

## Chapter III-8 — Curve Fitting

Most curve fits can be accomplished using the Curve Fitting dialog. If you need to do many fits using the same fit function fitting to numerous data sets you will probably want to write a procedure in Igor's programming language to do the job.

The facility for creating a user-defined fitting function using the Curve Fitting dialog will handle most common cases, but is probably not the best way to create a very complex fitting function. In such cases, you will need to write a fitting function in a procedure window. This is described later under **User-Defined Fitting Functions** on page III-250.

Some very complicated user-defined fitting functions may not work well with the Curve Fitting dialog. In some cases, you may need to write the fitting function in the Procedure window, and then use the dialog to set up and execute the fit. In other cases it may be necessary to enter the operation manually using either a user procedure or by typing on the command line. These cases should be quite rare.

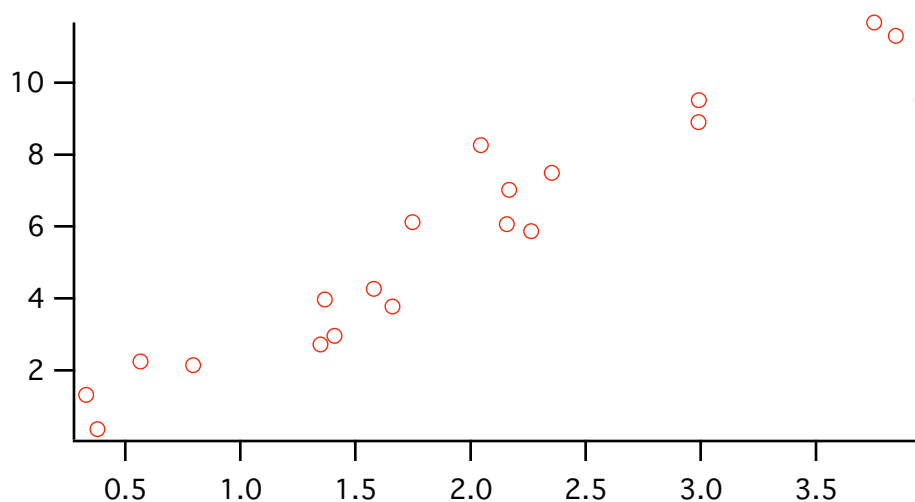
### A Simple Case — Fitting to a Built-In Function: Line Fit

To get started, we will cover fitting to a simple built-in fit: a line fit. You may have a theoretical reason to believe that your data should be described by the function  $y = ax + b$ . You may simply have an empirical observation that the data appear to fall along a line and you now want to characterize this line. It's better if you have a theoretical justification, but we're not all that lucky.

The Curve Fitting dialog is organized into four tabs. Each tab contains controls for some aspect of the fitting operation. Simple fits to built-in functions using default options will require only the Function and Data tab.

We will go through the steps necessary to fit a line to data displayed in a graph. Other built-in functions work much the same way.

You might have data displayed in a graph like this:



