Serie 3

Cholesky-Zerlegung

Aufgabe 3.1 (0.5 + 0.5 Punkte)**:** Lies die Beilage sorgfältig durch und beantworte folgende Theoriefragen:

a) Für welche Matrizen gibt es eine eindeutige Cholesky-Zerlegung? Wie sieht die Output-Matrix **L** einer Cholesky-Zerlegung für eine Matrix **A** aus?

b) Wofür kann eine Cholesky-Zerlegung benutzt werden? Betrachte für die Beantwortung dieser Frage auch Aufgabe 3.4.

Aufgabe 3.2 (1.5 Punkte)**:** Überlege dir drei Möglichkeiten, wie du in MATLAB die Summe

$$\sum_{i=1}^{n} v_i w_i$$

ausrechnen kannst, wobei v_i, w_i jeweils die Einträge von $\mathbf{v}, \mathbf{w} \in \mathbb{R}^n$ bezeichnen. Mindestens eine Möglichkeit soll ohne Schleifen auskommen. Diese Aufgabe soll dir bei Aufgabe 3.3 b) und d) helfen.

Aufgabe 3.3 (1 + 1 + 1 + 1 + 1 Punkte)**:** In dieser Aufgabe beschäftigen wir uns mit der Implementierung der Cholesky-Zerlegung.

a) Schreibe eine MATLAB-Funktion

`L = cholesky(A),`

welche die Cholesky-Zerlegung $\mathbf{A} = \mathbf{L}\mathbf{L}^\top$ einer Matrix **A** berechnet (siehe Beilage). Dabei soll die Funktion den unteren Dreiecksblock von **A** mit **L** überschreiben. Falls **A** keine Cholesky-Zerlegung besitzt soll die Funktion soll mit einer Fehlermeldung abbrechen und die Nullmatrix ausgeben. Dazu kannst du die Befehle `disp` und `break` verwenden.

b) Ändere nun den Algorithmus (siehe Beilage) so ab, dass du nur mit der äussersten `for`-Schleife auskommst. Nimm dir die Überlegungen von Aufgabe 3.2 zur Hilfe.

c) Teste deine Funktion für die Matrizen

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 3 & 6 & 10 & 15 \\ 1 & 4 & 10 & 20 & 35 \\ 1 & 5 & 15 & 35 & 70 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 36 & -30 & 24 & -18 \\ -30 & 74 & -62 & 50 \\ 24 & -62 & 116 & -98 \\ -18 & 50 & -98 & 164 \end{pmatrix}$$

$$\text{und} \quad \mathbf{C} = \begin{pmatrix} 2 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 2 \end{pmatrix}.$$

Berechne jeweils $\mathbf{A} - \mathbf{L}\mathbf{L}^\top$.

d) Ändere nun deine LR-Zerlegung mit Pivotisierung aus Serie 2 so ab, dass auch sie nur eine `for`-Schleife benötigt.

   Bemerkung: Falls du die LR-Zerlegung mit Pivotisierung in Serie 2 nicht implementiert hast, dann schreibe genau auf, welcher Teil und wie der Pseudocode aus Beilage 2 angepasst werden muss.

e) Generiere eine Toeplitz-Matrix $\mathbf{T}$ der Dimension $n = 1000$ mit dem erzeugenden Vektor $(1000, 999, \ldots, 2, 1)$. Der Befehl dafür lautet `toeplitz`. Zum Beispiel erzeugt die Eingabe `toeplitz(4:-1:1)` die Matrix

$$\begin{pmatrix} 4 & 3 & 2 & 1 \\ 3 & 4 & 3 & 2 \\ 2 & 3 & 4 & 3 \\ 1 & 2 & 3 & 4 \end{pmatrix}.$$

   Teste nun, ob die pivotisierte LR-Zerlegung mit der Modifizierung aus Teil d) oder die Cholesky-Zerlegung die Matrix $\mathbf{T}$ schneller zerlegt. Verwende die Befehle `tic` und `toc`, um die Zeiten zu testen. Was stellst du fest und wie erklärst du dir das Resultat?

   Bemerkung: Falls du die LR-Zerlegung mit Pivotisierung in Serie 2 nicht implementiert hast, dann vergleiche die Rechenzeit der in MATLAB implementierten Befehle `chol` und `lu`, die jeweils die Cholesky- bzw. die pivotisierte LR-Zerlegung berechnen. Da diese Befehle aufwändiger und dadurch deutlich effizienter implementiert wurden, setze $n = 5000$ um einen klaren Unterschied zwischen den Laufzeiten zur Zerlegung der Toeplitz-Matrix beobachten zu können.

Aufgabe 3.4 (0.5 + 1 + 1 Punkte)**:** Lade die Dateien `A.txt` und `b.txt` von der Website herunter und speichere die Dateien in deinem MATLAB-Ordner zur Serie 3. Sie enthalten die Einträge einer Matrix $\mathbf{A}$ und eines Vektors $\mathbf{b}$ (die Matrix stammt aus der zweidimensionalen Variante des Poisson-Problems, deren eindimensionale Variante wir in Serie 2 gesehen haben). Führe in deinem Skript zuerst die folgenden Befehle aus:

```
load A.txt
A = spconvert(A);
load b.txt
```

Dadurch werden die Matrix $\mathbf{A}$ und der Vektor $\mathbf{b}$ in MATLAB geladen.

a) Wie gross ist die Matrix $\mathbf{A}$?

b) Löse das Gleichungssystem $\mathbf{Ax} = \mathbf{b}$ auf zwei Arten: Bestimme sowohl die LR-Zerlegung als auch die Cholesky-Zerlegung von $\mathbf{A}$ und wende dann Vorwärts- und Rückwärtssubstitution an. Verifiziere deine Lösung, indem du sie mit dem von MATLAB durch `x = A\b` berechneten Lösungsvektor vergleichst (am einfachsten geht das, wenn man die Norm der Differenz der Lösungsvektoren betrachtet, d.h. `norm(x1-x2)` sollte nahe bei 0 sein).

   Bemerkung: Falls du die LR-Zerlegung mit Pivotisierung in Serie 2 nicht implementiert hast, kannst du die in MATLAB eingebaute Variante verwenden: `[L, R, P] = lu(A)`. Beachte aber, dass für die mit dem Befehl `lu` berechneten $\mathbf{L}$, $\mathbf{R}$ und $\mathbf{P}$ gilt: $\mathbf{P}^\top \mathbf{LR} = \mathbf{A}$.

c) Visualisiere die Besetzungsmuster der Matrix $\mathbf{A}$, der Matrizen $\mathbf{L}, \mathbf{R}$ und $\mathbf{P}$ aus der LR-Zerlegung und der Matrix $\mathbf{L}$ aus der Choleksy-Zerlegung jeweils mit dem Befehl `spy`. Benutze den Befehl `subplot`, um die Bilder in ein Fenster zu zeichnen. Was stellst du fest?

Allgemeine Informationen befinden sich auf der Website.
Zuletzt editiert am 15. März 2024.