Fitting tool v.1.0

Short user manual

The program is intuitively clear to start working with. However, the following steps summarized below can guide a user how to proceed.

Please follow these steps:

Step 1. Open or import ascii file with experimental data. If you open file, the latter has to have 2 columns with space as delimiter, e.g.: "3.56 7.886". Typical file looks as following, see Fig.1.

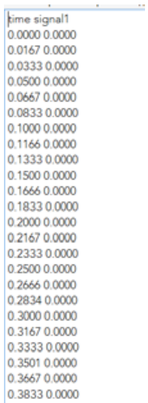

time signal1
0.0000 0.0000
0.0167 0.0000
0.0333 0.0000
0.0500 0.0000
0.0667 0.0000
0.0833 0.0000
0.1000 0.0000
0.1166 0.0000
0.1333 0.0000
0.1500 0.0000
0.1666 0.0000
0.1833 0.0000
0.2000 0.0000
0.2167 0.0000
0.2333 0.0000
0.2500 0.0000
0.2666 0.0000
0.2834 0.0000
0.3000 0.0000
0.3167 0.0000
0.3333 0.0000
0.3501 0.0000
0.3667 0.0000
0.3833 0.0000

Fig. 1. Example of acceptable experimental data format.

Once loaded successfully, the message window (Fig. 2-1) will pop up, showing the length of the data file (length of a column).

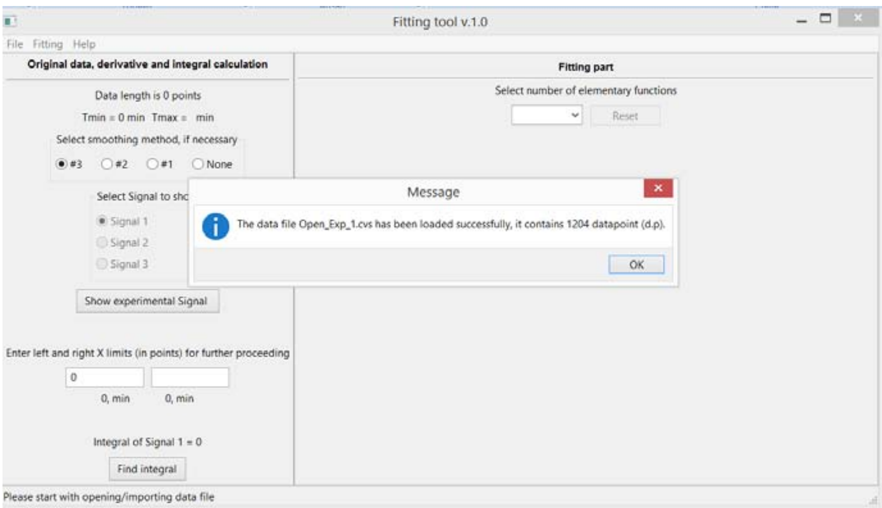
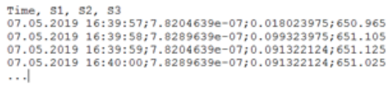


Fig. 2-1. Message notifies on successfully loaded data.

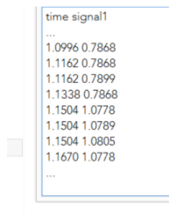
Note. It is possible to import a little bit different data format (Fig. 2-2), where instead of one - three signals: Y1, Y2 and Y3 with one X column are gives and user can choose which signal to proceed with. Moreover, X is given in a different time-date format: "07.05.2019 16:39:57". The program will automatically set first point in the X dataset as 0, whereas the following points will be recalculated in minutes passed after zero moment of time. And semicolon serves here as delimiter.



```
Time, S1, S2, S3
07.05.2019 16:39:57;7.8204639e-07;0.018023975;650.965
07.05.2019 16:39:58;7.8289639e-07;0.099323975;651.105
07.05.2019 16:39:59;7.8204639e-07;0.091322124;651.125
07.05.2019 16:40:00;7.8289639e-07;0.091322124;651.025
...
```

Fig.2-2. Another supported format to load experimental data file.

Step 2. Select one of the data smoothing methods, if necessary. It is advised to select one of them to minimize error and noise in signal and signal's derivative calculation. It is strongly recommended to use smoothing method #3, if your original experimental data contains noise some X -values being fully identical to each other, see Fig. 3.



```
time signal1
...
1.0996 0.7868
1.1162 0.7868
1.1162 0.7899
1.1338 0.7868
1.1504 1.0778
1.1504 1.0789
1.1504 1.0805
1.1670 1.0778
...
```

Fig.3. Example of repeated X values in the experimental data, where use of smoothing method #3 is strongly recommended.

