Cholesky-Zerlegung

Referent: Franz Brauße
Veranstaltung: Proseminar Numerik
Dozent: Dipl. Math. Christina Jager
Universität Trier, FB IV

16.01.2012

Gliederung

- $LDM^\top$-Zerlegung für quadratische Matrizen
- $LDL^\top$-Zerlegung für symmetrische Matrizen
- Stabilität von $LDM^\top = A$ pos. def. und nicht symmetrisch
- Cholesky-Zerlegung für A symmetrisch und pos. def.
  - Zwei Algorithmen zur Cholesky-Zerlegung
  - Erweiterung für A sym. pos. semidefinit
- Symmetrische Pivotierung

Rückblick

Zuletzt: Zerlegung von $A \in \mathbb{R}^{m \times n}$ in untere und obere Dreiecksmatrizen L und U mit $A = LU$ zur Lösung des GLS $Ax = b$, $b \in \mathbb{R}^m$.

Dann: $Ax = LUx = b$, also löse $Ly = b$ durch Rücksubstitutionen, gefolgt von $Ux = y$ ebenfalls durch Rücksubstitutionen.

- Rechenaufwand: $\frac{2}{3}n^3$ flops.
- Matrizen L, U können an den Stellen der alten Werte aus A während der Zerlegung abgespeichert werden.

Ziel: Cholesky-Zerlegung für A sym. pos. def.
Deshalb zunächst Betrachtung von $LDM^\top = A$, gefolgt von $LDL^\top = A$.

$LDM^\top$-Zerlegung

Sei im Folgenden $A \in \mathbb{R}^{n \times n}$.

Ist A nichtsingulär, so gibt es untere Dreiecksmatrizen $L, M \in \mathbb{R}^{n \times n}$ und eine Diagonalmatrix $D = \mathrm{diag}(d_1, \ldots, d_n)$, sodass $A = LDM^\top$. Diese Zerlegung ist eindeutig.

Ist die LU-Zerlegung von A bekannt, so kann daraus die $LDM^\top$-Zerlegung gewonnen werden. Setze

$$d_i = u_{ii}, \qquad i = 1, \ldots, n$$

Wegen $\mathrm{rg}(U) = \mathrm{rg}(A) = n$ folgt, dass $D = \mathrm{diag}(d_i)_i$ dann ebenfalls regulär ist.

Also existiert $D^{-1} = \operatorname{diag}\left(\frac{1}{d_i}\right)_i$, womit sich schließlich $M^\top$ ergibt zu

$$M^\top = D^{-1}U$$

L wird unverändert aus der LU-Zerlegung übernommen.

Zusammengesetzt ergibt sich

$$LDM^\top = LD(D^{-1}U) = LU = A.$$

Die Eindeutigkeit der $LDM^\top$-Zerlegung folgt direkt aus der der LU-Zerlegung: Seien $L'D'M'^\top = A = LDM^\top$ zwei Zerlegungen von A und $\overline{LU} = A$ ihre LU-Zerlegung. Dann

$$\begin{aligned}
d_i' = \overline{u}_{ii} = d_i \quad &\implies \quad D' = D \\
&\implies \quad D'^{-1} = D^{-1} \\
&\implies \quad M' = D'^{-1}\overline{U} = D^{-1}\overline{U} = M \\
\text{und nach Def.} \quad & \qquad\quad\ L' = \overline{L} = L.
\end{aligned}$$

Anderer Ansatz um $LDM^\top$ zu gewinnen

Induktiv: Berechne in jedem Schritt eine Spalte von L, eine Zeile von $M^\top$ und einen Diagonalwert von D.

Sind die Teilmatrizen $L^{(j)} = (l_{*k})_{k<j}$ und $M^{(j)} = (m_{i*})_{i<j}$, sowie die Diagonalwerte $d_1, \ldots, d_{j-1}$ von D im Schritt j bekannt, dann lässt sich daraus $L^{(j+1)}, M^{(j+1)}$ sowie d_j gewinnen:

Betrachte den Vektor v der j-ten Spalte der Gleichung $A = LDM^\top$:

$$(1) \qquad a_{*j} = Lv \qquad \text{und} \qquad v = DM^\top e_j \qquad (2)$$

Die „obere" Hälfte von (1) weist je v_i ($i = 1, \ldots, j$) als Lösung des LGS mit der bekannten unteren Dreiecksmatrix $L^{(j)}$ aus, d.h.

$$\sum_{k=1}^{n} l_{ik} v_k = \sum_{k=1}^{j} l_{ik} v_k = a_{ij}, \qquad i = 1, \ldots, j.$$

v_i $(i = 1, \dots, j)$ ist also berechnet, dies in $v = DM^\top e_j$ eingesetzt ergibt für $i = 1, \dots, j$

$$
\begin{aligned}
v_i &= \left(\sum_{k=1}^{n} d_{ik} m_{kl}^\top \right)_{l=1}^{n} \cdot e_j \\
&= \sum_{k=1}^{n} d_{ik} m_{kj}^\top \\
&= d_i m_{ij}^\top
\end{aligned}
$$

Damit lässt sich d_j und $M^{(j+1)}$ bestimmen durch

$$
\begin{aligned}
d_j &= v_j, \\
m_{ji} &= v_i / d_i, \qquad i = 1, \dots, j-1.
\end{aligned}
$$

Nach Definition $M = D^{-1} U$ gilt $m_{jj} = 1 \ \forall j$.

