Rechnernutzung in der Physik

Institut für Experimentelle Teilchenphysik
Institut für Theoretische Teilchenphysik

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

WS2018/19 – Blatt 10
Prog. bis Di., 29.01.2019

Wichtig: Bitte melden Sie sich bis spätestens 29.01.2019 im Studierendenportal zur Teilnahme an diesem Kurs an.

Simulation eines Poission-Prozesses

Aufgabe 19: Zufällige Ereignisse und Poisson-Verteilung * verpflichtend für Bachelor

In dieser Aufgabe soll eine Folge von Zeitpunkten erzeugt werden, zu denen ein Ereignis stattfindet, wobei die Eintreffwahrscheinlichkeit pro Zeitintervall konstant ist. Diese Voraussetzungen führen zu einer Poisson-Verteilung der Anzahl von Ereignissen in einem festen Zeitintervall. Die Anwendungsbeispiele dazu sind in der Physik sehr vielfältig, von der Emission eines Photons aus einer Einzel-Photon-Quelle, dem Zerfall eines Kerns in einem radioaktiven Präparat, oder auch der Produktion von Higgs-Bosonen am Large-Hadron-Collider.
Zum tieferen Verständnis solcher Poisson-Prozesse gehen wir zunächst von der Verteilung der sogenannten „Wartezeit“ aus, d. h. der zwischen dem Auftreten von zwei Zufallsereignissen verstrichenen Zeit.

a) Verteilung der Wartezeiten

Die Verteilungsdichte der Wartezeiten, t_w, zwischen zwei Ereignissen, die in der Zeit gleichverteilt sind, ist gegeben durch $f(t_w) = \frac{1}{\overline{t_w}} \exp\left(-t_w/\overline{t_w}\right)$, wobei $\overline{t_w}$ die mittlere Wartezeit ist und dem Kehrwert der Ereignisrate R entspricht. Also gilt auch $f(t_w) = R \cdot \exp\left(-R\, t_w\right)$ (zur Begründung s. Anhang).
Nutzen Sie diese Verteilung der Wartezeiten und schreiben Sie eine Funktion, die, ausgehend von $t_0 = 0$, eine Folge von N=50'000 Zeitpunkten t_i bestimmt, zu denen solche Ereignisse auftreten. Der Einfachheit halber sei die Rate $R = 1$/Zeiteinheit (ZE).
Histogrammieren Sie zur Kontrolle die Zeiten zwischen aufeinanderfolgenden Ereignissen und überlagern Sie zum Vergleich die Exponentialfunktion.

b) Hinzufügen von Rauschen

In der Realität wird neben einem Signal meist ein zusätzlicher Anteil aus Untergrund oder Rauschen beobachtet. Erzeugen Sie analog zu Aufgabenbeil a) Zeitpunkte für Rauschereignisse, deren Rate 10 % der Signalrate beträgt; die Wartezeit zwischen zwei Rauschereignissen ist also 10 mal größer als die Wartezeit zwischen zwei Signalereignissen, $t_w^{\text{noise}} = 10 \cdot t_w$. Erzeugen Sie Zeitpunkte für Rauschereignisse, die zwischen dem ersten und letzten Signalereignis liegen. Kombinieren Sie nun die Zeitpunkte der Signal- und der Rauschereignisse. Um eine aufsteigende Folge von Zeitpunkten zu erhalten, muss der kombinierte Datensatz in aufsteigender Reihenfolge sortiert werden (*siehe z. B.* `numpy.append()` und `numpy.sort()`).
Tragen Sie für die kombinierten Daten die Verteilung der Zeitdifferenzen zwischen zeitlich aufeinanderfolgenden Ereignissen auf (also die beobachteten Wartezeiten). Zeichnen Sie Ihre Erwartung ein und vergleichen Sie mit der Verteilung, die Sie in a) erhalten haben.

c) Überprüfung der zeitlichen Verteilung

Tragen Sie die im Aufgabenteil b) erzeugten Zeitpunkte t_i in ein Histogramm mit 500 Bins ein. Welche Verteilung erhalten Sie? Welche Ereignisrate/Bin lesen Sie ab? Zeichnen Sie die Erwartung für die Signalereignisse alleine und für die Summe aus Signal- und Rauschereignissen ein.

d) Verteilung der Bin-Inhalte
Stellen Sie die Bin-Inhalte des Histogramms aus Aufgabenteil c) als Häufigkeitsverteilung dar. Vergleichen Sie mit der Poisson-Verteilung.

e) Detektoreffizienz und gemessene Rate
Als letzter Schritt unserer kleinen Experimentsimulation soll nun das Ansprechverhalten eines Detektors und dessen Einfluss auf die gemessene Ereignisrate beschrieben werden. Nehmen Sie dazu an, ein Detektor habe nach dem Registrieren eines Ereignisses zum Zeitpunkt t_i eine Totzeit, die dazu führt, dass die Nachweiseffizienz nach der Funktion $\epsilon(t) = 5./\mathrm{ZE} \cdot (t - t_i)$ für $(t - t_i) < 0.2\,\mathrm{ZE}$ von Null bis Eins ansteigt. Welche Ereignisrate misst der Detektor? Wie groß ist der auf Grund der Totzeit notwendige Korrekturfaktor auf die gemessene Rate?

ANHANG: Hinweise

Verteilung der Wartezeiten

Man denke sich die Zeit t_w in n kleine, gleich große Intervalle $\Delta t_i = t_w/n$ zerlegt. Die Wahrscheinlichkeit, ein Ereignis in einem solchen Intervall zu beobachten, ist gegeben durch $p = R\,t_w\,/n$. Ein Zerfall nach der Zeit t_w bedeutet, dass in den Intervallen davor kein Ereignis beobachtet wurde, d.h. $p(t_w) = (1 - Rt_w/n)^n$, für $n \to \infty$ also $p(t_w) = \exp(-Rt_w)$.

Hinweise zur Programmierung

Die Nachweiswahrscheinlichkeit im Detektor von Aufg. d) hängt vom Zeitpunkt des letzten Ereignisses ab, der dazu gespeichert werden muss. In `C++` kann man dazu eine statische Variable verwenden (z.B. `static float tlast`), in `Python` eine globale Variable, die außerhalb einer Funktion definiert ist und innerhalb einer Funktion als `global tlast` deklariert werden muss. Obwohl für diese Übungsaufgabe akzeptabel, sind in längeren Programmen solche statischen bzw. globalen Variablen sehr problematisch. Die saubere Lösung ist es, den Detektor und seine Eigenschaften als eigene Klasse zu definieren, in `Python` z. B.

```
# Class defining detector properties
class SimpleDetector:
  """a class defining a simple detector with dead time"""
  def __init__(self, tau=0.2):
    self.tlast = -1.     # initialize variable storing argument of last call
    self.tfulleff = tau  # time after which full efficiency is regained

  def Efficiency(self, dt):
   # function defining detector characteristics,
   #   here the efficiency depending on time since last hit
    # ...
    return eff

  def Signal(self,t):
   # function returning True if signal at time t is detected
   #   uses random sampling of function "Efficiency()"
    # ...
    #SigSeen = ... # True/False
    return SigSeen
```

Eine Instanz der Klasse kann dann alle notwendigen Daten zum simulierten Detektor und zur Berechnung der Nachweiseffizienz speichern und die benötigten Funktionen bereit stellen.

Hinweis: Mit dem Rechnernamen `fphctssh.physik.uni-karlsruhe.de` können Sie von überall aus mittels ssh/scp Programm per Netzwerk auf einen Poolrechner zugreifen.
