

Adapted from
Elementary Mathematical Models, 2nd Edition
Technology Guide
 Dan Kalman, Sacha Forgoston, Albert Goetz

5. Curve Fitting

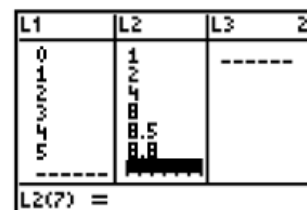
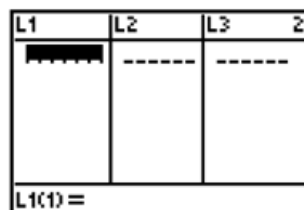
In the Elementary Mathematical Models text many examples and exercises involve fitting a model of a specified type to a set of data. This can be done in an informal trial-and-error process using spreadsheets as described in Section 4 of the Technology Guide. Here we briefly mention a few options for obtaining optimized fits for those students and instructors who wish to pursue this approach. Curve fitting is sometimes referred to as *regression*.

Calculators. Some graphing calculators have built in functions to obtain optimized fits of curves to data. In the example below a logistic function is fit to a set of points.

Example 5.1. We wish to fit a logistic growth model to the sequence 1, 2, 4, 8, 8.5, 8.8. The terms of the sequence and the position numbers appear in the table at right, where we have assumed that the first term, 1, has position number 0.

position number	term
0	1
1	2
2	4
3	8
4	8.5
5	8.8

On the TI-84, each column of the table will be entered as a list. Press the [STAT] key to obtain the first screen shown below. Then key in 1 to obtain the edit window, as depicted in the second screen below with no defined entries in any of the lists. If list entries have been defined previously, the window will look similar to the third screen below.



List values are entered as follows. First, use the ► and ◀ keys to move the insertion point into any desired column. If there are no defined entries in the column, the insertion point automatically goes to first position. Enter a numerical value and press either [enter] or ▼ to

continue with the next entry. Entries that have already been defined can be changed by moving the insertion point with ▲ or ▼ to the desired location, and then entering a new value. To append new values to a previously defined list move the insertion point to the line of dashes. The third image above shows the calculator screen after entering the position numbers for our example in list 1 and the sequence terms in list 2.

Once the data values have been entered, press the [STAT] key again. This time use ► to highlight CALC as shown in the first screen below. Scroll down to item B Logistic and press the enter key. That leads to the second screen below.



Observe that the Xlist and Ylist are identified as L_1 and L_2 .¹ No input has been defined on the next two lines representing optional parameters. Using the cursor keys, highlight the calculate command and press the enter key. This reveals the third screen above. It gives the form of the logistic equation, and the optimal values of the coefficients a , b , and c .

The values of a , b , and c can be used in additional calculations, or as coefficients in equations. To enter them into the display of the calculator, press the VARS key, then select and enter option 5, cursor over to the EQ column, and finally cursor down to the desired variable name, a , b , or c , or simply enter the appropriate number (2 for a , etc.).

Excel. There are two different curve fitting procedures in Excel, corresponding to linear and nonlinear least-squares optimization. These are included in two application packages that are available in Excel, but are not always loaded when the software is installed. To add them to your installation, open the file tab at the top of the Excel window and select Options from the menu along the left side of the screen. On the window that pops up, select Add-Ins on the menu at left. Select the Analysis Toolpak for linear least-squares and Solver Add-in for the nonlinear least-squares applications.

Linear Least-Squares. Linear least-squares finds the best coefficients in an equation that is a linear combination of known columns or functions. This cannot be used directly to fit a logistic equation.

Nonlinear Least-Squares. The nonlinear least-squares app can be used to optimize a collection of parameters that enter into the calculation of a cell on a spreadsheet. First the sheet is arranged to calculate the target cell's value for one set of parameter values. Then, when the nonlinear least-squares app is launched, the user identifies which cells are parameters, what the target cell

¹ These inputs can be edited if data have been entered in different columns of the list definition screen. Notice that L_1 through L_6 appear as labels above the number keys 1 through 6. To access list L_3 , for example, key in [2nd L_3].

is, and what the goal of the optimization is. For fitting a function to data, we select as the target cell value the computed value of the sum of squared deviations between the original y data and the model data, and our goal is to minimize that value. The optimization app is iterative in nature, and may find a local, rather than a global, minimum. We illustrate the process with the next example.

Example 5.2. We will again find a logistic model for the data considered in Example 5.1. This time, however, we will use the nonlinear optimizer, and will compute the model values in a single column.

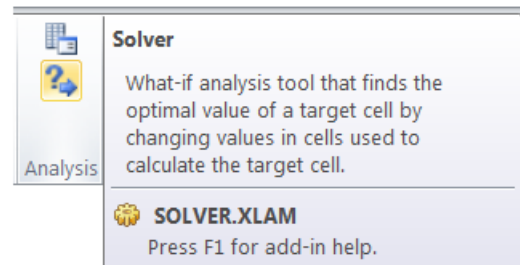
In the figure at right we have entered the y and x values in columns A and B, and initial values for coefficients a , b , and c below. The entries in column C are computed using the formula $c / (1 + ae^{-bx})$, where the coefficients are replaced with references to the cells A10, B10, and C10. The formula that appears in cell C2 is

