Computerphysik WS 2016/2017

Prof. Dr. Petra Imhof, FU Berlin

Übungsblatt 10: Fourieranalyse und Zufallszahlen

5. Januar 2017

10.1 Fouriertransformation (10 Punkte)

In einem neu entwickelten Auto führen bestimmte Fahrbedingungen zu starken Schwingungen des Autodachs. Mittels eines Beschleunigungssensors wurden diese Schwingungen gemessen (siehe signal.txt, Sampling Frequenz 1000 Hz, 1024 Samples). Allerdings ist das Signal der Messung sehr komplex und überlagert von anderen Signalen. Als Ursache für die Schwingungen haben die Konstrukteure eine Welle im Getriebe im Verdacht, welche bei diesen Fahrbedingungen mit 3.000 rpm rotiert.

a) (3 Punkte): Implementieren Sie eine Pythonfunktion `dft()`, die eine diskrete Fouriertransformation einer Datenreihe durchführt.

b) (3 Punkte): Implementieren Sie eine weitere Pythonfunktionen `fft()`, die eine Fast Fouriertransformation einer Datenreihe durchführt.

   Hinweis: Zur Kontrolle können Sie die Funktion `numpy.fft.fft` nutzen.

c) (3 Punkte): Bestimmen Sie mit ihren beiden Funktionen das Frequenzspektrum für das Signal.

d) (1 Punkt): Kann die Welle wirklich für die Schwingungen verantwortlich sein?

10.2 Zufallszahlen (5 Punkte)

a) (3 Punkte) Implementieren Sie den in der Vorlesung vorgestellten linearen kongruenten Zufallszahlengenerator mit der Iterationsvorschrift:

$$n_{i+1} = (a \cdot n_i + b)\, mod\, c. \tag{1}$$

   Der Modulo-Operator wird in Python als % geschrieben und kann sowohl für Ganzzahlen als auch für Gleitkommazahlen verwendet werden.

   Leiten Sie den Startwert (*seed*) n_0 von der Systemzeit t (`time.time()`) wie folgt her:

$$int\left(\frac{abs((64979 \cdot t \cdot (pid - 83)) mod\, 104729)}{104729} \cdot c\right) \tag{2}$$

   Hierbei ist pid die Process ID der laufenden Anwendung (`os.getpid()`). Nutzen Sie die *magic numbers* $a = 7^5$, $b = 0$ und $c = (2^{31} - 1)$.

b) (2 Punkte) Generieren Sie eine Folge von $N = 10.000$ Zufallszahlen $r_1, \dots, r_N$ im Bereich $[0, 1)$ mit der Vorschrift $r_i = n_i/c$. Berechnen Sie das arithmetische Mittel $\bar{r} = \frac{1}{N}\sum_{i=1}^{N} r_i$ (`np.mean()`) sowie die Standardabweichung

$$\sigma_r = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(r_i - \bar{r})^2} \tag{3}$$

(`np.std()`) dieser Zufallszahlen. Plotten Sie die Verteilung der Zufallszahlen in einem Histogramm.

10.3 Monte-Carlo-Integration (5 Punkte)

Bei der Stützstellen Monte Carlo Integration ergibt sich ein Schätzwert für das gesuchte Integral durch Auswertung des Integranden an N zufällig ausgewählten Stützstellen:

$$I_{MC} := \frac{b-a}{N}\sum_{i=1}^{N} f(a + (b-a)r_i) \approx \int_a^b f(x)\,\mathrm{d}x, \tag{4}$$

wobei $(r_1, ..., r_N)$ unabhängige, zwischen 0 und 1 gleichverteilte, Zufallszahlen sind.

Wir wollen diese Methoden nun mit der Rechteck- oder Mittelpunktsmethode,

$$I_M := \frac{b-a}{N}\sum_{i=1}^{N} f(a + (b-a)(i-0,5)/N) \approx \int_a^b f(x)\,\mathrm{d}x, \tag{5}$$

vergleichen.

a) (2 Punkte) Implementieren Sie die beiden Integrationsmethoden allgemein. Ihre Funktionen sollten die zu integrierende Funktion f, die Intervallgrenzen a, b, sowie die Anzahl Stützstellen/Stichproben N als Argumente haben.

   Hinweis: Zum Erzeugen von Zufallszahlen können Sie entweder ihren Generator aus Aufgabe 1 oder Funktionen aus der Klasse `numpy.random` verwenden.

b) (1 Punkte) Berechnen Sie das Integral

$$\int_0^1 \exp(-x^2)\ \mathrm{d}x = \frac{1}{2}\sqrt{\pi}\,\mathrm{erf}(1) \tag{6}$$

   mit den beiden Methoden für jeweils

$$N = 2^1, 2^2, ..., 2^{18}. \tag{7}$$

   Stellen Sie die jeweiligen absoluten Abweichung vom analytischen Ergebnis als Funktion von N in einem gemeinsamen doppellogarithmischen Plot dar.

c) (2 Punkte) Man kann Gleichung 4 leicht erweitern um mehrdimensionale Integrale zu lösen. Für ein Integral in zwei Dimensionen ergibt sich zum Beispiel

$$\int_a^b \int_c^d f(x,y)\,\mathrm{d}x\,\mathrm{d}y \approx \frac{(b-a)(d-c)}{N}\sum_{i=1}^{N} f(\tilde{x}_i, \tilde{y}_i)\ . \tag{8}$$

   Hier ist $\tilde{x}_i = a + (b-a)r_i$ und $\tilde{y}_i = c + (d-c)\rho_i$, wobei $(r_1, ..., r_N)$ und $(\rho_1, ..., \rho_N)$ zwischen 0 und 1 gleichverteilte, Zufallszahlen sind.

   Lösen Sie mit Hilfe der mehrdimensionalen Stützstellen Monte Carlo Integration das Integral

$$I = \int_0^1 dx_1 \int_0^1 dx_2 \dots \int_0^1 dx_{10}(x_1 + x_2 + \cdots + x_{10})^2 = \frac{155}{6}\ . \tag{9}$$

   und überprüfen Sie ihre Resultat durch Vergleich mit dem analytischen Ergebnis.

10.4 Berechnung von Pi (Bonusaufgabe 5 Punkte)

a) (2 Punkte) Schreiben Sie ein Programm, dass den Wert von π (3.141592...) mit Hilfe der Monte-Carlo-Methode berechnet. Nehmen sie einen Krei mit Radius $r = 0.5$, umschlossen von einem 1×1 Quadrat. Die Fläche des Keises ist $\pi r^2 = \pi/4$ und die des Quadrats 1. Erzeugen Sie eine Anzahl Zufallspunkte irgendwo innerhalb des Quadrats, d.h.zwischen (0,0) und (1,1) und zählen Sie, welche davon innerhalb des Kreises landen.

b) (1 Punkt) Erweitern sie Ihr Programm so, dass Sie die Fläche eines Rings berechnen können, dessen äußerer Kreis den Radius $r = 0.5$ und dessen innerer Kreis den Radius $r = 0.4$ hat.

c) (2 Punkte) Varrieren Sie die Anzahl Punkte 30, 60 ... 3000 und plotten Sie den erhaltenen Wert von π bzw. die Fläche des Rings, sowie den jeweiligen absoluten Fehler gegen die Anzahl Punkte. *Hinweis:* Zum Erzeugen von Zufallszahlen können Sie entweder ihren Generator aus Aufgabe 1 oder Funktionen aus der Klasse `numpy.random` verwenden.