Model Results for Multivariate Fitting

As with fitting to a function of one independent variable, Igor creates waves containing the model output and residuals automatically. This is done if you choose _auto_ for the destination and _auto trace_ for the residual on the Output Options tab. There are some differences in detail, however.

By default, the model curve for a univariate fit is a smooth curve having 200 points to display the model fit. This depends on being able to sensibly interpolate between successive values of the independent variable. Multicolumn independent variables, on the other hand, probably don't have successive values of all the independent variables in sequential order, so it is not possible to do this. Consequently, it calculates a model point for each point in the dependent variable data wave. If the data wave is displayed as a simple 1D trace in the top graph window, the fit results will be appended to the graph.

Residuals are always calculated on a point-for-point basis, so calculating residuals for a multicolumn multivariate fit is just like a univariate fit.

Displaying results of fitting to a multidimensional wave is more problematic. If the dependent variable has three or more dimensions, it is not easy to display the results. The model and residual waves will be created and the results calculated but not displayed. You can make a variety of 3D plots using Gizmo: just choose the appropriate plot type from the Windows→New→3D Plots menu.

Fits to a 2D matrix wave are displayed on the top graph if the Y Data wave is displayed there as either an image or a contour. The model results are plotted as a contour regardless of whether the data are displayed as an image or a contour. Model results contoured on top of data displayed as an image can be a very powerful visualization technique.

Residuals are displayed in the same manner as the data in a separate, automatically-created graph window. The window size will be the same as the window displaying the data.

Time Required to Update the Display

Because contours and images can take quite a while to redraw, the time to update the display at every iteration may cause fits to contour or image data to be very slow. To suppress the updates, click the Suppress Screen Updates checkbox on the Output Options tab.

Multivariate Fitting Examples

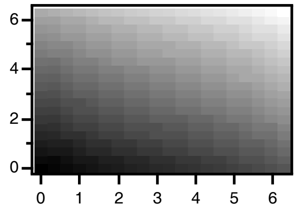
Here are two examples of fitting to a multivariate function — the first uses the built-in poly2D function to fit a plane to a gridded dataset to remove a planar trend from the data. The second defines a simplified 2D gaussian function and uses it to define the location of a peak in XY space using random XYZ data.

Example One — Remove Planar Trend Using Poly2D

Here is an example in which a matrix is filled with a two-dimensional sinusoid with a planar trend that overwhelms the sinusoid. The example shows how you might fit a plane to the data to remove the trend. First, make the data matrix, fill it with values, and display the matrix as an image:

```
Make/O/N=(20,20) MatrixWave
SetScale/I x 0,2*pi,MatrixWave
SetScale/I y 0,2*pi,MatrixWave
MatrixWave = sin(x) + sin(y) + 5*x + 10*y
Display;AppendImage MatrixWave
```

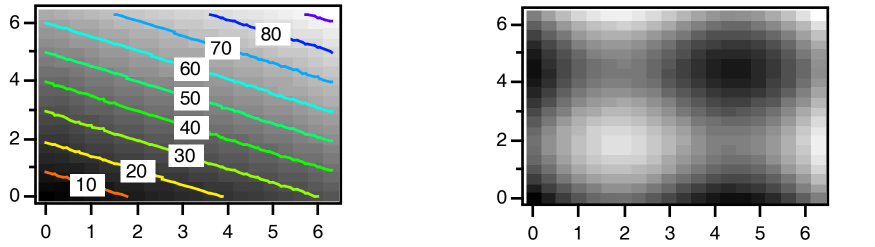
These commands make a graph displaying an image like the one that follows. Note the gradient from the lower left to the upper right:



We are ready to do the fit.

1. Choose Curve Fitting from the Analysis menu to bring up the Curve Fitting dialog.
2. If you have not already done so, choose Show Multivariate Functions from the Function menu.
3. Choose Poly2D from the Function menu.
4. Make sure the 2D Polynomial Order is set to 1.
5. Choose MatrixWave from the Y Data menu.
6. Click the Output Options tab.
7. Choose _auto trace_ from the Residual menu.
8. Click Do It.