$$= \$C\$10 / (1 + \$A\$10 * EXP(-\$B\$10 * B2)).$$

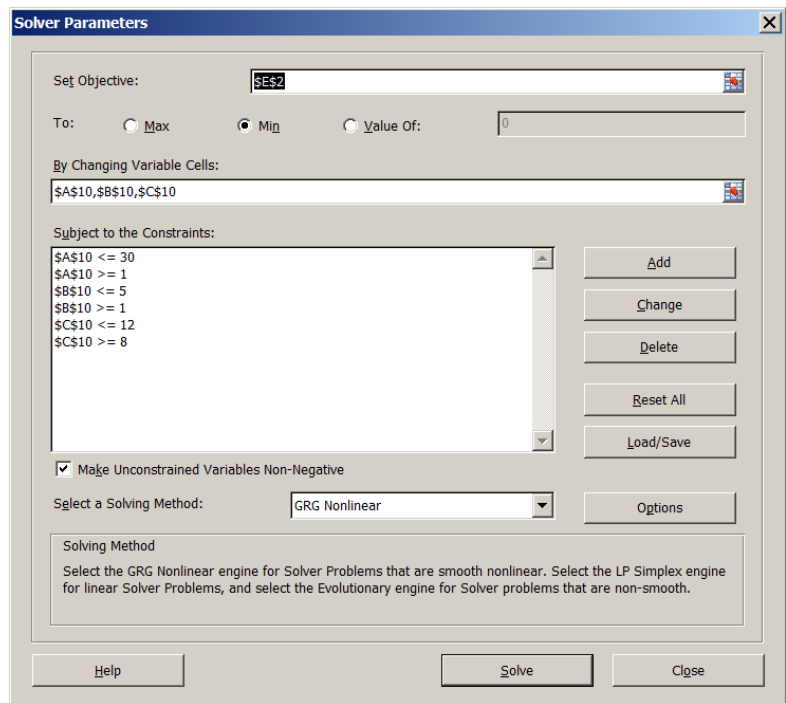
The dollar signs are included so that the references to the coefficient cells are absolute locations, and are not changed when the formula is copied to other cells. Thus we can copy cell C2 to cells C3 through C7 to populate those cells with correct values. Each value in column D is computed as the square of the difference between corresponding values of y and y -hat. Cell E2 shows the sum of the entries in column D. This is the sum of squared errors from our model, and is the value we wish to minimize. The initial coefficient values of 15, 1.5, and 9.5 were chosen by trial and error to roughly fit the data. We will restrict the range over which each coefficient is varied to an interval containing the initial value. In practice, the nonlinear regression app will perform better if we can limit the coefficient values in this way.

The optimization app is activated as follows: Select the *DATA* tab at the top of the excel window, and then select the solver icon in the analysis tile. That should look something like the figure at right. The question mark and arrow are the icon for the solver. Selecting this icon creates a window where we specify the inputs to the optimization app. See the image at the top of the next page.

	A	B	C	D	E	
1	y	x	yhat = $c / (1 + ae^{-bx})$	$(y - yhat)^2$	sum of sqr errors	
2		1	0.00	0.59375	0.165039	3.110712
3		2	1.00	2.185439159	0.034388	
4		4	2.00	5.438497384	2.069275	
5		8	3.00	8.143078532	0.020471	
6		8.5	4.00	9.159440263	0.434861	
7		8.8	5.00	9.421833963	0.386677	
8						
9	a	b	c			
10	15	1.5	9.5			



We have entered E2 as the target cell (*ie.* the value to be minimized), selected the Min option, and entered A10, B10, and C10 as the cells containing the parameters to be determined. Constraints have been added for each parameter, restricting its values to an interval. The app is executed by clicking on Solve.



That writes the optimized values of the parameters into cells A10, B10, and C10, and the optimal value of the target is then shown in E2. At the same time, a window appears, asking whether we wish to accept the results, or restore the spreadsheet to its state prior to running the app. For the purpose of this example, we accepted the results of the app. The results appear in the image below. The results of the optimization agree with those found earlier to five or six decimal places.

	A	B	C	D	E	
			yhat =		sum of	
1	y	x	$c/(1+ae^{-bx})$	$(y - \text{yhat})^2$	sqr errors	
2		1	0.00	0.519301527	0.231071	1.088392
3		2	1.00	1.810384268	0.035954	
4		4	2.00	4.595159208	0.354214	
5		8	3.00	7.351677167	0.420322	
6		8.5	4.00	8.610768519	0.01227	
7		8.8	5.00	8.985903557	0.03456	
8						
9	a	b	c			
10	16.55027	1.411588	9.113883594			

This approach to fitting a function to a set of data is more widely applicable than linear least-squares method, which cannot be applied for logistic functions. Here we minimized the sum of squared errors, but the method works in a similar way with other error measures. For example, if the target cell is computed as the sum of the absolute errors, then the optimization will minimize that value.

On the other hand, in working with this method we have found that the results it produces depend on the initial values of the parameters entered. In particular, running the app twice

sometimes produces better results than running it once. That is, if the “optimal” parameter values from the first run are used as the initial parameter values for the second run, a lower minimum is sometimes obtained, with modified “optimal” parameters. We have always found the optimized results to be better than what we have obtained with informal trial and error. But for serious optimization tasks, it would be prudent to consider the possible implications of results that are not necessarily best possible.