Übungen zu Computergrundlagen WS 2011/2012

Übungsblatt 7: Asymptotisches Verhalten, Matplotlib und Fließkommazahlen

29. November 2011

Allgemeine Hinweise

Abgabetermin für die Lösungen ist

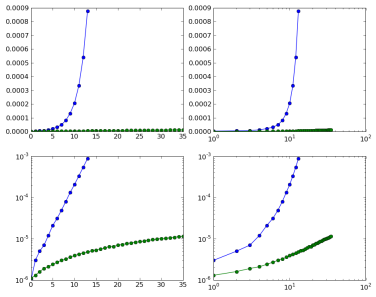
- **Montag, 5.12., 13:00** für die Übungsgruppen am Mittwoch und Donnerstag
- **Donnerstag, 8.12., 13:00** für die Übungsgruppen am Montag und Dienstag

Die Lösungen solltest Du in eine Kopie der Datei `/share/Courses/CG2011/07/vorlage07.txt` einfügen. Zur Abgabe schickst Du die Lösungsdatei im Anhang einer Email an Deinen Tutor.

Aufgabe 7.1: Aufwand der Berechnung von Fibonacci und Matplotlib (3 Punkte)

Das Skript `/share/Courses/CG2011/07/plotfib.py` enthält die Funktionen `fib1` und `fib2` vom letzten Übungsblatt und plottet die Fibonaccireihe.

- 7.1.1 (2 Punkte) Erweitere das Skript wie folgt:
  - 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.



  Füge das erweiterte Skript in die Lösungsdatei ein.

  Hinweis Verwende 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. Miss 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.

- 7.1.2 (1 Punkt) 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)?

Aufgabe 7.2: Effizienz und Konvergenz der numerischen Integration (3 Punkte)

Das Skript `/share/Courses/CG2011/07/plotpi.py` enthält die Funktionen `compute_pi1`, `compute_pi2` und `compute_pi3`. Diese schätzen die Kreiszahl π mit Hilfe der Algorithmen von Übungsblatt 4 ab. Außerdem berechnet das Skript den Fehler der Abschätzungen für jeweils N Schritte (also den Betrag der Differenz von π und der Abschätzung).

- 7.2.1 (2 Punkte) Erweitere das Skript wie folgt:
  - Es soll den Fehler der Abschätzung von π für unterschiedliche Werte von N messen. N soll die Werte der Zweierpotenzen von 1 bis 21 annehmen, also $N = 2^i$ für $i \leq 21$.
  - Der Fehler für jeden Wert von N soll über 10 Läufe gemittelt werden.
  - Es soll einen doppeltlogarithmischen Plot (`loglog`) erstellen, in dem der (über 10 Läufe gemittelte) Fehler der drei Funktionen über der Anzahl Schritte N aufgetragen ist.

  Füge das Skript in die Lösungsdatei ein.
- 7.2.2 (1 Punkt) Aus dem Plot kann man Aussagen über den asymptotischen Genauigkeit der Funktionen treffen. Welche asymptotische Genauigkeit zeigen die Funktionen (exponentiell, polynomial, linear)?

Aufgabe 7.3: Fließkommazahlen (4 Punkte)

- 7.3.1 (2 Punkt) Berechne die 32-bit Fließkommadarstellungen („einfache Genauigkeit“) der folgenden Zahlen nach IEEE-754-Standard (`http://de.wikipedia.org/wiki/IEEE_754`). Schreibe den Lösungsweg und das Ergebnis im Hexadezimalsystem in die Lösungsdatei.
  - $\pi = 3.14159265358979323846$
  - $\frac{22}{7} = 3.1428571428571428$

  Hinweis Um zu verstehen, wie das funktioniert, darfst Du die o. g. Wikipediaseite verwenden. Auch darfst Du Python verwenden, um z.B. mit Hilfe der Funktion `hex` Dezimalzahlen in Hexadezimalzahlen umzuwandeln.
- 7.3.2 (1 Punkt) Wieviele Dezimalstellen Genauigkeit haben diese Darstellungen?
- 7.3.3 (1 Punkt) Berechne $\frac{22}{7} - \pi$ mit Hilfe der IEEE-Fließkommaarithmetik und trage das Ergebnis im Hexadezimalsystem in die Lösungsdatei ein. Wie viele Dezimalstellen Genauigkeit hat das Ergebnis?