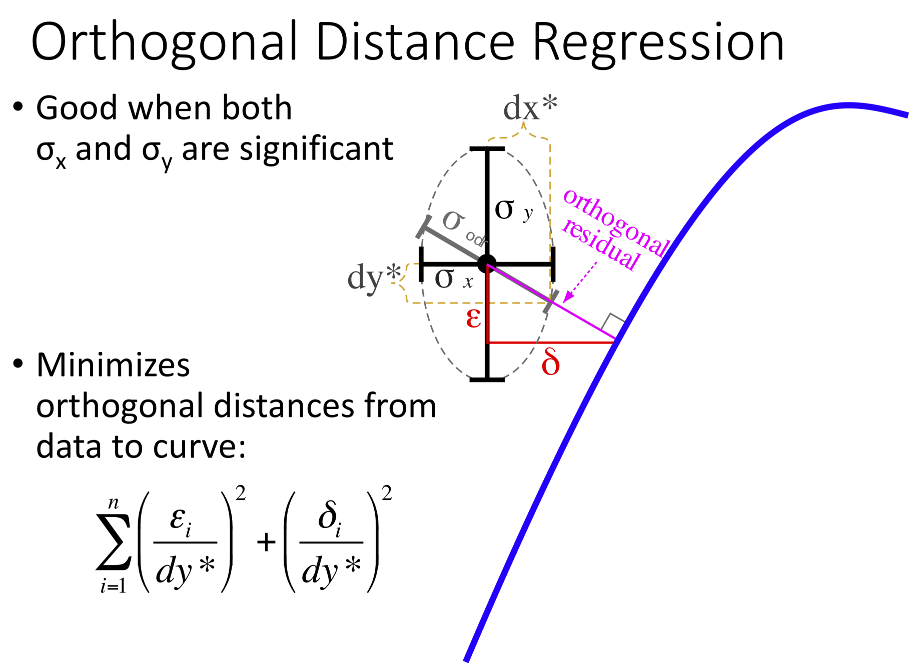
Using ODR Fitter

David Bailey

2018-08-14

ODR Fitter

The program odr_fit_extended.py uses Orthogonal Distance Regression to fit a mathematical model to two-dimensional data with uncertainties in both x and y. Both random and systematic uncertainties are allowed, and the uncertainties do not have to be Gaussian. The program should run on any version of Python 2.7+ or 3+ that has numpy, scipy, and matplotlib installed.



Orthogonal Distance Regression

- Good when both σ_x and σ_y are significant
- Minimizes orthogonal distances from data to curve:

$$\sum_{i=1}^{n}\left(\frac{\varepsilon_i}{dy*}\right)^2+\left(\frac{\delta_i}{dy*}\right)^2$$

What you must do

Initialize fit

To do your own fit, modify the code section labelled `#### Mandatory User Definitions` to define:

1. `func` : the model you are trying to fit (i.e. y=func(x, p)),
    - x are the data values of "independent variable
    - p are the parameters that the fit optimizes,

2. **data_file** : the name of the data file read in by numpy.loadtxt,
3. **p_guess** : the initial guesses for the fit parameters.
   - If the guess if poor, the fit may not converge to the best values.

Data Format

Data filse normally have one data point on each line in the format x, σ_x, y, σ_y; the values can be separated by whitespace or commas. A fifth optional *scale* parameter for each point is discussed later in the section on Systematic scaling. Datafile lines beginning with "#" are treated as comments (with an exception for systematic uncertainties, see below). A simple data file might look like:

```
#x     dx   y     dy
11.9   0.1  26.1  0.1
8.9    0.1  9.3   0.1
6.3    0.1  2.9   0.1
14.0   0.2  42.0  0.2
...
```

