LR-Zerlegung

Numerische Mathematik 1 (WS 2014/15)

6. November 2014

LR-Zerlegung

$$Ax = b, \quad A \in \mathbb{R}^{n \times n} \text{ nichtsingulär}, \; b \in \mathbb{R}^n$$

- Gauss-Algorithmus liefert eine LR-Zerlegung von A:

$$A = L \cdot R$$

mit

$$L = \begin{bmatrix} 1 & & & \\ * & \ddots & & \\ \vdots & \ddots & \ddots & \\ * & \dots & * & 1 \end{bmatrix}, \quad R = \begin{bmatrix} * & \dots & \dots & * \\ & \ddots & & \vdots \\ & & \ddots & \vdots \\ & & & * \end{bmatrix}$$

- Bei partieller Pivotisierung erhält man noch eine Permutationsmatrix P, so dass

$$PA = LR$$

LR-Zerlegung mit partieller Pivotisierung

Schema:

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix} \to \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \otimes & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix}$$

Pivot suchen

$$\to \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \otimes & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix} \to \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \otimes & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \end{bmatrix}$$

Permutieren und Eliminieren

Beispiel

$$A = \begin{bmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix} =: A^{[1]}$$

1. Schritt:

(a) Pivotisieren:

$$P_1 A^{[1]} = \begin{bmatrix} 4 & 9 & -3 \\ 2 & 4 & -2 \\ -2 & -3 & 7 \end{bmatrix} \text{ mit } P_1 := \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(b) Elimination

$$M_1 P_1 A^{[1]} = \begin{bmatrix} 4 & 9 & -3 \\ 0 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{3}{2} & \frac{11}{2} \end{bmatrix} \text{ mit } M_1 := \begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{bmatrix}$$

Beispiel

$$A^{[2]} := \begin{bmatrix} 4 & 9 & -3 \\ 0 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{3}{2} & \frac{11}{2} \end{bmatrix}$$

2. Schritt:

(a) Pivotisieren:

$$P_2 A^{[2]} = \begin{bmatrix} 4 & 9 & -3 \\ 0 & \frac{3}{2} & \frac{11}{2} \\ 0 & -\frac{1}{2} & -\frac{1}{2} \end{bmatrix} \text{ mit } P_2 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

(b) Elimination

$$M_2 P_2 A^{[2]} = \begin{bmatrix} 4 & 9 & -3 \\ 0 & \frac{3}{2} & \frac{11}{2} \\ 0 & 0 & \frac{4}{3} \end{bmatrix} \text{ mit } M_2 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{3} & 1 \end{bmatrix}$$

Beispiel

$$A^{[3]} := \begin{bmatrix} 4 & 9 & -3 \\ 0 & \frac{3}{2} & \frac{11}{2} \\ 0 & 0 & \frac{4}{3} \end{bmatrix} =: R$$

Rekonstruktion von R, L, P:

- da $P_2 P_2 = I_n$ erhält man

$$R = \underbrace{M_2}_{:=\hat{M}_2} \underbrace{P_2 M_1 P_2}_{=:\hat{M}_1} \underbrace{P_2 P_1}_{:=P} A$$

- und mit $L^{-1} := \hat{M}_2 \hat{M}_1$ ist

$$L = \hat{M}_1^{-1} \hat{M}_2^{-1}$$

Lösung von LGS mit LR-Zerlegung

$$Ax = b$$

Sei

$$PA = LR$$

dann löst man das lineare Gleichungssystem durch:

1. Löse $Ly = Pb$ (Vorwärtseinsetzen)
2. Löse $Rx = y$ (Rückwärtseinsetzen)

Beispiel

Für

$$b = \begin{bmatrix} 2 \\ 8 \\ 10 \end{bmatrix}, \; R = \begin{bmatrix} 4 & 9 & -3 \\ 0 & \frac{3}{2} & \frac{11}{2} \\ 0 & 0 & \frac{4}{3} \end{bmatrix}, \; L = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ \frac{1}{2} & -\frac{1}{3} & 1 \end{bmatrix}, \; P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

erhalten wir

1. $$\begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ \frac{1}{2} & -\frac{1}{3} & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 10 \\ 2 \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 14 \\ \frac{8}{3} \end{bmatrix}$$

2. $$\begin{bmatrix} 4 & 9 & -3 \\ 0 & \frac{3}{2} & \frac{11}{2} \\ 0 & 0 & \frac{4}{3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 14 \\ \frac{8}{3} \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 2 \\ 2 \end{bmatrix}$$

