

FittingFunctions2.0

Improved version of the fittingfunctions python module used by the Physics Institution and LTH at Lund University.

Made in python version 3.11.5

Authored by Erik Ewald & Gustav Sjövall, 2023-2024

Contents

- [FittingFunctions2.0](#)
 - [Contents](#)
 - [Importing](#)
 - **[Gaussian Functions](#)**
 - [gaussian class](#)
 - [Printing](#)
 - [Generic peak](#)
 - [Public members](#)
 - [Public Methods](#)
 - [Fitting Functions](#)
 - [fit_gaussian](#)
 - [fit_double_gaussian](#)
 - [Spectrum Calibration](#)
 - [calibrate](#)

Importing

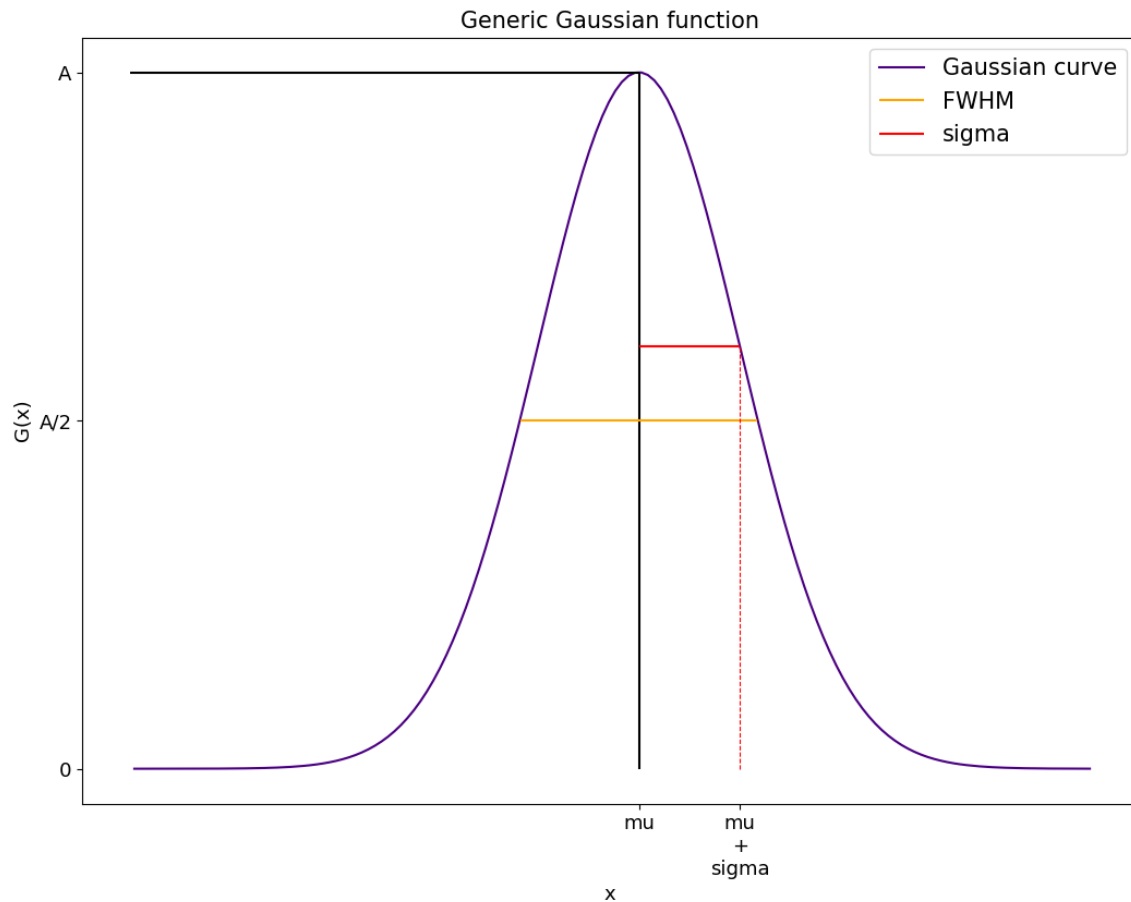
To import the module, this python file is placed in the working directory for the project and imported into the main file as

```
import fitfuncs2 as ff
```

Gaussian Functions

For the following scripts the definition of a Gaussian function is as follows:

$$G(x) = A \cdot \exp\left(-\frac{(x - \mu)^2}{2 \cdot (\sigma)^2}\right)$$



gaussian class

The gaussian class holds the return values from all functions that fit a Gaussian function and is not intended to be used independently. **Printing** a gaussian object gives rounded values. Raw values are accessed as public members or methods. Raw values for uncertainties have to be derived from the covariance matrix.