Step 3. To show the loaded signal - Y *vs* X and its derivative - $\frac{dY}{dX}$ *vs* X, please press the '*Show experimental signal*' button. If everything is OK with data, a graph similar to the following, Fig. 4 will be plotted.

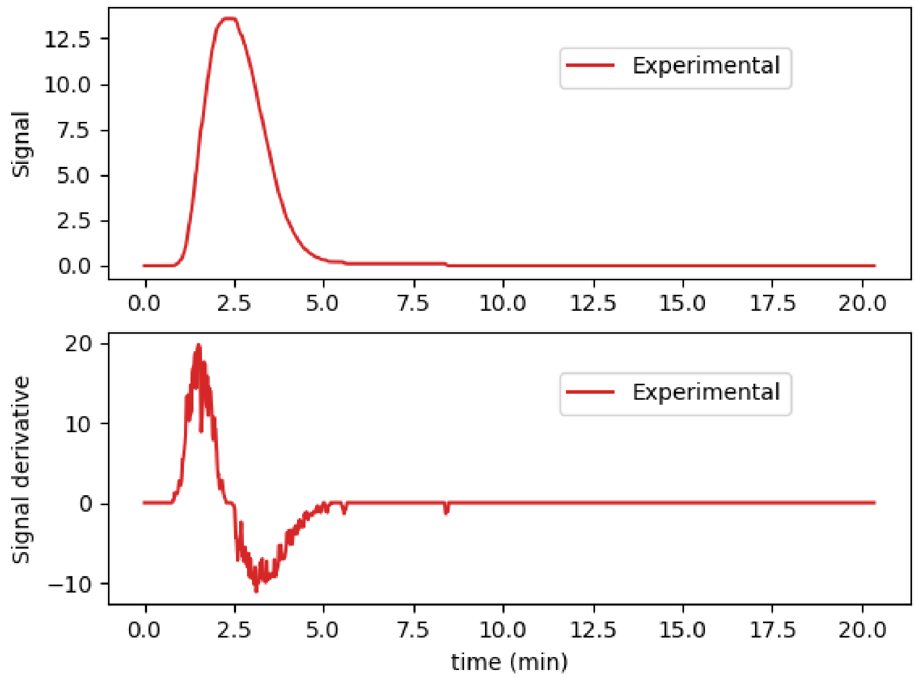


Fig.4. Example of experimental signal and its derivative graphs.

Step 4. With help of 'left' and 'right' X limits, see Fig. 5, it is advised to focus on the part of the curve which is of interest. By default the 'left' limit is set to 0 and the 'right' limit is set to last point in the X array. By pressing the same button '*Show experimental signal* ' zoomed graph will be displayed, see Fig. 5.

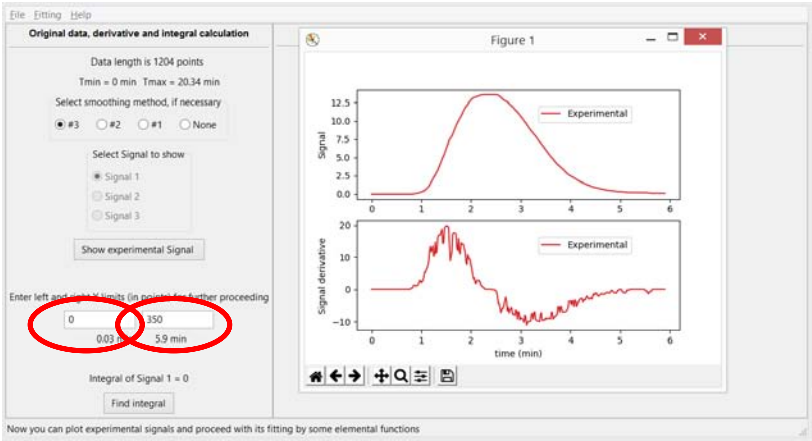


Fig. 5. Zoomed graph within the new time limits, corresponding to the entered left and right X limits.

Step 5. The button 'Find integral' calculates and shows integral of the signal within the entered left and the right limits, as is shown in Fig. 5.

Step 6. In the 'Fitting part' please choose how many elementary functions will be used for fitting. Minimum - 1, Maximum -4 (in this version). The corresponding number of drop down menus will appear, see Fig.6.

