Übungen zu Computergrundlagen WS 2014/2015

Übungsblatt 8: Python V

03.12.2014

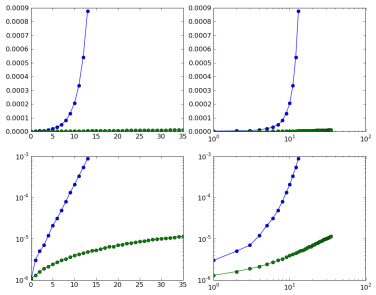
Allgemeine Hinweise

- Abgabetermin für die Lösungen ist
  - **Freitag, 12.12., 10:00**
- Schickt die Lösungen bitte per Email an Euren Tutor.

Aufgabe 8.1: Aufwand der Berechnung von Fibonacci und Matplotlib (3 Punkte)

Das Skript `/group/cgl/2014/08/plotfib.py` enthält die Funktionen `fib1` und `fib2` vom letzten Übungsblatt und plottet die Fibonaccireihe.

- **8.1.1** Erweitert das Skript wie folgt: (2 Punkte)
  - Es soll die Laufzeiten $t(n)$ messen, die benötigt werden, um die Funktion `fib1(n)` für Werte von $n \leq 17$ bzw. die Funktion `fib2(n)` für Werte von $n \leq 40$ zu berechnen.
  - Es soll mit Hilfe der `matplotlib` einen Plot erstellen, in dem die Laufzeiten $t(n)$ der verschiedenen Funktionen gegen n aufgetragen sind. Dieser soll aus vier Subplots bestehen, in denen jeweils die Plotfunktionen `plot`, `semilogx`, `semilogy` und `loglog` verwendet werden, um beide Kurven darzustellen. Der Plot sollte ungefähr so aussehen wie in nebenstehender Abbildung.



Hinweis: Verwendet die Funktion `time.clock()` zum Messen der Zeiten. Die Aufrufe der Funktion sind zu schnell und `time.clock()` zu ungenau, um die Zeiten einzelner Aufrufe der Funktion zu messen. Messt deswegen die Zeit, die Python benötigt, um `fib1(n)` jeweils 10000 mal und `fib2(n)` jeweils 100000 mal auszuführen, und berechne dann daraus die Zeit für einen einzelnen Funktionsaufruf.

- **8.1.2** Aus den Plots kann man Aussagen über den asymptotischen Aufwand der Funktionen `fib1` und `fib2` treffen. Welchen asymptotischen Aufwand zeigen die Funktionen (exponentiell, polynomial, linear)? (1 Punkt)

Aufgabe 8.2: Konvergenz der numerischen Integration (7 Punkte)

Die auf Blatt 4 beschriebene Methode zur Berechnung von π ist nicht besonders effizient. Nun wollen wir die Genauigkeit (und damit die Effizienz) der Methode erhöhen. Betrachten wir nun die Kreislinie als Funktion

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

dann 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 das Integral und damit 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. Diese Methode, eine Funktion mit Hilfe von Zufallszahlen zu integrieren, nennt sich *Monte Carlo-Methode*, benannt nach den Spielcasinos des Stadtteils von Monaco.

- **8.2.1** Schreibt eine Python-Funktion `compute_pi_mc(N)`, dass die Monte Carlo-Integration mit N Stützstellen verwendet, um π zu berechnen. (2 Punkte)

Die oben beschriebene Monte Carlo-Methode ist recht einfach und funktioniert bei beliebigen und auch hochdimensionalen Funktionen. 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, also $x_i = i/N$ für $i = 0,1,\ldots,N-1$.

- **8.2.2** Schreibe eine Python-Funktion `compute_pi_rect(N)`, dass anstelle von zufällig gezogenen Sütztstellen ein gleichmäßiges Gitter von N Stützstellen verwendet. (1 Punkt)

Da wir bereits gelernt haben, dass numpy auf effiziente Numerik ausgelegt ist, können wir die Berechnung durch die Verwendung von `numpy`-Arrays weiter beschleunigen.

- **8.2.3** Schreibe eine weitere Python-Funktion `compute_pi_rect_np(N)`, die π *ohne eine explizite Schleife* mit Hilfe von numpy berechnet. Erzeuge dazu zunächst einen Vektor von äquidistanten Stützstellen und berechne dann die Werte $f(x_i)$ in einem weiteren numpy-Array. Nutze dann `numpy.sum()`, um das Integral auszuwerten. (2 Punkte)

- **8.2.4** Bestimme für die drei Methoden zur Berechnung von π die Laufzeitabhängigkeit von der Anzahl der Stützstellen $N = 2^i$ für $i \leq 20$. Mittle hierfür über jeweils 100 Integrationen analog zu dem Hinweis in Aufgabe 8.1.1. Plotte die Ergebnisse ebenfalls wie in Aufgabe 8.1.1. (2 Punkte)

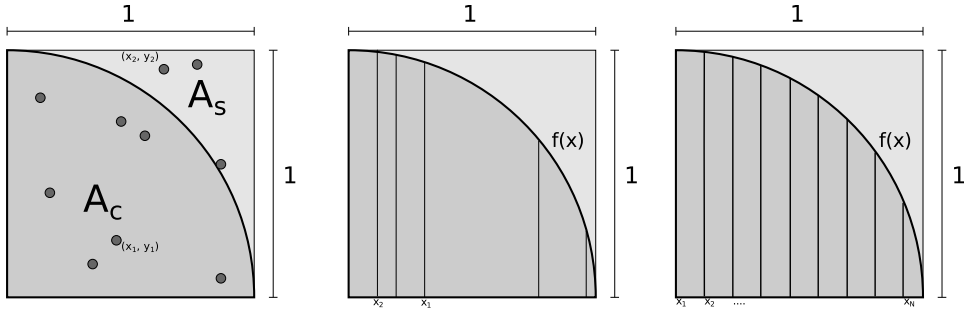


Abbildung 1: Die drei Methoden zur Abschätzung von π. Links die zweidimensionale Schießmethode, in der Mitte die Monte Carlo-Integration von $f(x) = \sqrt{1 - x^2}$ und rechts die Integration mittels äquidistanter Stützstellen.