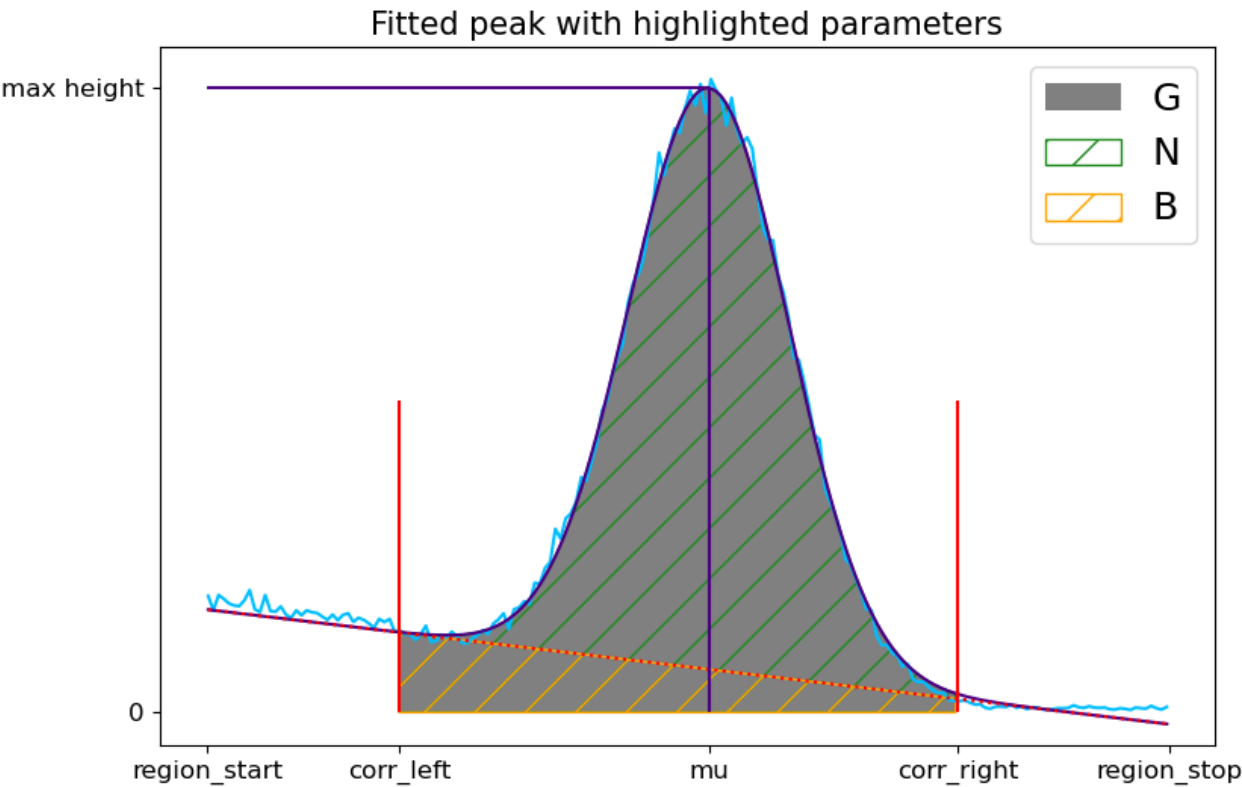
Printing

Example of a gaussian fitted to a Cs-137 peak in a gamma ray spectrum.

```
[Python3]: print(BG0["662kev"])
===== Gaussian Peak =====
Estimated parameters: A = 1740.5264, mu = 661.8352, sigma = 40.6196
Uncertainties:  $\sigma(A)$  = 7.7571,  $\sigma(\mu)$  = 0.2089,  $\sigma(\sigma)$  = 0.2096
Additional info: Max height = 1915.892, G = 79320, N = 65096, B = 14224

Covariance matrix:
[[ 6.01726192e+01  1.87206691e-03 -9.40087680e-01]
 [ 1.87206691e-03  4.36592014e-02 -9.98241359e-05]
 [-9.40087680e-01 -9.98241359e-05  4.39143608e-02]]
```

Generic peak



Public members

Member	Description
<i>gaussian.A</i> : float	The amplitude of the fitted function. OBS! This does not account for the scatter correction and will sometimes not match the peak height in a spectrum, see <i>gaussian.max_height()</i> .
<i>gaussian.mu</i> : float	The central value of the fitted gaussian curve.
<i>gaussian.sigma</i> : float	The fitted value of sigma.
<i>gussian.cov_matrix</i> 3x3-array	The covariance matrix from the fit corresponding to <i>A</i> , <i>mu</i> and <i>sigma</i> .
<i>gussian.G</i> : float	The gross area of the region enclosed by the scatter correction boundaries., i.e. the sum of all y-values in the selected region.
<i>gussian.B</i> : float	The scatter background calculated as the sum of the linear scatter correction function across the region enclosed by the scatter correction boundaries.
<i>gussian.N</i> : float	The net area of the peak, i.e. <i>gussian.G</i> - <i>gussian.B</i> .