Systematic uncertainties can be reported by adding lines beginning with `#  a_x, da_x, b_x, db_x =` or `#  a_x, da_x, b_x, db_x =`. Here is example of a full data file (gauss_ODR.txt):

```
## x calibration uncertainties (assuming x_true = a_x + b_x * x_measured)
#  a_x, da_x, b_x, db_x = 0, 0.05, 1.0, 0.006
## y calibration uncertainties (assuming y_true = a_y + b_y * y_measured)
#  a_y, da_y, b_y, db_y = 0, 0.05, 1.0, 0.006
#x     dx   y     dy
11.9   0.1  26.1  0.1
8.9    0.1  9.3   0.1
6.3    0.1  2.9   0.1
14.0   0.2  42.0  0.2
8.0    0.1  7.0   0.1
12.7   0.1  32.8  0.2
10.2   0.1  16.8  0.1
18.2   0.2  46.4  0.2
20.8   0.1  31.3  0.2
17.8   0.2  49.4  0.2
17.0   0.2  50.6  0.2
19.8   0.2  38.0  0.2
```

This assumes linear systematics without correlations between x and y, but the code could be modified if the systematics are non-linear.

Optional changes to code

You are, of course, welcome to modify the code in any way you wish, but a few more common changes can be modified in the **#### Mandatory User Definitions** code section:

1. **x_label, y_label** : labels for plot axes
2. **Number_of_MC_iterations** : number of Monte Carlo iterations. 100 is a good choice to start so the code will run quickly, but a > 1000 is better for final fits.
3. Decide whether you want some optional fitted curves on the output plot:
   - **plot_initial_guess** : fit function with initial guesses
   - **plot_MC_fits** : with Monte Carlo mean, median, or extreme parameters

4. The probability distributions associated with uncertainties that are used to generate Monte Carlo data values by "smearing" the observed or fit values. The default assumption is that the reported uncertainties (σ_x, σ_y) is the standard deviation of a gaussian distribution, but a uniform distribution (where $y \pm \sigma_y$ defines the full range of uncertainty of y) is sometimes appropriate (e.g. reading error of a digital instrument), and other distributions are sometimes more realistic.
   - **`smear_x_data`** : random x
   - **`smear_y_data`** : random y
   - **`smear_scale_x`** : systematic x scale
   - **`smear_scale_y`** : systematic y scale
   - **`smear_x_offset`** : systematic x offset
   - **`smear_y_offset`** : systematic y offset
5. Whether systematic uncertainties are correlated, e.g. if the same instrument is used to measure both x and y:
   - **`correlated_offset`** : any systematic offset error is the same for x & y
   - **`correlated_scale`** : any systematic scale error is be the same for x & y
6. The values of the uncertainties can be set in the code, e.g. if they appear to have been misevaluated.
   - **`x_sigma, y_sigma`** : random uncertainties
   - **`dx_offset, dx_scale, dy_offset, dy_scale`** : systematic uncertainties

Output from program

On the following pages is the text output from a Gaussian fit to the above data file. Both x and y were measured with the same instrument so any calibration scale error will be the same for both, so `correlated_scale=True` is set in the code. The output first identifies the data file, prints out the ODR model function, and lists any systematic uncertainties. It then lists the fitted parameters, their estimated uncertainties, the initial guesses, and the parameter correlation matrix. An quasi-χ^2 is calculated from the uncertainty normalized orthogonal distances of the data points from the fit line, and confidence level estimated assuming the quasi-χ^2 actually has a χ^2 distribution.

This initial ODR result ignores any systematic uncertainties. Small systematic offset uncertainties could be included in the fit via the covariance matrices of the input data, but scale (normalization) errors are more problematic[1], so the evaluation of the effect of systematics is left for Monte Carlo estimation.

The data are consistent with the Gaussian model. The median Monte Carlo uncertainties are significantly larger than the the original ODR uncertainties, reflecting the relatively large systematic uncertainties.