Die j-te Spalte von L lässt sich nun mit der unteren Hälfte von (1),

$$a_{ij} = L_{i*}^{(j)} v,$$

errechnen: Für $i = j + 1, \dots, n$ gilt

$$\begin{aligned} a_{ij} &= \sum_{k=1}^{j} l_{ik} v_k \\ \Longleftrightarrow \qquad l_{ij} v_j &= a_{ij} - \sum_{k=1}^{j-1} l_{ik} v_k . \end{aligned}$$

Beispiel

$$A = \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix}$$

$$\boxed{\begin{array}{rcll} v_i & = & \text{Lösung von } L^{(j)}_{i*} v = a_{ij}, & i = 1, \dots, j \\ d_j & = & v_j & \\ m_{ji} & = & \frac{v_i}{d_i}, & i = 1, \dots, j \\ l_{ij} & = & \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} v_k}{v_j}, & i = j+1, \dots, n \end{array}}$$

$j = 1$: $L = \left[\begin{smallmatrix} 1 & 0 \\ \cdot & 1 \end{smallmatrix}\right]$, $D = \left[\begin{smallmatrix} \cdot & 0 \\ 0 & \cdot \end{smallmatrix}\right]$, $M^\top = \left[\begin{smallmatrix} 1 & \cdot \\ 0 & 1 \end{smallmatrix}\right]$

$$\begin{array}{lcl} a_{11} = \sum_{k=1}^{1} l_{1k} v_k = l_{11} v_1 = v_1 & \Longrightarrow & v_1 = 1 = d_1 \\ l_{21} = \frac{a_{21} - \sum_{k=1}^{0} l_{ik} v_k}{v_1} = \frac{a_{21}}{1} = 4 & & \end{array}$$

$j = 2$: $L = \left[\begin{smallmatrix} 1 & 0 \\ 4 & 1 \end{smallmatrix}\right]$, $D = \left[\begin{smallmatrix} 1 & 0 \\ 0 & \cdot \end{smallmatrix}\right]$, $M^\top = \left[\begin{smallmatrix} 1 & \cdot \\ 0 & 1 \end{smallmatrix}\right]$

$$\begin{array}{lcl} a_{12} = \sum_{k=1}^{2} l_{1k} v_k = v_1 + 0 & \Longrightarrow & v_1 = 3 \\ a_{22} = \sum_{k=1}^{2} l_{2k} v_k = 4 \cdot v_1 + v_2 & \Longrightarrow & v_2 = -10 = d_2 \\ m_{21} = \frac{v_1}{d_1} = \frac{3}{1} = 3 & & \end{array}$$

$L = \left[\begin{smallmatrix} 1 & 0 \\ 4 & 1 \end{smallmatrix}\right]$, $D = \left[\begin{smallmatrix} 1 & 0 \\ 0 & -10 \end{smallmatrix}\right]$, $M^\top = \left[\begin{smallmatrix} 1 & 3 \\ 0 & 1 \end{smallmatrix}\right]$

Beispiel

$$A = \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix}$$

$$\boxed{\begin{array}{rcll} v_i & = & \text{Lösung von } L_{i*}^{(j)} v = a_{ij}, & i = 1, \dots, j \\ d_j & = & v_j & \\ m_{ji} & = & \frac{v_i}{d_i}, & i = 1, \dots, j \\ l_{ij} & = & \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} v_k}{v_j}, & i = j+1, \dots, n \end{array}}$$

$j = 1$: $L = \left[\begin{smallmatrix} 1 & 0 \\ \cdot & 1 \end{smallmatrix}\right]$, $D = \left[\begin{smallmatrix} \cdot & 0 \\ 0 & \cdot \end{smallmatrix}\right]$, $M^\top = \left[\begin{smallmatrix} 1 & \cdot \\ 0 & 1 \end{smallmatrix}\right]$

$$\begin{aligned} a_{11} &= \textstyle\sum_{k=1}^{1} l_{1k} v_k = l_{11} v_1 = v_1 \quad &\Longrightarrow \quad & v_1 = 1 = d_1 \\ l_{21} &= \frac{a_{21} - \sum_{k=1}^{0} l_{ik} v_k}{v_1} = \frac{a_{21}}{1} = 4 \end{aligned}$$

$j = 2$: $L = \left[\begin{smallmatrix} 1 & 0 \\ 4 & 1 \end{smallmatrix}\right]$, $D = \left[\begin{smallmatrix} 1 & 0 \\ 0 & \cdot \end{smallmatrix}\right]$, $M^\top = \left[\begin{smallmatrix} 1 & \cdot \\ 0 & 1 \end{smallmatrix}\right]$

$$\begin{aligned} a_{12} &= \textstyle\sum_{k=1}^{2} l_{1k} v_k = v_1 + 0 \quad &\Longrightarrow \quad & v_1 = 3 \\ a_{22} &= \textstyle\sum_{k=1}^{2} l_{2k} v_k = 4 \cdot v_1 + v_2 \quad &\Longrightarrow \quad & v_2 = -10 = d_2 \\ m_{21} &= \frac{v_1}{d_1} = \frac{3}{1} = 3 \end{aligned}$$

$$L = \left[\begin{smallmatrix} 1 & 0 \\ 4 & 1 \end{smallmatrix}\right], \ D = \left[\begin{smallmatrix} 1 & 0 \\ 0 & -10 \end{smallmatrix}\right], \ M^\top = \left[\begin{smallmatrix} 1 & 3 \\ 0 & 1 \end{smallmatrix}\right]$$

Beispiel

$$A = \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix}$$

$$\boxed{\begin{array}{rcll} v_i & = & \text{Lösung von } L_{i*}^{(j)} v = a_{ij}, & i = 1, \ldots, j \\ d_j & = & v_j & \\ m_{ji} & = & \frac{v_i}{d_i}, & i = 1, \ldots, j \\ l_{ij} & = & \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} v_k}{v_j}, & i = j+1, \ldots, n \end{array}}$$

$j = 1$: $L = \left[\begin{smallmatrix} 1 & 0 \\ \cdot & 1 \end{smallmatrix}\right]$, $D = \left[\begin{smallmatrix} \cdot & 0 \\ 0 & \cdot \end{smallmatrix}\right]$, $M^\top = \left[\begin{smallmatrix} 1 & \cdot \\ 0 & 1 \end{smallmatrix}\right]$

$$a_{11} = \sum_{k=1}^{1} l_{1k} v_k = l_{11} v_1 = v_1 \quad \Longrightarrow \quad v_1 = 1 = d_1$$

$$l_{21} = \frac{a_{21} - \sum_{k=1}^{0} l_{ik} v_k}{v_1} = \frac{a_{21}}{1} = 4$$

