“Small project #2” (the same for all)

Curve fitter

Say, we have an experiment running that registers data distributed as shown below in Figure 1:

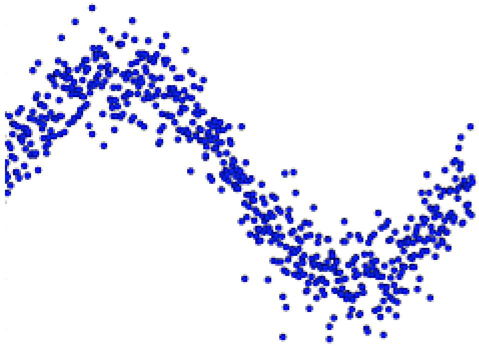

Figure 1. Data observed in our experiment.

The main aim of this project is to fit an appropriate model to this data set and check the quality of such fit. The solution should contain the following components:

- **Data generator** (the code should be flexible enough to be able to model different noise level etc.)
- **Fitter** (based on the **scipy.optimize** module with **curve_fit** method)
- **StatAnalyser** (statistical analysis engine, should implement the χ^2 test to check the quality of your fitting procedure)
- **Plotter** (for visualization)

This project is not so much about the programming challenge as it is about using Python as a tool for data analysis. I would like you to concentrate more on the “scientific” part this time. Play with the simulation that generates the data, vary the noise levels and check the impact on the fitting procedure. In this way, you can predict the behavior of your detection device and its sensitivity, e.g., if the noise level is too high the model you chose to describe the data may no longer be meaningful. Create the model and try to give its parameters some “physical” meaning – this is how we do things in experiments. Also, look at the χ^2 as a function of noise level for different models. Is there anything interesting about it?