In an ODR fit, even when the fitting function is linear in the coefficients, the fitting equations themselves introduce a nonlinearity. Consequently, the error estimates from an ODR fit are always an approximation. See the **Curve Fitting References** on page III-267 for detailed information.

ODR Fitting Examples

A simple example: a line fit with no weighting. If you run these commands, the call to SetRandomSeed will make your "measurement error" (provided by **gnoise** function on page V-323) the same as the example shown here:

```
SetRandomSeed 0.5       // so that the "random" data will always be the same...
Make/N=10 YLineData, YLineXData
YLineXData = p+gnoise(1)      // gnoise simulates error in X values
YLineData = p+gnoise(1)       // gnoise simulates error in Y values
// make a nice graph with errors bars showing standard deviation errors
Display YLineData vs YLineXData
ModifyGraph mode=3,marker=8
ErrorBars YLineData XY,const=1,const=1
```

Now we're ready to perform a line fit to the data. First, a standard curve fit:

```
CurveFit line, YLineData/X=YLineXData/D
```

This command results in the following history report:

```
fit_YLineData= W_coef[0]+W_coef[1]*x
W_coef={1.3711,0.78289}
V_chisq= 15.413; V_npnts= 10; V_numNaNs= 0; V_numINFs= 0;
V_startRow= 0; V_endRow= 9; V_q= 1; V_Rab= -0.797202; V_Pr= 0.889708;
V_r2= 0.791581;
W_sigma={0.727,0.142}
Coefficient values ± one standard deviation
    a = 1.3711 ± 0.727
    b = 0.78289 ± 0.142
```

Next, we will use /ODR=2 to request orthogonal distance fitting:

```
CurveFit/ODR=2 line, YLineData/X=YLineXData/D
```

which gives this result:

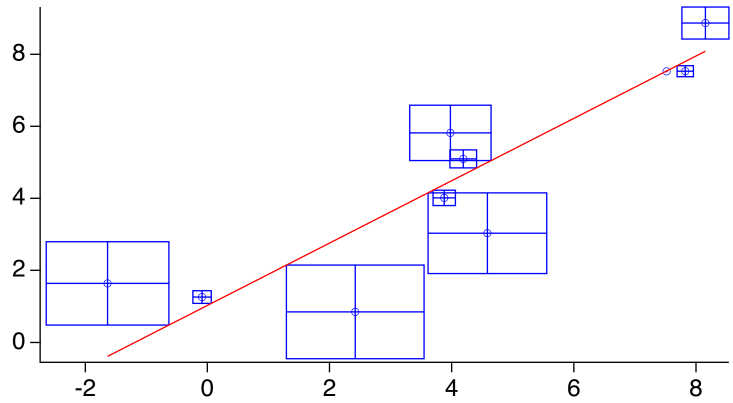
```
Fit converged properly
fit_YLineData= W_coef[0]+W_coef[1]*x
W_coef={1.0311,0.86618}
V_chisq= 9.18468; V_npnts= 10; V_numNaNs= 0; V_numINFs= 0;
V_startRow= 0; V_endRow= 9;
W_sigma={0.753,0.148}
Coefficient values ± one standard deviation
    a   = 1.0311 ± 0.753
    b   = 0.86618 ± 0.148
```

Add output of the X adjustments and Y residuals:

```
Duplicate/O YLineData, YLineDataXRes, YLineDataYRes
CurveFit/ODR=2 line, YLineData/X=YLineXData/D/XR=YLineDataXRes/R=YLineDataYRes
```

And a graph that uses error bars to show the residuals:

```
Display YLineData vs YLineXData
ModifyGraph mode=3,marker=8
AppendToGraph fit_YLineData
ModifyGraph rgb(YLineData)=(0,0,65535)
ErrorBars YLineData BOX,wave=(YLineDataXRes,YLineDataXRes),
wave=(YLineDataYRes,YLineDataYRes)
```



The boxes on this graph do not show error estimates, they show the residuals from the fit. That is, the differences between the data and the fit model. Because this is an ODR fit, there are residuals in both X and Y; error bars are the most convenient way to show this. Note that one corner of each box touches the model line.

