Łukasz Fulek
April 22, 2016

Curve fitter

The aim of this project was to generate a signal with some noise, fit a function to the data and implement a χ^2 test to check the quality of the fit.

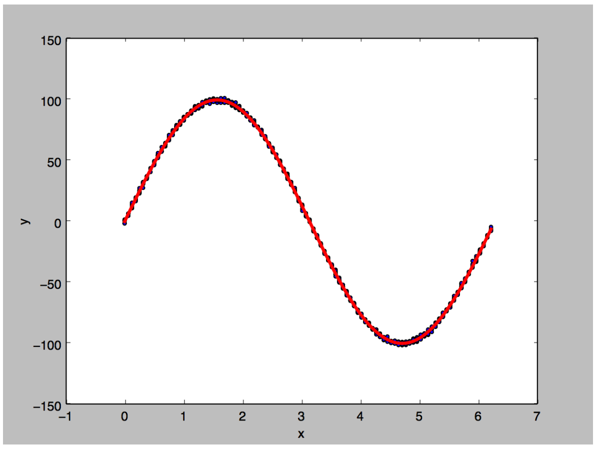
The application consists of the following components:

- Data generator – generate signal as a sine function (from 0 to 2π) and noise as a Gaussian (with σ given by the user and the mean assumed to be $\mu \in (-\sigma, \sigma)$). Also the amplitude of the signal and number of the positions (x) are given by the user. For a given position x there are generated 5 to 10 points. The code is flexible enough to replace the signal or the noise description with some other functions (Signl and Noise are the base classes and the subclasses for sine and Gaussian are used).
- Data preparation – prepares the data for the fit and analysis: the mean and std. are calculated for each x position.
- Fitter – the module for fitting the curve. It consists of one base class and three subclasses for three different curves – sine, straight line and 3rd order polynomial. The curve_fit method from scipy.optimize module is used for performing the fit. The sine is of the form: $a\sin(bx)$, straight line: $ax + b$ and 3rd order poly: $ax^3 + bx^2 + cx + d$
- StatAnalyzer – here the χ^2 test is performed. The output of this module is the χ^2/ndf.
- Plotter – module for the visualization. The data and the fitted function are drawn with the matplotlib.pyplot.
- Main module – user defines here number of x positions to be generated, the σ of the Gaussian noise, the amplitude of the signal and the function to be fitted. Because the program is written as tool for analysis and the author focused more on the scientific part of the project, the application does not contain any user interface. All changes should be made in the main module.

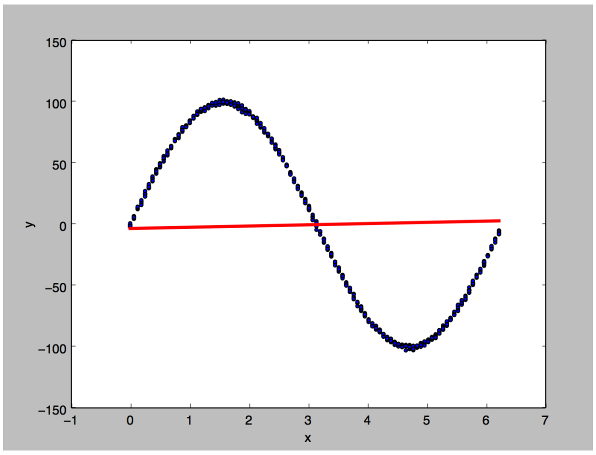
Below is the comparison between outputs for different functions fitted (sine, straight line and 3rd order polynomial) and various σ of the Gaussian noise. Each time 100 points were generated. The amplitude of the signal was assumed to be 100. It is seen that for very low noise level only the sine function is describing good the data. For larger σ values, the noise is so large that it’s hard to distinguish how the signal should look like, e.g. for $\sigma = 100$ all three functions are describing the data.

