Nonlinear Regression for a Gaussian Function

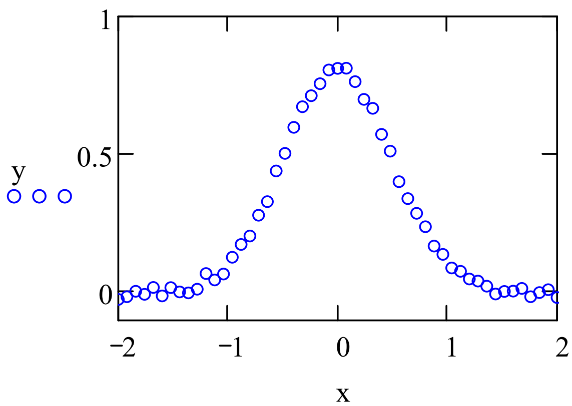
Physics 258, 259, and 281 - E. Eyler, 2006. A new contribution to a series of tutorials compiled by D. Hamilton in 2004.

This example worksheet uses a generalized least-squares fit in Mathcad to fit a peak to a Gaussian function. It should work with Mathcad 2001i and later.

Start by generating a data set of (x,y) points from a normal distribution, with a little random noise added for realism. Set the mean to zero and the standard deviation to 0.5.

$N := 50 \qquad i := 0..N$

$x_i := -2 + \frac{4 \cdot i}{N} \qquad\qquad y_i := \text{dnorm}(x_i, 0, 0.5) + \text{rnd}(0.05) - 0.025$



Always plot the data before attempting a fit.

$$\text{gauss}(x, A, \mu, \sigma) := \frac{A}{\sqrt{2 \cdot \pi} \cdot \sigma} \cdot \exp\left[-\left(\frac{x-\mu}{\sqrt{2} \cdot \sigma}\right)^2\right]$$

This is the fitting function, a Gaussian with an arbitrary amplitude.

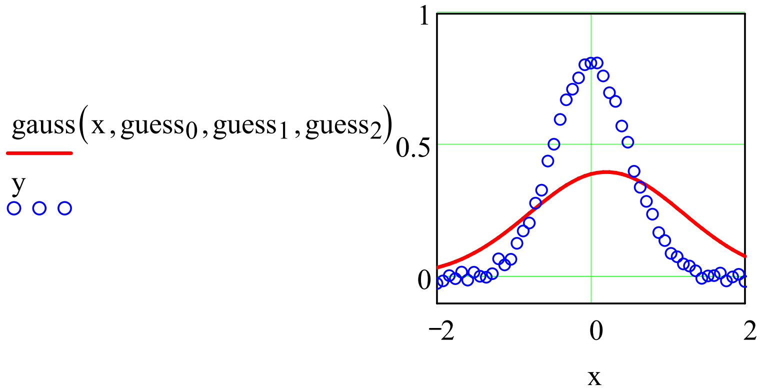
$$f(x, A, \mu, \sigma) := \begin{bmatrix} \text{gauss}(x, A, \mu, \sigma) \\ \frac{1}{A} \cdot \text{gauss}(x, A, \mu, \sigma) \\ \frac{x-\mu}{\sigma^2} \cdot \text{gauss}(x, A, \mu, \sigma) \\ \frac{(x-\mu)^2}{\sigma^3} \cdot \text{gauss}(x, A, \mu, \sigma) \end{bmatrix}$$

Vector containing the function and its derivatives with regard to each parameter. An additive background term is not included, but could easily be incorporated, with a derivative of one.

$$guess := \begin{pmatrix} 1 \\ 0.2 \\ 1 \end{pmatrix}$$

Enter initial guesses for the three parameters. Usually it's sufficient to make reasonable guesses based on a quick look at your plot. They don't have to be all that close to work.

Below we plot the data together with the current guess.



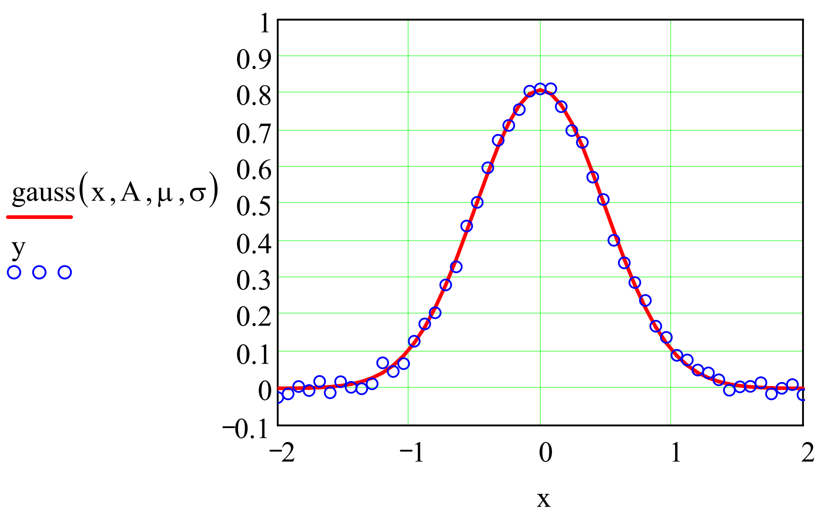
$$Results := genfit(x, y, guess, f)$$

Now perform the nonlinear least-squares fit and store the results in a vector.

$$Results = \begin{pmatrix} 1.004 \\ -1.528 \times 10^{-4} \\ 0.495 \end{pmatrix}$$

$$A := Results_0$$
$$\mu := Results_1$$
$$\sigma := Results_2$$

Plot these results, now stored in constants A, μ, and σ, together with the original data:



We can easily calculate the RMS difference between the data points and the fitting function. The built-in standard deviation function can be used to make this quick and painless. A vectorizing operator is included for completeness, although the result will be the same without it.

$$\mathrm{stdev}\left(\overrightarrow{\mathrm{gauss}(x, A, \mu, \sigma)} - y\right) = 0.013$$

We can also plot the residuals, the difference between the data and the fit. Because we used uniformly distributed noise, the residuals should reflect this.