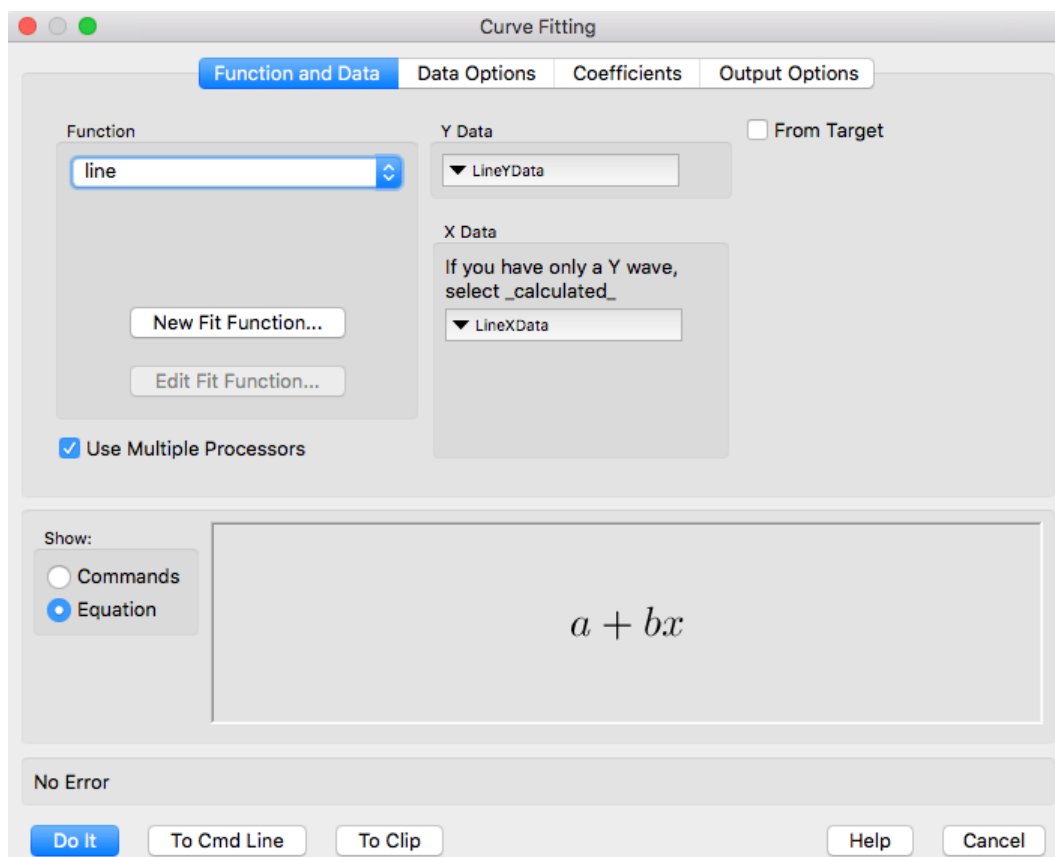
Now you wish to find the best-fitting line for this data. The following commands will make a graph like this one. The SetRandomSeed command is used so that the "random" scatter produced by the enoise function will be the same as shown above. If you would like to perform the actions yourself as you read the manual, you can make the data shown here and the graph by typing these commands on the command line:

```
Make/N=20/D LineYData, LineXData
SetRandomSeed 0.5 // So the example always makes the same "random" numbers
LineXData = enoise(2)+2 // enoise makes random numbers
LineYData = LineXData*3+gnoise(1) // so does gnoise
Display LineYData vs LineXData
ModifyGraph mode=3,marker=8
```

The first line makes two waves to receive our “data”. The second line sets the seed for Igor's pseudo-random number generators, resulting in reproducible noise. The third line fills the X wave with uniformly-distributed random numbers in the range of zero to four. The fourth line fills the Y wave with data that falls on a line having a slope of three and passing through the origin, with some normally-distributed noise added. The final two lines make the graph and set the display to markers mode with open circles as the marker.

### Choosing the Function and Data

You display the Curve Fitting dialog by choosing Curve Fitting from the Analysis menu. If you have not used the dialog yet, it looks like this, with the Function and Data tab showing:



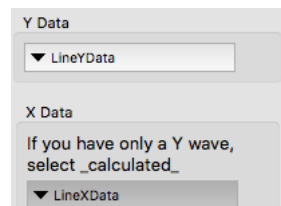
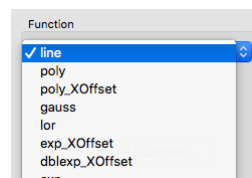
The first step in doing a curve fit is to choose a fit function. We are doing a simple line fit, so pop up the Function menu and choose “line”.

Select the Y data from the Y Data menu. If you have waveform data, be sure that the X data menu has “\_Calculated\_” selected.

If you have separate X and Y data waves, you must select the X wave in the X Data menu. Only waves having the same number of data points as the Y wave are shown in this menu. A mismatch in the number of points is usually the problem if you don't see your X wave in the menu.