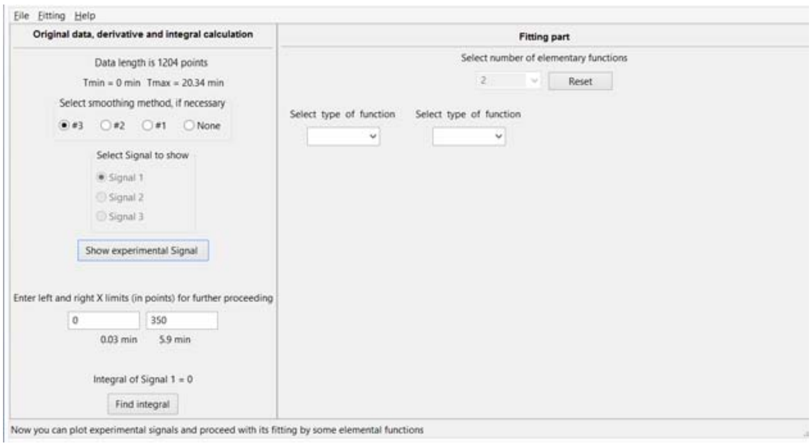
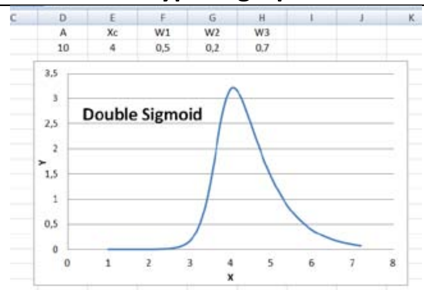


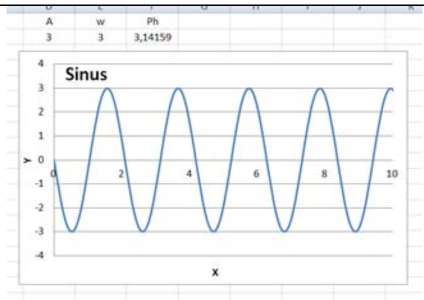
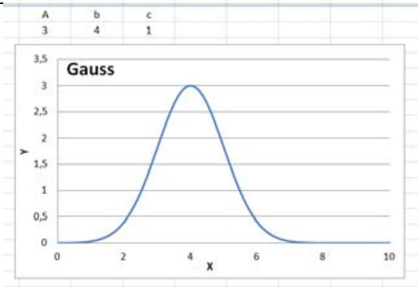
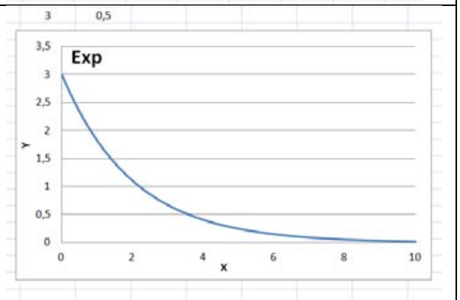
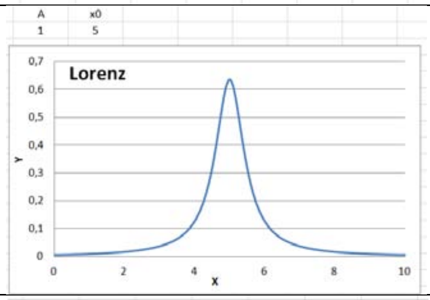
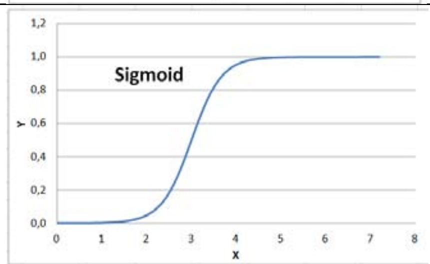
Fig. 6. Appearance of selected number of drop down menus to proceed.

Step 7. In each drop down menu, please, select type of mathematical function to be used for fitting. Or press the 'Reset' button to repeat Step 6.

Note: When it is active, pressing the 'Reset' button always brings you to the **Step 6**.

The following six elemental functions are built in the program.

