Dirk Praetorius,
Michele Ruggeri

Wintersemester 2016/17
17.11.2016

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 7

Aufgabe 7.1. Die Frobeniusnorm einer Matrix $A \in \mathbb{R}^{m\times n}$ ist durch

$$\|A\|_F := \Big(\sum_{j=1}^{m}\sum_{k=1}^{n} A_{jk}^2\Big)^{1/2}$$

definiert. Schreiben Sie eine Funktion `frobeniusnorm`, die für gegebene Matrix A und gegebene Dimensionen $m,n \in \mathbb{N}$ die Frobeniusnorm berechnet. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem die Zeilen- und Spaltendimensionen $m,n \in \mathbb{N}$ und A eingelesen werden und $\|A\|_F$ ausgegeben wird. Die Matrix A soll dabei als dynamische Matrix (vom Typ `double**`) realisiert werden. Speichern Sie den Source-Code unter `frobeniusnorm.c` in das Verzeichnis `serie07`.

Aufgabe 7.2. Der Laplacesche Entwicklungssatz für Determinanten besagt, dass für ein beliebiges $j \in \{1,\dots,n\}$ gilt, dass

$$\det A = \sum_{i=1}^{n} (-1)^{i+j} \cdot a_{ij} \cdot \det A_{ij} \tag{1}$$

wobei A_{ij} die $(n-1)\times(n-1)$-Untermatrix von A ist, die durch Streichen der i-ten Zeile und j-ten Spalte entsteht. Schreiben Sie eine rejursive Funktion `detlaplace`, die die Determinante $\det(A)$ einer Matrix $A \in \mathbb{R}^{n\times n}$ mit Hilfe des Laplaceschen Entwicklungssatzes berechnet. Speichern Sie den Source-Code unter `detlaplace.c` in das Verzeichnis `serie07`.
Hinweis: Für eine 1×1-Matrix $A \in \mathbb{R}^{1\times 1}$ gilt $\det(A) = A$.

Aufgabe 7.3. Nicht jede Matrix $A \in \mathbb{R}^{n\times n}$ hat eine normalisierte LU-Zerlegung $A = LU$, d.h.

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ \ell_{21} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \ell_{n1} & \dots & \ell_{n,n-1} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & u_{n-1,n} \\ 0 & \dots & 0 & u_{nn} \end{pmatrix}.$$

Wenn aber A eine normalisierte LU-Zerlegung besitzt, so gilt

$$\begin{aligned} u_{ik} &= a_{ik} - \sum_{j=1}^{i-1} \ell_{ij} u_{jk} \quad \text{für } i = 1,\dots,n, \quad k = i,\dots,n, \\ \ell_{ki} &= \frac{1}{u_{ii}} \Big(a_{ki} - \sum_{j=1}^{i-1} \ell_{kj} u_{ji}\Big) \quad \text{für } i = 1,\dots,n, \quad k = i+1,\dots,n, \\ \ell_{ii} &= 1 \quad \text{für } i = 1,\dots,n, \end{aligned}$$

wie man leicht über die Formel für die Matrix-Matrix-Multiplikation zeigen kann. Alle übrigen Einträge von $L,U \in \mathbb{R}^{n\times n}$ sind Null. Schreiben Sie eine Funktion `computeLU`, die die LU-Zerlegung von A berechnet und zurückgibt. Dazu überlege man, in welcher Reihenfolge man die Einträge von L und U berechnen muss, damit die angegebenen Formeln wohldefiniert sind (d.h. alles was benötigt wird, ist bereits zuvor berechnet worden). Schreiben Sie ein main-Programm, in dem Sie die Funktion `computeLU` an einen geeigneten Beispiel testen. Speichern Sie den Source-Code unter `computeLU.c` in das Verzeichnis `serie07`.

Aufgabe 7.4. Um die Determinante einer Matrix $A \in \mathbb{R}^{n\times n}$ zu berechnen, ist der Laplacesche Entwicklungssatz aus Aufgabe 7.2 kein geeignetes Mittel (Sie sind eingeladen, dies praktisch zu erfahren!). Es ist besser, man berechnet die normalisierte LU-Zerlegung aus Aufgabe 7.3. Es gilt nämlich $\det(A) = \det(L)\det(U) = \det(U) = \prod_{j=1}^{n} u_{jj}$. Schreiben Sie eine Funktion `det(A)`, die die Determinante einer Matrix A mittels der normalisierten LU-Zerlegung berechnet.

Aufgabe 7.5. Schreiben Sie eine Funktion `dec2float`, die für eine gegebene Dezimalzahl $x \in \mathbb{R}_{>0}$ und eine Mantissenlänge $M \in \mathbb{N}$ die Ziffern $a_1, \dots, a_M \in \{0, 1\}$ und den Exponenten $e \in \mathbb{Z}$ der normalisierten Gleitkommadarstellung (d.h. $a_1 = 1$) berechnet und zurückgibt. Schreiben Sie ein aufrufendes Hauptprogramm, in dem x eingelesen und die Gleitkommadarstellung von x ausgegeben wird. Speichern Sie den Source-Code unter `dec2float.c` in das Verzeichnis `serie07`.

Aufgabe 7.6. Schreiben Sie eine Funktion `palindrom`, welches überprüft ob ein Wort (String) ein Palindrom ist. Ein Palindrom ist ein Wort, welches von vorne und hinten gelesen gleich lautet, z.B.: Anna, Otto, Reliefpfeiler. Schreiben Sie ferner ein aufrufendes Hauptprogramm, welches das Wort einliest und `palindrom` aufruft. Testen Sie ihr Programm entsprechend! Speichern Sie den Source-Code unter `palindrom.c` in das Verzeichnis `serie07`.

Aufgabe 7.7. Schreiben Sie eine Struktur `Vector` zur Speicherung von `double`-Vektoren der Länge n. Die Struktur enthalte neben der Dimension n den dynamischen Datenvektor. Im Gegensatz zur üblichen Indizierung in `C` (bzw. zur Indizierung in der Vorlesung) soll die Indizierung der Vektorkoeffizienten in Ihrer Struktur `Vector` von 1 bis n laufen (wie in der Mathematik üblich). Ferner schreibe man die zugehörigen Funktionen `newVector`, `delVector`, `getVectorLength`, `setVectorLength`, `getVectorEntry`, `setVectorEntry`, wobei `setVectorLength` den Speichervektor reallokieren soll. Stellen Sie mittels `assert` sicher, dass die Dimension n in `Vector` immer positiv ist. Testen Sie Ihren Code entsprechend. Speichern Sie den Source-Code unter `vector.c` in das Verzeichnis `serie07`.

Aufgabe 7.8. Schreiben Sie eine Funktion `vectorCut`, die für gegebenes $C > 0$ und gegebenen Vektor $x \in \mathbb{R}^n$ einen Vektor y erzeugt, bei dem alle Einträge x_j mit $|x_j| > C$ aus den Vektor $x \in \mathbb{R}^n$ gestrichen sind, und den gekürzten Vektor y zurückgibt. Als Beispiel liefere die Funktion für $x = (4, 9, -2, 7, 1, -8)$ und $C = 4.2$ den Vektor $y = (4, -2, 1)$. Verwenden Sie die Struktur `Vector` von Aufgabe 7.7. Testen Sie Ihren Code entsprechend. Speichern Sie den Source-Code unter `vectorcut.c` in das Verzeichnis `serie07`.