In general, the Monte Carlo parameter medians are more robust than the means which are sensitive to just a few (or even one) extreme values, so the median fit values should be taken as the best estimates of the parameters and their uncertainties. The code can easily be modified to print out 95% CI (or any other) uncertainties, or a histogram of the distribution of Monte Carlo parameters could be added.

Figure 1 shows the plot output of the above nice fit. The original ODR fit (in blue) and curves plotted with the mean and median Monte Carlo parameters all fit the data very closely. The cyan dotted curve shows the single Monte Carlo fit (out of 1001 in total) with the largest y value, so you can see an extreme example of the possible range of variance at the 1/1001 chance level.

[1] *On the use of the covariance matrix to fit correlated data*, G. D'Agostini, Nucl. Instrum. Methods Phys. Res. A 346 (1994) 306-311; doi:10.1016/0168-9002(94)90719-6. (also http://inspirehep.net/record/361137/files/desy93-175.pdf).

```
***********************************************************
               ORTHOGONAL DISTANCE REGRESSION
***********************************************************

Fitting 12 Data points from file gauss_ODR.txt to following Model function
  def gaussian(x,*p) :
    from numpy import exp
    mu     = p[0]    #   Peak position
    sigma  = p[1]    #   Peak width
    y_mu   = p[2]    #   Peak height scale
    return y_mu*exp(-(x-mu)**2/(2*sigma**2))

Systematics
   correlated_offset =  False
   correlated_scale  =  True
   x_true = 0.0±0.05 + (1.0±0.006)*x_measured
   y_true = 0.0±0.05 + (1.0±0.006)*y_measured

**** ODR has finished with: Sum of squares convergence

 Estimated parameters, uncertainties, and starting guesses
   p[0] =     16.643 +/-    0.05991          Guessed: 15
   p[1] =     4.2778 +/-   0.042811          Guessed: 5
   p[2] =     50.686 +/-    0.27017          Guessed: 10

 ODR Correlation Matrix
        [[ 1.          0.73326796 -0.34473356]
         [ 0.73326796  1.         -0.46575296]
         [-0.34473356 -0.46575296  1.        ]]

 Quasi Chi-Squared/dof   =    1.23347, Quasi CDF =   26.88333%

**** Running Monte Carlo CDF Estimator ****
    1001 successful MC simulations in 2.15797 seconds.

 Fraction of Monte Carlo quasi-chi-squared values larger than value for ODR fit:
    Monte Carlo CDF =   26.2%

 MC Fit parameters Average + Standard Deviation; Median and 68.3% interval
   p[0] =      16.6486 +/-     0.122837 ;      16.6485 +     0.121426 -     0.120198
   p[1] =      4.27964 +/-    0.0479119 ;      4.27985 +    0.0484368 -    0.0483026
   p[2] =      50.7412 +/-     0.394779 ;      50.7204 +      0.41728 -     0.357883

 Monte Carlo Correlation Matrix
        [[ 1.          0.70209012  0.55616856]
         [ 0.70209012  1.          0.25758756]
         [ 0.55616856  0.25758756  1.        ]]

 Check For Bias in MC Fit parameters
   (Monte Carlo median)/(fit value) - 1
      p[0] Bias : +0.0001806 +/- 0.0002921 (   0.62 SD)
      p[1] Bias : +0.0001247 +/- 0.0004407 (   0.28 SD)
      p[2] Bias : +0.0002257 +/- 0.0002149 (   1.05 SD)```
```

