

## Fitting Your Data to Find a “Best Fit” Equation

Due to many factors, your data will *never* perfectly match what you expect. There are three main ways that Excel can help you determine an equation that closely matches your data. But before you can use any of them, you must already have in mind a general idea of the form that the “best fit” equation should have (i.e., is it linear, parabolic, etc.?)

### Trendline

The first method is called “Trendline”. This feature is very nice if you want to quickly see the “best fit” line to your plot; however, it has the disadvantage that it does not provide uncertainties. Also, it gives you a fairly limited choice of what kind of fitted curve you want to use (linear, polynomial, exponential, logarithmic, or power law). To use Trendline, right-click on any data point on your plot, and choose “add trendline” from the menu that appears. You first specify the form of the fitted curve you want. Excel will give you any equation you ask for, even if you ask for something that doesn’t make sense. So, make sure that choose the appropriate equation based on your data and theory. Before you finish, click on the “options” tab, and then the “display equation on chart” button to have Excel write the resulting equation in a text box on your plot. *Be cautious:* Excel will incorrectly call your variables “x” and “y”, even though it is unlikely that these are the actual names for your axes.

### Linest

The second method to determine a “best-fit” equation uses an Excel function called “linest”. This tool will provide not only the equation you want, but the uncertainties as well. But, it can only be used to find polynomial fits to the data. We will only use it for linear and quadratic equations, which are both types of polynomial. To use this function, you must first select an unused region of your spreadsheet where you want Excel to place the answers. Since a straight line is described by two quantities (a *slope* and an *intercept*), you will need two columns for linear equations. Quadratic equations require 3 columns. In either case, you must select two rows (one for the value, and one for the uncertainty).

	A	B	C
1			
2	#	t (s)	x(cm)
3	0	0.0000	4.0
4	1	0.0167	4.3
5	2	0.0333	5.9
6	3	0.0500	8.3
7	4	0.0667	11.6
8			
9		slope	intercept
10	value	115.2000	2.9800
11	uncertainty	19.9720	0.8154
12			

You enter the function as a formula **while all 4 (or 6) of these cells are still highlighted**. For linear equations, the form of the function is:

```
=linest(y_axis_data, x_axis_data,1,1)
```

Note that the vertical axis data comes first, not the horizontal axis data. For the data shown in the above figure, this would be:

```
=linest(C3:C7, B3:B7,1,1)
```

*After it is entered, you may not press **enter!*** Instead, you must press CTRL–SHIFT–ENTER, all at one time. This tells Excel that you want 4 (or 6) total results, not just 1. For the example shown, and using correct significant digits, the slope is  $(11.5 \pm 2.0) \times 10^1$  cm/s, and the intercept is  $(2.98 \pm 0.82)$  cm.

For quadratic equations, the format is:

```
=linest(y_axis_data, x_axis_data^{1,2},1,1)
```

Notice the curly brackets. For the data shown in the example, this would be:

```
=linest(C3:C7, B3:B7^{1,2},1,1)
```

Again, you must use CTRL-SHIFT-ENTER. The results are listed in the order *ABC* where the equation is:  $Ax^2 + Bx + C$ . You can also compare these results to the results of trendline to make sure you know which value is which.

## Analysis Using Solver

Sometimes, you'll need to find best-fit functions that are not polynomials. Although trendline has some non-polynomial options, it doesn't have many. Linest has only polynomial options, so it can't help us at all in this circumstance. Excel includes a tool called "Solver" that can help with this kind of problem. Unlike linest, which a monkey can learn to use, Solver requires a competent user. Here's the big idea: Solver works by making a lot of guesses for the unknowns until it stumbles across a guess that makes the equation match the original data very closely. To practice using it, download the "solver demo" worksheet from my outbox. This worksheet already has a lot of stuff in it, but typically you would only start with your raw data (which I've put in columns B and C, and highlighted in light blue).

Suppose that I have already done a theoretical analysis of this problem, which tells me to expect the data to follow the form  $f = A \sin(\omega(x-x_0))$ . The unknowns are  $A$ ,  $\omega$  and  $x_0$ . Our first step is to make a cell for every unknown. I've done this in column H (green cells). Currently, all of the unknowns are set to zero, which is obviously not right!

Next, we have to use the basic equation to compute the results for your guess. I've used column D for this purpose; click on cell D10 (for example) and examine the equation. Note that the yellow cell (D4) is there for *your* reference only; Excel does not use it. You'll see several "\$" characters in the equations. Although they are not necessary, they are very helpful. This character means "don't change the following number when dragging equations into new cells". That way, you can type your basic equation only once (in cell E9, for example), and then drag this equation down through the entire column, and Excel will put the correct equation into each cell. Compare the result to what happens when the "\$" character is missing! Make absolutely sure that you understand the *purpose* of every character in the equations in column D.

The next step is to type in some guesses for our three unknowns before we even try to use the Solver tool. If you guess badly, there is a chance that Solver won't work at all. To evaluate whether a guess is good, we should compare the guess to the original data. The plot shows the original data as points, and the fitted line based on your guesses (so far) as a pink line. Try changing the values in the green boxes until you get something that is in the right neighborhood. From inspection, the amplitude  $A$  looks it should be near 4.0, and by trial and error, the frequency  $\omega$  looks like it should be near 11.0. Once those two are chosen, it looks like an  $x_0$  of 0.1 moves the plot to the right a more or less sufficient amount.

