Rechnernutzung in der Physik: Zusatz Python-Einführung

Institut für Experimentelle Teilchenphysik
Institut für Theoretische Teilchenphysik

Prof. Dr. G. Quast, Prof. Dr. M. Steinhauser
Dr. Th. Chwalek, Dr. A. Mildenberger
http://comp.physik.kit.edu

WS2018/19 – Zusatzblatt[1] 04
Bearbeitungszeitraum: bis Di, 27.11.2018

Aufgabe 110: Signal-Darstellung und Verarbeitung (*) **Testat**

In der Datei `signal.csv` zu diesem Aufgabenblatt finden Sie Daten eines typischen Signals im `CSV`-Format, wie es z. B. ein Digitaloszilloskop ausgeben würde.

Bei der Lösung dieser Aufgabe orientieren Sie sich bitte an den Beispielen aus den Dateien: `a110-test_meanFilter.py`, `a110-test_Fourier.py` und `a110-test_AutoCorrelation.py`.

(a) Öffnen Sie die Datei `a110-signal.csv` mit einem Text-Editor und „erraten" Sie das Datenformat. Lesen Sie die Datei in `numpy-arrays t` und `a` ein und stellen Sie den Signalverlauf grafisch dar.
Tipp: Lassen Sie sich vom Code in `PhyPraKit.readCSV` inspirieren!

(b) Sie werden feststellen, dass das Signal durch zufällige Beiträge zur Amplitude an jedem Messpunkt sehr verrauscht ist. Glätten Sie das Signal mit einem einfachen Mittelwertfilter (s. Bsp. `a110-test_meanFilter.py` oder `PhyPraKit.meanFilter()`). Wählen Sie einen passenden Glättungsparameter und stellen Sie den geglätteten Signalverlauf dar. Bilden Sie die Differenz des geglätteten Signals und des Originals und stellen Sie es ebenfalls grafisch dar.

(c) Sie haben sicherlich schon bemerkt, dass es sich um ein periodisches Signal handelt. In solch einem Fall ist eine Bestimmung der Frequenz bzw. der Periodendauer von Interesse. Die genaueste Methode zur Frequenzbestimmung eines Signals mit fester Grundfrequenz ist die Autokorrelationsanalyse. Führen Sie diese mit dem geglätteten Signal durch (Codebeispiel `a110-test_AutoCorrelation.py`, das das Modul `PhyPraKit.autocorrelate()` verwendet). Bestimmen Sie wie im Beispiel die Maxima und Minima der Autokorrelationsfunktion und tragen sie die Differenzen benachbarter Minima und Maxima als Häufigkeitsverteilung auf. Die statistische Analyse dieses Histogramms erlaubt die Bestimmung der Periodendauer und der Unsicherheit (aus Mittelwert und Unsicherheit des Mittelwerts, s. `PhyPraKit.histstat()`).

(d) (freiwillig) Bei nicht klar erkennbarer Periodizität des Signalverlaufs empfiehlt sich die Durchführung einer Fourier-Analyse. Stellen Sie die Fourier-Transformierte des geglätteten Signals dar (Code-Beispiel `PhyPraKit.FourierSpectrum()`).

Aufgabe 111: Korrelation (*) **Testat**

In dieser Aufgabe soll die Verteilung der Bin-Inhalte eines Histogramms sowie die Korrelation zwischen ihnen näher untersucht werden. Dazu gibt es eine Vorlagen, die Sie ergänzen können (`a111-Covariance.py`).

(a) (Testat) Untersuchung der Häufigkeitsverteilung von Bin-Inhalten
Füllen Sie als Experiment $N = 100$ in $[0, 1]$ gleichverteilten Zufallszahlen in ein Histogramm mit $n_b = 4$ Bins zwischen $[0, 1]$. Die Bins enthalten nun im Mittel je 100/4 Einträge. Diese Häufigkeit n_i in den einzelnen Bins soll nun weiter untersucht werden. Wiederholen Sie dazu dieses Experiment 10'000 mal und erzeugen Sie für zwei der Bins (z.B. Bin 1 und Bin 3) ein Histogramm der dabei in diesem Bin gefundenen Häufigkeiten n_i.

Welche Verteilung der Bin-Inhalte erwarten Sie? Vergleichen Sie, indem Sie die erwartete Verteilung einzeichnen.

Da in jedem Experiment $N = 100$ Zahlen gefüllt werden, ist die Wahrscheinlichkeit eines Eintrags im Bin i also $p_i = 0.25$. Für die Verteilung der Gesamtzahl der Einträge in jedem einzelnen Bin wird also eine Binomialverteilung mit Mittelwert $Np_i = 25$ und Standard-Abweichung $\sigma = \sqrt{Np_i(1 - p_i)} \approx 4.33$. erwartet.

(b) (freiwillig) Berechnung der Korrelation von Bin-Inhalten

[1]Die Python-Einführung (5 Aufgabenblätter) ist von denjenigen Teilnehmern zu bearbeiten, die die Rechnernutzung im Umfang von 6 LP absolvieren möchten. In diesem Teil sind 80% der Pflichtaufgaben erfolgreich zu bearbeiten.

Berechnen Sie die Kovarianz und den Korrelationskoeffizienten der Anzahl von Einträgen in zwei Bins:

$$\begin{aligned}
\mathrm{cov}(x,y) &= \Big\langle (x - \langle x\rangle)\ (y - \langle y\rangle) \Big\rangle = \langle x \cdot y\rangle - \langle x\rangle\langle y\rangle, \\
\rho_{xy} &= \frac{\mathrm{cov}(x,y)}{\sqrt{\mathrm{var}(x)\,\mathrm{var}(y)}}.
\end{aligned}$$

Verallgemeinern Sie dazu den Programmcode aus Aufg. 106 von Extra-Blatt 2 so, dass Kovarianz und Korrelationskoeffizient von zwei Arrays berechnet werden.

Bestimmen Sie den Korrelationskoeffizienten der in **(a)** gefundenen Inhalte n_i und n_j zweier Bins, also z.B. o.B.d.A. Bin 1 und Bin 3.

(c) (Testat) Grafik der Korrelation von Bin-Inhalten
Stellen Sie die Einträge in den Bins 1 und 3 als zweidimensionales Histogramm dar.

Eine anschauliche Methode, Korrelationen zwischen den Bin-Inhalten zu untersuchen, besteht darin, zweidimensionale Histogramme zu füllen. Wenn Sie sich die aus dem zwei-dimensionalen Histogramm bestimmten Korrelationskoeffizienten ausgeben lassen, können Sie Ihren eigenen Code zur Berechnung aus Teil **(b)** validieren. Siehe dazu die `Python`-Funktion `hist2dstat` in `PhyPraKit`.

Zur Darstellung der Korrelation kann auch die Funktion `hist2d` der Matplotlib verwendet werden.
