

---

# **Imfit Documentation**

***Release 1.0***

**Jörn Werdecker**

September 27, 2013



# CONTENTS

<b>1</b>	<b>lmfit Reference</b>	<b>3</b>
<b>2</b>	<b>Indices and tables</b>	<b>5</b>
	<b>Python Module Index</b>	<b>7</b>
	<b>Index</b>	<b>9</b>



Contents:



# LMFIT REFERENCE

**class** `lmfit.lmfit` (*func*, *xdata*, *ydata*, *p0*, *yerror=None*, *lm\_options={}*, *verbose=True*, *plot=False*,  
*plot\_options={}*)

Class handling non-linear least squares fitting of 2d datasets.

Based on scipy's leastsq function lmfit implements the Levenberg-Marquardt-Algorithm provided by the Fortran MINPACK library.

**func** [function-type] Testfunction to be fitted.

**xdata, ydata: array-like** Datasets with equal dimensions.

**p0** [dict or list] Set of initial parameters. When passing p0 as a list the ordering of the parameters must be the same as in the function definition. So for `func = lambda x, a, b: a*x + b` either is possible: `p0={'a':1, 'b':2}` or `p0=[1, 2]`.

**yerror** [array-like, optional] Weights for individual data points. Must have the same dimensions as x/ydata arrays.

**lm\_options** [dict, optional] Options passed to `scipy.optimize.leastsq`. Cf. scipy reference for possible options.

**verbose, plot** [bool, optional] Toggle verbose output (default: True) and plot window (default: False).

**plot\_options** [dict, optional] Options passed to this classes plot method.

`xdata, ydata func P StdDev CovMatr Chi2 RMSChi2 Residuals full_results fig` : class instance

Instance of matplotlib's figure class. Only available after plotting.

`scipy.optimize.leastsq` : Wrapper for MINPACK's fit functions.

**Chi2**  
Value of  $\chi^2$ .

**CovMatrix**  
The Covariance Matrix.

**P**  
Dictionary containing the resulting fit parameters.

**RMSChi2**  
Root mean square value of  $\chi^2$ .

**Residuals**  
Array containing the Residuals.

**StdDev**  
Dictionary containing the standard deviations of the resulting fit parameters.

**\_\_call\_\_** (*x*)

Evaluates the testfunction at *x* with the current set of parameters

**\_\_weakref\_\_**

list of weak references to the object (if defined)

**bootstrap** (*n=500, plot=False*)

Performs a bootstrapping analysis of the Residuals.

The Residuals are randomly resampled and superimposed on the fitted testfunction. This artificial dataset is then fitted again and the final parameters are stored. This is repeated *n*-times. In the end the mean values and standard deviations of the fit parameters from all fits are calculated and returned.

**n** [int, optional] Number of bootstrapping runs (default=500).

**plot** [bool, optional] Plot all fits (default=False). Be careful! Can be slow for large values of *n*.

**Mean** [dict] Mean values of the fit parameters from all bootstrap fits.

**StdDev: dict** Standard Deviations of the fit parameters determined from all bootstrap fits.

**outlist** [dict] List of dictionaries containing all final parameter sets from the bootstrap fits.

**fit** (*p0, lm\_options={}, verbose=True, plot=False, plot\_options={}*)

Carries out the non-linear fit.

**p0** [dict or list] Set of initial parameters. When passing *p0* as a list the ordering of the parameters must be the same as in the function definition. So for `func = lambda x, a, b: a*x + b` either is possible: `p0={'a':1, 'b': 2}` or `p0=[1, 2]`.

**weight** [array-like, optional] Weights for individual data points. Must have the same dimensions as *x/ydata* arrays.

**lm\_options** [dict, optional] Options passed to `scipy.optimize.leastsq`. Cf. `scipy` reference for possible options.

**verbose, plot** [bool, optional] Toggle verbose output (default: True) and plot window (default: False).

**plot\_options** [dict, optional] Options passed to this classes plot method.

**FloatingPointError** If the testfunction can be evaluated given the initial parameters.

**Exception** If `scipy.optimize.leastsq` fails without raising an own exception.

**full\_results**

Dictionary with all results from the fit and additional information.

**func**

The original testfunction.

**plot** (*residuals=True, acf=True, lagplot=True, histogramm=True*)

Creates a plot of the data and the test function using current parameters.

**residuals** [bool, optional] Plot the residuals

**acf** [bool, optional] Plot the autocorrelogramm

**lagplot** [bool, optional] Show a lagplot

**histogramm** [bool, optional] Plot histogramm of residuals

**report** ()

Prints a report about the results of the last fitting procedure.



# INDICES AND TABLES

- *genindex*
- *modindex*
- *search*



# PYTHON MODULE INDEX

I

lmfit, 3



# INDEX

## Symbols

`__call__()` (lmfit.lmfit method), 3  
`__weakref__` (lmfit.lmfit attribute), 4

## B

`bootstrap()` (lmfit.lmfit method), 4

## C

`Chi2` (lmfit.lmfit attribute), 3  
`CovMatrix` (lmfit.lmfit attribute), 3

## F

`fit()` (lmfit.lmfit method), 4  
`full_results` (lmfit.lmfit attribute), 4  
`func` (lmfit.lmfit attribute), 4

## L

`lmfit` (class in lmfit), 3  
`lmfit` (module), 3

## P

`P` (lmfit.lmfit attribute), 3  
`plot()` (lmfit.lmfit method), 4

## R

`report()` (lmfit.lmfit method), 4  
`Residuals` (lmfit.lmfit attribute), 3  
`RMSChi2` (lmfit.lmfit attribute), 3

## S

`StdDev` (lmfit.lmfit attribute), 3