$j = 2$: $L = \left[\begin{smallmatrix} 1 & 0 \\ 4 & 1 \end{smallmatrix}\right]$, $D = \left[\begin{smallmatrix} 1 & 0 \\ 0 & \cdot \end{smallmatrix}\right]$, $M^\top = \left[\begin{smallmatrix} 1 & \cdot \\ 0 & 1 \end{smallmatrix}\right]$

$$a_{12} = \sum_{k=1}^{2} l_{1k} v_k = v_1 + 0 \quad \Longrightarrow \quad v_1 = 3$$

$$a_{22} = \sum_{k=1}^{2} l_{2k} v_k = 4 \cdot v_1 + v_2 \quad \Longrightarrow \quad v_2 = -10 = d_2$$

$$m_{21} = \frac{v_1}{d_1} = \frac{3}{1} = 3$$

$L = \left[\begin{smallmatrix} 1 & 0 \\ 4 & 1 \end{smallmatrix}\right]$, $D = \left[\begin{smallmatrix} 1 & 0 \\ 0 & -10 \end{smallmatrix}\right]$, $M^\top = \left[\begin{smallmatrix} 1 & 3 \\ 0 & 1 \end{smallmatrix}\right]$

Beispiel

$$A = \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix}$$

$$\boxed{\begin{array}{rcll} v_i & = & \text{Lösung von } L_{i*}^{(j)} v = a_{ij}, & i = 1, \dots, j \\ d_j & = & v_j & \\ m_{ji} & = & \frac{v_i}{d_i}, & i = 1, \dots, j \\ l_{ij} & = & \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} v_k}{v_j}, & i = j+1, \dots, n \end{array}}$$

$j = 1$: $L = \left[\begin{smallmatrix} 1 & 0 \\ \cdot & 1 \end{smallmatrix}\right]$, $D = \left[\begin{smallmatrix} \cdot & 0 \\ 0 & \cdot \end{smallmatrix}\right]$, $M^\top = \left[\begin{smallmatrix} 1 & \cdot \\ 0 & 1 \end{smallmatrix}\right]$

$$\begin{aligned} a_{11} &= \textstyle\sum_{k=1}^{1} l_{1k} v_k = l_{11} v_1 = v_1 \quad &\Longrightarrow \quad v_1 = 1 = d_1 \\ l_{21} &= \frac{a_{21} - \sum_{k=1}^{0} l_{ik} v_k}{v_1} = \frac{a_{21}}{1} = 4 \end{aligned}$$

$j = 2$: $L = \left[\begin{smallmatrix} 1 & 0 \\ 4 & 1 \end{smallmatrix}\right]$, $D = \left[\begin{smallmatrix} 1 & 0 \\ 0 & \cdot \end{smallmatrix}\right]$, $M^\top = \left[\begin{smallmatrix} 1 & \cdot \\ 0 & 1 \end{smallmatrix}\right]$

$$\begin{aligned} a_{12} &= \textstyle\sum_{k=1}^{2} l_{1k} v_k = v_1 + 0 \quad &\Longrightarrow \quad v_1 = 3 \\ a_{22} &= \textstyle\sum_{k=1}^{2} l_{2k} v_k = 4 \cdot v_1 + v_2 \quad &\Longrightarrow \quad v_2 = -10 = d_2 \\ m_{21} &= \frac{v_1}{d_1} = \frac{3}{1} = 3 \end{aligned}$$

$L = \left[\begin{smallmatrix} 1 & 0 \\ 4 & 1 \end{smallmatrix}\right]$, $D = \left[\begin{smallmatrix} 1 & 0 \\ 0 & -10 \end{smallmatrix}\right]$, $M^\top = \left[\begin{smallmatrix} 1 & 3 \\ 0 & 1 \end{smallmatrix}\right]$

Beispiel

$$A = \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix}$$

$$\boxed{\begin{array}{rcll} v_i & = & \text{Lösung von } L^{(j)}_{i*} v = a_{ij}, & i = 1, \dots, j \\ d_j & = & v_j & \\ m_{ji} & = & \frac{v_i}{d_i}, & i = 1, \dots, j \\ l_{ij} & = & \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} v_k}{v_j}, & i = j+1, \dots, n \end{array}}$$

$j = 1$: $L = \left[\begin{smallmatrix} 1 & 0 \\ \cdot & 1 \end{smallmatrix}\right]$, $D = \left[\begin{smallmatrix} \cdot & 0 \\ 0 & \cdot \end{smallmatrix}\right]$, $M^\top = \left[\begin{smallmatrix} 1 & \cdot \\ 0 & 1 \end{smallmatrix}\right]$

$$\begin{array}{ll} a_{11} = \sum_{k=1}^{1} l_{1k} v_k = l_{11} v_1 = v_1 & \Longrightarrow \quad v_1 = 1 = d_1 \\ l_{21} = \frac{a_{21} - \sum_{k=1}^{0} l_{ik} v_k}{v_1} = \frac{a_{21}}{1} = 4 & \end{array}$$

$j = 2$: $L = \left[\begin{smallmatrix} 1 & 0 \\ 4 & 1 \end{smallmatrix}\right]$, $D = \left[\begin{smallmatrix} 1 & 0 \\ 0 & \cdot \end{smallmatrix}\right]$, $M^\top = \left[\begin{smallmatrix} 1 & \cdot \\ 0 & 1 \end{smallmatrix}\right]$

$$\begin{array}{ll} a_{12} = \sum_{k=1}^{2} l_{1k} v_k = v_1 + 0 & \Longrightarrow \quad v_1 = 3 \\ a_{22} = \sum_{k=1}^{2} l_{2k} v_k = 4 \cdot v_1 + v_2 & \Longrightarrow \quad v_2 = -10 = d_2 \\ m_{21} = \frac{v_1}{d_1} = \frac{3}{1} = 3 & \end{array}$$

$L = \left[\begin{smallmatrix} 1 & 0 \\ 4 & 1 \end{smallmatrix}\right]$, $D = \left[\begin{smallmatrix} 1 & 0 \\ 0 & -10 \end{smallmatrix}\right]$, $M^\top = \left[\begin{smallmatrix} 1 & 3 \\ 0 & 1 \end{smallmatrix}\right]$