1. $\sigma = 1$

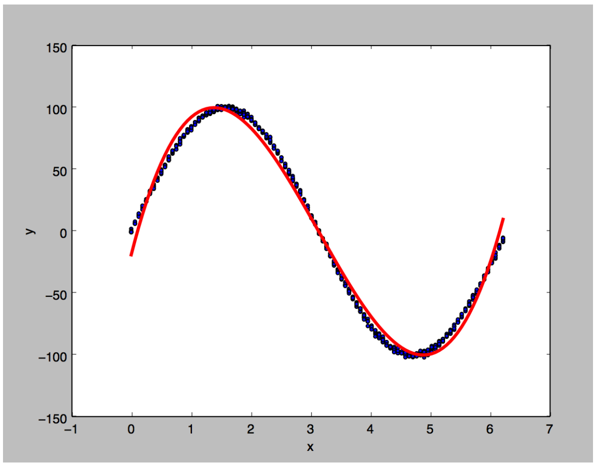
Sine fit: $\chi^2/ndf\ = 0.24$



Linear fit: : $\chi^2/ndf\ = 6857.17$

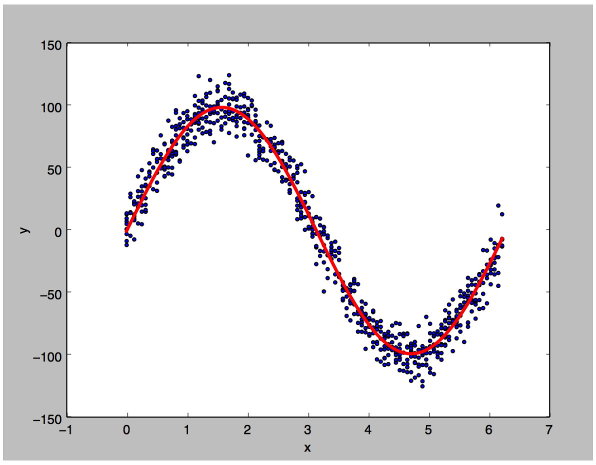


3rd order poly fit: $\chi^2/ndf\ = 55.48$

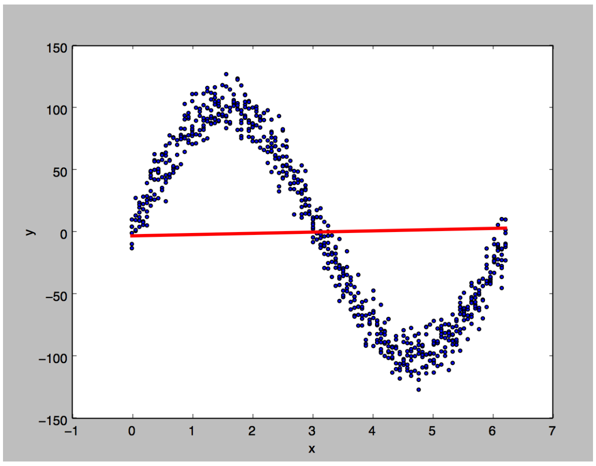


2. $\sigma = 10$

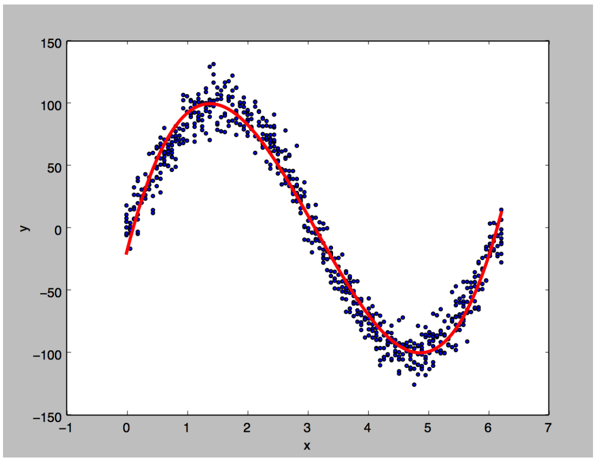
Sine fit: $\chi^2/ndf = 0.29$



Linear fit: : $\chi^2/ndf = 74.78$

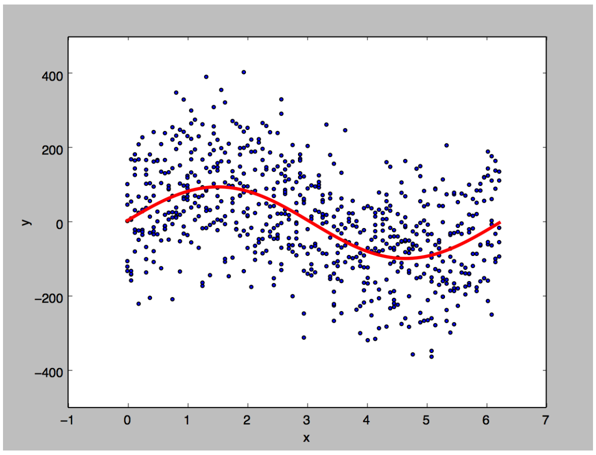


3rd order poly fit: $\chi^2/ndf = 0.79$

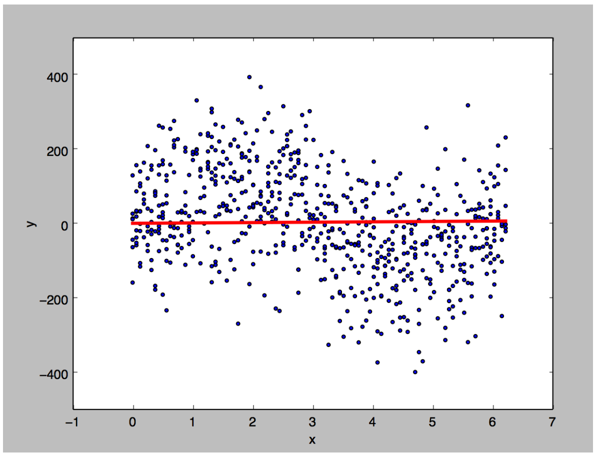


3. $\sigma = 100$

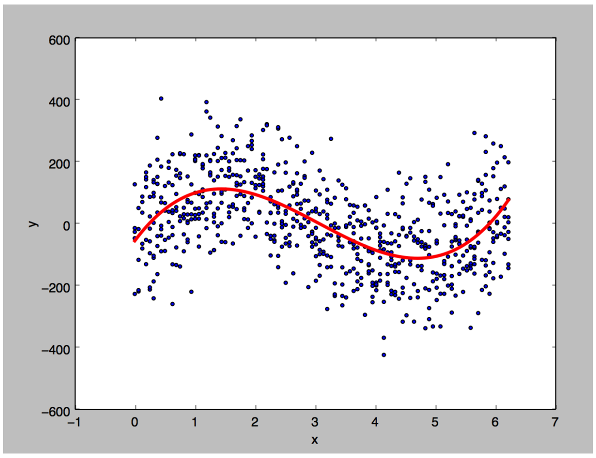
Sine fit: $\chi^2/ndf = 0.20$



Linear fit: : $\chi^2/ndf = 0.88$



3rd order poly fit: $\chi^2/ndf = 0.18$



SUMMARY

The curve fitter application has been presented. The data and noise are generated as a sine and Gaussian functions. The application is flexible, so the user can add some other models. Also there are implemented three functions to describe the data (also can be expanded): sine, straight line and 3rd order polynomial. It was checked how the functions describe the data with different noise level. The application has a possibility to draw the data with the fitted function.