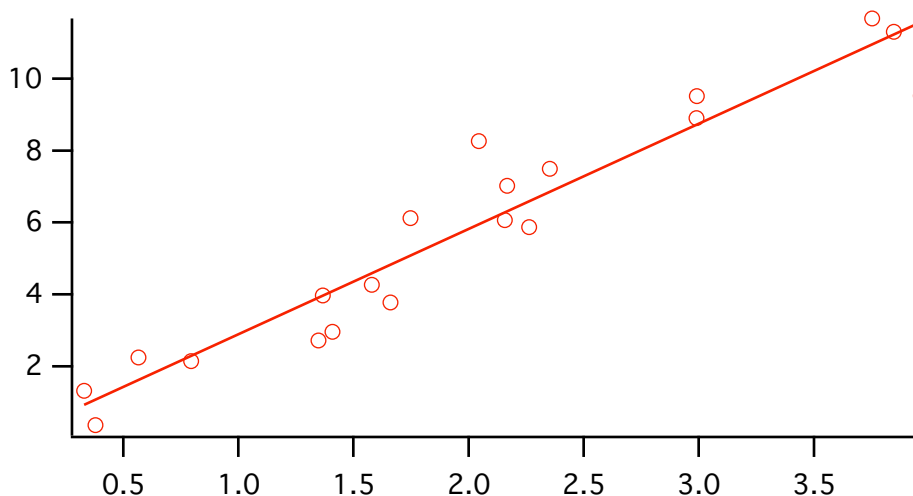
For the line fit example, we select LineYData from the Y Data menu, and LineXData from the X Data menu.

If you have a large number of waves in your experiment, it may be easier if you select the From Target checkbox. When it is selected only waves from the top graph or table are shown in the Y and X wave menus, and an attempt is made to select wave pairs used by a trace on the graph.



## Chapter III-8 — Curve Fitting

At this point, everything is set up to do the fit. For this simple case it is not necessary to visit the other tabs in the dialog. When you click Do It, the fit proceeds. The line fit example graph winds up looking like this:



In addition to the model line shown on the graph, various kinds of information appears in the history area:

```
•CurveFit line LineYData /X=LineXData /D
  fit_LineYData= W_coef[0]+W_coef[1]*x
  W_coef={-0.037971,2.9298}
  V_chisq= 18.25; V_npnts= 20; V_numNaNs= 0; V_numINFs= 0;
  V_startRow= 0;V_endRow= 19;V_q= 1;V_Rab= -0.879789;
  V_Pr= 0.956769;V_r2= 0.915408;
  W_sigma={0.474,0.21}
  Coefficient values ± one standard deviation
    a  =-0.037971 ± 0.474
    b  =2.9298 ± 0.21
```

Command line generated by the dialog.

This line can be copied and used to reevaluate the model curve.

Fit coefficients as a wave.

Standard deviations of the Fit coefficients as a wave.

Coefficient values in a list using the names shown in the dialog.

### Two Useful Additions: Holding a Coefficient and Generating Residuals

Well, you've done the fit and looked at the graph and you decide that you have reason to believe that the line should go through the origin. Because of the scatter in the measured Y values, the fit line misses the origin. The solution is to do the fit again, but with the Y intercept coefficient held at a value of zero.

You might also want to display the residuals as a visual check of the fit.

Bring up the dialog again. The dialog remembers the settings you used last time, so the line fit function is already chosen in the Function menu, and your data waves are selected in the Y Data and X Data menus.