Acronym	Full name	Formula and parameters	Typical graph
DbSigmoind	Double sigmoid	$Y(x) = A * \dfrac{1 - \dfrac{1}{1 + e^{-\frac{x - x_c + w1/2}{w3}}}}{1 + e^{-\frac{x - x_c + w1/2}{w2}}}$ Parameters: A, X_c, W1, W2, W3	

Sinus	Sinus	Y (x)= $A * Sin(w * x + Ph)$ Parameters: A, w, Ph	
Gauss	Gaussian function	$$Y(x) = A * e^{-\frac{(x-b)^2}{2c^2}}$$ Parameters: A, b, c	
Exp	Exponential function	$$Y(x) = A * e^{-b*x}$$ Parameters: A, b	
Lorenz	Lorentzian Function	$$Y(x) = \frac{A/_{2\pi}}{(x - x_0)^2 + A^2/_4}$$ Parameters: A, x_0	
Sigmoid	Sigmoid	$$Y(X) = \frac{A}{1 + e^{-2B*(x-C)}}$$ Parameters: A, B, C	

Step 8. If necessary, in the appeared table(s) (Fig. 8) one can modify 'Min', 'Max' and 'Def' values of each parameter of the chosen function(s). 'Def' - determines default or initial value of the parameter to be optimized. 'Min' and 'Max' values determine the range as the left and the right limits, where user can look for the optimal value of the parameter.

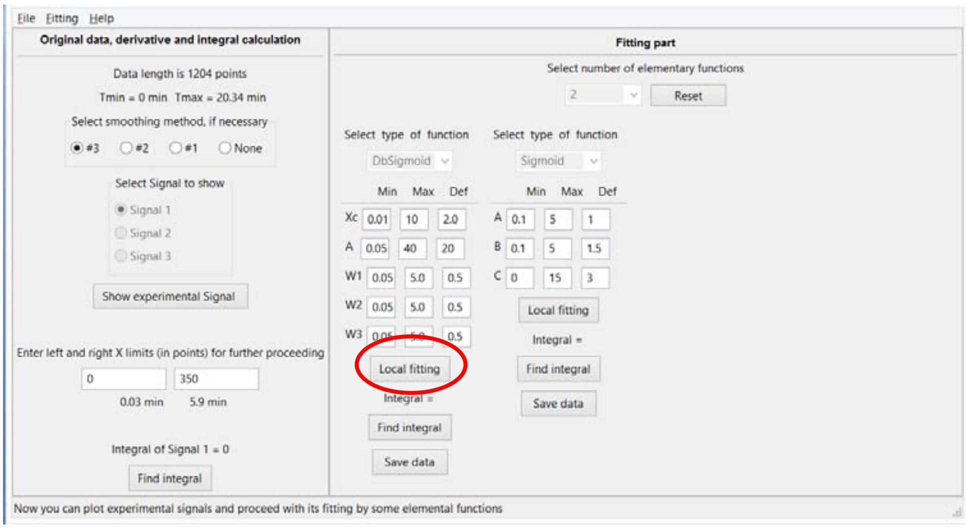


Fig. 8. In the appeared table one can modify 'def' (current) values of each parameter of each selected elemental function. The range, determined by 'min' and 'max' values, where the optimum value of the parameter will be looked for.

Step 9. Press any 'Local fitting' button in the 'Fitting part', to fit the experimental data with **only one** of earlier selected function, Fig. 8.

Note, **only one** selected function will be used here. Most likely it will not be enough to get a good fitting, but it gives a kind of taste of what can be achieved in fitting with one function only. Sometimes fitting with only one function helps to find faster the optimum parameters in case of multiple function fitting. In fact, the next (second, third, etc) function can be added to the first one with already almost optimized parameters.

Alternatively, one can start fitting with e.g. two functions (press 'Fitting\Global fitting'), but deliberately make amplitude of the first function equal to zero (or close), Fig. 9

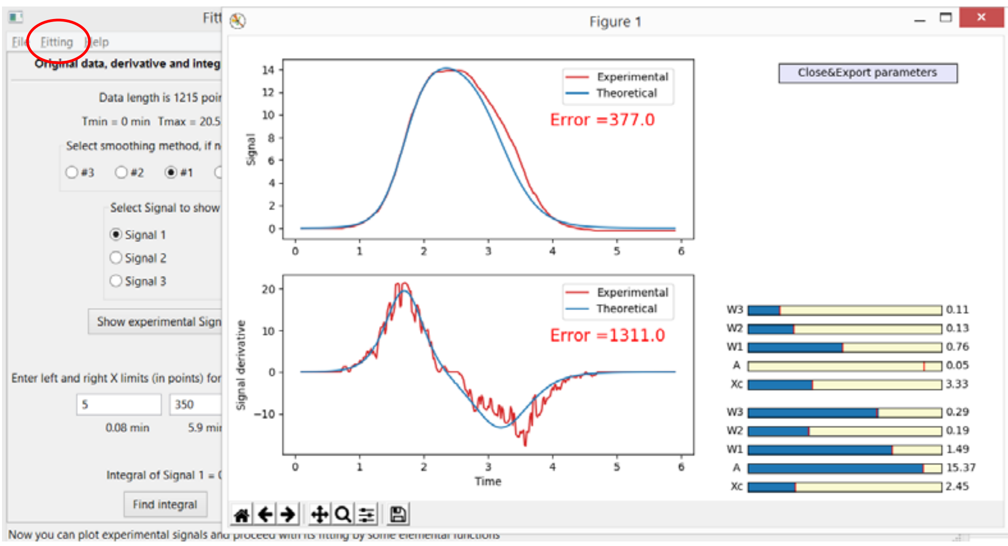


Fig. 9. Example of one function fitting. Note matching the shape of main features in the signal's derivative graph.

