Serie 1

LR-Zerlegung und Permutationsmatrizen

Aufgabe 1.1 (2 + 2 + 2 Punkte)**:** In dieser Aufgabe wollen wir ein lineares Gleichungssystem $\mathbf{Ax} = \mathbf{b}$ mithilfe der LR-Zerlegung der Matrix $\mathbf{A}$ lösen.

a) Schreibe eine Matlab-Funktion

```
[L, R] = lr_band(A, p, q),
```

welche die LR-Zerlegung von $\mathbf{A}$ berechnet, wobei $\mathbf{A}$ eine Bandmatrix mit oberer Bandbreite p und unterer Bandbreite q ist.

Verwende diese, um die LR-Zerlegung der Matrizen

$$\text{(i)} \quad \mathbf{A} = \begin{pmatrix} 2 & 3 & 5 & 0 \\ -4 & 6 & 1 & 0 \\ 0 & 0 & 9 & 3 \\ 2 & 0 & 0 & 5 \end{pmatrix} \qquad \text{(ii)} \quad \mathbf{A} = \begin{pmatrix} 7 & 8 & 0 & 0 & 0 \\ 0 & 3 & 9 & 0 & 0 \\ 0 & 0 & 5 & 2 & 0 \\ 0 & 0 & 0 & 4 & 3 \\ 0 & 0 & -2 & 0 & 1 \end{pmatrix}$$

zu berechnen. Kontrolliere deine Lösung, indem du testest, ob $\mathbf{A} - \mathbf{LR} = \mathbf{0}$.

b) i) Benutze den Befehl `spy`, um dir das Besetzungsmuster, d.h. die Nichtnulleinträge, der Matrizen $\mathbf{A}, \mathbf{L}$ und $\mathbf{R}$ aus a) (i) und a) (ii) anzeigen zu lassen. Benutze dazu den Befehl `subplot`. Was fällt dir auf?

ii) Was passiert, wenn du die LR-Zerlegung auf die Matrix

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

anwendest?

c) Schreibe zwei Matlab-Funktionen

```
y = forward_sub(L, b, q),
x = backward_sub(R, y, p),
```

welche eine Vorwärts- beziehungsweise Rückwärtssubstituion durchführen sollen, um das Gleichungssystem zu lösen. Nutze dabei die Struktur der Bandmatrizen aus, um die Vorwärts- und die Rückwärtssubstitution effizient zu implementieren!

Verwende alle drei Matlab-Funktionen, um folgende lineare Gleichungssysteme $\mathbf{Ax} = \mathbf{b}$ numerisch zu lösen:

i) $\mathbf{A}$ aus a) (i) und $\mathbf{b} = (7, 2, 0, -14)^\top$,

ii) $\mathbf{A}$ aus a) (ii) und $\mathbf{b} = (24, 64, 5, -23, 97)^\top$.

Berechne zur Kontrolle jeweils $\mathbf{Ax} - \mathbf{b}$.

Aufgabe 1.2 (1 + 1 + 1 + 1 Punkte)**:**

a) Stelle (von Hand) die Permutationsmatrix $\mathbf{P}$ zu folgender Permutation auf:

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 1 & 2 & 5 & 4 \end{pmatrix}$$

b) Schreibe dann eine Matlab-Funktion `P = per_mat(p)`, welche diese Matrix erzeugt. Dabei bezeichnet `p` die Bilder der Zahlen 1 bis n. Im obigen Beispiel wäre `p` $= (31254)$. Wende diese Funktion auf das Beispiel in a) an und vergleiche die Ausgabe mit der Lösung von a).

c) Gegeben seien nun der Vektor $\mathbf{x}$ und die Matrix $\mathbf{A}$:

$$\mathbf{x} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{pmatrix}$$

Berechne die Produkte $\mathbf{Px}$, $\mathbf{x}^\top\mathbf{P}$, $\mathbf{PA}$, $\mathbf{AP}$, $\mathbf{AP}^\top$, $\mathbf{PAP}$ und $\mathbf{PAP}^\top$. Was fällt dir bei den Produkten auf?

d) Schreibe eine Matlab-Funktion `s = per_sign(P)`, welche die Determinante s einer Permutationsmatrix $\mathbf{P}$ (also das Signum der Permutation π) mit Hilfe der Zyklenzerlegung berechnen kann.

Betrachte dafür das Bild des ersten Basisvektors $\mathbf{e}_1 = (1, 0, \ldots, 0)$, nachdem du die Permutationsmatrix $\mathbf{P}$ darauf angewendet hast, dann das Bild des Ergebnisses u.s.w., *solange, bis* das Bild wieder der erste Basisvektor selbst ist. Dann tust du dasselbe für den nächsten Basisvektor $\mathbf{e}_i$, der selbst in noch keinem Zyklus enthalten war. Dies tust du *solange, bis* alle Basisvektoren einmal abgebildet wurden. Finde also einen Weg dir zu merken, welche der Vektoren bereits abgebildet wurden. Der Zyklus selbst muss dabei nicht abgespeichert werden. Wichtig ist nur, dass du das Signum, nachdem du einen Zyklus mit gerader Länge gefunden hast, mit dem richtigen Vorzeichen versiehst.

Hinweis: Der Befehl `find` ist hilfreich.

Teste dein Programm nun mit der Permutationsmatrix aus Teil b).

Allgemeine Informationen befinden sich auf der Website.
Zuletzt editiert am 1. März 2024.