Select the Coefficients tab. Each of the coefficients has a row in the Coefficients list:

Curve Fitting

Function and Data   Data Options   **Coefficients**   Output Options

Coefficient Wave:    Graph Now   Auto guess

Constraints:

Coef Name	Initial Guess	Hold?	Lower Constraint	Upper Constraint
a	0	<input checked="" type="checkbox"/>		
b		<input type="checkbox"/>		

Show:  
☐ Commands  
☒ Equation

$$a + bx$$

No Error

Do it   To Cmd Line   To Clip   Help   Cancel

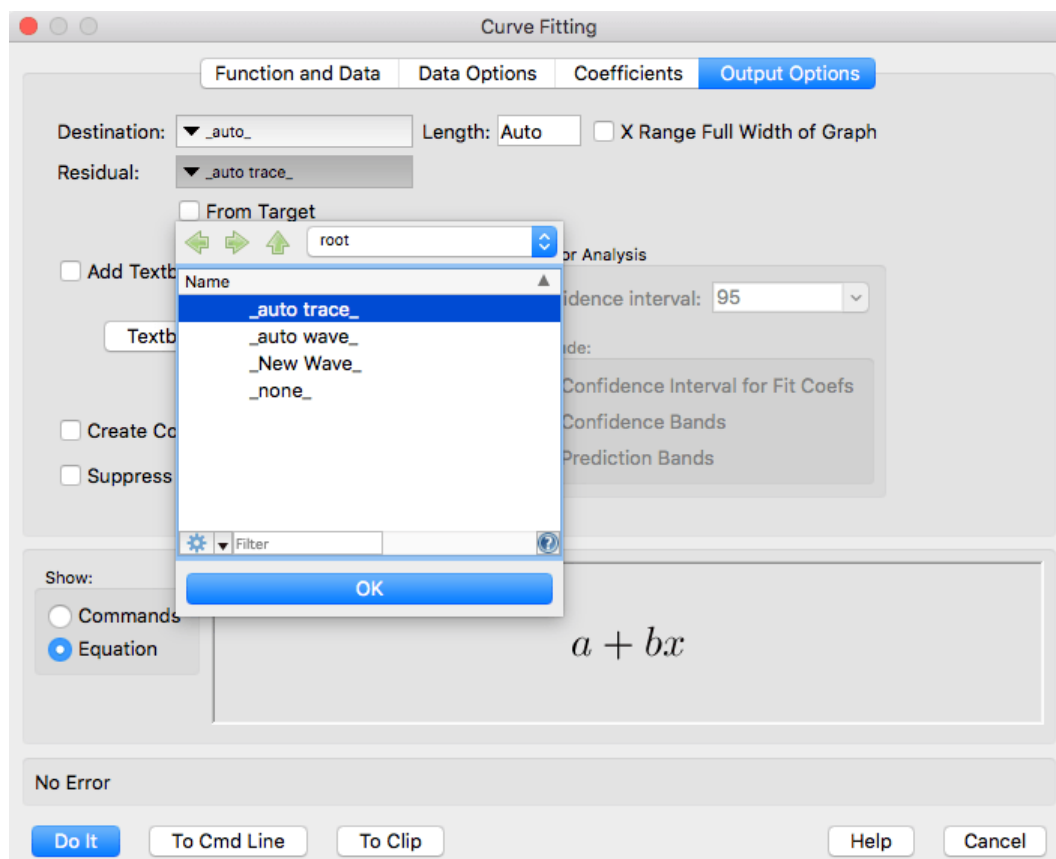
Click the checkbox in the column labeled “Hold?” to hold the value of that coefficient.

To specify a coefficient value, fill in the corresponding box in the Initial Guess column. Until you select the Hold box the initial guess box is not available because built-in fits don’t require initial guesses.

To fill in a value, click in the box. You can now type a value. When you have finished, press Enter (*Windows*) or Return (*Macintosh*) to exit editing mode for that box.

## Chapter III-8 — Curve Fitting

Now we want to calculate the fit residuals and add them to the graph. Click the Output Options tab and choose `_auto trace_` from the Residual menu:



There are a number of options for the residual. We chose `_auto trace_` to calculate the residual and add it to the graph. You may not always want the residuals added to your graph; choose `_auto wave_` to automatically calculate the residuals but *not* display them on your graph. Both `_auto trace_` and `_auto wave_` create a wave with the same name as your Y wave with “Res\_” prefixed to the name. Choosing `_New Wave_` generates commands to make a new wave with your choice of name to fill with residuals. It is not added to your graph.