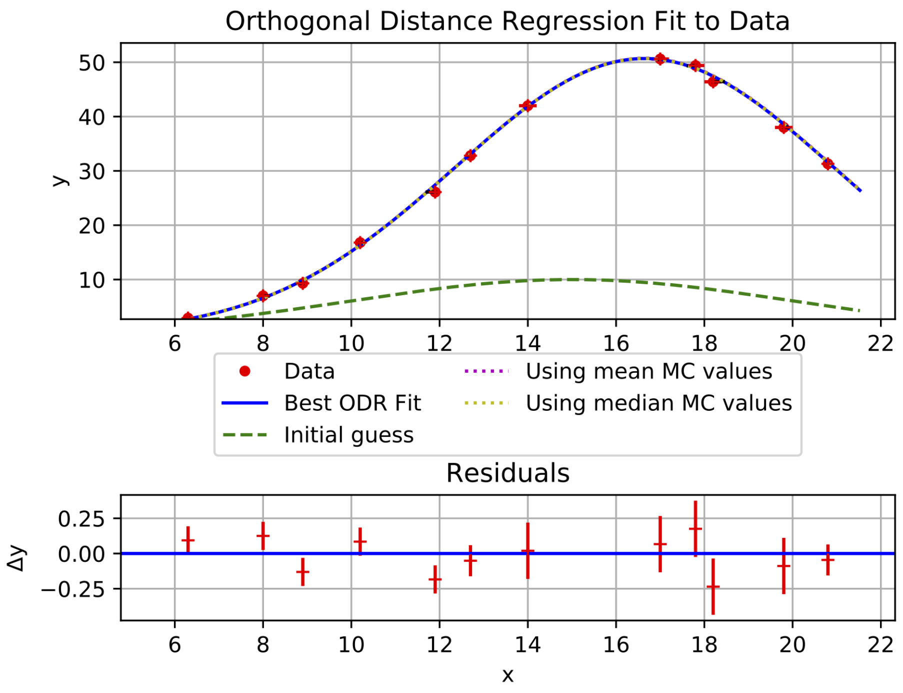


Figure 1: Nice ODR output plot example.

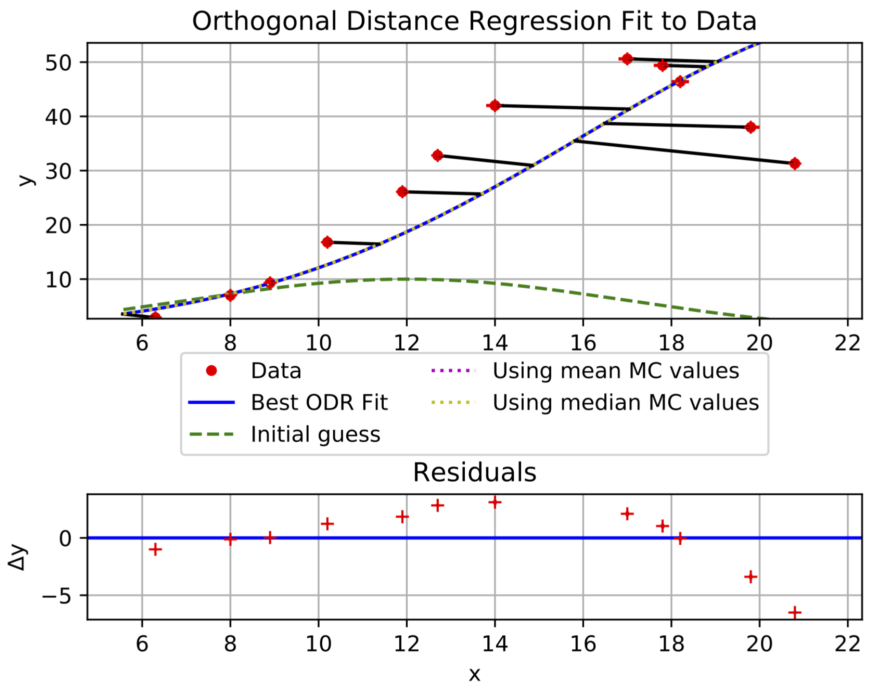


Figure 2: ODR fit to same data as Fig. 1 but with a poor initial guess for the parameters.