- Da die Spalte j von A nach Schritt j nicht mehr benötigt wird, können – analog zur LU-Zerlegung – dort die berechneten l_{*j}, d_j sowie $m_{*j}^\top$ abgespeichert werden:

$$\begin{bmatrix} d_1 & m_{12}^\top & \cdots & m_{1j}^\top & a_{1,j+1} & \cdots & a_{1n} \\ l_{21} & d_2 & \ddots & \vdots & \vdots & & \vdots \\ \vdots & \ddots & \ddots & m_{j-1,j}^\top & \vdots & & \vdots \\ \vdots & & \ddots & d_j & \vdots & & \vdots \\ \vdots & & & l_{j+1,j} & \vdots & & \vdots \\ \vdots & & & \vdots & \vdots & & \vdots \\ l_{n1} & \cdots & \cdots & l_{nj} & a_{n,j+1} & \cdots & a_{nn} \end{bmatrix}$$

Algorithmischer Aufwand

$$\begin{array}{llcll}
1. & v_i & = & \text{Lösung von } L_{i*}^{(j)} v = a_{ij}, & i = 1, \dots, j \\
2. & d_j & = & v_j & \\
3. & m_{ji} & = & \frac{v_i}{d_i}, & i = 1, \dots, j \\
4. & l_{ij} & = & \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} v_k}{v_j}, & i = j+1, \dots, n
\end{array}$$

1. Je j-fache Rücksubstitution $\Longrightarrow \sum_{j=1}^{n} \sum_{k=1}^{j} 2k \in \mathcal{O}(\frac{n^3}{3})$
2. Je eine Zuweisung $\Longrightarrow \sum_{j=1}^{n} 1 = n$ flops
3. Je j-fache Division $\Longrightarrow \sum_{j=1}^{n} j \in \mathcal{O}(\frac{n^2}{2})$ flops
4. Je $(n-j)$-faches, j-maliges Aufsummieren und eine Division $\Longrightarrow \sum_{j=1}^{n} (n-j) \cdot 2j = n^2(n+1) - \frac{n(n+1)(2n+1)}{3} = \frac{n^3 - n}{3}$

Zusammen also $\mathcal{O}(n^3)$. In der Literatur findet sich ein Wert von $\mathcal{O}(\frac{2}{3} n^3)$ für den gesamten Algorithmus.

$LDL^\top$-Zerlegung für symmetrisches A

Sei $A \in \mathbb{R}^{n\times n}$ symmetrisch und regulär. Dann gilt für ihre $LDM^\top$-Zerlegung $L = M$.

Betrachte $L = \triangle$ und die Produkte $C = LAL^\top$, $B = LA$.
Behauptung: C ist ebenfalls symmetrisch.

$$
\begin{aligned}
c_{ij} &= \sum_f b_{if} l_{fj}^\top = \sum_f \left(\sum_k l_{ik} a_{kf} \right) l_{jf} \\
&= \sum_f \sum_k l_{ik} a_{kf} l_{jf} \overset{a_{kf}=a_{fk}}{=} \sum_k \sum_f l_{jf} a_{fk} l_{ik} \\
&= \sum_k \left(\sum_f l_{jf} a_{fk} \right) l_{ki}^\top = \sum_k b_{jk} l_{ki}^\top \\
&= c_{ji}
\end{aligned}
$$

Also $C = LAL^\top$ ist symmetrisch.

Damit

$$\begin{aligned} & A = LDM^\top \text{ symmetrisch} \\ \Longrightarrow \quad & M^{-1}AM^{-\top} = \underbrace{M^{-1}}_{\text{◺}} \cdot \underbrace{LD}_{\text{◺}} = \text{◺} \text{ ebenfalls symmetrisch} \\ \Longrightarrow \quad & M^{-1}LD \text{ ist Diagonalmatrix.} \end{aligned}$$

Da D regulär, muss auch $M^{-1}L$ eine Diagonalmatrix sein. Wegen $m_{ii} = l_{ii} = 1$ folgt $M^{-1}L = I$, also $M^{-1} = L^{-1} \implies M = L$.

Also existieren $L, D \in \mathbb{R}^{n \times n}$, $L = \text{◺}$, $D = \operatorname{diag}(d_1, \dots, d_n)$ mit $A = LDL^\top$.

Ist im Voraus bekannt, dass A symmetrisch ist, können beim Algorithmus sowohl Speicherplatz als auch flops eingespart werden.

Da im j-ten Schritt bereits l_{jk} für $k < j$ bekannt ist, entfällt das Lösen des $(j \times j)$-Gleichungssystems $Lv = A$, da v auch durch $v = DM^\top e_j = DL^\top e_j$, d.h. ein Skalieren der bereits bekannten Werte l_{jk} durch d_k, berechnet werden kann:

$$v = \begin{pmatrix} d_1 \cdot l_{j1} \\ \vdots \\ d_{j-1} \cdot l_{j,j-1} \\ d(j) \end{pmatrix}$$

Es ergibt sich folgender Algorithmus.

```
1 for j = 1 : n
2     for i = 1 : j − 1
3         v(i) = L(j, i)d(i)
4     end
5     v(j) = A(j, j) − L(j, 1 : j − 1)v(1 : j − 1)
6     d(j) = v(j)
7     L(j + 1 : n, j) = (A(j + 1 : n, j) − L(j + 1 : n, 1 : j − 1)v(1 : j − 1)) / v(j)
8 end
```

- Die Laufzeit der Zeilen 2-4 und 5 ergeben wieder eine Arithmetische Reihe, d.h. $\mathcal{O}(n^2)$
- Zeile 6 hat lineare Laufzeit in n
- An Zeile 7 hat sich im Vergleich zu vorher nichts verändert, sie bleibt in $\mathcal{O}(\frac{n^3}{3})$

Damit hat der Algorithmus eine geringere Laufzeit von nur noch $\mathcal{O}(\frac{n^3}{3})$, was auf die Symmetrie von A zurückzuführen ist.

