Dirk Praetorius
Christoph Überhuber

Sommersemester 07
09. Mai 2007

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 7

Die Aufgaben mit Stern (*) sind bis zur nächsten Übung vorzubereiten und werden dort abgeprüft. Die übrigen Aufgaben dienen nur Ihrer Übung und mir als zusätzliche Grundlage für den Prüfungsstoff in den schriftlichen Tests. Kopieren Sie bitte den Source-Code in ein Unterverzeichnis `serie07` *Ihres Home-Verzeichnisses. Überprüfen Sie bitte vor der Übung, ob Ihre Source-Codes mit* `matlab` *interpretiert werden können. In den folgenden Aufgaben sollen im wesentlichen* ***Arithmetik****,* ***Verzweigungen*** *und* ***Funktionen*** *geübt werden.*

Aufgabe 61*. Die Summe $r = p + q$ zweier Polynome p, q ist wieder ein Polynom. Man schreibe eine Funktion `addPolynomials`, die die Summe r berechnet. Dabei sollen $p(x) = \sum_{k=1}^{m} a_k x^{k-1}$ und $q(x) = \sum_{k=1}^{n} b_k x^{k-1}$ in Form der Zeilenvektoren $a \in \mathbb{R}^m$ und $b \in \mathbb{R}^n$ ihrer Koeffizienten gespeichert werden. Verwenden Sie keine Schleifen, sondern lediglich die Matlab-Arithmetik!

Aufgabe 62*. Man schreibe eine Funktion `evalPolynomial`, die den Funktionswert $p(x) = \sum_{k=1}^{n} a_k x^{k-1}$ zurückgibt. Dabei sollen keine Schleifen, sondern lediglich die Matlab-Arithmetik verwendet werden. Falls x ein Spaltenvektor der Länge m ist, soll $p(x)$ ebenfalls ein Spaltenvektor der Länge m sein.

Aufgabe 63*. Schreiben Sie eine Funktion `differentiatePolynomial`, die den Koeffizientenvektor der Ableitung $p'(x)$ des Polynoms $p(x) = \sum_{k=1}^{n} a_k x^{k-1}$ zurückgibt. Dabei sollen keine Schleifen, sondern lediglich die Matlab-Arithmetik verwendet werden.

Aufgabe 64*. Schreiben Sie eine Funktion `minabs`, die von zwei Werten $x, y \in \mathbb{R}$ denjenigen zurückliefert, dessen Absolutbetrag kleiner ist. Der Absolutbetrag wird in Matlab durch `abs` gegeben. Realisieren Sie die Funktion mit und ohne Verwendung von `min`.

Aufgabe 65. Schreiben Sie eine Funktion `rundung`, die für eine gegeben Zahl $x \in \mathbb{R}$ die Zahl $n \in \mathbb{N}$ zurückliefert, die x am nächsten liegt. Falls x genau in der Mitte zwischen zwei ganzen Zahlen liegt, werde die größere zurückgeliefert.

Aufgabe 66. Schreiben Sie eine Funktion `cut`, die zu gegebenem $k \in \mathbb{N}$ aus einem Vektor $x \in \mathbb{R}^n$ alle Einträge x_j mit $|x_j| \geq k$ streicht. Anstatt Schleifen soll der Befehl `find` verwendet werden.

Aufgabe 67. Schreiben Sie eine Funktion `skalarprodukt`, die das Skalarprodukt zweier Vektoren $x, y \in \mathbb{R}^n$ berechnet, ohne Schleifen zu verwenden. Dabei dürfen x und y Spalten- oder Zeilenvektoren sein, die ggf. mittels `reshape` auf passende Form gebracht werden.

Aufgabe 68. Schreiben Sie eine Funktion `pnorm`, die für $p \in [1, \infty)$ die ℓ_p-Norm

$$\|x\|_p := \Big(\sum_{j=1}^{n} |x_j|^p \Big)^{1/p}$$

eines Vektors $x \in \mathbb{R}^n$ berechnet. Dabei sollen keine Schleifen verwendet werden. Die Summation realisiere man mittels `sum`.

Aufgabe 69. Man schreibe eine rekursive Funktion `binomial`, die den Binomialkoeffizienten $\binom{n}{k}$ berechnet. Dazu verwende man das Additionstheorem $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$. Ferner schreibe man eine Funktion `binomial2`, die den Binomialkoeffizienten mittels $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ berechnet. Dazu ist eine rekursive Funktion `faktorielle` zu entwickeln, die zu gegebenem $n \in \mathbb{N}$ die Faktorielle $n!$ berechnet.

Aufgabe 70. Es sei $L \in \mathbb{R}^{n \times n}$ eine untere Dreiecksmatrix mit $\ell_{jj} \neq 0$ für alle $j = 1, \ldots, n$. Dann ist L invertierbar, und die Inverse läßt sich wie folgt rekursiv berechnen: Wir schreiben L in Block-Form

$$L = \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix}$$

mit $L_{11} \in \mathbb{R}^{p \times p}$, $L_{21} \in \mathbb{R}^{q \times p}$ und $L_{22} \in \mathbb{R}^{q \times q}$, wobei $n = p + q$ gilt. Man beachte, dass L_{11} und L_{22} wieder untere Dreiecksmatrizen sind mit $\ell_{jj} \neq 0$. Üblicherweise wählt man $p = n/2$, falls n gerade ist, bzw. $p = (n-1)/2$, falls n ungerade ist. Offensichtlich wird die Inverse L^{-1} dann durch

$$L^{-1} = \begin{pmatrix} L_{11}^{-1} & 0 \\ -L_{22}^{-1} L_{21} L_{11}^{-1} & L_{22}^{-1} \end{pmatrix}$$

gegeben. Man schreibe eine Funktion `invertL`, die die Inverse L^{-1} rekursiv berechnet. Sie können die Korrektheit Ihrer Funktion mit Hilfe der Matlab-Funktion `inv` überprüfen.