Effect of a poor initial guess

The initial guess of the Gaussian parameters for the fit shown in Figure 1 was `p=(15,5,10)` (which produces the green dashed curve). Figure 2 shows what happens if the the initial guess for the peak position is changed from 15 to 12, i.e. `p=(12,5,10)`. The fit converges to parameter values that produce a curve far from the data. The fit presumably found a local minimum of the summed orthogonal distances (as a function of `p`) and could not find its way to the global best fit. (The black lines from the red data points to the blue curve show the Orthogonal Distances that the fitting program was trying to minimize. These are also plotted in Figure 1 but are too small to see.)

Systematic uncertainties

To see the effect of the systematic uncertainties we can set them to zero by changing the input data file so that

```
#  a_x, da_x, b_x, db_x = 0, 0, 1, 0
#  a_y, da_y, b_y, db_y = 0, 0, 1, 0
```

(Simply deleting these lines also works.) If we rerun the fit on the modified file (gauss_ODR_no_systematics.txt), we see that the Monte Carlo fit parameters are now consistent with the initial ODR uncertainties, i.e.

```
...
 Estimated parameters, uncertainties, and starting guesses
   p[0] =     16.643 +/-    0.05991          Guessed: 15
   p[1] =     4.2778 +/-   0.042811          Guessed: 5
   p[2] =     50.686 +/-    0.27017          Guessed: 10
...
 MC Fit parameters Average + Standard Deviation; Median and 68.3% interval
   p[0] =      16.642 +/-    0.052245 ;      16.642 +    0.051553 -    0.053513
   p[1] =      4.2762 +/-    0.037367 ;      4.2764 +     0.03715 -     0.03839
   p[2] =      50.724 +/-     0.23134 ;      50.711 +     0.23992 -     0.21328
...
```

Although the difference is small, it does appear that the initial ODR uncertainties are slight overestimates, but this is not important since the contribution from the systematic uncertainty is so large.

Non-gaussian uncertainties

Not all uncertainties are associated with a normal Gaussian probability distribution. For example, if there is neglible noise in the signal and the random uncertainties are just the reading error of a digital meter, then these uncertainties are associated with a uniform random distribution. For example, if the meter reading was `354`, we might report the result as 354 ± 0.5 since expect the "true" value to be somewhere in the range 353.5 to 354.5, with uniform probability inside that interval and zero probability outside.

If the data random uncertainties of both x and y describe the full range of a uniform uncertainty, we should change

```
smear_x_data   = gaussian_distribution  # Random x
smear_y_data   = gaussian_distribution  # Random y
```

to

```
smear_x_data   = uniform_distribution  # Random x
smear_y_data   = uniform_distribution  # Random y
```

Rerunning the ODR fit with the systematics still turned off produces:

```
***********************************************************
               ORTHOGONAL DISTANCE REGRESSION
***********************************************************

Fitting 12 Data points from file gauss_ODR_no_systematics.txt to following Model function
  def gaussian(x,*p) :
    from numpy import exp
    mu     = p[0]    #   Peak position
    sigma  = p[1]    #   Peak width
    y_mu   = p[2]    #   Peak height scale
    return y_mu*exp(-(x-mu)**2/(2*sigma**2))


**** ODR has finished with: Sum of squares convergence

 Estimated parameters, uncertainties, and starting guesses
   p[0] =     16.643 +/-    0.05991          Guessed: 15
   p[1] =     4.2778 +/-   0.042811          Guessed: 5
   p[2] =     50.686 +/-    0.27017          Guessed: 10

 ODR Correlation Matrix
        [[ 1.          0.73326796 -0.34473356]
         [ 0.73326796  1.         -0.46575296]
         [-0.34473356 -0.46575296  1.        ]]

 Quasi Chi-Squared/dof   =    1.23347, Quasi CDF =   26.88333%

**** Running Monte Carlo CDF Estimator ****
    1001 successful MC simulations in  1.9493 seconds.

 Fraction of Monte Carlo quasi-chi-squared values larger than value for ODR fit:
    Monte Carlo CDF is less than    0.1%
             and is consistent with 0.0%
    For a better limit run with more iterations!

 MC Fit parameters Average + Standard Deviation; Median and 68.3% interval
   p[0] =      16.6419 +/-    0.0313624 ;       16.641 +    0.0337829 -    0.0314054
   p[1] =      4.27715 +/-    0.0219635 ;      4.27735 +    0.0219851 -    0.0225303
   p[2] =      50.6921 +/-      0.13023 ;      50.6929 +     0.131281 -     0.136698

 Monte Carlo Correlation Matrix
        [[ 1.          0.7266205  -0.29216103]
         [ 0.7266205   1.         -0.39443881]
         [-0.29216103 -0.39443881  1.        ]]

 Check For Bias in MC Fit parameters
   (Monte Carlo median)/(fit value) - 1
      p[0] Bias : -3.278e-05 +/- 7.655e-05 (  -0.43 SD)
      p[1] Bias : -9.318e-05 +/- 0.0002147 (  -0.43 SD)
      p[2] Bias : +0.0001557 +/- 0.000117  (   1.33 SD)
```

