Vorsemesterkurs
Programmieren mit Scilab

Die Lösungen sind bis Freitag, 09.10.2015, 12:00 Uhr mit dem Betreff “Vorsemesterkurs Scilab Übungsblatt 4” an die Adresse `scilab2015@math.uni-frankfurt.de` einzusenden. Bitte vermerken Sie die Namen und die Matrikelnummer aller beteiligten Studenten in der Einsendung.

Aufgabe 4.1

Gegeben sei der folgende Pseudo-Code:

```
Eingabe:  Matrix A
    // n-1 Iterationsschritte
   for i := 1 to n-1
       // Zeilen der Restmatrix werden durchlaufen
       for k := i+1 to n
          // Berechnung von L
          A(k,i) := A(k,i) / A(i,i) // Achtung:  Division durch 0
             // Spalten der Restmatrix werden durchlaufen
             for j := i+1 to n
                // Berechnung von R
                A(k,j) := A(k,j) - A(k,i) * A(i,j)
Ausgabe:  Matrix A (in modifizierter Form)
```

a) **(3 Punkte)** Übertragen Sie den Pseudo-Code in eine Scilab-Code und testen Sie diesen mit der Matrix

$$A = \begin{pmatrix} 8 & 2 & 9 \\ 4 & 9 & 4 \\ 6 & 7 & 9 \end{pmatrix}.$$

b) **(2 Punkte)** Beschreiben Sie, was in der i-ten Iteration abläuft und wie sich die Zeilen bzw. Spalten der Matrix A verändern.

c) **(3 Punkte)** Geben Sie an für welche Matrizen $A \in \mathbb{R}^{d\times d}$ der Code nicht funktioniert. Modifizieren Sie ihr Programm derart, dass, falls A nicht vom Code angenommen wird, ein Fehler ausgegeben wird.

   Bitte nutzten Sie jewals eine eigene Datei für die Teilaufgaben **a)** und **c)**.

Aufgabe 4.2

Falls das Riemann-Integral $\int_a^b f(x)\,\mathrm{d}x$ einer Funktion $f : [a,b] \to \mathbb{R}$ existiert, gilt

$$\int_a^b f(x)\,\mathrm{d}x \approx \sum_{n=0}^{N-1} f(\xi_n)\frac{b-a}{N},$$

wenn man $\xi_n \in [a+(b-a)\frac{n}{N}, a+(b-a)\frac{n+1}{N}]$ und N gröss genug wählt. Wir können also die rechte Seite als simple Approximation des Integrals verwenden.

a) **(3 Punkte)** Schreiben Sie eine Funktion `integriere(f,a,b,N)`, die das Integral von f in diesem Sinne approximiert. Verwenden Sie die Scilabroutine `sum`, um Ihre Funktion möglichst einfach zu gestalten.

b) **(1 Punkt)** Testen Sie Ihre Funktion an der Sinusfunktion und den Daten `a=0`, `b=pi` und `N=20` sowie `a=pi`, `b=0` und `N=20`.

c) **(2 Punkte)** Vergleichen Sie die Werte ihrer Riemann-Integral-Funktion mit der Scilab-Funktion `integrate`. Verwenden Sie dazu die Daten `a=0` und `b=pi` sowie `N=20, 30, 40, ..., 1000`.

Aufgabe 4.3 (Optional)

Die Fibonacci-Zahlen genügen der folgenden Rekursion:

$$f_n = f_{n-1} + f_{n-2}$$

für $n \geq 2$ mit $f_0 = 0$ und $f_1 = 1$. Die ersten 10 Zahlen lauten also: $0, 1, 1, 2, 3, 5, 8, 13, 21$ und 34.

a) Implementieren sie eine Funktion `fibo`, welche die Fibonacci-Zahlen rekursiv berechnet.

b) Eine weitere Möglichkeit die Laufzeit zu ermitteln ist die vollständige Induktion. Der Induktionsanfang für unsere rekursive Funktion ist in diesem Fall $fibo(0) < 2^0$ und $fibo(1) < 2^1$. Zeigen sie den Induktions- -schritt für $fibo(n)$ und geben sie eine obere Schranke der Laufzeit in der O-Notation an.

c) Messen sie wiederum die Laufzeit mittels `tic()` und `toc()` und erzeugen sie den loglog-Plot für $n = 10, \ldots, 30$.

d) Die tatsächliche Laufzeit beträgt $l(n) = c \cdot \left(\frac{1+\sqrt{5}}{2}\right)^n$. Bestimmen sie die Konstante c und fügen sie l ihrem Plot aus Teilaufgabe **b)** hinzu. Schätzen sie wie lange ihr Algorithmus für $n = 100$ laufen würde (in Jahren).