A positiv definit

Eine Matrix $A \in \mathbb{R}^{n \times n}$ heißt positiv definit, falls für alle $0 \neq x \in \mathbb{R}^n$ gilt

$$x^\top A x > 0$$

Anwendungen positiv definiter Matrizen umfassen bspw.

- Quadriken $x^\top A x + b^\top x = 1$: A bei Ellipsoiden hat nur positive Eigenwerte, ist deshalb auch pos. def.
- $X^\top X$ ist pos. def. falls X vollen Rang hat.
- Optimierungsprobleme, strikt konvexe Funktion mit einem globalen Minimum

Positive Definitheit wird eher selten ohne gleichzeitige Symmetrie betrachtet, obwohl es solche Matrizen gibt, bspw. $\left[\begin{smallmatrix} 1 & 1 \\ -1 & 1 \end{smallmatrix}\right]$.

Einschub: Rundungsfehler und Auslöschung

Arithmetische Operationen $+, -, \cdot, /$, aber auch $\sqrt[2]{x}$, $\operatorname{sincos} x$, $2^x - 1$, $y \cdot \log_2 x$, sowie $x^\top y$ für $x, y \in \mathbb{R}^n, n \leq 8$ usw. können als Maschinenbefehle ($\stackrel{?}{=}$ flop) ausgeführt werden, wenn man mit einer Approximation (Maschinengenauigkeit $\mathbf{u}$) der Zahlen arbeitet:

- Single (`float`, $\mathbf{u} = \frac{2^{-23}}{2} \approx 10^{-7}$) bzw.
- Double Precision (`double`, $\mathbf{u} = \frac{2^{-52}}{2} \approx 10^{-16}$)

Standard IEEE-754-2008 beschreibt deren Aufbau und Rundungsverhalten.

$\leadsto$ Geschwindigkeit aber auch Rundungsfehler v.a. wenn große und kleine Zahlen miteinander verrechnet werden.

Beispiel zu Auslöschung: 8-stellige Dezimalarithmetik

$$\begin{array}{rcl} 0{,}12345679 & \leftarrow & \text{relativer Fehler } \approx 10^{-8} \\ -0{,}12345678 & & \\ \hline 0{,}00000001 & \leftarrow & \text{relativer Fehler } \approx 1 \end{array}$$

A pos. def., nicht symmetrisch

Sei $\varepsilon > 0$, dann ist A positiv definit:

$$A = \begin{bmatrix} \varepsilon & m \\ -m & \varepsilon \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\frac{m}{\varepsilon} & 1 \end{bmatrix} \begin{bmatrix} \varepsilon & 0 \\ 0 & \varepsilon + \frac{m^2}{\varepsilon} \end{bmatrix} \begin{bmatrix} 1 & \frac{m}{\varepsilon} \\ 0 & 1 \end{bmatrix} = LDM^\top$$

Falls $\frac{m}{\varepsilon} \gg 1$

- $LDM^\top$-Zerlegung wird instabil
- stärkere Rundungsfehlern und damit verbunden Auslöschungen

Betrachtet man die Anteile der Symmetrie, $T = (A + A^\top)/2$ und die der Asymmetrie, $S = (A - A^\top)/2$, dann lässt sich die absolute Größe der Einträge in $LDM^\top = A$ abschätzen durch

$$\| |L| \cdot \underbrace{|D|}_{(|d_{ij}|)_{ij}} \cdot |M^\top| \|_F \leq n(\|T\|_2 + \|ST^{-1}S\|_2).$$

Beweis siehe Golub, Van Loan: „Unsymmetric Positive Definite Linear Systems“ (1979).

Wurde $\hat{x}$ berechnet als Lösung zu $Ax = b$, dann beschreibt dieses Rundungsfehler-behaftete $\hat{x}$ die Lösung des um $E \in \mathbb{R}^{n \times n}$ gestörten Systems $(A + E)\hat{x} = b$.

E lässt sich nun abschätzen durch (c abhängig von $\||\hat{L}||\hat{D}||\hat{M}^\top|\|_F$)

$$\|E\|_F \in \mathcal{O}(n\|A\|_F + \mathbf{u}cn^2(\|A\|_2 + \|ST^{-1}S\|_2) + \mathbf{u}^2).$$

Golub und van Loan kommen zu dem Schluss, dass falls

$$\Omega = \frac{\|ST^{-1}S\|_2}{\|A\|_2}$$

„nicht zu groß“ ist, nicht pivotiert werden muss.

D.h. die Norm des asymmetrischen Anteils S darf im Vergleich zu dem des symmetrischen Teils T nicht zu groß werden.

Ist A symmetrisch, dann ist $S = 0$ und damit ebenfalls $\Omega = 0$.

A symmetrisch und positiv definit

Eine Eigenschaft symmetrisch positiver Matrizen ist, dass ihre Elemente auf der Hauptdiagonalen alle anderen dominieren (ohne Beweis).

Für A sym. pos. def. existiert $LDL^\top = A$ und ist stabil zu berechnen.

L ist eine strikte untere Dreiecksmatrix, d.h. für ihre Einträge auf der Hauptdiagonalen gilt $l_{ii} = 1$.

Die Diagonalmatrix D enthält wegen A pos. def. nur positive Einträge auf der Diagonalen.

Deshalb lässt sich D nun in die beiden Faktoren L bzw. $L^\top$ integrieren, sodass eine neue Zerlegung von A entsteht.

Cholesky-Zerlegung

Sei $G = L\,\mathrm{diag}(\sqrt{d_1}, \ldots, \sqrt{d_n})$, dann $\mathbb{R}^{n\times n} \ni G = \triangleleft$ und es gilt:

$$A = GG^\top.$$

G wird auch Cholesky-Dreieck genannt.

Offenbar

$$\begin{aligned} GG^\top &= (L\,\mathrm{diag}(\sqrt{d_i})_i)(\mathrm{diag}(\sqrt{d_i})_i L^\top) \\ &= L\,\mathrm{diag}(d_i)_i L^\top = LDL^\top. \end{aligned}$$

Das System $Ax = b$ lässt sich nun wieder durch zweifache Rücksubstitution $Gy = b$ und $G^\top x = y$ lösen. Dann

$$b = Gy = G(G^\top x) = (GG^\top)x = Ax.$$

Ein Vorteil ist, dass nur eine Dreicksmatrix G abgespeichert werden muss.