We see that the Monte Carlo parameter uncertainties are now roughly half the size of those reported by the initial ODR fit which assumes the uncertainties are standard deviations, and the Monte Carlo CDF is consistent with zero. This is unsurprising. The standard deviation of a uniform unit distribution is $1/\sqrt{12}$, so in the linear limit we expect the ODR fit to overestimate the uncertainty by $\sim \sqrt{12}/2 \approx 1.7$. If the uncertainties really do correspond to a uniform distribution with sharp edges, then every error bar in the residual plot should touch the fit line, and if even one misses the CDF will be zero.

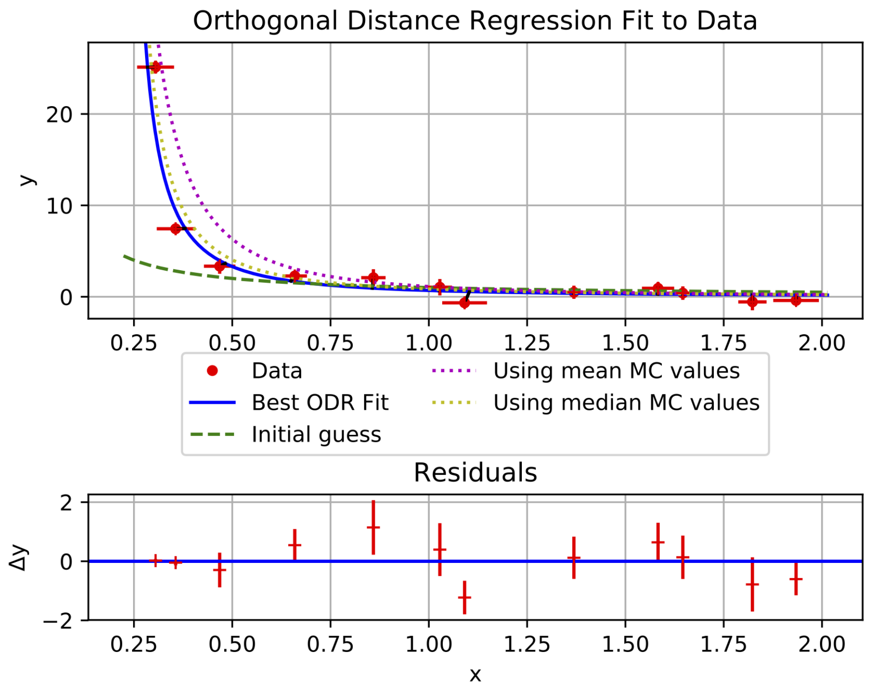
Asymmetric uncertainties

Even if the input data uncertainties are symmetric about the data values,[2] the uncertainties of the output parameters may not be. For example, consider this output from a power law fit to a data file (power_data.txt) with gaussian uncertainties and no systematics:

```
***********************************************************
               ORTHOGONAL DISTANCE REGRESSION
***********************************************************

Fitting 12 Data points from file power_data.txt to following Model function
  def absolute_power(x,*p) :
    # A "peak like" (hence the minus sign) absolute power function
    mu    = p[0]    #  Central position
    nu    = p[1]    #  Power exponent
    y_mu  = p[2]    #  Normalization
    return y_mu*abs(x-mu)**nu


**** ODR has finished with: Sum of squares convergence

 Estimated parameters, uncertainties, and starting guesses
   p[0] =    0.20045 +/-    0.13953          Guessed: 0
   p[1] =    -1.5997 +/-    0.89354          Guessed: -1
   p[2] =    0.47992 +/-    0.23242          Guessed: 1

 ODR Correlation Matrix
        [[ 1.          0.90884147  0.22292667]
         [ 0.90884147  1.          0.5666132 ]
         [ 0.22292667  0.5666132   1.        ]]

 Quasi Chi-Squared/dof   =    1.22822, Quasi CDF =   27.20293%

**** Running Monte Carlo CDF Estimator ****
    1001 successful MC simulations in 3.34612 seconds.

 Fraction of Monte Carlo quasi-chi-squared values larger than value for ODR fit:
    Monte Carlo CDF =   25.9%

 MC Fit parameters Average + Standard Deviation; Median and 68.3% interval