Step 10. Manually move the sliders within the earlier fixed limits ('Min' and 'Max') for every parameter of the function to minimize the fitting error and get the best possible (so far) matching with derivative curve's shape, Fig. 9.

Note. The fitting error ($Error = 1000 * \sqrt{\frac{\sum_{i=1}^{N}(Yexp_i - Ytheory_i)^2}{N(N-1)}}$) is automatically calculated for fitting of the original signal and its derivative.

Note. One can export the current values of all parameters back to the main window by pressing the 'Close&Export parameters' button, Fig. 9.

Step 11. Either one can press 'Fitting\Static Graph' or Fitting\'Global fitting' to see static or dynamic (with possibility to modify all parameters) superposition of all functions involved in the fitting or simply make amplitude of the second function to be above zero. Often adding second or third, etc. function helps to improve fitting results in term of matching the shape of both curves (original signal and its derivative) and the error values, Fig. 10.

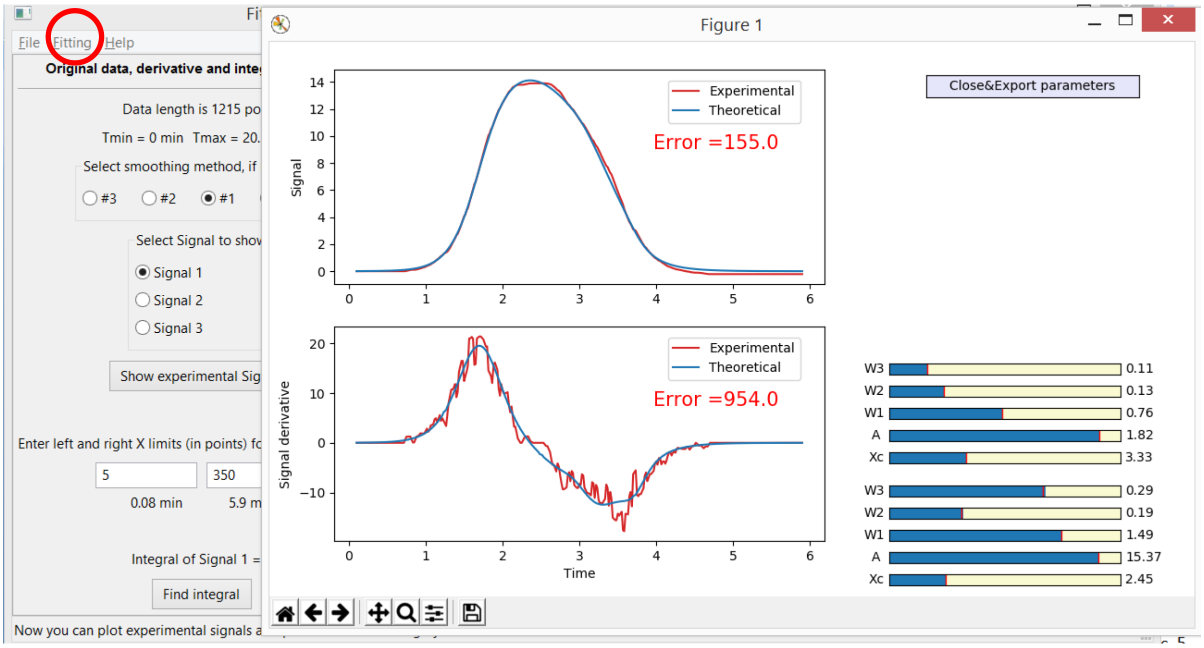


Fig. 10. Example of dynamic fitting with three function of the same type (DbSigmoid). Note by being more accurate and passion one can even better fitting.

Step 12. When the best fitting is achieved and the optimal parameters are found and exported to the main window, one can calculate integral of each individual elemental function (with optimal parameters and within earlier defined X range). One can also save X, Y values of the individual function in easy readable format by pressing 'Save data' button.