Public Methods

Method	Description
--------	-------------

Method	Description
<i>gussian.area()</i> : -> float	The analytically calculated area of the gaussian. OBS! This may deviate from <i>gussian.G</i> as the bin size in the x-data is not accounted for by this function.
<i>gussian.FWHM(uncertainty = False : bool)</i> : -> float/tuple	Return the full-width-half-max value of the peak. Setting <i>uncertainty</i> till <i>True</i> return a tuple with the FWHM as the first element the uncertainty in FWHM as the second.
<i>gussian.max_height()</i> : -> float	Returns the real height of the gaussian at the point <i>mu</i> . Calculated as <i>corr_f(mu) + A</i> .
<i>gussian.plot(xlabel = None : str, ylabel = None : str)</i> : -> None	Plots the gaussian over the selected region.
<i>gussian.value(x : float)</i> : -> float	Returns the value of the gaussian at the point <i>x</i> .

Fitting Functions

fit_gaussian

```
fit_gaussian(X, Y, region_start, region_stop,
             corr_left=None, corr_right=None,
             mu_guess=None, A_guess=None, sigma_guess=None,
             scatter_corr="auto", scatter_corr_points=3,
             corr_thresh=0.05)
```

Attempts to fit a gaussian function to a data set using the *curve_fit* routine from *scipy.optimize*.

Parameter	Description
<i>X : array_like</i>	An iterator corresponding to the x-axis of the data set.
<i>Y : array_like</i>	An iterator corresponding to the y-axis of the data set.
<i>region_start : float</i>	The lower bound of the region on which the gaussian will be fitted.
<i>region_stop : float</i>	The upper bound of the region on which the gaussian will be fitted.
<i>corr_left : float, optional</i>	Left point at which scatter correction will be made. This only affects the scatter correction, the fit will always be made over the entire region.
<i>corr_right : float, optional</i>	Right point at which scatter correction will be made. This only affects the scatter correction, the fit will always be made over the entire region.
<i>mu_guess : float, optional</i>	Manual guess for <i>mu</i> .

Parameter	Description
<code>A_guess : float, optional</code>	Manual guess for <i>A</i> .
<code>sigma_guess : float, optional</code>	Manual guess for <i>sigma</i> .
<code>scatter_corr : str/bool, optional</code>	Sets method of scatter correction. <code>'auto'</code> attempts automatic scatter correction, if this fails <i>region_start</i> and <i>region_stop</i> will be set as correction boundaries. Manually setting boundaries will overrule the automatically set boundaries. <code>True</code> makes scatter correction without automatically set limits. Limits are manually set by <i>corr_left</i> and <i>corr_right</i> , if one or both of these are not set <i>region_start</i> and <i>region_stop</i> will be set as correction points. <code>False</code> no scatter correction is made.
<code>scatter_corr_points : int, optional</code>	The number of points over which the bounds of the scatter correction is averaged. The default 3 means the point is chosen as the average of the first 3 inside the gaussian region.
<code>corr_thresh : float, optional</code>	Sets the gradient threshold for determining the edges of the peak when <code>sactter_corr='auto'</code> .
Returns	Description
Peak : <i>gaussian</i>	An object containing the data from the fit. See the section on the <i>gaussian</i> class.

`fit_double_gaussian`

Attempts to fit a sum of two gaussian functions to a data set using the *curve_fit* routine from `scipy.optimize`.

```
fit_double_gaussian(X, Y, region_start, region_stop, split_point,
                    corr_left=None, corr_right=None,
                    plotting = False,
                    mu1_guess=None, mu2_guess=None,
                    A1_guess=None, A2_guess=None,
                    sigma1_guess=None, sigma2_guess=None,
                    scatter_corr=True, scatter_corr_points=3)
```

Spectrum Calibration

Calibrates a spectrum by marking peaks and providing energies corresponding to the marked peaks.

`calibrate`

```
calibrate(Y, peak_regions, energies, plot=False, gauss=True)
```

Parameter	Description
<code>Y : array_like</code>	The dataset corresponding to the y-axis of the spectrum.
<code>peak_regions : array_like</code>	An iterator containing start and stop values for the peak regions to be used in the calibration. A <i>gaussian</i> object can be given instead of a start and stop region. Example: <code>[[150, 300], (500, 600), gaussian_peak]</code> .
<code>energies : array_like</code>	Iterator with the energies of the peaks at the corresponding index in <i>peak_regions</i> .
<code>plot : bool, optional</code>	Plots the calibrated spectrum and highlight the points used for calibration.
<code>gauss : bool, optional</code>	If <code>True</code> a gaussian will be fitted to each selected region using the <i>fit_gaussian</i> function and the <i>mu</i> value is used as the calibration point. If <code>False</code> the maximum point in the region is used as the calibration point.
Returns	Description
<code>calib_x : numpy.array</code>	The calibrated x-axis corresponding to the given y-axis.
<code>(k, m) : tuple</code>	The calibration coefficients.