Übungen zu Computergrundlagen WS 2011/2012

Übungsblatt 4: πthon

6. November 2011

Allgemeine Hinweise

Abgabetermin für die Lösungen ist

- **Dienstag, 15.11., 13:00** für die Übungsgruppen am Mittwoch und Donnerstag
- **Donnerstag, 17.11., 13:00** für die Übungsgruppen am Montag und Dienstag

Die Lösungen solltest Du in eine Kopie der Datei `/share/Courses/CG2011/04/vorlage04.txt` einfügen. Zur Abgabe schickst Du die Lösungsdatei im Anhang einer Email an Deinen Tutor.

Bitte achtet dabei auf die korrekte Kodierung der Umlaute. Wenn das „Ü“ im Wort „Übungsblatt“ in Deinem Editor nicht korrekt dargestellt wird, dann ändert entweder die Kodierung des Editors auf UTF-8, oder verwende keine Umlaute oder sonstige Sonderzeichen in dem, was Du schreibst!

Aufgabe 4.1: Variablennamen (1 Punkt)

Python (und auch bash) erlaubt keine Variablennamen, die das Zeichen „`-`“ enthalten (Beispiel: `max-size`). Warum nicht?

Aufgabe 4.2: πthon 1 (3 Punkte)

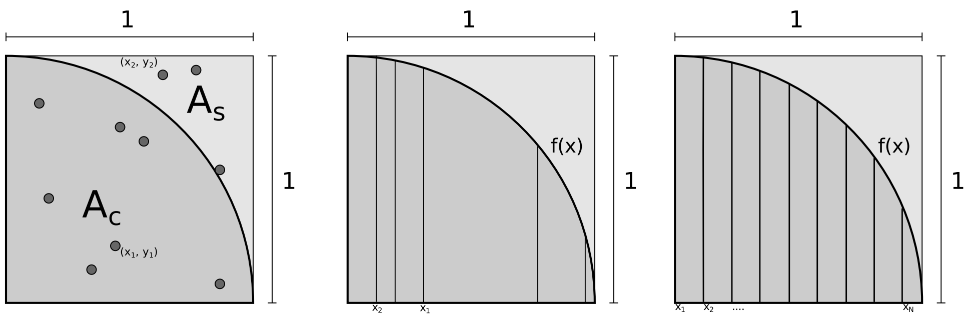

Abbildung 1: Drei Methoden zur Abschätzung von π (siehe Aufgaben 4.2, 4.3 und 4.4)

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

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

Wir nähern damit die Fläche des Kreissegments an. 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

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

Dabei gilt: je größer N, desto genauer ist die Abschätzung.

Aufgabe: Das Python-Skript `compute_pi.py` soll die Zahl π annähern. Leider ist es fehlerhaft. Korrigiere das Skript und füge das korrigierte Skript in die Lösungsdatei ein.

Du kannst ausprobieren, wie sich die Genauigkeit und die Rechenzeit verändert, wenn Du die Anzahl der gezogenen Punkte veränderst.

Aufgabe 4.3: πthon 2 (3 Punkte)

Die in Aufgabe 4.2 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 also eine Reihe x_i von zufällige 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.

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: Verändere das in Aufgabe 4.2 verwendete Python-Skript `compute_pi.py` so, daß es die Zahl π mit Hilfe der hier beschriebenen Methode annähert. Füge das veränderte Skript in die Lösungsdatei ein.

Aufgabe 4.4: πthon 3 (3 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.

Aufgabe: Verändere das in Aufgabe 4.3 verwendete Python-Skript `compute_pi.py` so, daß es die Zahl π mit Hilfe der hier beschriebenen Methode annähert. Füge das veränderte Skript in die Lösungsdatei ein.