



# Howto: Fit sigmoid functions in OpenOffice Calc and Excel

SVEN KOCHMANN

READ REVIEWS

WRITE A REVIEW

CORRESPONDENCE:

DATE RECEIVED:  
June 10, 2015

© Kochmann This article is distributed under the terms of the [Creative Commons Attribution 4.0 International License](#), which permits unrestricted use, distribution, and redistribution in any medium, provided that the original author and source are credited.



Last week, I posted an [article about sigmoid functions](#) and how to use them. Nevertheless, it is hard to guess the parameters for a given problem. So, people use software such as Origin [1] or QtiPlot to fit.

Personally, I use Origin/QtiPlot only for plotting and Excel/OO–Calc for evaluation/calculation, because both programs are much more comfortable and powerful. However, both lack the possibility to fit sigmoid functions, automatically.

It is possible to do such fits for nearly any function using the *solver*, though, but only a few people I've met knew how to do this or that it is even possible. But, I think the *solver* is a very handy feature and, therefore, I want to give here a short introduction into using it for fitting a sigmoid function to a set of data.

First, we need a set of data. We take the points from the [article last week](#):

pH, F: 3.00, 0.019 ; 3.80, 0.199 ; 4.00, 0.283 ; 4.30, 0.500 ; 4.70, 1.100 ; 5.10, 1.600 ; 5.50, 2.300 ; 5.92, 3.161 ; 6.30, 4.499 ; 6.70, 5.300 ; 7.12, 5.500 ; 7.50, 5.981 ; 8.00, 5.993 ; 8.52, 6.075 ; 9.00, 6.059

Open a blank spreadsheet in Excel or OO–Calc and insert these values in two columns *pH* and *F* (*meas.*). Besides, add another two columns labeled *F* (*calc.*), and  $\Delta F$ . Below the last column add a cell for the sum of  $\Delta F$ . Last, add a list of parameter cells labeled *a*, *b*, *c*, *d*, *f*, and *t*. Fill them with values of 1.0 (for *a*, *b*, *c*, *d*, and *f*) and 0.0 (for *t*).

It is time to fill in our function. Just put the formulas below into the *F* (*calc.*) column. For *pH* take the corresponding cell of the first column. For the parameters take the corresponding cells – don't forget to fix them! If you have marked a cell in a formula, you can press F4 (in Excel) or SHIFT+F4 (in OO–Calc) to fix the cell.

$$F_{calc.}(pH) = \frac{a}{b + c \cdot e^{-d \cdot (pH - f)}} + t$$

Next, we fill the  $\Delta F$  column with the following formula. This will be our error indicator. It is just the difference between the measure *F* and the calculated *F*. We use the square function to get rid of the sign of this difference.

$$\Delta F = (F_{calc.} - F_{meas.})^2$$

If not already done, sum up the values of column  $\Delta F$  by the sum-macro. In the end, the spreadsheet should look like in the following figure.

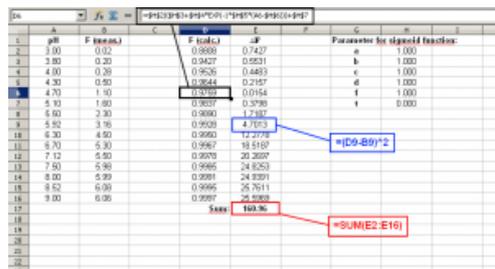


Figure 1. Our basis spreadsheet showing the necessary columns and the inserted formulas for the sigmoid calculation (black), the error calculation (blue), and the sum of the errors (red).

We have included everything we need to fit our function parameters. The *solver* works by varying parameters to optimize (=minimize) a function. The function to minimize is NOT our sigmoid function but our error sum. Since the error is calculated by adding the differences between our calculated  $F_s$  and the measured  $F_s$ , the sum will directly judge the fitting of the sigmoid function and its parameters. If the error is zero, there is no difference between the calculated and the measured  $F_s$  anymore and we found the perfect fit. In reality, of course, the error will just get very small but never reach zero.

First, mark the cell you want to minimize. In my case it is E17. Then start the *solver* tool. You find it under Tools/Solver in OO-Calc or under Data/Solver in Excel (if it is not visible then you have not activated the Add-in. See [2] or [3].). The solver dialog box should open. The target cell is the cell we want to minimize (here: E17). Also, select the 'Min(imum)' option under the target cell. The solver should vary our parameters, so select the corresponding cells (here: H2:H7). We do not need any further conditions, so leave the list empty.

The standard solver algorithm in OO-Calc is only usable for linear problems. Since this is not the case here, you need an extension (add-in) [4]. Install and active it. Then you can select two new algorithms for the solver engine under 'Options'. I used 'DEPS Evolutionary Algorithm' with standard parameters.

Finally, click on 'Solve' to start. It starts calculating and varying the parameters. If it has found a solution it will ask to keep it. When you keep it, the values of your parameters should have changed and the calculated  $F_s$  are now similar to the measured one. The error should be much lower now (0.25 instead of 160.96).

Now, you can plot the data points and the function. I made two extra columns for plotting the calculated function, since plotting only few points will make the curve looking angular. My final spreadsheet is depicted in Figure 2.

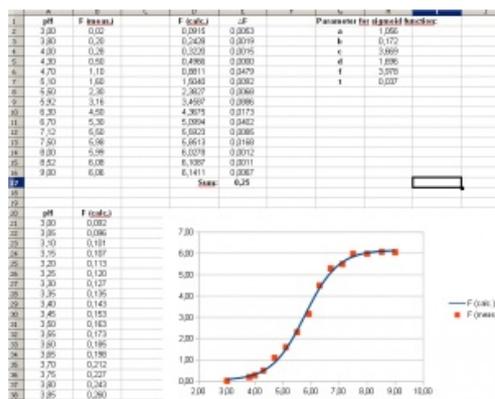


Figure 2. The final spreadsheet with the fitted sigmoid function plotted.

That's it! As stated in the beginning, this procedure is not limited to sigmoid functions and can be used for other fittings, too.

It should be noted that the solver uses numerical analysis (mathematical optimizing [5]) to find solutions. It may happen that the algorithm does not find a feasible solution or that it converges to a local optimum. In these cases you have to play around with the parameters of the algorithm (e.g. increase the steps). Also, the start values have an huge impact on the behavior: the better the guess of starting values the more likely it is that a (feasible) solution is found.

#### REFERENCES

1. " OriginLab - Origin and OriginPro - Data Analysis and Graphing Software " <http://www.originlab.com/>
2. "Load the Solver Add-in" <http://office.microsoft.com/en-us/excel-help/load-the-solver-add-in-HP010021570.aspx>
3. "Load the Solver Add-in" <http://office.microsoft.com/en-us/excel-help/load-the-solver-add-in-HP001127725.aspx>
4. "Solver for Nonlinear Programming [BETA] | Apache OpenOffice Extensions", *Apache OpenOffice Extensions* <http://extensions.services.openoffice.org/project/NLPSolver>
5. "Mathematical optimization - Wikipedia, the free encyclopedia", *Wikipedia* [http://en.wikipedia.org/wiki/Mathematical\\_optimization](http://en.wikipedia.org/wiki/Mathematical_optimization)