Übungen zu Computergrundlagen WS 2015/2016

Übungsblatt 11: Python III

15.01.2016

Allgemeine Hinweise

- Abgabetermin für die Lösungen ist
  - **Freitag, 22.01., 11:00**
- Schickt die Lösungen bitte per Email an Euren Tutor.

Aufgabe 11.1: Datenanalyse mit Numpy(6 Punkte)

Ziel dieser Aufgabe ist es Messdaten mit numpy zu bearbeiten. Es sollen Mittelwerte und Standardabweichungen von Daten berechnet werden die aus Dateien eingelesen werden sollen. Die Dateien mit den Daten liegen unter `/group/cgl/2015/11`. Schreibe ein python-Skript, das folgende Aufgaben erfüllt.

- **11.1.1** Lade die Daten aus den Dateien `sim_data_dens_0.0001.txt`, `sim_data_dens_0.001.txt`, `sim_data_dens_0.01.txt` und `sim_data_dens_0.1.txt` in ein numpy-Array. Die Daten in der ersten Spalte sind die Zeiten an dem die Messpunkte augenommen wurden, in der 2. Spalte stehen die Energien, in der 3. Spalte der Druck und in der 4. Spalte steht eine weitere Observable. (2 Punkte)
- **11.1.2** Berechne für alle drei Observablen die Mittelwerte und Standardabweichungen. (1 Punkt)
- **11.1.3** Plotte den zeitlichen Verlauf dieser Daten für ein System mithilfe des `matplotlib.pyplot`-Paket. Stelle in diesen Plots auch zusätzlich den Mittelwert als rote horizontale Linie dar und deute die Standardabweichung ebenfalls als horizontale Linien in grün an, so dass die obere und untere Linie genau einmal die berechnete Standardabweichung auseinander liegen. (1 Punkt)
- **11.1.4** Schaue dir für eine dieser Dateien, die Histogramme der Observablen an. Erzeuge hierfür für alle drei Observablen ein Histogramm der Messwerte mithilfe von `numpy.histogram()`. Plotte diese dann wieder mit der `matplotlib`. (2 Punkte)

Aufgabe 11.2: Fits in python(4 Punkte)

Für den zweiten Teil dieses Aufgabenblatts soll die `curve_fit()`-Routine von `scipy`-Paket verwendet werden. Die Daten die gefittet werden sollen liegen in `/group/cgl/2015/11/fit_data.txt`. Das Datenlayout ist `x:y:yerr`.

- **11.2.1** Lade die Daten wieder als numpy-Array in python und verwende die `curve_fit()` Funktion um eine Parabel $f(x) = ax^2 + bx + c$ an die Messdaten zu fitten. Beachte dabei, dass die Messdaten nicht alle auf einer Parabel liegen, stelle hierfür den Fit-Bereich entsprechend ein. (3 Punkte)
- **11.2.2** Plotte dann die "experimentellen" Daten und gefittete Parabel zusammen. (1 Punkt)

Hinweise:

- Horizontale Linien in können via `axhline()` erzeugt werden.
- Die Grenzen der dargestellten Achsen kann mit `xlim()` und `ylim()` eingestellt werden.

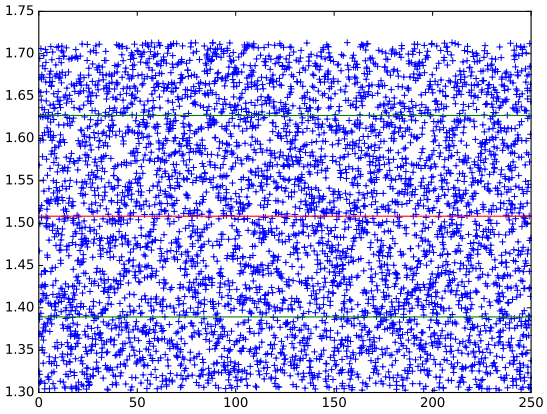

Abbildung 1: Beispielhafte Darstellung der Messdaten mit Mittelwert (rot) und Standardabweichung (grün).

- `numpy.histogram()` gibt 2 numpy-Arrays zurück. In dem ersten stehen die Anzahl der Treffer und in dem Zweiten die Ränder der bins.
- Um `curve_fit()` verwenden zu können muss `from scipy.optimize import curve_fit` am Beginn des Skriptes stehen.
- `curve_fit()` gibt 2 Arrays zurück. Im ersten stehen die Werte für die Fitparameter und in dem Zweiten die Kovarianzmatrix, die Imformationen über die statistischen Fehler für die Fitparamter enthält.
- Für `curve_fit()` muss eine Funktion definiert werden für ein Polynom mit 3 Paramtern die Definition könnte zum Beispiel `def fit_func(x, a, b, c)` lauten.
- Um die Plot Funktionalität der `matplotlib` Bibliothek verwenden zu können, kann mann alle nötigen Funktionen via `from matplotlib.pyplot import *` einbinden.
- Um Plots mit Fehlerbalken zu erzeugen, kann die Funktion `errorbar()` der `matplotlib` Bibliothek verwendet werden.