Beispiel

$A = \begin{bmatrix} 2 & -2 \\ -2 & 5 \end{bmatrix}$ positiv definit und symmetrisch.

```
for j = 1 : n
    for i = 1 : j − 1
        v(i) = L(j, i)d(i)
    end
    v(j) = A(j, j) − L(j, 1 : j − 1)v(1 : j − 1)
    d(j) = v(j)
    L(j + 1 : n, j) = (A(j + 1 : n, j) −
        L(j + 1 : n, 1 : j − 1)v(1 : j − 1)) / v(j)
end
```

$LDL^\top$-Zerlegung:

- $j = 1$: $v_1 = a_{11} - 0 = 2 = d_1$,
  $l_{21} = \dfrac{a_{21} - 0}{v_1} = -1$
- $j = 2$: $v_1 = l_{21} d_1 = -2$,
  $v_2 = a_{22} - l_{21} v_1 = 3 = d_2$

$$A = \underbrace{\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}}_{L^\top} = \underbrace{\begin{bmatrix} \sqrt{2} & 0 \\ -\sqrt{2} & \sqrt{3} \end{bmatrix}}_{G} \underbrace{\begin{bmatrix} \sqrt{2} & -\sqrt{2} \\ 0 & \sqrt{3} \end{bmatrix}}_{G^\top}$$

Beispiel

$A = \begin{bmatrix} 2 & -2 \\ -2 & 5 \end{bmatrix}$ positiv definit und symmetrisch.

```
for j = 1 : n
    for i = 1 : j − 1
        v(i) = L(j, i)d(i)
    end
    v(j) = A(j, j) − L(j, 1 : j − 1)v(1 : j − 1)
    d(j) = v(j)
    L(j + 1 : n, j) = (A(j + 1 : n, j) −
        L(j + 1 : n, 1 : j − 1)v(1 : j − 1)) / v(j)
end
```

$LDL^\top$-Zerlegung:

- $j = 1$: $v_1 = a_{11} - 0 = 2 = d_1$,
  $l_{21} = \dfrac{a_{21} - 0}{v_1} = -1$
- $j = 2$: $v_1 = l_{21} d_1 = -2$,
  $v_2 = a_{22} - l_{21} v_1 = 3 = d_2$

$$A = \underbrace{\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}}_{L^\top} = \underbrace{\begin{bmatrix} \sqrt{2} & 0 \\ -\sqrt{2} & \sqrt{3} \end{bmatrix}}_{G} \underbrace{\begin{bmatrix} \sqrt{2} & -\sqrt{2} \\ 0 & \sqrt{3} \end{bmatrix}}_{G^\top}$$

Beispiel

$$A = \begin{bmatrix} 2 & -2 \\ -2 & 5 \end{bmatrix}$$ positiv definit und symmetrisch.

```
for j = 1 : n
    for i = 1 : j − 1
        v(i) = L(j, i)d(i)
    end
    v(j) = A(j, j) − L(j, 1 : j − 1)v(1 : j − 1)
    d(j) = v(j)
    L(j + 1 : n, j) = (A(j + 1 : n, j) −
        L(j + 1 : n, 1 : j − 1)v(1 : j − 1)) / v(j)
end
```

$LDL^\top$-Zerlegung:

- $j = 1$: $v_1 = a_{11} - 0 = 2 = d_1$,
  $l_{21} = \dfrac{a_{21} - 0}{v_1} = -1$
- $j = 2$: $v_1 = l_{21} d_1 = -2$,
  $v_2 = a_{22} - l_{21} v_1 = 3 = d_2$

$$A = \underbrace{\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}}_{L^\top} = \underbrace{\begin{bmatrix} \sqrt{2} & 0 \\ -\sqrt{2} & \sqrt{3} \end{bmatrix}}_{G} \underbrace{\begin{bmatrix} \sqrt{2} & -\sqrt{2} \\ 0 & \sqrt{3} \end{bmatrix}}_{G^\top}$$

Beispiel

$$A = \begin{bmatrix} 2 & -2 \\ -2 & 5 \end{bmatrix}$$ positiv definit und symmetrisch.

```
for j = 1 : n
    for i = 1 : j − 1
        v(i) = L(j, i)d(i)
    end
    v(j) = A(j, j) − L(j, 1 : j − 1)v(1 : j − 1)
    d(j) = v(j)
    L(j + 1 : n, j) = (A(j + 1 : n, j) −
        L(j + 1 : n, 1 : j − 1)v(1 : j − 1)) / v(j)
end
```

$LDL^\top$-Zerlegung:

- $j = 1$: $v_1 = a_{11} - 0 = 2 = d_1$,
  $l_{21} = \dfrac{a_{21} - 0}{v_1} = -1$
- $j = 2$: $v_1 = l_{21} d_1 = -2$,
  $v_2 = a_{22} - l_{21} v_1 = 3 = d_2$

$$A = \underbrace{\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}}_{L^\top} = \underbrace{\begin{bmatrix} \sqrt{2} & 0 \\ -\sqrt{2} & \sqrt{3} \end{bmatrix}}_{G} \underbrace{\begin{bmatrix} \sqrt{2} & -\sqrt{2} \\ 0 & \sqrt{3} \end{bmatrix}}_{G^\top}$$

Beispiel

$A = \begin{bmatrix} 2 & -2 \\ -2 & 5 \end{bmatrix}$ positiv definit und symmetrisch.

```
for j = 1 : n
    for i = 1 : j − 1
        v(i) = L(j, i)d(i)
    end
    v(j) = A(j, j) − L(j, 1 : j − 1)v(1 : j − 1)
    d(j) = v(j)
    L(j + 1 : n, j) = (A(j + 1 : n, j) −
        L(j + 1 : n, 1 : j − 1)v(1 : j − 1)) / v(j)
end
```

$LDL^\top$-Zerlegung:

- $j = 1$: $v_1 = a_{11} - 0 = 2 = d_1$,
  $l_{21} = \dfrac{a_{21} - 0}{v_1} = -1$
- $j = 2$: $v_1 = l_{21} d_1 = -2$,
  $v_2 = a_{22} - l_{21} v_1 = 3 = d_2$

$$A = \underbrace{\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}}_{L^\top} = \underbrace{\begin{bmatrix} \sqrt{2} & 0 \\ -\sqrt{2} & \sqrt{3} \end{bmatrix}}_{G} \underbrace{\begin{bmatrix} \sqrt{2} & -\sqrt{2} \\ 0 & \sqrt{3} \end{bmatrix}}_{G^\top}$$

Beispiel

$A = \begin{bmatrix} 2 & -2 \\ -2 & 5 \end{bmatrix}$ positiv definit und symmetrisch.

```
for j = 1 : n
    for i = 1 : j − 1
        v(i) = L(j, i)d(i)
    end
    v(j) = A(j, j) − L(j, 1 : j − 1)v(1 : j − 1)
    d(j) = v(j)
    L(j + 1 : n, j) = (A(j + 1 : n, j) −
        L(j + 1 : n, 1 : j − 1)v(1 : j − 1)) / v(j)
end
```

$LDL^\top$-Zerlegung:

- $j = 1$: $v_1 = a_{11} - 0 = 2 = d_1$,
  $l_{21} = \dfrac{a_{21} - 0}{v_1} = -1$
- $j = 2$: $v_1 = l_{21} d_1 = -2$,
  $v_2 = a_{22} - l_{21} v_1 = 3 = d_2$

$$A = \underbrace{\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}}_{L^\top} = \underbrace{\begin{bmatrix} \sqrt{2} & 0 \\ -\sqrt{2} & \sqrt{3} \end{bmatrix}}_{G} \underbrace{\begin{bmatrix} \sqrt{2} & -\sqrt{2} \\ 0 & \sqrt{3} \end{bmatrix}}_{G^\top}$$

Effizienter Algorithmus für Cholesky-Zerlegung

Der Algorithmus wird $A \in \mathbb{R}^{n \times n}$ zerlegen in $\beta \in \mathbb{R}$, $v \in \mathbb{R}^{n-1}$ und eine neue Matrix $A' \in \mathbb{R}^{n-1 \times n-1}$, mit denen diese Prozedur dann wiederholt wird.

Betrachte die Partitionierung

$$A = \begin{bmatrix} \alpha & v^\top \\ v & B \end{bmatrix} = \underbrace{\begin{bmatrix} \beta & 0 \\ \frac{v}{\beta} & I_{n-1} \end{bmatrix}}_{G_1} \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & B - \frac{1}{\alpha} v v^\top \end{bmatrix}}_{A'} \underbrace{\begin{bmatrix} \beta & \frac{v^\top}{\beta} \\ 0 & I_{n-1} \end{bmatrix}}_{G_1^\top},$$

dann ist $\beta = \sqrt{\alpha}$. Wegen A pos. def. gilt $\alpha > 0$.

Es ergibt sich A' aus der Teilmatrix B aktualisiert durch das skalierte äußere Produkt $-\frac{1}{\alpha} v v^\top$.

Soll diese Vorschrift induktiv fortgesetzt werden, so bleibt zu zeigen, dass A' immernoch positiv definit ist.

Zeige zunächst allgemeiner:

- Für $A \in \mathbb{R}^{n \times n}$ positiv definit und $X \in \mathbb{R}^{n \times k}$ mit Rang k gilt $B = X^\top A X \in \mathbb{R}^{k \times k}$ ist positiv definit.

Falls für $z \in \mathbb{R}^k$ gilt

$$0 \geq z^\top B z = (Xz)^\top A (Xz),$$

dann $Xz = 0$, da A positiv definit ist.

Da X aber vollen Spaltenrang k hat, ist sein Kern $\{0\}$, d.h. $Xz = 0 \iff z = 0$.

Also $z^\top B z \leq 0 \iff z = 0$, damit ist die neue Matrix B ebenfalls positiv definit.

$A' = B - \frac{1}{\alpha}vv^\top$ positiv definit

Sei $X = \begin{bmatrix} \frac{1}{\beta} & -\frac{v^\top}{\beta^2} \\ 0 & I_{n-1} \end{bmatrix} \in \mathbb{R}^{n\times n}$.

Für $X^\top A X$ gilt mit Obigem positive Definitheit. Andererseits ist

$$X^\top G_1 = \begin{bmatrix} \frac{1}{\beta} & 0 \\ -\frac{v}{\beta^2} & I_{n-1} \end{bmatrix} \begin{bmatrix} \beta & 0 \\ \frac{v}{\beta} & I_{n-1} \end{bmatrix} = I,$$

d.h. $B - \frac{1}{\alpha}vv^\top$ ist eine Teilmatrix von
$X^\top A X = X^\top G_1 A' G_1^\top X = A'$ und damit auch positiv definit.

Also sind alle Voraussetzungen erfüllt und es lässt sich nun G_2 und ein A'' mit dem selben Verfahren aus A' erzeugen. Damit

$$A = G_1 G_2 \dots G_{n-1} G_n \cdot I_n \cdot G_n^\top G_{n-1}^\top \dots G_2^\top G_1^\top$$

Cholesky-Zerlegung mittels äußeres-Produkt-Updates

Es ergibt sich folgender Algorithmus.

1 **for** $k = 1 : n$
2 $\quad A(k,k) = \sqrt{A(k,k)}$ $\quad$ /* β */
3 $\quad A(k+1:n,k) = A(k+1:n,k)/A(k,k)$ $\quad$ /* $\frac{\mathbf{v}}{\beta}$ */
4 $\quad$ **for** $j = k+1 : n$
5 $\quad\quad A(j:n,j) = A(j:n,j) - A(j:n,k)A(j,k)$ $\quad$ /* $B_{*j} - vv_j^\top/\beta$ */
6 $\quad$ **end**
7 **end**

Zeile 5 führt die Anpassung von B mittels des äußeren Produkts $vv^\top$ durch.