In the next example, we do an exponential fit in which the Y values and errors are small compared to the X values and errors. The curve fit history report has been edited to include just the output of the solution. First, fake data and a graph:

```
SetRandomSeed 0.5   // so that the "random" data will always be the same…
Make/D/O/N=20 expYdata, expXdata
expYdata = 1e-6*exp(-p/2)+gnoise(1e-7)
expXdata = p+gnoise(1)
display expYdata vs expXdata
ModifyGraph mode=3,marker=8
```

A regular exponential fit:

```
CurveFit exp, expYdata/X=expXdata/D

Coefficient values ± one standard deviation
    y0        =-1.0805e-08 ± 4.04e-08
    A         =7.0438e-07 ± 9.37e-08
    invTau    =0.38692 ± 0.116
```

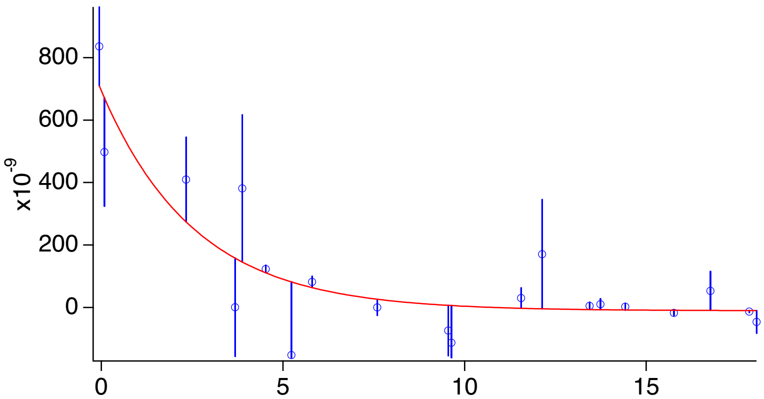
An ODR fit with no weighting, with X and Y residuals:

```
Duplicate/O expYdata, expYdataResY, expYdataResX
expYdataResY=0
expYdataResX=0
CurveFit/ODR=2 exp, expYdata/X=expXdata/D/R=expYdataResY/XR=expYdataResX

Coefficient values ± one standard deviation
    y0     =-1.0541e-08 ± 4.03e-08
    A      =7.0443e-07 ± 9.37e-08
    invTau =0.38832 ± 0.116
```

And a graph:

```
Display /W=(137,197,532,405) expYdata vs expXdata
AppendToGraph fit_expYdata
ModifyGraph mode(expYdata)=3
ModifyGraph marker(expYdata)=8
ModifyGraph lSize(expYdata)=2
ModifyGraph rgb(expYdata)=(0,0,65535)
ErrorBars expYdata
BOX,wave=(expYdataResX,expYdataResX),wave=(expYdataResY,expYdataResY)
```

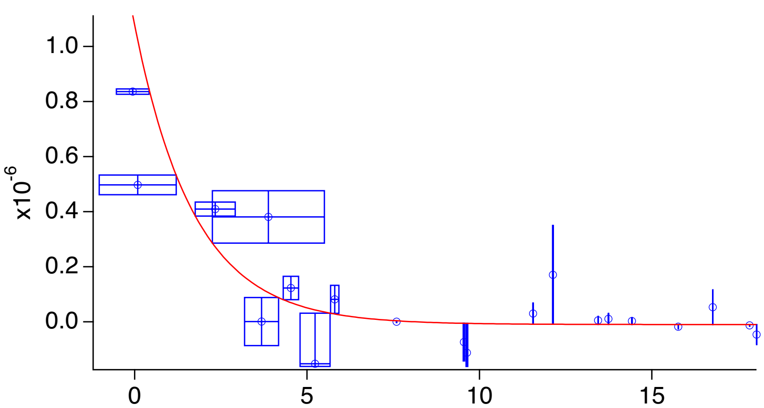


Because the Y values are very small compared to the X values, and we didn't use weighting to reflect smaller errors in Y, the residual boxes are tall and skinny. If the vertical graph scale were the same as the horizontal scale, the boxes would be approximately square. The data line would be very nearly a horizonal line. One way to understand this is to remember that the ODR method is essentially geometric. In this example, the vertical scale on the graph has been expanded very greatly, but the ODR method works in an unexpanded scale where the perpendicular lines to the fit curve are very nearly exactly vertical.

Now we can add appropriate weighting. It's easy to decide on the correct weighting since we added "measurement error" using gnoise():

```
Duplicate/O expYdata, expYdataWY
expYdataWY=1e-7
Duplicate/O expYdata, expYdataWX
expYdataWX=1
// Caution: Next command wrapped to fit on page.
CurveFit/ODR=2 exp, expYdata/X=expXdata/D/R=expYdataResY/XR =expYdataResX/W=expYdataWY
/XW=expYdataWX/I=1

Coefficient values ± one standard deviation
    y0     =-9.8498e-09 ± 3e-08
    A      =1.0859e-06 ± 5.39e-07
    invTau =0.57731 ± 0.248
```



One way to think about the weighting waves for ODR fitting is that they provide geometric scaling. In this example, the vertical dimension is about 10^7 times smaller than the horizontal dimension. When the vertical dimension is scaled by the weighting waves, the dimensions are similar and the perpendicular distances from the fit curve to the data points are no longer merely vertical.