Implementation

- Die Subdiagonal-Elemente von $L = [l_{ij}]$ werden in den erzeugten “Nullen” der Matrix A gespeichert, d.h. im i-ten Schritt

$$A^{[i]} = \begin{bmatrix} r_{11} & \cdots & \cdots & \cdots & \cdots & r_{1n} \\ 0 & \ddots & & & & \vdots \\ \vdots & \ddots & r_{i-1,i-1} & \cdots & \cdots & r_{i-1,n} \\ \vdots & & 0 & a_{ii}^{[i]} & \cdots & a_{in}^{[i]} \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & a_{ni}^{[i]} & \cdots & a_{nn}^{[i]} \end{bmatrix}$$

Implementation

- Die Subdiagonal-Elemente von $L = [l_{ij}]$ werden in den erzeugten “Nullen” der Matrix A gespeichert, d.h. im i-ten Schritt

$$A^{[i]} = \begin{bmatrix} r_{11} & \dots & \dots & \dots & \dots & r_{1n} \\ \ell_{21} & \ddots & & & & \vdots \\ \vdots & \ddots & r_{i-1,i-1} & \dots & \dots & r_{i-1,n} \\ \vdots & & \ell_{i,i-1} & a_{ii}^{[i]} & \dots & a_{in}^{[i]} \\ \vdots & & \vdots & \vdots & & \vdots \\ \ell_{n1} & \dots & \ell_{n,i-1} & a_{ni}^{[i]} & \dots & a_{nn}^{[i]} \end{bmatrix}$$

- Es wird nur der zu P gehörige Permutationsvektor *piv* gespeichert.
  [in Matlab: $b(piv)$ liefert Pb]

Beispiel

$$
\begin{aligned}
A = \begin{bmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix} & \qquad piv = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \\
\to \begin{bmatrix} 4 & 9 & -3 \\ 2 & 4 & -2 \\ -2 & -3 & 7 \end{bmatrix} & \qquad piv = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix} \\
\to \left[\begin{array}{c|cc} 4 & 9 & -3 \\ \hline \color{red}{\frac{1}{2}} & -\frac{1}{2} & -\frac{1}{2} \\ \color{red}{-\frac{1}{2}} & \frac{3}{2} & \frac{11}{2} \end{array}\right] & \qquad piv = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix} \\
\to \left[\begin{array}{c|cc} 4 & 9 & -3 \\ \hline \color{red}{-\frac{1}{2}} & \frac{3}{2} & \frac{11}{2} \\ \color{red}{\frac{1}{2}} & -\frac{1}{2} & -\frac{1}{2} \end{array}\right] & \qquad piv = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} \\
\to \left[\begin{array}{c|c|c} 4 & 9 & -3 \\ \hline \color{red}{-\frac{1}{2}} & \frac{3}{2} & \frac{11}{2} \\ \cline{2-3} \color{red}{\frac{1}{2}} & \color{red}{-\frac{1}{3}} & \frac{4}{3} \end{array}\right] & \qquad piv = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}
\end{aligned}
$$

Implementation

- Dreieckssysteme kann man **in-place** lösen, d.h. man überschreibt die rechte Seite direkt mit der Lösung.
- Um zum Beispiel

$$\begin{bmatrix} 3 & -1 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 9 \\ 9 \\ 8 \end{bmatrix}$$

zu lösen, kann man die Folge

$$\begin{bmatrix} 9 \\ 9 \\ 8 \end{bmatrix} \to \left[\begin{array}{c} 9 \\ 9 \\ \hline \mathbf{4} \end{array}\right] \to \left[\begin{array}{c} 9 \\ \hline \mathbf{5} \\ \mathbf{4} \end{array}\right] \to \left[\begin{array}{c} \hline \mathbf{2} \\ \mathbf{5} \\ \mathbf{4} \end{array}\right] =: z$$

durchlaufen.

PA 2: LR-Zerlegung

```
function fact = lr_zerlegung(A)
n = size(A, 1);
piv = 1:1:n;

for i=1:n-1
   % Step i.a : Pivoting

   % Step i.b : Elimination
end
fact.LR = A; fact.piv = piv; fact.n  = n;
```

PA 2: Vorwärts-\ Rückwärtseinsetzen

```
function x = vor_rueck(fact, b)
LR = fact.LR; piv = fact.piv; n  = fact.n;

b = b(piv); % Compute Pb

% Solve Ly = Pb by forward substitution
for i=2:n
   b(i) = ...; % compute i-th entry of y
end

% Solve Rx = y by backward substitution
for i=n:-1:1
   b(i) = ...; % compute i-th entry of x
end

x=b;
```

Matlab Befehle

- Mit `[L,R,P]=lu(A)` bekommt man L, R und P.
- `x=A\B` löst das lineare Gleichungssystem $Ax = b$.
- `cond(A)` liefert die Konditionszahl von A bzgl. $\|.\|_2$
  `cond(A,p)` liefert die Konditionszahl von A bzgl. $\|.\|_p$, $p \in \{1, 2, \infty, F\}$