Die Matrix $G = G_1 G_2 \dots G_n$ wird an alten Positionen im unteren Dreieck von A gespeichert, wo sich je die Vektoren v im Schritt k befanden.

Laufzeit wieder in $\mathcal{O}(\frac{n^3}{3})$, wegen der Neuberechnung von $B \in \mathbb{R}^{n-k \times n-k}$ in jedem Schritt $1 \leq k \leq n$ (Zeilen 4-6).

Cholesky-Zerlegung mittels GAXPY

$\exists$ weitere Algorithmen, mit Wertlegung auf sogenannte GAXPY-Operationen (Ax $\underline{\text{p}}$lus y)

- häufig anzutreffende Operation in numerischen Verfahren
- schnell in Hardware, siehe oben:
  $x^\top y$ für $x, y \in \mathbb{R}^n$, $n \le 8$ existiert bereits als Maschinenbefehl auf neueren CPUs – solche mit AVX-Instruktionserweiterung

```
for j = 1 : n
    if j > 1
        A(j : n, j) = A(j : n, j) − A(j : n, 1 : j − 1) A(j, 1 : j − 1)^T  /* GAXPY */
    end
    A(j : n, j) = A(j : n)/sqrt(A(j, j))  /* scaled GAXPY operation */
end
```

Dieser Algorithmus benötigt nur halb so viele Vektoroperationen wie der Obige, läuft allerdings auch in $\mathcal{O}(\frac{n^3}{3})$.

A sym. pos. def. $\implies$ Cholesky-Zerlegung existiert und Algorithmus findet je immer positive α, sodass $\sqrt{\alpha}$ existiert.

In der anderen Richtung ergibt sich, dass falls ein nichtpositives α im Algorithmus gefunden wird, A nicht sym. pos. def. war.

Zur Stabilität:

Der Einträge in G sind begrenzt durch die Diagonaleinträge von A:

$$\|A\|_2 = \|GG^\top\|_2 = \|G\|_2^2 \implies g_{ij}^2 \leq \sum_{k=1}^{i} g_{ik}^2 = a_{ii}$$

Der Rundungsfehler einer Lösung $\hat{x}$ von $(A+E)\hat{x} = b$ wird begrenzt durch

$$\|E\|_2 \leq c_n \mathbf{u} \|A\|_2,$$

wobei c_n eine kleine, von n abhängige Konstante ist (siehe Wilkinson 1968).

A symmetrisch und positiv semidefinit

A ist positiv semidefinit, falls $\forall x \in \mathbb{R}^n : x^\top Ax \geq 0$.

Einige Eigenschaften:

1. $|a_{ij}| \leq (a_{ii} + a_{jj})/2$
2. $|a_{ij}| \leq \sqrt{a_{ii} a_{jj}}$
3. $\max\limits_{i,j} |a_{ij}| = \max\limits_{i} a_{ii}$
4. $a_{ii} = 0 \implies a_{i*} = a_{*i} = 0$

Denn:

1. $x = e_i + e_j \implies 0 \leq x^\top Ax = a_{ii} + a_{jj} + 2a_{ij}$ und
   $x = e_i - e_j \implies 0 \leq x^\top Ax = a_{ii} + a_{jj} - 2a_{ij}$.
2. O.E. $i = 1$, $j = 2$, dann da $A(1:2, 1:2)$ ebenfalls pos. semidef. ist, gilt $\forall x$

$$0 \leq \begin{pmatrix} x \\ 1 \end{pmatrix}^\top \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix} = a_{11}x^2 + 2a_{12}x + a_{22}$$

   $\implies$ Diskriminante $4a_{12}^2 - 4a_{11}a_{22}$ der quadratischen Gleichung muss negativ sein.

3. bzw. 4. folgen aus 1. bzw. 2.

Eine Cholesky-ähnliche Operation lässt sich auch auf symmetrisch pos. semidef. Matrizen anwenden.

Angenommen $a_{kk} = 0$, dann folgt mit 4., dass $a_{ik} = a_{ki} = 0$ $(k \leq i \leq n)$ und es bleibt im Algorithmus für G_k nichts zu tun.

Es ergibt sich durch einfache Abfrage auf $a_{kk} > 0$

```
1 for k = 1 : n
2     if A(k,k) > 0
3         A(k,k) = sqrt(A(k,k))
4         A(k+1 : n, k) = A(k+1 : n, k)/A(k,k)
5         for j = k+1 : n
6             A(j : n, j) = A(j : n, j) - A(j : n, k)A(j,k)
7         end
8     end
9 end
```

Rundungsfehler können bei der Abfrage jedoch fehlerhafte „wahr“-Instanzen liefern, die im Folgenden dann Zeile 4 deutlich falsche Ergebnisse liefern.

Symmetrisches Pivotieren

Um sich stark auswirkende Rundungsfehler zu vermeiden wurde bisher mit Pivotierung gearbeitet.

$P \in \mathbb{R}^{n \times n}$ Permutationsmatrix $\implies$ PA hat dieselben Zeilen wie A, jedoch anders sortiert $\implies$ Symmetrie bleibt nicht erhalten.

Eine die Symmetrie erhaltende Permutation wird durch $PAP^\top$ erreicht, bei der die Diagonalelemente auf der Diagonalen bleiben, jedoch anders angeordnet werden.

P von links sortiert die Zeilen um, die Inverse $P^\top$ von rechts sortiert die Spalten um.

Beispiel zum symmetrischen Pivotieren

$$
\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}
\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}
\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}
$$

$$
= \begin{bmatrix} a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \\ a_{11} & a_{12} & a_{13} & a_{14} \end{bmatrix}
\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}
$$

$$
= \begin{bmatrix} a_{22} & a_{23} & a_{24} & a_{21} \\ a_{32} & a_{33} & a_{34} & a_{31} \\ a_{42} & a_{43} & a_{44} & a_{41} \\ a_{12} & a_{13} & a_{14} & a_{11} \end{bmatrix}
$$

Quellen

- Prof. Dr. Schulz, Skript Numerik ’08/’09, Trier
- L.N. Trefethen, D. Bau, „Numerical Linear Algebra“
- G.H. Golub, C.T. van Loan, „Matrix computations“