   p[0] =     0.141039 +/-     0.194061 ;     0.196883 +    0.0907961 -     0.188061
   p[1] =     -1.98581 +/-      1.27211 ;     -1.62672 +     0.618818 -      1.23818
   p[2] =     0.824935 +/-       2.0448 ;     0.534004 +     0.245712 -     0.174193

 Monte Carlo Correlation Matrix
        [[ 1.          0.94660266 -0.56915545]
         [ 0.94660266  1.         -0.46866823]
         [-0.56915545 -0.46866823  1.        ]]
```

[2] The input data uncertainty smearing functions can, of course, be recoded to be asymmetric if needed.

The uncertainties of the Monte Carlo medians have large asymmetries which are not obvious from the standard deviation uncertainties of the Monte Carlo means or the initial ODR fit parameters.



Systematic scaling

It is best to take all the data using just one range, but if this is not the case, then a *scale* factor can be appended to each data point that will be used to scale the systematic offset. For example, consider a multi-range voltmeter that has a 1% calibration offset uncertainty that is expected to be proportional on different ranges. i.e. If the systematic offset is -0.07 Volts on the 10 volt range, it will be -0.7 Volts on the 100 volt range. Here is the data file with systematics and scale factors.

```
## x calibration uncertainties (assuming x_true = a_x + b_x * x_measured)
#  a_x, da_x, b_x, db_x = 0, 0.05, 1.0, 0.006
## y calibration uncertainties (assuming y_true = a_y + b_y * y_measured)
#  a_y, da_y, b_y, db_y = 0, 0.05, 1.0, 0.006
#x     dx    y     dy
11.9   0.1  26.1  0.1    1
8.9    0.1  9.3   0.1    1
6.3    0.1  2.9   0.1    1
14.0   0.2  42.0  0.2    2
8.0    0.1  7.0   0.1    1
12.7   0.1  32.8  0.2    1
10.2   0.1  16.8  0.1    1
18.2   0.2  46.4  0.2    2
20.8   0.1  31.3  0.2    1
17.8   0.2  49.4  0.2    2
17.0   0.2  50.6  0.2    2
19.8   0.2  38.0  0.2    2
```

For this particular data file (gauss_ODR_with_scale.txt), the effect of the data point scale factors is negligible because they do not affect the systematic scale uncertainties which turn out to be dominant in this example.

In this example, both x and y have the same correlated systematics and scale factors. The code can, of course, be modified for more complex relationships among the systematics.