Übungen zu Computergrundlagen WS 2018/2019

Übungsblatt 10: Python

10. Januar 2019

Allgemeine Hinweise

- Abgabetermin für die Lösungen ist **Freitag, 18.01.2019, 11:00 Uhr**
- Schickt die Lösungen bitte per Email an Euren Tutor:
  - Montag 14:00–15:30: Grant Cates (`gcates@icp.uni-stuttgart.de`)
  - Dienstag 9:45–11:15: Kai Szuttor (`kai@icp.uni-stuttgart.de`)
  - Dienstag 15:45–17:15: Julian Michalowsky (`jmichalowsky@icp.uni-stuttgart.de`)
  - Mittwoch 15:45–17:15: Michael Kuron (`mkuron@icp.uni-stuttgart.de`)
  - Donnerstag 9:45–11:15: Frank Maier (`fmaier@icp.uni-stuttgart.de`)
- Die Übungen sollen in Gruppen von jeweils *zwei bis drei* Leuten bearbeitet werden. Abgaben von Einzelpersonen werden nicht akzeptiert. Bitte gebt *nur eine Lösung pro Gruppe* ab und nennt in eurer Abgabe alle Mitglieder eurer Gruppe!

In diesem Blatt geht es um die (näherungsweise) Berechnung der Kreiszahl π. Drei Methoden dafür sind in Abbildung 1 skizziert.

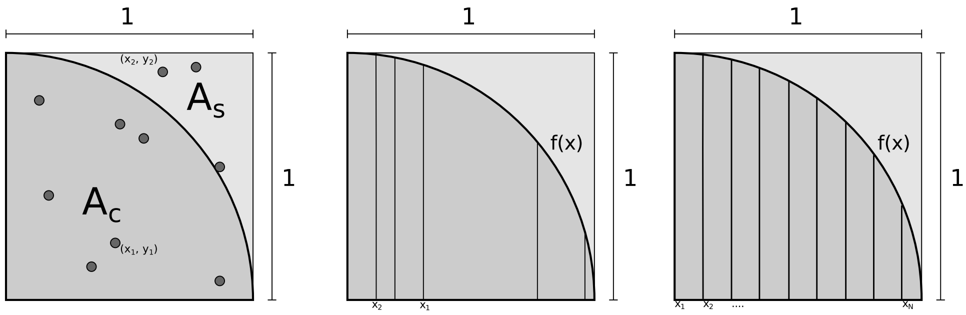

Abbildung 1: Drei Methoden zur Abschätzung von π (siehe Aufgaben 10.1, 10.2 und 10.3)

Aufgabe 10.1: π-thon 1 (3 Punkte)

Eine einfache Methode, um eine Abschätzung für die Kreiszahl π zu erhalten, ist die folgende. Die Idee dabei ist es, zufällig N_s Punkte (x_i, y_i) aus einem Einheitsquadrat (Quadrat mit Kantenlänge 1) zu ziehen ($0 \leq x_i \leq 1$, $0 \leq y_i \leq 1$). N_c sei die Anzahl der Punkte davon, die in einem Viertel des Einheitskreises (Kreis mit Radius 1) liegen (also bei denen $x^2 + y^2 < 1$ ist). Da wir wissen, dass die Fläche des Einheitsquadrats $A_s = 1$ ist und die Fläche des Einheitskreises $A_c = \pi$, sollte also gelten:

$$\frac{N_c}{N} \approx \frac{\frac{1}{4}A_c}{A_s} = \frac{1}{4}\pi \qquad \Rightarrow \pi \approx 4\frac{N_c}{N}$$

Wir nähern damit die Fläche des Kreissegments an. Dabei gilt: je größer N, desto genauer ist die Abschätzung. Wir können π also durch folgende Formel annähern:

$$\pi \approx 4\frac{1}{N}\sum_{i=1}^{N}\chi_{x^2+y^2<1}(x_i, y_i)$$

wobei

$$\chi_{x^2+y^2<1}(x, y) = \begin{cases} 1, & \text{falls } x^2 + y^2 < 1 \\ 0, & \text{sonst} \end{cases}$$

Das Python-Skript `/group/cgl/2018/10/compute_pi.py` soll die Zahl π mit Hilfe der beschriebenen Methode annähern. Leider ist es fehlerhaft. Korrigiere das Skript!

Aufgabe 10.2: π-thon 2 (3 Punkte)

Die in Aufgabe 10.1 beschriebene Methode ist nicht besonders effizient. Nun wollen wir die Genauigkeit (und damit die Effizienz) der Methode erhöhen. Dazu betrachten wir nun die Kreislinie als Funktion

$$f(x) = \sqrt{1 - x^2}\,.$$

Wieder wollen wir die Fläche des Kreissegments (also die Fläche unter der Funktion) ausrechnen, d. h. wir *integrieren* die Funktion $f(x)$. Dazu ziehen wir diesmal eine Reihe x_i von zufälligen Werten $(0 \leq x_i \leq 1)$ und nähern π wie folgt an:

$$\pi \approx 4\frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

Wieder gilt: Je größer N, desto genauer ist die Abschätzung.

Implementiere die hier beschriebene Methode zur Berechnung der Zahl π. Das Skript aus Aufgabe 10.1 kann hierfür als Basis dienen.

Hinweis: Die oben beschrieben Methode, eine Funktion mit Hilfe von Zufallszahlen zu integrieren, nennt sich *Monte-Carlo-Methode* (benannt nach den Spielcasinos des Stadtteils von Monaco).

Aufgabe 10.3: π-thon 3 (4 Punkte)

Die oben beschriebene Monte-Carlo-Methode ist recht einfach, funktioniert bei beliebigen Funktionen und ist in vielen Fällen daher sehr nützlich. Im konkreten Falle der Berechnung von π ist es aber noch sehr viel effizienter, die Werte von x_i nicht zufällig zu ziehen, sondern stattdessen ein gleichmässiges Gitter von Werten zwischen 0 und 1 zu benutzen.

Kopiere das in Aufgabe 10.2 erstellte Python-Skript und verändere es so, dass es die Zahl π mit Hilfe der hier beschriebenen Methode annähert.