential shifts ΔE for the 6 components

Icor (42 x 121): matrix corrected for potential shifts

Irep (42 x 121): reproduced matrix

c (42 x 6): concentration profiles for the 6 components

slope (6 x 2): first column: slope of linear ΔE *vs*. pH evolution or maximum slope in the case of sigmoid ΔE *vs*. pH evolution; second column: associated standard deviations

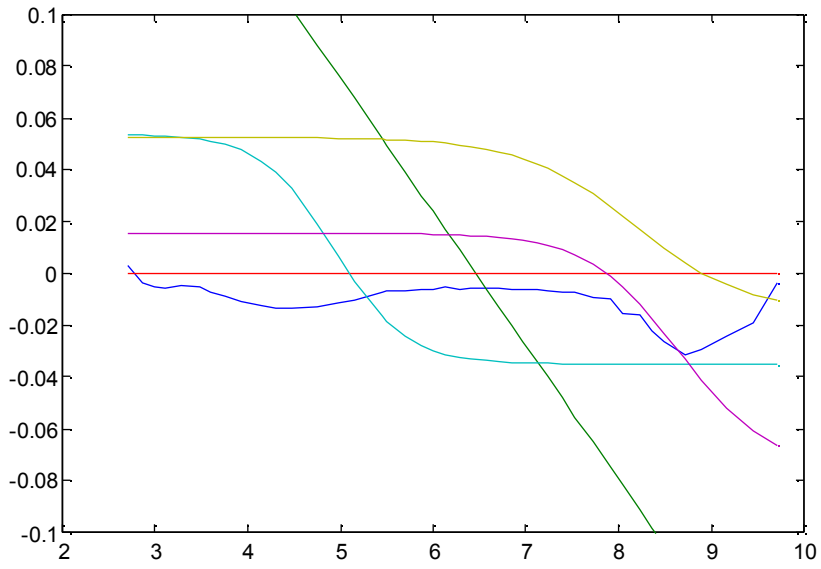
infpoint (6 x 2): first column: inflexion point of sigmoid ΔE *vs*. pH evolution; second column: associated standard deviations

Some of this information appears in the command window when the program finishes and additionally, the lack of fit is given:

```
maximum slope (in absolute value):

slope =

         0         0
    0.0515    0.0030
         0         0
    0.0573    0.0003
    0.0463    0.0014
    0.0282    0.0002

inflexion point

infpoint =

         0         0
         0         0
         0         0
    4.9308    0.0115
    8.6526    0.0576
    8.1601    0.0169

Current plot held
Current plot released

lack of fit (%) :
   15.4552
```

When many components are involved, it is interesting to plot again the optimized potential shift values in order to have a clearer view of them without the disturbance of initial estimates. To do this, just type:

```
plot(pH,deltaE)
```



You can practise this methodology and compare the results with the reference matrices obtained in the present work (e.g. **deltaE_reference**). They should not be too different, unless you choose a different position for the reference voltammograms in **esv,** or different potential, existence or fitting ranges. In this case you would probably get a quite higher lack of fit.