The result is the original graph with a contour plot showing the fit to the data, and a new graph of the residuals, showing the sinusoidal signal left over from the fit:



Original graph

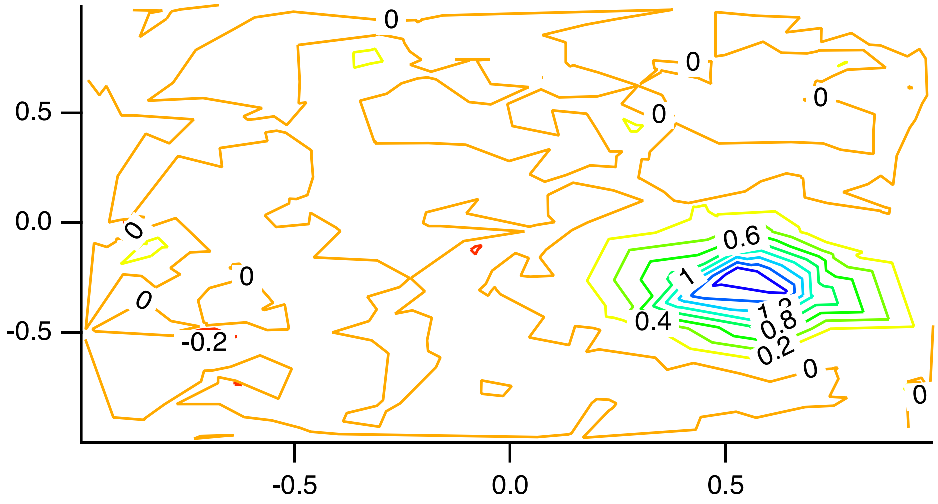
Residual graph showing sinusoidal signal

Similarly, you can use the ImageRemoveBackground operation, which provides a one-step operation to do the same fit. With an image plot as the top window, you will find Remove Background in the Image menu.

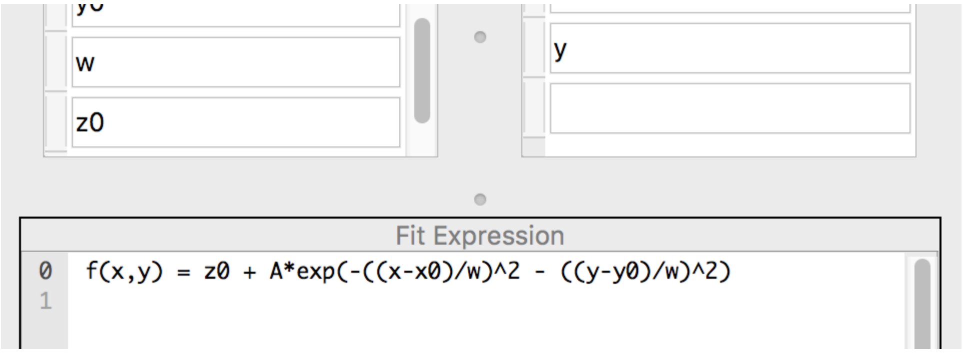
Example Two — User-Defined Simplified 2D Gaussian Fit

In this example, we have data defining a spot which we wish to fit with a 2D Gaussian to find the center of the spot. For some reason this data is in the form of XYZ triplets with random X and Y coordinates. These commands will generate the example data:

```
Make/N=300 SpotXData, SpotYData, SpotZData
SetRandomSeed 0.5
SpotXData = enoise(1)
SpotYData = enoise(1)
// make a gaussian centered at {0.55, -0.3}
SpotZData = 2*exp(-((SpotXData-.55)/.2)^2 -((SpotYData+.3)/.2)^2)+gnoise(.1)
Display; AppendXYZContour SpotZData vs {SpotXData,SpotYData}
```



Now bring up the Curve Fitting dialog and click the New Fit Function button so that you can enter your user-defined fit function. We have reason to believe that the spot is circular so the gaussian can use the same width in the X and Y directions, and there is no need for the cross-correlation term. Thus, the new function has a z0 coefficient for the baseline offset, A for amplitude, x0 and y0 for the X and Y location and w for width. Here is what it looks like in the New Fit Function dialog:



```
f(x,y) = z0 + A*exp(-((x-x0)/w)^2 - ((y-y0)/w)^2)
```

Click Save Fit Function Now to save the function in the Procedure window and return to the Curve Fitting dialog. The new function is selected in the Function menu automatically.

To perform the fit choose:

1. SpotZData in the Y Data menu.
2. SpotXData in the X Data menu.
3. SpotYData in the X Data menu.

At this point, the data selection area of the Function and Data tab looks like this: