Dirk Praetorius
Christoph Überhuber

Wintersemester 2007/08
12. Dezember 2007

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 9

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 `serie09` *Ihres Home-Verzeichnisses. Überprüfen Sie bitte vor der Übung, ob Ihre Source-Codes mit* `matlab` *auf der* `cad.zserv.tuwien.ac.at` *interpretiert werden können. Dabei soll neben der* MATLAB*-Arithmetik auch die Zählschleife (siehe* `help for`*) geübt werden.*

Aufgabe 81*. 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. Realisieren Sie die Funktion sowohl mittels geeigneter Schleifen analog zu C als auch unter Vermeidung von Schleifen mithilfe der MATLAB-Arithmetik.

Aufgabe 82*. Man schreibe eine Funktion `evalPolynomial`, die den Funktionswert $p(x) = \sum_{k=1}^{n} a_k x^{k-1}$ zurückgibt. Dabei werde das Polynom in einem Zeilenvektor $a \in \mathbb{R}^n$ der Länge n gespeichert. Falls x ein Spaltenvektor der Länge m ist, soll $p(x)$ ebenfalls ein Spaltenvektor der Länge m sein. Realisieren Sie die Funktion wieder unter Verwendung sowie unter Vermeidung von Schleifen.

Aufgabe 83*. Man schreibe eine Funktion `matrixvectorU`, die das Matrix-Vektor-Produkt $y = Ux$ mit einer oberen Dreiecksmatrix $U \in \mathbb{R}^{n\times n}$ mittels geeigneter Schleifen berechnet, vgl. Aufgabe 78 zur Definition einer oberen Dreiecksmatrix. Dabei soll auf die offensichtlichen Nulleinträge von U nicht zugegriffen werden, um den Rechenaufwand gering zu halten. Sie können Ihre Funktion mittels `U*x` verifizieren. Eine zufällige obere Dreiecksmatrix $U \in \mathbb{R}^{n\times n}$ können Sie beispielsweise mit `[tmp,U] = qr(rand(n));` erzeugen. Was machen diese beiden Befehle?

Aufgabe 84*. Gegeben sei eine obere Dreiecksmatrix $U \in \mathbb{R}^{n\times n}$ mit $u_{jj} \neq 0$ für alle $j = 1, \dots, n$. Zu gegebenem $y \in \mathbb{R}^n$ existiert dann ein eindeutiges $x \in \mathbb{R}^n$ mit $Ux = y$. Man schreibe mittels geeigneter Schleifen eine Funktion `solveU`, die x berechnet. Dabei soll auf die offensichtlichen Nulleinträge von U nicht zugegriffen werden, um den Rechenaufwand gering zu halten. Sie können Ihre Funktion mittels `U\y` verifizieren. — Um den Algorithmus herzuleiten, schreibe man das Matrix-Vektor-Produkt $y = Ux$ komponentenweise für y_j mit $j = 1, \dots, n$

als Summe hin. Man überlege, wie die spezielle Gestalt von U die Laufindizes der Summe vereinfacht und löse diese Gleichung nach x_j auf.

Aufgabe 85. Das Produkt $U = AB$ zweier oberer Dreiecksmatrizen $A, B \in \mathbb{R}^{n\times n}$ ist wieder eine obere Dreiecksmatrix. Man beweise diese Aussage zunächst mathematisch, indem man sich die Formel für das Matrix-Matrix-Produkt hinschreibe und mittels der Voraussetzung an A und B die Indizes vereinfache. Danach schreibe man eine Funktion `matrixmatrixU`, die die Produktmatrix berechnet und zurückgibt. Dabei sollen natürlich nur die nicht-trivialen Einträge von U, d.h. U_{jk} für $j \le k$, berechnet werden. Ferner soll auf die trivialen Einträge von A und B nicht zugegriffen werden, d.h. man verwende die anfangs hergeleitete Formel. — Sie können Ihre Funktion mittels `A*B` verifizieren.

Aufgabe 86. 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. Realisieren Sie die Funktion wieder in zwei Varianten, erstens mit und zweitens ohne Verwendung von Schleifen.

Aufgabe 87. 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. Realisieren Sie die Funktion erstens mittels geeigneter Schleifen und zweitens ohne Verwendung von Schleifen. Im zweiten Fall realisiere man die Summation mittels `sum`.

Aufgabe 88. 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 $|x_j| \ge k$ streicht. Anstatt Schleifen soll der Befehl `find` verwendet werden.

Aufgabe 89. Man schreibe eine Funktion `matrixvectorL`, die das Matrix-Vektor-Produkt $y = Lx$ mit einer unteren Dreiecksmatrix L berechnet, vgl. Aufgabe 78 zur Definition einer unteren Dreiecksmatrix. Dabei soll auf die offensichtlichen Nulleinträge von L nicht zugegriffen werden, um den Rechenaufwand gering zu halten.

Aufgabe 90. Gegeben sei eine untere Dreiecksmatrix $L \in \mathbb{R}^{n\times n}$ mit $\ell_{jj} \neq 0$ für alle $j = 1, \dots, n$. Zu gegebenem $y \in \mathbb{R}^n$ existiert dann ein eindeutiges $x \in \mathbb{R}^n$ mit $Lx = y$. Man schreibe eine Funktion `solveL`, die x berechnet. Dabei soll auf die offensichtlichen Nulleinträge von L nicht zugegriffen werden, um den Rechenaufwand gering zu halten.