We should enter those three values into the appropriate (green) cells. Now, are these the *best* values? Probably not. But, we still haven't even actually used Solver yet. To find the best values, we'll use the "Method of Least Squares". We'll compute the square of the difference between each original data point and the computed value (done in column E), and then add them up (done in cell E30). The bigger this value, the worse our guess is. If this cell is zero, then the guess is perfect (which can only happen if the data itself is perfect). So, we want to minimize this value. The differences are squared because I want all the errors to be positive, instead of some positive and some negative, which could accidentally add up to zero and make the data seem perfect. This method is the same method used by linest.

Now, we finally get to actually use the "Solver" tool. Go to the Tools/Solver... menu. The "target" cell is E30. The "By changing" cells are our guesses (H3, H4, and H5). We want to *minimize* cell E30, so we'll choose the "min" option. Now, when you press the "Solve" button, Solver will guess values of  $A$ ,  $\omega$  and  $x_0$ , thousands of times. Each time, it will look at cell E30, and it will remember the combination of  $A$ ,  $\omega$  and  $x_0$  that results in the smallest value of cell E30. When it thinks it can't make a better guess anymore, it will stop.

If your initial guesses weren't very good, Solver might not give a good result at all. To check the result, look at the two plots to see if they are similar. Some of the original data should be above the best-fit curve (pink), and some below it. Now that you know the answer ( $A = 4.002$ ,  $\omega = 10.911$ , and  $x_0 = 0.0363$ ), try it again with worse guesses (maybe 1, 1, and 0). Notice that solver gives you garbage (including an impossible negative amplitude). You *must* have reasonably good starting guesses. Within the Solver menu, you can also specify restrictions on the guesses. For example, you might specify that  $H3 \geq 0$ . But, this kind of tinkering doesn't help as much and is harder to do than simply starting with good guesses.

## Uncertainty Using Solver

Did you ever wonder how *linest* actually computes uncertainties? We're going to learn part of it today, and see how to compute uncertainties for results obtained from using Solver. Download the "SolverUncert.xls" spreadsheet from my outbox. The basic idea is this: you use Solver twice. The first pass tells us the best-fit parameters as we've already leaned. On the second pass, we look at how much the answer changes with imperfect guesses and use that information to determine the uncertainty.

We have some measured data (columns B & C, conveniently colored orange for you), and we want to find the best-fit equation of the form shown in cell H3 ( $Ax^m + B$ ; cell H3 itself isn't used by Excel for anything; it is there for your benefit only). Columns A through K are things you already know how to do using Solver. Notice that Column F depends on Column E. While we could have done these two operations in one step (as we did when originally learning how to use Solver), we're going to need both columns E and F later on.

First, use solver to compute the best values for  $A$ ,  $B$ , and  $m$ , using initial guesses of 0.3, 200, and 3 in cells I6 – I8. The correct answers are shown with fewer sig-figs in column Q. Solver might quit a little bit too early, so you might need to run it more than once to get the best values. Let's look a little closer at what's already happened so far:

Notice the "chi squared" ( $\chi^2$ ) cell (I16): 
$$\frac{\sum error^2}{stdev(error)^2} \frac{N-n}{N-1}$$

Basically,  $\chi^2$  is a kind of measure of how far our guesses are from the "best possible" values. It is calculated from the total number of data points ( $N$ ), the number of parameters we are fitting ( $n$ ), the standard deviation of the difference between the data and the computation, and the sum of the squares of these differences (which, as a reminder, is our Solver "target"). When the guesses are indeed the "best" fit,  $\chi^2 = N - n$ . So, as a reassurance, we notice that  $\chi^2 = 19$ , and  $N - n = 19$ , too.

But, when the fit is worse,  $\chi^2$  gets bigger. In particular, it can be shown by people smarter than me that if:

- you change **one** of the free parameters a little, and then
- you run Solver a second time by varying only the *other* free parameters, while
- calculating  $\chi^2$  using the standard deviation of the first "best" pass but using the error squared of all the new values, then
- if  $\chi^2$  increases by exactly 1.0, then
- it must be that in part (a) you changed the free parameter by exactly one standard deviation.

Wow, that seems confusing. The big idea is that since the uncertainty is the same as one standard deviation, this method can be used to find the uncertainty of our guesses. Let's work through an example where we find the uncertainty of  $m$ . So far, all we know is that  $m = 2.965$ .

Columns L through N use the same equation ( $Ax^m + B$ ) with new guesses (cells Q6-Q8). This time, I'll let solver change the guesses for  $A$  and  $B$ , but I'll be in control of changing  $m$ . I'll let Solver start with the results of the first pass for initial guesses for  $A$  and  $B$ . For my first guess of  $m$ , I'll try 3.0. Now, we can run solver and see what happens. Be sure that Solver changes only  $A$  and  $B$ , but not  $m$ !

Take a very, very close look at the formula in cell Q13 (the new  $\chi^2$ ). It uses the old  $N$ , the old  $n$ , and the old standard deviation of the differences, but uses the new  $\Sigma$  difference<sup>2</sup>. If the new chi  $\chi^2$  changes from 19.0 to 20.0, then you guessed correctly. Unfortunately, I notice that the new  $\chi^2$  is only 19.73, so I guess I didn't change  $m$  enough. So, I have to try another  $m$ , *and then run solver again*. I keep doing this until the new  $\chi^2$  is 20.0 **after another run of Solver**. Your final guess for  $m$  should be near 3.006 (or 2.925) when  $\chi^2$  reaches 20.

Since the best value of  $m$  was 2.965, we have determined that the uncertainty on  $m$  is either:

$$\Delta m = |2.965 - 3.006| = 0.041, \text{ or}$$

So, in appropriate sig-figs, my result for  $m$  is:  $m = 2.965 \pm 0.041$ .