



Numerische Mathematik I

3. Übung

1. Cholesky-Zerlegung: 3×3

Gegeben sei folgende Matrix:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 7 \\ 3 & 7 & 26 \end{pmatrix}$$

- Berechnen Sie die Cholesky-Zerlegung. Entscheiden Sie aus dem Algorithmus heraus, ob die Matrix positiv definit ist oder nicht.
- Lösen Sie mit Hilfe der Cholesky-Zerlegung folgendes Gleichungssystem:

$$Ax = b = \begin{pmatrix} -3 \\ -9 \\ 20 \end{pmatrix}.$$

2. Rationale Cholesky-Zerlegung

Für $A \in \mathbb{R}^{n \times n}$ betrachten wir die “Rationale Cholesky-Zerlegung”

$$A = LDL^T,$$

mit einer unipotenten unteren Dreiecksmatrix L und einer positiven Diagonalmatrix D . Im Gegensatz zu der “klassischen Cholesky-Zerlegung”

$$A = \tilde{L}\tilde{L}^T$$

mit einer unteren Dreiecksmatrix \tilde{L} , bietet die rationale Cholesky-Zerlegung in der Praxis Vorteile. Insbesondere vermeidet sie die Berechnung von Wurzeln.

Leiten Sie den Algorithmus der rationalen Cholesky-Zerlegung her.

- Geben Sie hierzu zunächst das Matrixprodukt $A = LDL^T$ für eine allgemeine positiven Diagonalmatrix D bzw. unipotenten unteren Dreiecksmatrix L an.
- Konstruieren Sie daraus rekursive Formeln zur Berechnung der rationalen Cholesky-Zerlegung und formulieren Sie einen Pseudocode.

3. Rationale Cholesky-Zerlegung II

- (a) Wie würden Sie aufbauend auf der rationalen Cholesky-Zerlegung effizient ein lineares Gleichungssystem $Ax = b$ lösen? Geben Sie dazu den Pseudocode an.
- (b) Schätzen Sie die Anzahl der für die rationale Cholesky-Zerlegung nötigen Rechenoperationen in Abhängigkeit der Dimension n ab.

4. Programmieraufgabe: LR -Zerlegung/Gauß-Algorithmus (Abgabe bis Mo, 24.10.2016, 23:59 Uhr)

Eine ausgereifte Implementierung der LR -Zerlegung einer Matrix $A = (a_{ij})$ ist gegeben durch den Algorithmus

Algorithm 1: Gauß-Elimination mit relativer Spaltenmaximumstrategie und Berechnung der Determinante

Input: Feld A
Output: $D = \det(A)$, R und L im Feld A , Zeilentausch-vektor p

```

1  $D = 1$ ;
2 for  $k = 1 : n - 1$  do                                     // Gauß-Schritte
3     max = 0;
4     for  $i = k : n$  do                                     // Pivotstrategie
5         for  $j = k : n$  do                                 // Zeilensumme  $s$  berechnen
6              $s = s + |a_{ij}|$ ;
7          $q = |a_{ik}|/s$ ;                                     // Relativieren
8         if  $q > \text{max}$  then                             // Maximum bestimmen
9              $\text{max} = q$ ;     $p_k = i$ ;
10    if max = 0 then STOP;                                 // Matrix nicht invertierbar
11    if  $p_k \neq k$  then                                    // Zeilentausch von Zeile  $k$  mit Zeile  $p_k$ 
12         $D = -D$ ;
13        for  $j = 1 : n$  do
14             $h = a_{kj}$ ;     $a_{kj} = a_{p_k j}$ ;     $a_{p_k j} = h$ ;
15     $D = D * a_{kk}$ ;
16    for  $i = k + 1 : n$  do                                  // Elimination in Spalte  $k$ 
17         $a_{ik} = a_{ik}/a_{kk}$ ;                               //  $l_{ik}$  berechnen und speichern in  $a_{ik}$ 
18        for  $j = k + 1 : n$  do
19             $a_{ij} = a_{ij} - a_{ik} * a_{kj}$ ;
20  $D = D * a_{nn}$ ;

```

Angewendet wird eine fortgeschrittene Spaltenpivotstrategie, die relative Spaltenmaximumstrategie. Dabei wird die Zeile ausgewählt, deren Pivotelement relativ (bezogen auf die Zeilensumme) am größten ist.

- (a) Programmieren Sie diese LR -Zerlegung für allgemeine Dimension $n \in \mathbb{N}$ als function in MATLAB mit dem Syntax

```
function [D,A,p] = lr_zerlegung_ZZ(A)
```

wobei ZZ Ihre Teamnummer ist.

(b) Entwerfen und programmieren Sie eine zweite `function`

```
function [x] = lr_solve_ZZ(A,b,p)
```

welche anschließend das lineare Gleichungssystem $Ax = b$ effizient löst. Beachten Sie die Vertauschungen der Zeilen.

(c) Testen Sie Ihren Code am gegebenen Skript `lr_main.m`. Achten Sie auf die Korrektheit der Ergebnisse.

Abgabe der Programmieraufgabe:

- Abgabe per email an `armin.rund@uni-graz.at`
- Betreff: Programmieraufgabe LR
- In email bitte die Namen und Matrikelnr aller beteiligten Studierenden des Teams aufführen
- Jedes Team erhält eine eindeutige Teamnummer ZZ z.B. 14
- Bitte testen Sie ihr Programm am vorgegebenen Skript `lr_main.m`
- Abzugeben sind 2 `function` Files mit vorgegebenen Namen unter Verwendung der Teamnummer, also z.B. `lr_zerlegung_14.m` und `lr_solve_14.m`
- Ihr Programm wird nach der Abgabe auf anderen Beispielen (allgemeiner Dimension n) getestet und bewertet. Der schnellste Code erhält einen Sonderpunkt.